

А. И. ЯКИМОВ

**ТЕХНОЛОГИЯ ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ
ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ**

Монография

Могилев
ГУ ВПО «Белорусско-Российский университет»
2010

УДК 658.012.011.56(043.3)

Рекомендовано к опубликованию Советом Белорусско-Российского университета «29» мая 2009 г., протокол № 12

Рецензенты: д-р техн. наук, проф., зав. кафедрой «Математические проблемы управления» Гомельского государственного университета им. Ф. Скорины И. В. Максимей ; д-р техн. наук, доц., проф. кафедры «Автоматизация технологических процессов и производств» Могилевского государственного университета продовольствия Н. Н. Дорогов

Якимов, А. И.

Технология имитационного моделирования систем управления промышленных предприятий : монография / А. И. Якимов. – Могилев : Белорус.-Рос. ун-т, 2010. – 304 с. : ил.
ISBN 978-985-492-072-6.

Детально рассмотрен метод моделирования деятельности промышленных предприятий. Представлен программно-технологический комплекс имитации сложных систем (ПТКИ) BelSim 2, реализованный на основе применения современных информационных технологий UML, IDEF0, XML и др. Приведены примеры использования ПТКИ при системном анализе деятельности предприятий.

Монография предназначена для научных и инженерно-технических работников, решающих вопросы системного анализа, моделирования и автоматизации деятельности предприятий, а также для аспирантов, слушателей курсов повышения квалификации, студентов соответствующих специальностей.

УДК 658.012.011.56(043.3)

ISBN 978-985-492-072-6

© Якимов А. И., 2010

© ГУ ВПО «Белорусско-Российский университет», 2010

Перечень условных обозначений, терминов и сокращений

MPI (Message Passing Interface) – технология программирования параллельных компьютеров с распределенной памятью. Основной способ взаимодействия параллельных процессов – передача сообщений друг другу [17, с. 29].

XML (Extensible Markup Language) – язык разметки, описывающий класс объектов данных, называемых XML-документами. Используется в качестве средства для описания грамматики других языков и контроля правильности составления документов [197, с. 39].

Абстракция (abstraction) – характеристика сущности, которая отличает ее от других сущностей [106, с. 19].

Валидация – проверка правильности концептуальной модели на точное соответствие отображению системы для конкретных целей исследования [188, с. 43].

Верификация – проверка правильности реализации алгоритмов имитационной модели [188, с. 43].

ИМ – имитационная модель.

Инкапсуляция – принцип объектно-ориентированного проектирования; характеризует сокрытие отдельных деталей внутреннего устройства классов от внешних по отношению к нему объектов или пользователей [106, с. 20].

Класс – абстракция совокупности реальных объектов, которые имеют общий набор свойств и обладают одинаковым поведением. **Объект** в контексте объектно-ориентированного проектирования рассматривается как экземпляр соответствующего класса [106, с. 19].

Координация – проблема принятия соответствующих решений блоком верхнего уровня, который должен так воздействовать на блоки нижнего уровня, чтобы была достигнута цель, поставленная перед всей системой [119, с. 13].

Критерий – показатель или функционал, который дает возможность количественно или качественно оценить то или иное свойство системы [82, с. 83].

Модель (model) – абстракция физической системы, рассматриваемая с определенной точки зрения и представленная на некотором языке или в графической форме [106, с. 18].

Наследование – принцип объектно-ориентированного проектирования, в соответствии с которым знание о наиболее общей категории разрешается применять для более частной категории [106, с. 19].

Неопределенность целей – это неопределенность выбора и достижения целей в многокритериальных задачах принятия решений [82, с. 142].

Объектно-ориентированный анализ и проектирование (ООАП, Object-Oriented Analysis/Design) – технология разработки программных

систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов [106, с. 23].

Параметр – количественный показатель, который определяет некоторое свойство системы [82, с. 83].

ПО – программное обеспечение.

Полиморфизм (греч. *Poly* – много, *morfos* – форма) – свойство объектов принимать различные внешние формы в зависимости от обстоятельств [106, с. 21].

ПТКИ – программно-технологический комплекс имитации сложных систем.

Рациональный – разумно обоснованный, целесообразный [131, с. 659].

Система – упорядоченное множество структурно взаимосвязанных и функционально взаимозависимых элементов [82, с. 69].

Системная задача – задача анализа определенной совокупности свойств объекта исследования с единой позиции целостного подхода для достижения заданных целей в определенных условиях [82, с. 76].

Системный анализ – прикладная научная методология, которая опирается на множество структурно взаимозависимых и функционально взаимодействующих эвристических процедур, методических приемов, программных и вычислительных средств, что обеспечивает формирование целостных междисциплинарных знаний об исследуемом объекте [82, с. 60].

Сложная иерархическая система – целостный объект, созданный из функционально разнотипных элементов, структурно взаимосвязанных отношением подчинения и функционально объединенных для решения заданных целей в определенных условиях. Пример: современное производственное предприятие [82, с. 70].

Сложная система – упорядоченное множество структурно взаимосвязанных и функционально взаимозависимых систем [82, с. 69].

СМ – система моделирования.

Событие – итог, который может произойти при осуществлении комплекса условий [166, с. 7].

Управление – процесс организации такого целенаправленного воздействия на объект, в результате которого этот объект переводится в требуемое (целевое) состояние [153, с. 12].

Фактор – внешнее или внутреннее воздействие, которое влияет на определенные свойства и поведение сложных систем [82, с. 83].

Введение

В настоящее время состояние предприятий характеризуется большим количеством, интенсивностью и разнонаправленностью информационных связей между подсистемами и элементами; слабой формализацией технологий управления основными, вспомогательными, обслуживающими процессами предприятия; изолированностью систем управления бизнес-процессами и производственными процессами; непрозрачностью различных сторон деятельности предприятий; отсутствием методов и средств оптимизации систем управления бизнес-процессами и производственными процессами.

Современные научные исследования показывают, что неэффективность деятельности многих предприятий, в частности, нерациональные решения, принимаемые руководством, связаны с недостаточной автоматизацией управленческих процессов при наличии больших объемов разнородной информации, учитываемой при принятии решений. Совершенствование экономико-математических моделей, используемых при выработке плановых и управленческих решений, а также прогнозирование технико-экономических показателей на предприятиях, в министерствах, ведомствах и объединениях является одним из приоритетных направлений научно-технической деятельности в Республике Беларусь на 2006–2010 годы. Приоритетными направлениями фундаментальных и прикладных исследований являются развитие теоретико-методологических основ информатики и информационных технологий, разработка аппаратных и программных комплексов и систем для информационного обеспечения управления.

Комплексная информационная система (КИС) промышленного предприятия, являясь средой информационной поддержки целенаправленной коллективной деятельности, решает задачу эффективного управления всеми ресурсами предприятия (материально-техническими, финансовыми, технологическими, трудовыми и интеллектуальными) для получения максимальной прибыли и удовлетворения материальных и профессиональных потребностей всех сотрудников предприятия. Она включает в себя совокупность различных программно-аппаратных платформ, универсальных и специализированных программных приложений, интегрированных в единую информационно-однородную систему. КИС должна накапливать определенный опыт и формализованные знания, постоянно совершенствоваться и развиваться, быстро адаптироваться к изменяющимся условиям внешней среды и новым потребностям предприятия [111, с. 90–92].

На развитом промышленном предприятии руководитель принимает решение по результатам анализа различной информации от других подразделений. При этом невозможно принять эффективное решение ввиду сложности и многообразия вопросов, подлежащих анализу. Декомпозиция функциональных задач приводит к созданию структуры управления,

включающей различные отделы: производственный, отдел сбыта, финансовый, отдел кадров и пр. При этом отделы имеют разные *цели функционирования*, во многом *взаимно противоположные* [35, с. 8]. Например, *производственный отдел* желает, чтобы продукция была однообразной (малономенклатурной) и, если даже нет сбыта, продолжался выпуск продукции. Цель – максимальный выпуск продукции с узкой номенклатурой, чтобы не перенастраивать станки для снижения затрат при переходе на новый ассортимент. *Отдел сбыта* требует широкого ассортимента продукции, чтобы легче ее реализовывать, чтобы были в запасе товары, даже редко пользующиеся спросом, т. к. они все равно могут понадобиться. Поэтому этот отдел не возражает против запасов, если даже нет производства. *Финансовый отдел*, однако, возражает против запасов, т. к. это связанные деньги, а его задача – минимизировать эти связанные деньги (деньги в запасах), что означает, в свою очередь, минимизацию запасов. Финансовый отдел требует производства продукции, даже если нет в настоящий момент продаж товара. *Отдел кадров* против сокращения производства, если отсутствуют продажи продукции, т. к. это связано с увольнением работников, а это всегда социально неприятная процедура.

Для производства и реализации продукции предприятие заключает множество контрактов. От эффективности этих контрактов напрямую зависят результаты деятельности предприятия. Это особенно актуально в условиях финансово-экономического кризиса, когда большинство промышленных предприятий балансируют на грани выживания. При отсутствии резервов финансовых средств одно единственное неверное решение, один заключенный невыгодный контракт могут привести к банкротству. Поэтому требуется тщательная проработка и оценка последствий принимаемых решений.

В *маркетинговой деятельности* промышленного предприятия необходимо принимать решения для определения отпускной цены на продукцию в условиях различных видов платежей. К ним относят векселя, ценные бумаги, проведение взаимозачетов, бартерных сделок, предварительную оплату, отсрочку платежа и ряд других. При этом отсутствие математически и экономически обоснованных методик расчета отпускных цен на товарную продукцию в условиях различных видов платежей и инфляционных процессов приводит к снижению рентабельности и потерям прибыли. Поэтому оценка и прогнозирование эффективности контрактов являются одними из важнейших задач в маркетинговой деятельности предприятий.

Сегодняшние белорусские предприятия работают в условиях серьезной конкуренции. При этом на базе уже существующих технологий важно оптимизировать две составляющие их деятельности: организацию и управление предприятием. Для решения этой задачи на предприятии требуется внедрение информационных технологий, сопровождаемое серьезными ин-

вестициями. Но в то же время оно должно повысить эффективность работы предприятия и в конечном счете увеличить его прибыли.

Для внедрения комплексных информационных систем (например, SAP R/3, Галактика) предприятия приходят к использованию современных MRP II- и ERP-концепций управления. MRP II – метод планирования всех ресурсов производственного предприятия на основе данных, полученных от поставщиков и потребителей; осуществляет прогнозирование, планирование и контроль за производством. ERP – это информационная система, ориентированная на бухгалтерский учет, для идентификации и планирования ресурсов по всему предприятию, необходимых для принятия и учета заказов клиентов, изготовления и отгрузки продукции. Одной из функций ERP-системы управления является математическое моделирование технологий на предприятии. Поэтому на основе КИС, используя единое хранилище данных, внедряются системы поддержки принятия решений, включающие в своей структуре аналитические модели (например, SAP BW, модуль Контроллинг в КИС Галактика). В отличие от аналитических моделей имитационная модель (ИМ) служит средством изучения динамики явлений в сложной производственно-экономической системе (СПЭС) и в ряде случаев обеспечивает выбор более экономичного варианта задачи исследования из-за высокого уровня детализации объекта исследования. Обычно ИМ используется в тех случаях, когда требуется наблюдение за поведением компонентов СПЭС в течение определенного периода времени. В ряде случаев ИМ является единственным способом исследования, когда невозможны наблюдения динамики явлений в реальной СПЭС либо когда изучаются новые ситуации, о которых мало что известно или неизвестно ничего. Часто ИМ используется для предсказания возможных трудностей, появляющихся в поведении СПЭС при введении в нее новых компонентов.

Задачи математического моделирования и оптимизации деятельности предприятий, дальнейшего совершенствования методов имитационного моделирования для системного представления динамики функционирования промышленного предприятия с учетом принятия решений представляются весьма актуальными как с научной, так и с практической точек зрения. При моделировании или анализе функций органов управления предприятием одной из центральных проблем является мера соотношения централизации и децентрализации управления. Указанная проблема особенно сильно проявляется в органах управления с иерархической структурой при наличии нескольких уровней принятия решений.

Важной является задача оптимального планирования ресурсов предприятия, которая решается в комплексных информационных ERP-системах, а эффективным инструментом ее решения может выступать имитационное моделирование. Проблема разработки и внедрения методов, средств и технологий имитационного моделирования в ERP-системы управления про-

мышленным предприятием ранее комплексно не рассматривалась. Поэтому весьма актуальным является создание методологии системного анализа процессов на предприятии, включающей в себя методы имитационного моделирования и технологии использования методов и средств для решения практических задач системного анализа функционирования предприятия.

Целью настоящей работы является решение научно-технической проблемы создания эффективных методов, средств и технологий имитационного моделирования для системного анализа функциональной деятельности в многоуровневой иерархической структуре системы на примере промышленного предприятия с ERP-системой управления ресурсами.

В *первом разделе* монографии дается аналитический обзор литературы по методам и средствам моделирования, их применению для принятия решений в управлении промышленным предприятием.

Во *втором разделе* излагается метод поиска рациональных решений в ERP-системе управления промышленным предприятием на основе ИМ.

В *третьем разделе* разработан метод имитационного моделирования ERP-системы управления ресурсами промышленного предприятия, предполагающий комплексное использование современных информационных технологий и ориентацию на применение комплекса современных коммерческих программных систем.

В *четвертом разделе* представлены состав и структура программно-технологического комплекса имитации сложных систем (ПТКИ) BelSim, который позволяет реализовать метод построения имитационных моделей производственно-экономической деятельности промышленных предприятий.

В *пятом разделе* представлены результаты апробации этого метода, средств и технологии имитационного моделирования ERP-системы управления.

В *шестом разделе* представлены программные средства для реализации метода поиска рациональных решений.

Седьмой раздел содержит основные результаты апробации метода поиска рациональных решений в ERP-системе управления ресурсами промышленного предприятия.

В заключении обобщены основные научные результаты по созданию методологии системного анализа промышленного предприятия на основе имитационного моделирования.

1 Обзор методов и средств моделирования и принятия решений в управлении промышленным предприятием

1.1 Обзор современных концепций управления промышленным предприятием

В конце 1960-х годов крупные компании с множеством автоматизированных рабочих мест стали искать способ упростить управление производственными процессами. Первым шагом на этом пути стало появление идеи единой модели данных в масштабе всей организации. Так появилась концепция системы управления *MRP (Material Requirements Planning)* – автоматизированное планирование потребности сырья и материалов для производства. Главное достижение *MRP*-системы – минимизация издержек, связанных со складскими запасами. *MRP* – планирование потребностей в материалах по замкнутому циклу, составление производственной программы и ее контроль на цеховом уровне. В ходе такого планирования даются рекомендации о выпуске заказов для пополнения материальных ресурсов. В дальнейшем, поскольку процесс планируется по времени, выдаются рекомендации по перепланированию открытых заказов в случае, если даты готовности и даты потребности не совпадают. *MRP* на определенный период начинается с указания изделий, перечисленных в объемно-календарном плане, затем определяются количество всех материалов и компонентов, необходимых для производства этих изделий, и данные по этим материалам и компонентам [143].

Дальнейшим развитием является *MRP II (Manufacturing resource planning)* – метод планирования уже всех ресурсов производственного предприятия на основе данных, полученных от поставщиков и потребителей; выполнение прогнозирования, планирования и контроля за производством. *MRP II* включает следующие функции: *Sales and Operation Planning* – Планирование продаж и производства; *Demand Management* – Управление спросом; *Master Production Scheduling* – Составление плана производства; *Material Requirement Planning* – Планирование потребностей в сырье и материалах; *Bill of Materials* – Спецификации продукции; *Inventory Transaction Subsystem* – Складская подсистема; *Scheduled Receipts Subsystem* – Отгрузка готовой продукции; *Shop Flow Control* – Управление производством на цеховом уровне; *Capacity Requirement Planning* – Планирование производственных мощностей; *Input/output control* – Контроль входа/выхода; *Purchasing* – Материально-техническое снабжение; *Distribution Resource Planning* – Планирование запасов сбытовой сети; *Tooling Planning and Control* – Планирование и управление инструментальными средствами; *Financial Planning* – Финансовое планирование; *Simulation* – Моделирование; *Performance Measurement* – Оценка результатов деятельности.

Планирование производственных ресурсов – это прямое расширение концепции *MRP* (планирования потребности в материальных ресурсах) с замкнутым циклом планирования. Производители, которые успешно внедрили систему *MRP*, имеют уменьшенные складские запасы, уменьшенное время выхода на рынок новых продуктов, увеличение прибыльности [48].

ERP (Enterprise Resource Planning) – планирование ресурсов в масштабе предприятия. Класс *ERP*, в отличие от *MRP* и *MRP II* (для которых имеются строгие определения и формализованные перечни требований), представлен только на уровне концепции. Термин *ERP* появился, когда в список учитываемых при планировании ресурсов добавились планирование для распределения и финансовое планирование. Различие между концепциями *MRP II* и *ERP* в том, что первая ориентирована на производство, а вторая – на бизнес. Например, условия кредитования заказчика по отгрузке готовой продукции попадают в *ERP*, но не в *MRP II*. Инструментарий *OLAP* [24, с. 49–66], средства поддержки принятия решений – также принадлежность *ERP*, но не *MRP/MRP II*-систем.

ERP – это информационная система, ориентированная на бухгалтерский учет, для идентификации и планирования ресурсов по всему предприятию, необходимых для принятия, изготовления, отгрузки и учета заказов клиентов. Система *ERP* отличается от типичной системы *MRP II* по техническим требованиям, таким как графический интерфейс пользователя, реляционная база данных, использование языка четвертого поколения и новейших компьютерных программных средств конструирования, архитектура клиент/сервер и мобильность открытой системы. Использование *ERP* содействует сокращению ошибок, уменьшению числа ненужных операций, улучшает способности к прогнозу и планированию, что может обеспечить значительное сокращение издержек и улучшение процесса производства.

CRM (Customer Relations Management) и *SCM (Supply Chain Management)* – управление отношениями с заказчиками и с поставщиками соответственно. *CRM* – методология управления ресурсами предприятия, ориентированная на продажи и взаимоотношения с клиентами. К настоящему времени эти задачи решаются благодаря использованию браузера с поддержкой Java. В качестве рабочего места эта система изначально ориентирована на доступ к приложениям из локальной сети предприятия и извне через Интернет. Место *CRM* и *SCM* в концепции *ERP II* показано на рисунке 1.1 [221, с. 5].

WCM (World Class Manufacturing) – идея глобального производства; предприятия, даже действующие строго на местном рынке, фактически конкурируют с компаниями по всему миру.

MES (Manufacturing Execution Systems) – исполнительные производственные системы; поддерживают интеграцию систем на заводе, концентрируясь на оперативной деятельности: включают центральную базу данных, системы качества, управления документооборотом, планирования.

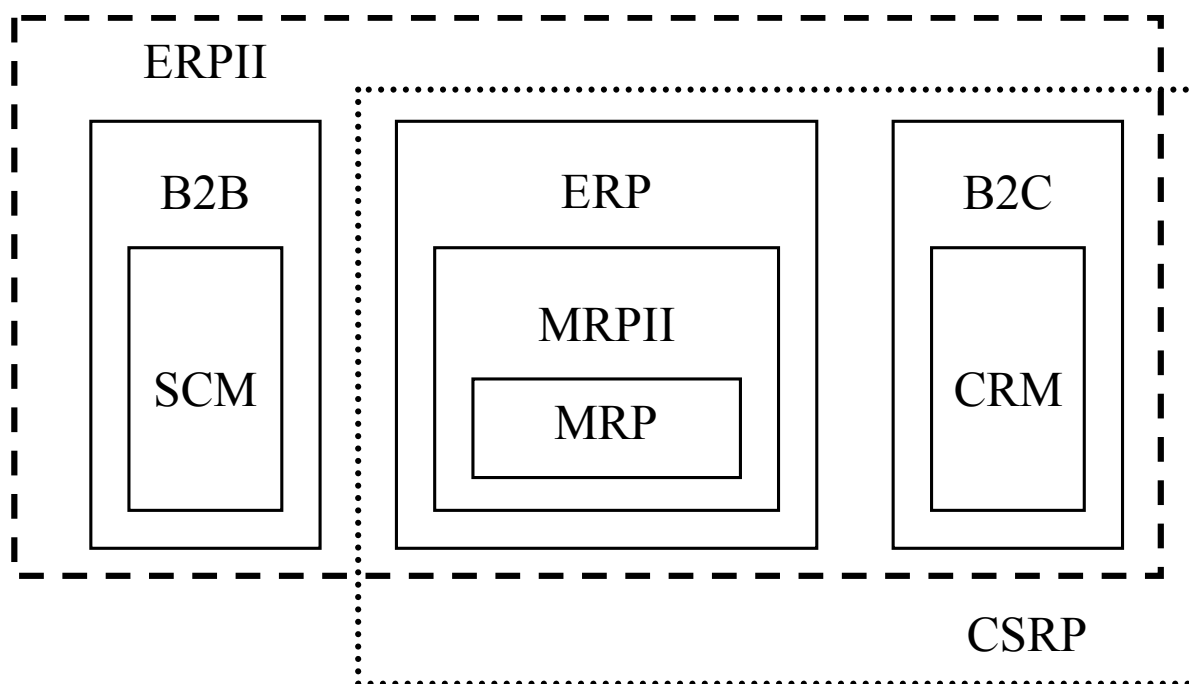


Рисунок 1.1 – Схема основных концепций управления предприятием

JIT (Just-In-Time) – точно в срок; является попыткой идентификации и устранения потерь, определяемых как «все, что увеличивает цену, не расширяя функциональности»: уменьшились запасы, поменялась планировка заводов, начата автоматизация производства [9, с. 44–45].

B2C (Business to Customer) и *B2B (Business to Business)* – обозначения широких классов программных продуктов, обслуживающих взаимоотношения предприятий с покупателями (*B2C*) и между собой (*B2B*) [16].

Концепция *CSRP (Customer Synchronized Resource Planning)* [63] направлена на расширение функциональности в сфере взаимодействия предприятия с его заказчиками. Корпоративные ресурсы, охватываемые *CSRP*-системой, обслуживают такие этапы производственной деятельности, как проектирование будущего изделия с учетом специфических требований заказчика, гарантийное и сервисное обслуживание [121].

В рассмотренных концепциях, направленных на повышение эффективности управления предприятиями, важное место отводится моделированию. Однако использование имитационного моделирования сдерживается трудностями построения модели сложной системы и отсутствием средств автоматизации построения таких моделей [219, с. 113–115; 222, с. 106–110].

В работе С. А. Сенченко [172, с. 4] отмечается необходимость информационного сопровождения управления. Представлен алгоритм принятия управленческих решений, построенный на основе работы Е. В. Попова [172, с. 14], в котором применяется имитационное моделирование. Однако практические рекомендации по интеграции имитационного моделирования с комплексной информационной системой (КИС) предприятия отсутствуют,

не развита целостная концепция управления экономической системой предприятия, основанная на эффективном формировании и использовании информационных ресурсов в управлении инновационными процессами.

1.2 Математическая постановка системной задачи управления сложным иерархическим объектом

Промышленные предприятия (ПП) имеют явно различимую иерархическую структуру разделения ресурсов. Поэтому предложена содержательная (вербальная) постановка системной задачи рационального управления функционированием предприятия как сложным иерархическим объектом. *Дано:* для сложной иерархической системы определены структуры всех уровней, требования к основным характеристикам и показателям управления процессами функционирования системы, которые в заданных условиях отличаются неполнотой и нечеткостью. *Требуется:* определить структуру системы управления, разработать проектные решения всех иерархических уровней при условии достижения необходимого качества управления системой в прогнозируемых штатных и нештатных ситуациях.

Математическая постановка задачи. В общем случае при исследовании ПП в целом возникает потребность в определении его целей на разных уровнях иерархии. При этом для одних целей оптимальные решения соответствуют минимальному значению определенного критерия, а для других – максимальному. Однако заменой переменных эти задачи легко привести к единому типу критериев и одному типу задач оптимизации. Их можно рассматривать как задачу многокритериальной оптимизации:

$$f_1(\bar{x}) \rightarrow \max_{\bar{x} \in D}; f_2(\bar{x}) \rightarrow \max_{\bar{x} \in D}; \dots; f_m(\bar{x}) \rightarrow \max_{\bar{x} \in D}, \quad (1.1)$$

где \bar{x} – вектор принимаемого решения;

D – область допустимых решений.

Очевидно, что наилучшим решением задачи раскрытия неопределенности будет такое значение \bar{x} , для которого условия (1.1) выполняются одновременно для всех целевых функций. Однако на практике это не выполняется, т. к. функции $f_k(\bar{x})$, $k = 1, 2, \dots, m$ разные по природе. Поэтому экстремума функция достигает при своем значении \bar{x} и зачастую невозможно найти такое значение \bar{x}^0 , для которого условия (1.1) выполняются одновременно для всех целевых функций. Поэтому задача сводится к нахождению такого значения \bar{x}^0 , для которого обеспечивается *рациональный компромисс* заданных целей [82, с. 143].

Для сложной многоуровневой системы структурная взаимосвязь функциональных элементов (ФЭ) разных иерархических уровней определена в виде

$$f_\ell : V_L \rightarrow \bigcup_{\ell=1}^L V_\ell; \quad f_{\ell p} : V_\ell \rightarrow \bigcup_{p=1}^{P_0} V_{\ell p}; \quad (1.2)$$

$$f_{\ell p}^{-1} : \bigcup_{p=1}^{P_0} V_{\ell p} \rightarrow V_\ell; \quad f_\ell^{-1} : \bigcup_{\ell=1}^L V_\ell \rightarrow V_L, \quad (1.3)$$

где V_L – множество ФЭ объекта в целом;

V_ℓ – множество ФЭ ℓ -го иерархического уровня;

$V_{\ell p}$ – p -й функциональный элемент ℓ -го иерархического уровня.

Функциональная взаимосвязь показателей качества функционирования ПП в целом с параметрами ФЭ определена в виде функции с монотонным включением переменных:

$$\bar{K}_L = \tilde{F}_L(\bar{X}_L, \tilde{F}_{L-1}(\bar{X}_{L-1}, \tilde{F}_{L-2}(\dots \tilde{F}_\ell(\bar{X}_\ell, \tilde{F}_{\ell-1}(\dots, \tilde{F}_1(\bar{X}_1)) \dots))), \quad (1.4)$$

где \bar{K}_L – вектор показателей качества ПП;

\bar{X}_L – вектор параметров функциональных элементов высшего уровня иерархической структуры системы (уровень ПП в целом);

\bar{X}_ℓ – вектор параметров функциональных элементов ℓ -го иерархического уровня, $\ell = 1, \dots, L$;

\bar{X}_1 – вектор параметров функциональных элементов нижнего иерархического уровня (например, уровень элементов технологического процесса);

$\tilde{F}_L, \dots, \tilde{F}_\ell, \dots, \tilde{F}_1$ – функции взаимосвязи показателей функциональных элементов иерархический уровней; эти функции определены не точно, нечетко, неполно.

Требуется: построить такую последовательность преобразований, которая определит структуру системы управления и параметры ее функциональных элементов с учетом достижения необходимого уровня работы ПП в некоторый момент времени $t \in [0, T]$ в прогнозируемых ситуациях риска $s_k \in S_0, k = 1, \dots, N$:

$$\Phi_\ell : G_L \rightarrow \bigcup_{\ell=1}^{L_w} G_\ell; \quad \Phi_{\ell p} : G_\ell \rightarrow \bigcup_{p=1}^{P_w} G_{\ell p}; \quad (1.5)$$

$$\Phi_{\ell p}^{-1} : \bigcup_{p=1}^{P_w} G_{\ell p} \rightarrow G_\ell; \quad \Phi_\ell^{-1} : \bigcup_{\ell=1}^{L_w} G_\ell \rightarrow G_L, \quad (1.6)$$

где G_L – множество преобразований для ПП в целом;
 G_ℓ – множество преобразований ℓ -го иерархического уровня;
 $G_{\ell p}$ – p -е преобразование ℓ -го иерархического уровня.

Общие требования к ПП должны быть рационально трансформированы в требования к ФЭ каждого иерархического уровня системы. Для этого необходимо иметь описание взаимосвязи ФЭ в форме (1.2), которое также является основой для реализации процедуры декомпозиции. Необходимость взаимосвязи ФЭ в форме (1.3) вытекает из технологии проектирования (построения) реальных сложных иерархических систем, начиная с разработки ФЭ нижнего уровня. Также формула (1.3) является основой для реализации процедуры агрегирования в системном анализе. Для системы управления соотношения (1.5) аналогичны формуле (1.2), а (1.6) – формуле (1.3) [82, с. 406–407].

Принципиальная особенность этой системной задачи в том, что она имеет одновременно свойства задачи выбора структуры системы управления и задачи системного анализа и минимизации рисков ситуации. Для определения факторов риска и управляющих воздействий связь вектора показателей качества с векторами параметров ФЭ, факторов риска и управляющих воздействий задается в общем виде:

$$\bar{K}_L = \tilde{F}_L(\bar{\chi}_L, \tilde{F}_{L-1}(\bar{\chi}_{L-1}, \tilde{F}_{L-2}(\dots, \tilde{F}_\ell(\bar{\chi}_\ell, \tilde{F}_{\ell-1}(\dots, \tilde{F}_1(\bar{\chi}_1)) \dots))))), \quad (1.7)$$

где $\bar{\chi}_\ell$ – кортеж для ℓ -го иерархического уровня сложной иерархической системы, определенный соотношением

$$\bar{\chi}_\ell = \langle \bar{X}_\ell, \bar{u}_\ell, \bar{p}_\ell \rangle, \quad (1.8)$$

где \bar{X}_ℓ – вектор параметров функциональных элементов ℓ -го иерархического уровня, $\ell = 1, \dots, L$;

\bar{u}_ℓ – вектор управления;

\bar{p}_ℓ – вектор факторов риска.

В общем случае для реальных сложных систем задача управления как для объекта в целом, так и для каждого ℓ -го уровня его иерархической структуры состоит в одновременном выполнении двух условий:

1) обеспечить стабильное значение некоторых показателей качества $\bar{K}_{1\ell}$ в фиксированный момент времени t_r заданного интервала $[0, T]$;

2) обеспечить изменение определенных показателей качества $\bar{K}_{2\ell}$ по заданным программам.

Первое условие в идеальном случае означает

$$\bar{K}_{1\ell} = \text{const} \quad \forall t_r \in [0, T]. \quad (1.9)$$

В реальных условиях допускается определенное изменение показателей при воздействии дестабилизирующих факторов риска на ℓ -м уровне, что определяется соотношением

$$\bar{K}_{1\ell}(\bar{u}_\ell^r, \bar{p}_\ell^r, t_r) \in [\bar{K}_{1\ell}^+, \bar{K}_{1\ell}^-] \quad \forall t_r \in [0, T], \ell = 1, \dots, L, \quad (1.10)$$

где $\bar{K}_{1\ell}^+$, $\bar{K}_{1\ell}^-$ – заданные верхняя и нижняя границы показателей качества соответственно.

Другое условие в идеальном случае означает, что каждая компонента вектора $\bar{K}_{2\ell}$ является заданной функцией времени. В реальных условиях из-за воздействия дестабилизирующих факторов риска и программных управляющих воздействий в момент времени $t_r \in [0, T]$ требуется обеспечить выполнение условия

$$\bar{K}_{2\ell}(\bar{u}_\ell^r, \bar{p}_\ell^r, t_r) \in [\bar{K}_{2\ell}^+(t_r), \bar{K}_{2\ell}^-(t_r)] \quad \forall t_r \in [0, T], \ell = 1, \dots, L, \quad (1.11)$$

где $\bar{K}_{2\ell}^+(t_r)$, $\bar{K}_{2\ell}^-(t_r)$ – заданные верхняя и нижняя границы, соответственно, допустимого интервала отклонения функции $\bar{K}_{2\ell}(t_r)$ в момент t_r .

Многоуровневая иерархическая система управления требует рационального распределения функций между уровнями иерархии, вследствие чего на каждом уровне появляется иерархическая структура целей и задач управления. Отсюда вытекает необходимость такой иерархической системы моделей, на основе которой можно создать рациональную структуру управления и обеспечить достижение поставленных целей. Такая модель должна отображать не только свойства объекта, но и его структуру [82, с. 69].

1.3 Аналитический обзор моделей и способов моделирования промышленного предприятия

Промышленные предприятия имеют явно различимую структуру ступенчатого разделения ресурсов, именуемую иерархией. Под иерархической системой, или иерархией, понимают систему, состоящую из взаимосвязанных подсистем, каждая из которых, в свою очередь, является иерархической по своей структуре.

Г. Саймон (H. Simon, 1972) выдвинул гипотезу, что сложные системы представляются как иерархические и путь эволюции сложных систем осуществляется через организацию устойчивых связей и образование подсистем на промежуточном уровне [167]. В иерархических системах Г. Саймон различает взаимодействия между подсистемами и взаимодействия внутри

подсистем, т. е. взаимодействия между частями этих подсистем.

Появление иерархической структуры управления обусловлено возрастающей сложностью управляемых систем, создающей большие трудности для централизованного управления. Поэтому появилась необходимость разделения всего процесса принятия решений на такое число уровней, чтобы решение задачи оптимизации на каждом из них было несложным. Но с возникновением многоуровневых иерархических систем управления появилась и новая задача согласования и координации решений, принимаемых на всех уровнях управления [6].

Общая задача оптимального управления иерархическими системами обычно ставится как статическая оптимизационная задача, т. к. рассматривается задача функционирования производства на достаточно больших интервалах времени (сутки и более), во время которых динамикой протекания процессов можно пренебречь. Высокочастотные возмущения материальных потоков, как предполагается в этом случае, обрабатываются системами автоматической стабилизации работы отдельных установок и диспетчерскими службами нижнего уровня. Решение отдельных локальных задач оптимизации для ряда подсистем и элементов без решения общей задачи оказывается мало или совершенно неэффективным, т. к. не определены переменные, согласующие режимы работ подсистем между собой, и не скоординированы критерии эффективности [6].

М. Модарес (M. Modarres, 2006) анализирует состояния статического и динамического моделирования сложных систем, представленных иерархическими структурами, и отмечает множество ошибок при реинжиниринге бизнес-процессов [332, с. 598–603]. Он также отмечает (со ссылкой на Г. Саймона [167]), что изменения отношений между подсистемами должны повлечь за собой изменения внутри подсистем и, наоборот, изменения внутри подсистем должны повлечь изменения на макроуровне. Стратегия изменений на предприятии должна проводиться в двух направлениях: административная реорганизация и реинжиниринг бизнес-процессов.

Дж. Форрестер (Jay W. Forrester, 1971) предлагает метод моделирования предприятия как динамической экономической системы с целью выработки усовершенствованных форм организации и общего руководства [195]. В своей модели Дж. Форрестер использует шесть взаимосвязанных потоков, которые отражают деятельность промышленного предприятия. Пять из них – это потоки материалов, заказов, денежных средств, оборудования и рабочей силы (трудовых ресурсов). Шестой – информационный поток, связывающий другие потоки в единую сеть. Динамическая структура модели представлена рядом резервуаров или уровней, связанных между собой управляемыми потоками. В основе динамического моделирования лежат: теория информационных систем с обратной связью, исследование процессов принятия решений, экспериментальное моделирование сложных систем, ЭВМ как средство имитации реальных процессов на их математи-

ческих моделях.

На основе системной динамики Г. Шульц (G. Zülch, 2006) представил четыре модели анализа решающих действий в иерархической структуре промышленного предприятия: с пассивным координатором, решающие действия на основе глобальной цели, локальной цели, решения на основе глобальной цели с учетом взаимодействия с другими координаторами, именуемыми агентами [393, с. 948].

С. А. Жданов [74] отмечает, что основные положения рыночной экономики изложены в работах, выполненных под редакцией В. Д. Горфинкеля, Е. М. Купрякова, И. Н. Герчиковой, В. П. Грузинова, А. С. Булатова, Е. И. Бородиной, М. И. Баканова, А. Д. Шеремета, Е. П. Голубкова, А. Я. Лившица, Г. Х. Попова, Б. В. Сребника, И. В. Липсица. В литературе по рыночной экономике недостаточно уделено внимания вопросам автоматизации получения и использования экономической информации, применению математических моделей и алгоритмов для системного анализа экономической ситуации, их комплексному использованию для обоснования и принятия решений по объему выпускаемой предприятием продукции, ценам, требуемой прибыли, необходимым финансовым ресурсам, ускорению сбыта, сбалансированному использованию оборудования.

На необходимость создания динамических моделей для прогнозирования развития экономических ситуаций, особенностей функционирования производства обращалось внимание в работах А. Г. Гранберга, А. А. Кугаенко, П. В. Авдулова, А. А. Спирина, В. С. Июзайтиса, Л. Л. Терехова, В. И. Терехина, Я. М. Уринсона. Этими авторами разработаны модели определения объема производства продукции предприятия, определения объема собственных и привлекаемых финансовых средств для обеспечения производства планируемых изделий, определения требуемой прибыли и цены для накопления планируемого объема собственных финансовых средств, для оценки влияния скачков в издержках на объемы производства продукции, ресурсного планирования объема выпускаемой продукции на промышленном предприятии, оптимальной загрузки производственных мощностей, выбора технологии изготовления заказной продукции, обеспечивающей выполнение заказа при минимальных затратах, прогнозирования спроса на товары длительного использования с помощью логистической функции [74].

С. В. Сильновой и Г. Р. Валеевым [173, с. 40–46] предложена модель реализации продукции в условиях рынка. Модель включает подмодели распределения ресурсов между видами выпускаемой продукции, производства продукции всех видов, формирования затрат на производство и сбыт, реализации продукции, расчета выручки и прибыли.

А. А. Григорьева, Ю. М. Осипов [58, с. 36–39] для решения задачи определения конкурентоспособности продукции промышленного предприятия предложили аддитивно-мультипликативную модель, модель на

основе метода порогов несравнимости, модель на основе теории нечетких множеств.

Математические методы и модели используются в планировании и управлении на металлургических предприятиях [53], при управлении предприятием и его развитием в условиях рыночной экономики на предприятиях легкой промышленности [134], при оптимизации производственных запасов в условиях перехода к рынку [208] и развитии модели запасов готовой продукции предприятия на складе при заключении договоров на изготовление продукции с субподрядчиками [277, с. 819–824], при управлении сложными иерархическими системами в строительной промышленности [330, с. 766–771], для оптимизации деятельности ремонтных предприятий [276, с. 857–861]. Отмечается большое внимание разработчиков моделей управлению строительством и финансовой сфере [284, с. 2044–2050].

С. М. Саад (S. M. Saad, 2003) и др. предложили методику распределенного моделирования распределенных производств с решением вопросов интеграции моделей различных предприятий для анализа их совместной деятельности [359, с. 1168–1173]. При этом использованы широко распространенные технологии DCOM и CORBA.

Л. Витман, К. Рамачандра, В. Кеткар (L. Whitman, K. Ramachandran, V. Ketkar, 2001) определили основные свойства действующей модели предприятия [384, с. 848–855]:

- *работоспособность*: модель должна допускать внесение изменений в соответствии с изменениями на предприятии с течением времени;
- *динамичность*: в соответствии с изменениями предприятия как системы модель должна оценивать степень изменений и причины изменений;
- *расширяемость*: модель должна поддерживать включение новых подсистем на предприятии;
- *декомпозиция*: модель поддерживает множество уровней детализации, на каждом уровне обеспечивается не только понимание системы, но и принятие решений по управлению системой;
- *совместимость с ключевыми показателями предприятия*: оценивая основные показатели предприятия, сама модель становится неотъемлемой частью предприятия;
- *использование фактических данных предприятия*: на входе модели должны быть фактические данные предприятия. Модель должна сопровождать предприятие, и предприятие должно сопровождать модель. Это обеспечивает реальность и правдоподобность модели.

Рассматривают *три категории* моделируемых процессов:

- 1) процессы, преобразующие внешние ограничения в набор внутренних указаний;
- 2) процессы, которые требуют ресурсов и готовят их для последующего использования;
- 3) процессы, которые используют ресурсы для получения результа-

тов производственной деятельности.

В модели предприятия могут быть представлены:

- информация (*Information*): определяет объекты, управляемые предприятием и правила управления их отношениями и взаимодействиями;
- деятельность (*Activity*): определяет функции, выполняемые предприятием (что делается);
- бизнес-процесс (*Process*): определяет упорядоченный во времени набор бизнес-процессов;
- ресурс (*Resource*): определяет ресурсы и возможности, управляемые предприятием;
- организация (*Organization*): представляет структуру организации и набор ограничений и управляющих правил внутри предприятия.

Тогда модель Q предприятия представляется формулой

$$Q = \langle A(S, D, E); P(S, D, E); I(S, D, E); R(S, D, E); O(S, D, E) \rangle, \quad (1.12)$$

- где A – деятельность;
 P – бизнес-процесс;
 I – информация;
 R – ресурс;
 O – организация;
 S – возможности модели;
 D – динамичность модели;
 E – использование модели.

Возможности S модели классифицируются следующим образом: 5 – объединение предприятий (все три категории процессов смоделированы для объединения); 4 – множество участков предприятия (все три категории процессов смоделированы для участков предприятия); 3 – предприятие (все три категории процессов смоделированы для предприятия); 2 – система (модели становятся все более активными в деятельности предприятия и используются для отдельных важных решений); 1 – начальный уровень (модели не используются активно в деятельности предприятия и применяются для решения отдельных небольших задач).

Динамичность D модели классифицируют аналогично возможностям S модели (1.12).

Использование E модели классифицируют следующим образом: 5 – оптимизация (набор моделей активно используется и поддерживается предприятием); 4 – управление (формальное использование модели и поддержка ее предприятием); 3 – описание (формальное использование модели или поддержка ее предприятием); 2 – одномоментно (модели используются нечасто, менее одного раза в год); 1 – не узаконены (модели не используются в активной деятельности предприятия и применяются для незначительных решений, впоследствии не поддерживаются предприятием).

В системе промышленного предприятия не следует усматривать только организационный аспект, целесообразнее на первый план выдвинуть *функциональный*. Это даст возможность вариантного использования системного анализа в зависимости от способа и цели исследования. Г. Ортнер [135] отмечает, что в качестве целевой функции может быть задано следующее уравнение производственно-экономической оценки системы:

$$E_0 - C_0 \rightarrow \max, \quad (1.13)$$

где E_0 – общий доход;

C_0 – общие затраты.

В преобразованном (в зависимости от иерархической организационной структуры системы) виде это уравнение представится следующим образом:

$$\sum_{i=1}^m \sum_{j=1}^n e_{ij} - \sum_{i=1}^m \sum_{j=1}^n c_{ij} \rightarrow \max, \quad (1.14)$$

где e_{ij} – доход от деятельности j -ступени i -й подсистемы;

c_{ij} – сумма затрат на j -ступени i -й подсистемы;

m – число подсистем;

n – число ступеней системы.

На примере взаимоисключающих целевых функций подсистем глобальной системы «Предприятие» следует указать возможность возникновения ошибок при оценке системы с точки зрения максимизации разницы между доходами и затратами при несоответствии между собой целей оптимизации подсистем и общей цели оптимизации системы в целом. Для *производства* целевая функция – максимальная эластичность оснащения производства, отвечающего требованиям рынка при заданном уровне инвестиций; для *складирования* целевая функция – минимальный уровень складских запасов по отношению к формируемому спросу (поступающим заказам).

Таким образом, разработанные модели промышленного предприятия позволяют исследовать различные аспекты его функционирования. Однако отсутствуют сведения об использовании комплексной модели предприятия, включающей модели различных аспектов его функционирования на различных ступенях иерархической структуры (с учетом целевых функций подсистем и глобальной системы). Такая ситуация может объясняться сложным характером взаимодействия подсистем предприятия, особенно в условиях неопределенности и риска современного рынка. В этих условиях решением проблемы исследования функционирования промышленного предприятия является использование методов имитационного моделирования [91, 110, 159].

1.4 Обзор методов и средств имитационного моделирования сложных систем

Наиболее полная классификация способов формализации и организации имитации приведена в [113, 114]. Это имитация активностями, событийное моделирование, имитация транзактами, агрегатами и процессами. Не существует универсального способа имитации. Каждый из способов имеет свои преимущества и недостатки и ориентирован на создание моделей определенного класса систем. При построении имитационных моделей производственно-экономической деятельности предприятия требуется высокий уровень детализации выполнения функциональных действий, при этом сама имитационная модель часто используется для поиска узких мест в системе. Метод имитационного моделирования на основе процессного способа формализации позволяет обеспечить очень важное в подобной ситуации сходство структуры модели и объекта исследования.

Основные подходы в имитационном моделировании по данным зарубежных источников – это системная динамика (СД), дискретно-событийное моделирование (ДС), агентное моделирование (АМ). Системная динамика и дискретно-событийное моделирование – традиционные устоявшиеся подходы, агентное моделирование – относительно новый подход. Область моделирования динамических систем, являясь инженерной дисциплиной, останется в стороне от рассмотрения. Математически системная динамика и динамические системы оперируют в основном с непрерывными во времени процессами, тогда как дискретно-событийное моделирование и агентное моделирование – с дискретными [367, с. 167–177; 381, с. 49–60; 309, с. 926–933; 388, с. 889–896].

Каждая имитационная модель представляет собой некоторую комбинацию составляющих: компоненты, переменные, параметры, функциональные зависимости, ограничения, целевые функции. Под компонентами понимают составные части, которые образуют систему. Параметрами являются величины, которые исследователь может выбирать произвольно. Функциональные зависимости описывают поведение переменных и параметров в пределах компонента. Ограничения представляют собой устанавливаемые пределы значений переменных или условия их изменений. Целевая функция – это точное отображение целей или задач системы и необходимых правил оценки их выполнения.

Чтобы построить имитационную модель сложной системы, должны быть предусмотрены:

- способы организации данных, обеспечивающие простое и эффективное моделирование;
- удобные средства формализации и воспроизведения динамических свойств моделируемой системы;
- возможности имитации стохастических систем, включающие в се-

бя процедуры моделирования и анализа случайных систем.

Применение специальных систем имитационного моделирования обладает следующими преимуществами:

- меньшие затраты времени на программирование;
- более эффективные методы выявления ошибок имитации;
- точность выражения понятий, характеризующих имитационные процессы;
- существование возможности для некоторой предметной области исследований заранее построить стандартные компоненты (библиотеки имитационных моделей);
- автоматическое формирование типов данных, соответствующих принятому способу имитации и необходимым в ходе имитационного эксперимента;
- удобство накопления и представления выходных данных;
- возможность учета субъективных предпочтений эксперта и его опыта в процессе принятия решений;
- возможность динамического анализа сценария развития.

В настоящее время известен ряд систем имитационного моделирования, обладающих функциональными возможностями для моделирования деятельности промышленного предприятия. Например, *Simkit* является пакетом моделирования с открытым кодом, написанным на Java. Основан на графовом представлении дискретных событий, компоненто-ориентированном моделировании, является платформу-независимым [288, с. 243–249]. Однако основным недостатком, ограничивающим его применение для моделирования сложного промышленного предприятия, является отсутствие средств коллективного управления процессом моделирования, что приводит к увеличению сроков проектирования и значительным ошибкам в проекте модели.

Flexsim – объектно-ориентированная среда для разработки, моделирования, исследования, визуализации и мониторинга динамических потоков. Использует C++ непосредственно или Flexs-скрипты. Вся мультипликация использует OpenGL. Результаты моделирования могут быть экспортированы во внешнее программное обеспечение через DDE, DLL, ODBC, SQL [337, с. 250–252]. Несмотря на значительные функциональные возможности, *Flexsim* ограничен исследованием непрерывных динамических процессов.

РДО (Ресурсы–Действия–Операции) – пакет имитационного моделирования, созданный в МГТУ им. Н. Э. Баумана; ориентирован на создание производственных моделей, обладает развитыми средствами компьютерной графики, применяется при моделировании сложных технологий и производств [70, с. 13]. Однако в нем моделирование функционирования промышленного предприятия ограничено отсутствием средств интеграции с комплексной информационной системой предприятия.

Pilgrim – пакет имитационного моделирования для создания дискретно-непрерывных моделей; поддерживается коллективное управление процессом моделирования, модели переносятся на любую другую плат-

форму при наличии компилятора C++. Средством построения моделей в системе *Pilgrim* является графический конструктор Gem (Generator models). Достоинства: ориентация на совместное моделирование материальных, информационных и «денежных» процессов; наличие развитой CASE-оболочки, позволяющей конструировать многоуровневые модели в режиме структурного системного анализа; наличие интерфейсов с базами данных [70, с. 13; 71]. Недостатки: исследование сложных имитационных моделей функционирования промышленного предприятия ограничивается отсутствием средств распределенных вычислений.

DESMO относится к системам распределенного имитационного моделирования, использует генетический алгоритм для оптимизации. Разработанные модели являются Java-программами, не зависят от используемой платформы. Эксперименты успешно проведены под Solaris, Linux, Windows NT/98/ME и MacOS [302, с. 508–514]. Недостатком является отсутствие средств автоматизации построения концептуальной модели исследуемого объекта, что является актуальным для комплексного моделирования подсистем в многоуровневой иерархической структуре промышленного предприятия.

AweSim включает встроенный язык моделирования Visual SLAM для сетей, дискретных и непрерывных систем. Сети моделируются в визуальном режиме. Другие системы могут быть смоделированы на Visual Basic, C или C++. В рассматриваемой системе моделирования имеются встроенные средства мультипликации, иллюстрирующие работу модели на разных уровнях иерархии (декомпозиции). *AweSim* интегрируется с приложениями Windows, использует реляционную базу данных, интегрируясь с Dbase, Access, FoxPro и Excel [341, с. 221–224]. Недостаток: ориентирован прежде всего на моделирование технических систем.

В Таганрогской научной школе «Высокопроизводительные проблемно-ориентированные вычислительные системы для структурно-процедурного моделирования и интеллектуального управления» под руководством д-ра техн. наук, профессора В. Ф. Гузика и д-ра техн. наук, профессора В. Е. Золотовского реализована программно-аппаратная система структурного моделирования *СимНП* на базе вычислительной сети и многопроцессорных акселераторов. Моделируемые объекты описываются при помощи иерархических структурных схем, состоящих из компонентов, которые задаются в форме математической нотации. Разработана специализированная форма многоуровневого представления моделируемых технических систем для автоматизации их обработки [154, с. 12]. Несмотря на использование современных информационных технологий в *СимНП*, его применение ограничено в основном техническими системами.

Другие наиболее известные системы имитационного моделирования и их основные характеристики представлены в таблице 1.1, взятой за основу из работы В. Б. Резникова [154].

Таблица 1.1 – Системы имитационного моделирования и их основные характеристики

Система имитационного моделирования	Визуальное программирование	Текстовое программирование	Нерывные модели	Дискретные модели	Генерация кода исполнения	Отображение графиков функций	Поддержка анимации	Интеграция с информационной системой	Ориентация на распределенные вычисления	Встроенные средства оптимизации	Коллективное управление моделированием
2D-Sim	Есть	Есть	Есть	Нет	Есть	Есть	Есть	Нет	Нет	Нет	Нет
ACSL	Нет	Есть	Есть	Нет	Есть	Есть	Нет	Нет	Нет	Нет	Нет
AnyLogic	Есть	Есть	Есть	Есть	Есть	Есть	Нет	Нет	Нет	Есть	Есть
Awesim	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Нет	Нет	Нет
BelSim2	Нет	Есть	Есть	Есть	Есть	Есть	Нет	Есть	Есть	Есть	Есть
Dymola	Есть	Есть	Есть	Нет	Нет	Есть	Есть	Нет	Нет	Нет	Нет
Dynast	Есть	Есть	Есть	Нет	Нет	Есть	Есть	Нет	Нет	Нет	Нет
Desmo	Нет	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Нет
Easy-5	Есть	Есть	Есть	Есть	Нет	Есть	Есть	Есть	Нет	Нет	Нет
Flexsim	Есть	Есть	Есть	Нет	Есть	Есть	Есть	Есть	Нет	Нет	Нет
LabView	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Нет	Есть	Нет
MICIC4	Нет	Есть	Нет	Есть	Есть	Есть	Есть	Нет	Нет	Нет	Нет
MVS	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Нет	Есть
Pilgrim	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Нет	Нет	Нет
Simulink	Есть	Нет	Есть	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет
Simkit	Есть	Есть	Нет	Есть	Есть	Есть	Нет	Нет	Нет	Нет	Нет
SYMBOLS-2000	Есть	Нет	Есть	Нет	Нет	Есть	Есть	Нет	Нет	Нет	Нет
VisSim	Есть	Нет	Есть	Есть	Нет	Есть	Нет	Нет	Нет	Есть	Нет
VTB	Есть	Нет	Есть	Нет	Нет	Есть	Есть	Есть	Нет	Есть	Нет
РДО (МВТУ)	Есть	Нет	Есть	Есть	Нет	Есть	Нет	Нет	Нет	Нет	Нет
СимНП	Есть	Есть	Есть	Есть	Есть	Есть	Есть	Нет	Есть	Нет	Нет

Таким образом, обзор методов и средств имитационного моделирования показывает, что известные системы имитационного моделирования по разным причинам не полностью отвечают требованиям моделирования функционирования промышленного предприятия в структуре комплексной информационной системы.

С. Онго (S. Onggo, 2006) и др. отмечают, что в будущем моделирование будет обладать свойствами модульности, повторного использования моделей, иерархичностью структуры; системы имитационного моделирования должны быть масштабируемыми, портативными, взаимодействующими между собой; модель должна быть удобной для использования и выполняться в распределенном режиме, через Интернет [340, с. 954–959].

Многообразие средств имитационного моделирования ставит перед исследователями задачу разработки стандартов для обеспечения взаимодействия программного обеспечения имитационного моделирования с другими программами, что позволит расширить область применения технологии моделирования [282, с. 2033–2043; 331, с. 2019–2026].

1.5 Обзор технологий имитационного моделирования

1.5.1 Формализация концептуальной модели. В моделировании широко используется правило «40x20x40», которое определяет трудоемкость этапов имитационного моделирования:

- а) **40 %** – описание проблемы моделирования, планирование проекта, описание системы, формализация концептуальной модели, предварительное планирование эксперимента и обработка входных данных;
- б) **20 %** – построение модели;
- в) **40 %** – выполнение верификации и валидации модели, проектирование и проведение экспериментов, анализ, интерпретация, реализация и документирование [306, с. 801].

Признается актуальной задача автоматизации построения концептуальной модели [392, с. 809–814]. С. Робинсон (S. Robinson, 2006) указывает, что один из путей ее решения – использование программного обеспечения для инженерного проектирования [355, с. 792–800].

Определены требования к технологии автоматизации построения концептуальной модели, которая должна:

- обеспечивать детальное описание исследуемой системы;
- быть понятной неспециалистам по моделированию;
- обладать хорошими семантическими возможностями для эффективного использования на следующих этапах проектирования модели;
- обладать хорошими возможностями визуализации;
- поддерживать командный метод проектирования модели.

При этом возможными вариантами ее использования являются следующие информационные технологии: IDEF (Integrated Definition for Func-

tional Modeling), UML (Unified Modeling Language) – диаграммы состояний и диаграммы деятельности, сети Петри, ACDs (Activity Cycle Diagrams), RADs (Role Activity Diagrams), IEM (Integrated Enterprise Modelling), EDPC (Event Driven Process Chains), DEVS (Discrete Event System Specification) [306, с. 802–808].

Средствами решения задач в методологиях структурного анализа являются:

- DFD (Data Flow Diagrams) – диаграммы потоков данных совместно со словарями данных и спецификациями процессов;
- ERD (Entity-Relationship Diagrams) – диаграммы «сущность-связь»;
- STD (State Transition Diagrams) – диаграммы переходов состояний.

Диаграммы потоков данных (DFD) являются основным средством моделирования функциональных требований системы. Эти требования разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных. Главная цель таких средств – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами [87].

С. Робинсон (S. Robinson, 2006) подчеркивает важность этапа концептуального моделирования и определяет четыре главных требования к концептуальной модели: точность, достоверность, полезность и выполнимость [355, с. 792–800].

1.5.2 Построение модели. Для быстрого налаживания эффективного процесса управления программированием сложных систем, к которым относятся и имитационные модели промышленного предприятия, подразделением Пентагона SPMN разработан подход СВР (Critical Best Practices). В самом общем виде подход СВР предлагает определять количественные параметры проекта (дата, бюджет, объем), создать быстро реализуемую стратегию выполнения проекта, количественно оценивать продвижение к цели, количественно оценивать (измерять) активность разработки [370].

Существуют девять правил, которыми следует руководствоваться для успешной реализации сложного программного проекта:

- 1) любой проект по разработке программного обеспечения (ПО) – рискованный, поэтому первое правило – формализованное управление проектом;
- 2) соглашение о пользовательских, внутренних (межмодульных) и внешних интерфейсах для стыковки с другими компонентами и приложениями;
- 3) формальные проверки проекта;
- 4) управление проектом на основе метрик;
- 5) качество продукта должно контролироваться на детальном уровне;
- 6) информация о ходе проекта должна быть общедоступной;
- 7) чтобы добиться высокого качества, надо отслеживать причины

возникновения ошибок;

8) конфигурационное управление;

9) управление персоналом.

Поддержка образовательных программ, создание условий для профессионального роста персонала, предпочтение индивидуальной работе с ним, создание организационной структуры, пользующейся поддержкой персонала – элементы эффективной практики антикризисного управления персоналом [176, с. 81–84].

Особенности ПО имитационных моделей сложных объектов приводят к необходимости анализа жизненного цикла модели. С. Робинсон (S. Robinson, 2001) предлагает три стадии жизненного цикла модели:

1) концептуализация, включающая постановку задачи, определение целей моделирования, концептуальное моделирование;

2) разработка модели, включающая кодирование модели, верификацию, валидацию завершенной модели, калибровку;

3) использование, включающее изучение модели, идентификацию основных результатов, разработку рекомендаций.

Допускаются обратные связи между стадиями и непрерывная валидация по ходу реализации проекта [363, с. 863–871].

А. М. Лоу (A. M. Law, 2005) предложил семь этапов для построения адекватной модели [321, с. 24–32]: формулировка проблемы; обобщение информации об объекте и построение концептуальной модели; проверка валидации концептуальной модели; программирование модели; проверка программной модели; проектирование, проведение и анализ результатов эксперимента; документирование и использование результатов моделирования.

Наибольшее значение для развития программного обеспечения имитационного моделирования в последнее время имеют модели жизненного цикла имитационной модели Крейцера (Kreutzer), Бэлси-Ненса (Balci-Nance), Саргента (Sargent) [336; 364, с. 130–143].

1.5.3 Выполнение верификации и валидации модели. Выбор методов верификации и валидации модели определяется следующей классификацией Клейнена (Kleijnen):

– реальные данные недоступны;

– имеются реальные выходные данные;

– имеются реальные входные и выходные данные [292, с. 911].

При недоступных реальных данных, что характерно для промышленных предприятий, верификация и валидация могут быть проведены на основе процедуры Клейнена: строится теоретическая матрица отношений между входами и выходами, по результатам прогона имитационной модели оценивают значения входов и выходов. Если имеется разница в характере рассматриваемых отношений, то оценивают коррелированность вхо-

дов. При ее наличии пересматривают входные и выходные переменные. При отсутствии коррелированности входов уточняют матрицу отношений [292, с. 914].

При наличии альтернативных моделей и доступе к данным реальной системы становится возможным применение нейронных сетей. В данном случае достоинство их в том, что они исследуют одновременно все альтернативные модели. Модель с самым высоким согласованием с реальными данными принимается в качестве действующей [328, с. 905–910].

А. М. Лоу (A. M. Law, 2004) указывает на следующие ошибки, допускаемые при анализе результатов моделирования: анализ результатов моделирования по данным одного прогона модели; отказ от оценки переходного режима в модели при исследовании установившихся процессов; отказ от определения статистической точности результатов моделирования с помощью доверительного интервала [323, с. 67–72].

Учитывая важность построения адекватной имитационной модели, используются в комплексе следующие из разработанных технологий валидации:

- *анимация* – иллюстрация поведения объектов модели во времени;
- *сравнение с другими моделями* – простые имитационные модели сравниваются с аналитическими моделями или уже с проверенными моделями;
- *вырожденные тесты* – подбор специальных параметров значений входных параметров и параметров внешней среды для исследования вырожденного поведения модели;
- *валидация событий* – проверка наличия случайных событий в модели с реальными данными исследуемого объекта;
- *чрезвычайные тесты* – структура модели должна отражать правильное поведение объекта при чрезвычайных условиях системы;
- *экспертная валидация* – выяснение о разумности поведения модели у специалистов;
- *валидация со статистическими данными* – часть данных используется для построения модели и часть данных используется для валидации;
- *исторические методы*: *рационализм* предполагает, что каждый знает, что основные положения модели верны и логические вычисления используются для развития модели; *эмпиризм* требует, чтобы предположение и результат были практически подтверждены; *положительная экономика* требует только, чтобы модель независимо от структуры обладала прогнозирующими свойствами;
- *внутренняя валидация* – несколько прогонов модели со случайными параметрами, чтобы определить стохастическую изменчивость в модели. Большое количество изменчивости может поставить под сомнение достоверность модели или проводимых исследований;
- *многоступенчатая валидация* использует три исторических мето-

да: разработка модели на предположениях теории, наблюдениях и общих знаниях; проверка предположений модели практически, где это возможно; сравнение отношений вход-выход модели с реальными данными системы;

– *операционная графика* – величины измерений различных характеристик, процент занятости обслуживающего устройства и др. показывают графически, и динамическое поведение таких указателей характеризует поведение модели;

– *анализ чувствительности* (изменчивость параметров) состоит в изменении входных и внутренних параметров модели, чтобы определить эффект в поведении или изменении откликов модели. Наиболее чувствительные параметры должны быть более точными до исследования модели;

– *прогнозирующая валидация* – сравнение результатов модели с итогами экспериментов на реальном объекте;

– *трассировка* – проверка правильности логического поведения объектов модели;

– *тест Тьюринга* – анализ данных модели и реального объекта экспертами. При правильной модели эти данные экспертами не должны различаться [365, с. 37–48].

Дж. С. Карсон (J. S. Carson II, 2002) предлагает *стресс-тест*, т. е. прогон модели по широкому диапазону входных данных, а также проверку модели для редких случаев и чрезвычайных ситуаций, чтобы оценить ее поведение, идентификацию выходных данных, сообщающих об ошибке модели или ее подозрительном поведении, целесообразность использования мультипликации на микроуровне и коротких интервалах времени [290, с. 52–58].

Тестирование программного обеспечения может достигать 50 % стоимости всего проекта. Полагают, что особенности, присущие имитационному моделированию, делают тестирование еще более сложной проблемой. Общая проблема тестирования – обнаружение непредвиденных взаимодействий между компонентами. Это же относится и к ИМ. Одна из проблем тестирования – снижение стоимости. Поэтому важно провести полное тестирование наименьшим числом тестов. Для этого тесты должны *охватывать* различные характеристики модели. *Идея охвата* предполагает две цели: выполнить проверку, чтобы все характеристики программного кода были протестированы, и избегать двойных тестов, которые проверяют одну ту же характеристику. Эти вопросы охвата особенно важны для моделирования, поскольку выполнение обычно состоит из последовательности выполнения компонентов, чей порядок определен разработчиком модели, реализацией имитационной модели и случайными числами [342, с. 641–647].

Основные особенности, отличающие имитационное моделирование от разработки программного обеспечения, связаны с формулировкой проблемы, уровнем детализации, верификацией, выбором закона распределе-

ния входных величин, проектированием и анализом результатов натуральных экспериментов с моделью, руководством проекта [294, с. 1633–1640].

Для достаточно сложных программных продуктов при современных технологиях невозможно гарантировать 100-процентной точности. Невозможно протестировать модель на 100 %, поэтому технологии тестирования направлены на построение доверия к модели. При этом полагают, что *верификация* имеет дело с преобразовательной точностью; *валидация* имеет дело с поведенческой или представительной точностью; *аккредитация* определена как официальное удостоверение, что модель или совокупность моделей и технологий моделирования приемлемы при использовании для достижения конкретной цели [281, с. 653–663].

1.6 Обзор применения методов оптимизации в имитационном моделировании

1.6.1 Обзор пакетов оптимизации в имитационном моделировании.

В последние годы обращается внимание на решение оптимизационных задач в имитационном моделировании. На основании аналитических работ А. М. Лоу (A. M. Law, 2002) [322, с. 41–44], Дж. Эприла (J. April, 2003) [280, с. 71–78], М. С. Фу (M. C. Fu, 2005) [301, с. 83–95] и др. построена таблица 1.2, в которой указаны наиболее известные оптимизационные пакеты, применяемые в имитационном моделировании.

Таблица 1.2 – Основные характеристики оптимизационных пакетов имитационного моделирования

Пакет оптимизации	Поставщик (URL)	Используемые эвристические процедуры	Система имитационного моделирования
AutoStat	AutoSimulations, Inc. (www.autosim.com)	Эволюционные стратегии, генетические алгоритмы	AutoMod
Evolutionary Optimizer	ImagineThat, Inc. (www.imaginethatinc.com)	Эволюционные стратегии, генетические алгоритмы	Extend
OptQuest	OptTek Systems, Inc. (www.opttek.com)	Случайный поиск, поиск с запретами, нейронные сети	AnyLogic; Arena; Crystal Ball; CSIM19; Enterprise Dynamics; Micro Saint; ProModel; Quest; SimFlex; SIMPROCESS; SIMUL8; TERAS
RISKOptimizer	Palisade Corp. (www.palisade.com)	Генетические алгоритмы	@RISK
WITNESS Optimizer	Lanner Group, Inc. (www.lanner.com/corporate)	Моделируемый отжиг, поиск с запретами	WITNESS

Как отмечает Н. Н. Венцов [46], развитие эволюционных подходов к решению оптимизационных задач в значительной мере определяется работами М. Л. Цетлина, Л. Фогеля (L. J. Fogel), Г. Шефеля (H. Schwefel), Дж. Холланда (John H. Holland), К. К. Морозова, В. М. Курейчика и др.

Преимущества эволюционных подходов к решению оптимизационных задач:

- повышенное быстродействие;
- высокая надежность и помехоустойчивость;
- высокая робастность, т. е. малая чувствительность к нерегулярностям поведения целевой функции;
- сравнительно простая внутренняя реализация;
- малая чувствительность к росту размерности задачи оптимизации;
- возможность естественного ввода в процесс поиска операции обучения и самообучения;
- легкое построение новых алгоритмов, реализующих различные эвристические процедуры адаптации в рамках известных схем случайного поиска.

Недостатки их использования:

- отсутствие гарантий качества получаемых решений;
- невозможность проверки на всех мыслимых классах задач;
- необходимость предварительной настройки параметров алгоритма для его эффективной работы.

1.6.2 Краткий обзор основных алгоритмов случайного поиска, используемых при оптимизации в имитационном моделировании. В соответствии с (1.1) задача поиска наилучшего решения в многоуровневой иерархической системе сводится к рациональному компромиссу заданных целей. Такая постановка задачи приводит к использованию универсальных механизмов оптимизации, к которым относятся алгоритмы случайного поиска, позволяющие найти рациональное решение для нескольких подсистем разной природы с антагонистическими целевыми функциями.

Наиболее простым и распространенным алгоритмом случайного поиска является *метод Монте-Карло*, иногда именуемый *грубым случайным поиском*. Реализация метода Монте-Карло может быть представлена следующим образом:

$$G \rightarrow U \rightarrow X \rightarrow Y \rightarrow \hat{\theta}.$$

Каждая из четырех стрелок – детерминированная функция. Генератор случайных чисел G используется для получения множества случайных чисел U , которые формируют множество входных данных X , используемых для вычисления множества выходных данных Y , по которым определяется статистическая оценка $\hat{\theta}$ [366, с. 39–46].

Это самый простой и в то же время самый известный алгоритм случайного поиска, состоящий из равномерного случайного «бросания» точек в пространство поиска. Основное его достоинство – простота, и в теории глобальной оптимизации этот алгоритм применяется в основном в качестве эталона при теоретическом или численном сравнении алгоритмов и в качестве составной части некоторых алгоритмов глобального случайного поиска [296, с. 774–782; 375, с. 394–400].

Например, А. Шапиро (A. Shapiro, 2001) предлагает делать выборку и работать над ней с использованием детерминированных методов оптимизации [8, 23]. Затем множество подобных выборок усредняют, получая оптимальное решение [368, с. 428–431].

Генетические алгоритмы (ГА) (Genetic Algorithms) можно отнести к группе адаптивных методов, которые используются для решения проблем поиска и оптимизации. Они сочетают в себе элементы детерминистического и стохастического подходов. В силу этого генетические алгоритмы относятся не только исключительно к методам случайного поиска. Более того, они успешно применяются в комбинациях с аналитическими методами или другими алгоритмами поиска и оптимизации.

Принцип работы генетических алгоритмов основан на моделировании некоторых механизмов популяционной генетики: манипулирование хромосомным набором при формировании генотипа новой биологической особи путем наследования участков хромосомных наборов родителей (кроссинговер), случайном изменении генотипа, известном в природе как мутация. Другим важным механизмом, заимствованным у природы, является процедура естественного отбора, направленная на улучшение от поколения к поколению приспособленности членов популяции путем большей способности к «выживанию» особей, обладающих определенными признаками.

В процессе поиска анализируются сразу несколько ветвей эволюции. Применяя так называемую «функцию приспособленности», определяющую качество найденного решения проблемы, ГА «выращивают» новые популяции объектов, генная структура которых более приспособлена к текущей ситуации. Таким образом, генетическая модель имитирует эволюцию приспособления через механизмы изменчивости объектов [51; 100; 130, с. 2–5; 351, с. 231–250; 352, с. 151–196].

Сфера применения ГА очень широка: от автоматизированного проектирования, решения комбинаторных задач и проектирования нейронных сетей до применения в экспертных и обучающихся системах. Приводятся результаты совместного применения имитационного моделирования, генетических алгоритмов и целочисленного программирования для определения оптимальной сети поставки сырья и материалов [380, с. 1268–1275]; применения имитационного моделирования и генетического алгоритма для выбора поставщика с учетом рыночного спроса, параметров хранилища запасов, характеристик поставщика и транспортных потоков [295, с. 1260–1267].

Моделируемый отжиг (Simulated Annealing). Этот метод был разработан С. Киркпатриком (S. Kirkpatrick, 1983). Как в основе генетических алгоритмов лежит идея биологической эволюции, так моделируемый отжиг появился в виде аналогии термодинамического процесса нагревания и медленного охлаждения субстанции для получения кристаллической структуры.

Это существенно усовершенствованная версия метода наискорейшего спуска. Начиная со случайно выбранной точки в пространстве поиска, делается шаг в случайном направлении. Если этот шаг приводит в точку с более низким уровнем значения функции оптимизации, то он принимается. Если же он приводит в точку с большим значением функции оптимизации, то он принимается с вероятностью $P(t)$, где t – время. Функция $P(t)$ сначала близка к единице, но затем постепенно уменьшается до нуля – по аналогии с охлаждением твердого тела. Таким образом, в начале процесса моделирования принимаются любые ходы, но, когда «температура» падает, вероятность совершения негативных шагов уменьшается [317, с. 671–680].

Негативные шаги иногда необходимы в том случае, когда нужно избежать локального оптимума, но принятие слишком многих негативных шагов может увести в сторону от глобального оптимума. На сегодняшний день этот метод является областью активных исследований (быстрый «переотжиг», параллельный отжиг).

Поиск с запретами (Tabu search) является еще одним стохастическим методом глобального поиска. Он основан на аналогии с человеческим поведением, т. е. на присутствии в поведенческой схеме человека случайных элементов, которые означают, что в одной и той же ситуации человек может поведи себя по-разному. Сохранение листа запретов, в котором, например, может храниться набор уже рассмотренных точек пространства поиска, является одной из основных характеристик данного подхода.

Алгоритм заключается в выборе случайной точки в пространстве поиска, рассмотрении точек из окрестностей данной точки и, при достижении определенного критерия, выборе новой точки в другом регионе поиска, который еще не рассматривался [307, с. 417–421].

Все рассмотренные методы обладают общими свойствами: они все имеют адаптивные, итеративные, стохастические алгоритмы. На каждом шаге работы все они оценивают значение функции качества, и для каждого случая можно доказать его сходимости к глобальному оптимуму.

Д. Волперт, В. Макреди (D. H. Wolpert, W. G. Macready, 1996) опубликовали NFL (No Free Lunch) теорему, в которой утверждается, что для всех алгоритмов, ищущих экстремум функции качества, производительность одинакова, если усреднить результаты по всевозможным функциям качества [385, с. 67–82; 391, с. 783–793]. Практическое значение этой теоремы состоит в том, что не существует панацеи на все случаи жизни, несомненный успех какого-либо оптимизационного метода в определенной области знаний не гарантирует такого же успеха в другой области. Это озна-

чает, что для каждой специфической области необходимо проводить такие исследования и отыскивать тот оптимизационный метод, которые подходят ей более всего.

Эволюционные стратегии (Evolution Strategies). Тоже эволюционный подход, использующий при конструировании структуры индивида адаптивный уровень мутации, что позволяет приспособиться к любой переменной модели. Применяется в области комбинаторики, экспертных системах и при машинном обучении [307, с. 417–421].

Для решения задач глобальной оптимизации и построения оптимизатора при имитационном моделировании разработан алгоритм SPSA (Simultaneous-Perturbation Stochastic Approximation), показывающий лучшие результаты в сравнении с генетическим алгоритмом и моделируемым отжигом, которые часто рекомендуются для глобальной оптимизации. Алгоритм SPSA основан на *стохастической аппроксимации SA (stochastic approximation)* с введением дополнительного шума. Использование SPSA часто приводит к рекурсии, что намного более экономично в терминах функции потерь, чем стандартная версия SA [329, с. 307–312].

Р. Аль-Аомар (R. Al-Aomar, 2002) предложил *метод робастной многокритериальной оптимизации* на основе имитационного моделирования, включающий имитационную модель, генетический алгоритм, метод энтропии [96, с. 24–25] и модуль робастности, дающий информационные оценки параметрам и откликам модели. Продукция производства может быть сделана робастной к изменению материалов, производственного процесса, условий применения. Производственные процессы также могут быть робастными к изменению материалов, окружающей среды, параметров машин. Метод направлен на то, чтобы многоуровневые производства/бизнес-процессы сделать робастными к изменению системно-уровневых факторов, таких как нормы поставок сырья и материалов, время обработки/обслуживания, отказы оборудования.

Для параметров и откликов модели введены три информационные оценки на основе отношения сигнал/шум. В *аддитивной функции полезности* при многокритериальной оптимизации метод энтропии используется для оценки веса важности отдельного критерия. Критерии функции полезности должны быть взаимно независимыми, чтобы полезность одного критерия не оказывала влияние на полезность других критериев. При взаимной зависимости отмечаются ошибочные результаты по итогам их исследований. Чтобы исключить подобную погрешность в принятом методе, функция полезности имеет динамическую структуру. Структура изменяется на каждом шаге итерации ГА, изменяя величины весов. Чтобы функция полезности могла оценивать различные альтернативы, проведено нормирование критериев [275, с. 1931–1939].

Известен метод принятия решений, основанный на *визуальном интерактивном имитационном моделировании* с применением нейронной сети

и экспертной системы. Метод реализуется следующими основными стадиями: пониманием процесса принятия решения; сбором данных; определением стратегий принятия решений экспертами; вычислением результатов стратегии принятия решения; поиском усовершенствований. Апробирован метод на заводе по сборке двигателей [356, с. 913–920].

В дополнение к хорошей модели требуется технология сжатия данных моделирования для принятия решения. Приводится алгоритм последовательного отбора с памятью *SSM (Sequential Selection with Memory)* [347, с. 401–407].

Одна из трудностей использования методов оптимизации на основе имитационного моделирования состоит в том, что методы достаточно сложны и требуют значительных знаний об объекте со стороны пользователя. Предложена структура оптимизации, которая позволяет эффективно решать сложные проблемы комбинаторного типа на основе имитационного моделирования в автоматизированном режиме. Для поиска оптимального решения, когда целевая функция не может быть задана аналитически, приводится алгоритм, в котором область решений разбивается на подмножества. Поиск проводится в наиболее многообещающих областях; выполняется случайная выборка из каждой подобласти для оценки всей области в целом; улучшаются оценки каждого подмножества, которые используются как отправные точки для поиска лучшего решения в пределах исследуемой области [339, с. 300–306].

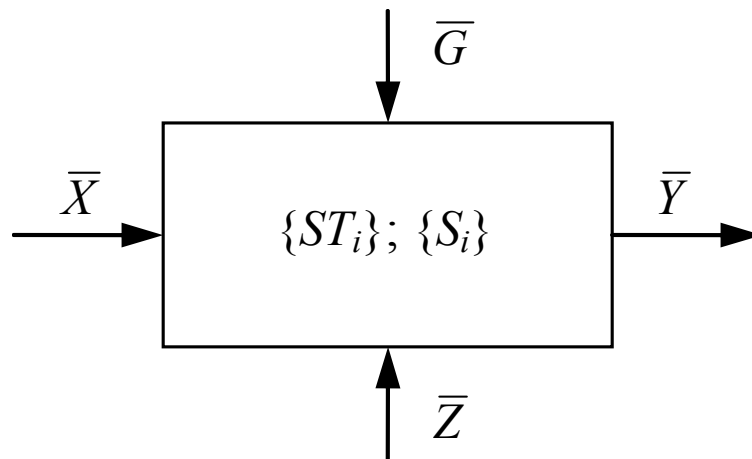
В [297, с. 467–475] предлагаются новые подходы к моделированию связанных между собой процессов в многоуровневых системах. Отмечается, что решение такой задачи представляется чрезвычайно сложным, хотя многие предметные области извлекли бы выгоду. Рассмотрен метод решения задачи, состоящей из нескольких представлений описания. При этом находится решение для каждого из уровней, целевая функция представлена линейным уравнением с весовыми коэффициентами.

Установлены шесть основных областей, в которых ожидается развитие методов оптимизации при имитационном моделировании: оптимизаторы, используемые в системах имитационного моделирования; интерфейсы между оптимизатором и пользователем и между оптимизатором и имитационной моделью; формализация проблемы, связанная с конструированием целевой функции и ограничений; классификация проблемы оптимизации для выбора метода оптимизации; стратегия и тактика использования оптимизации в имитационном моделировании для эффективного использования вычислительных средств (выбор методов оптимизации для класса проблем, способа отображения показателей, способа имитационного моделирования, выбор метода предварительной обработки данных, многочисленные сравнивающие тесты и др.); интеллектуализация (развитые логические возможности) оптимизатора для выбора стратегии и тактики использования различных методов [285, с. 1466–1469].

2 Метод поиска рациональных решений в ERP-системе управления ресурсами промышленного предприятия

2.1 Формальное описание многоуровневой иерархической системы

В практике имитационного моделирования [114, с. 121] для построения концептуальной модели объекта исследования используют, как правило, графическое описание, представленное на рисунке 2.1. Однако для формализации промышленного предприятия как многоуровневой иерархической системы такое представление является недостаточным.



\bar{X} – вектор управляемых параметров моделирования; \bar{G} – вектор задаваемых параметров; \bar{Z} – вектор неуправляемых параметров; \bar{Y} – вектор выходных параметров (откликов), входящих в состав функции качества, $\bar{Y} = \Phi(\bar{X}; \bar{G}; \bar{Z})$; $\{ST_i\}$ – множество статистик моделирования; $\{S_i\}$ – множество состояний объекта моделирования

Рисунок 2.1 – Графическое представление концептуальной модели

В соответствии с формализацией сложной системы, предложенной в [338, с. 480–487], любая многоуровневая система на ℓ -м уровне S^ℓ описывается следующей символьной конструкцией:

$$S^\ell \leftrightarrow \{w, S_0, \sigma\}^\ell, \quad (2.1)$$

где w^ℓ – функциональные действия элементов системы;
 σ^ℓ – структура, конструкция системы;
 S_0^ℓ – координатор на ℓ -м уровне;
 ℓ – индекс уровня, $\ell \in L$.

Для системы S^ℓ с подсистемами ℓ -го уровня функциональные действия и структура представлены множествами:

$$w^\ell \leftrightarrow \{\tilde{w}, S_0\}^\ell; \quad (2.2)$$

$$\sigma^\ell \leftrightarrow \{S_0, \tilde{\sigma}\}^\ell. \quad (2.3)$$

Координирующий элемент S_0^ℓ определяется внутренней ${}_0S^\ell$ и окружающей средой ${}_\varepsilon S^\ell$. При этом для ${}_0S^\ell$ определены функциональные действия и структура ${}_{0\pi}S^\ell$ внутриуровневых взаимодействий в окружающей среде ${}_\varepsilon S^\ell$ и функциональные действия и структура ${}_{\pi\varepsilon}S^\ell$ межуровневых взаимодействий в окружающей среде ${}_\varepsilon S^\ell$:

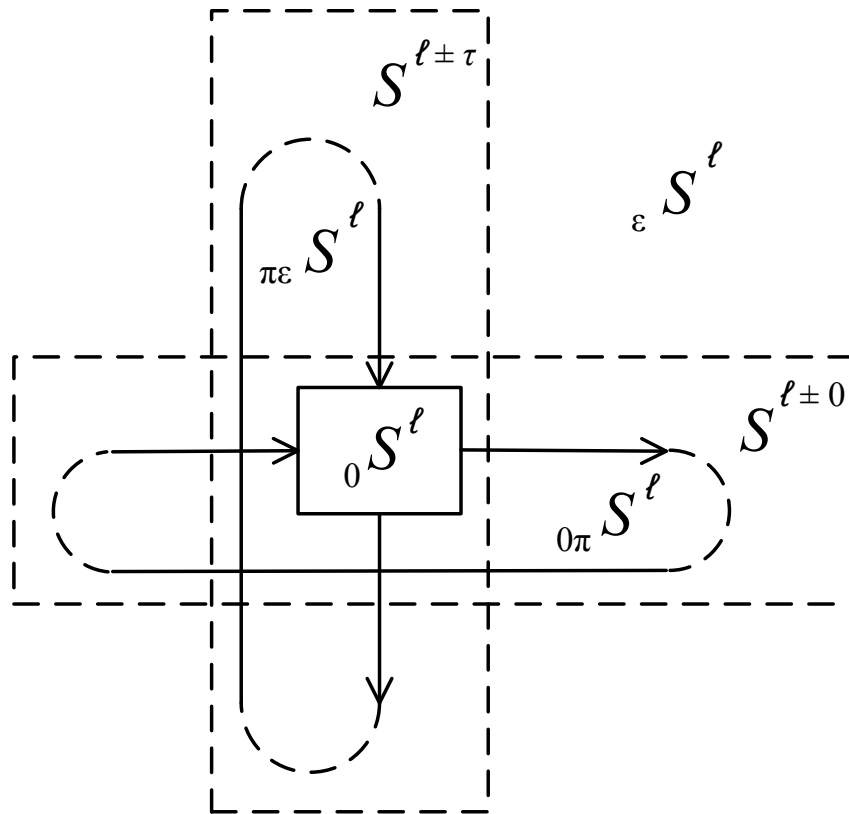
$$\{{}_0S, {}_{0\pi}S\}^\ell \leftrightarrow S^{\ell \pm 0}; \quad (2.4)$$

$$\{{}_{\pi\varepsilon}S, {}_\varepsilon S\}^\ell \leftrightarrow S^{\ell \pm \tau}, \quad (2.5)$$

где $S^{\ell \pm 0}$ – система внутриуровневых отношений;

$S^{\ell \pm \tau}$ – система межуровневых отношений;

τ – индекс уровня многоуровневой системы (рисунок 2.2) [238, с. 62–63].



$S^{\ell \pm 0}$ – система внутриуровневых отношений; ${}_0S^\ell$ – внутренняя среда; ${}_{0\pi}S^\ell$ – структура внутриуровневых взаимодействий в окружающей среде ${}_\varepsilon S^\ell$; $S^{\ell \pm \tau}$ – система межуровневых отношений; ${}_{\pi\varepsilon}S^\ell$ – структура межуровневых взаимодействий в окружающей среде ${}_\varepsilon S^\ell$

Рисунок 2.2 – Формализация многоуровневой системы

В системе на рисунке 2.2 функциональные действия и структура на ℓ -уровне представляются взаимодействующими с окружающими уровнями:

$$\tilde{w}^\ell \leftrightarrow \{w^{\ell\pm\tau}, {}_w\gamma \mid \tau \in L\}, \quad (2.6)$$

где $w^{\ell\pm\tau}$ с учетом системных действий на $\ell\pm\tau$ - уровне и отношений ${}_w\gamma^\ell$ представлены в виде

$$\begin{aligned} w^{\ell\pm\tau} &\leftrightarrow \{\tilde{w}^{\ell\pm 0}, \tilde{w}^{\ell\pm\tau}, {}_w\gamma^\ell \mid \tau \in L, \tau \neq 0\} \leftrightarrow \\ &\leftrightarrow \{\{{}_0w, {}_{0\pi}w, {}_w\gamma_0\}^{\ell\pm 0}, \{\pi\varepsilon w, {}_\varepsilon w, {}_w\gamma_\varepsilon\}^{\ell\pm\tau}, {}_w\gamma^\ell \mid \tau \in L, \tau \neq 0\}. \end{aligned} \quad (2.7)$$

Отношения ${}_w\gamma^\ell$ определяются множествами состояний ${}_kC^\ell$, входов ${}_kX^\ell$ и выходов ${}_kY^\ell$:

$${}_w\gamma^\ell \leftrightarrow \{\{X, C, Y\}_k \mid k \in {}_kL, {}_kL \leftrightarrow \{0, 0\pi, \pi\varepsilon, \varepsilon\}\}^\ell. \quad (2.8)$$

С учетом системных (внутриуровневых и межуровневых) взаимодействий динамика функциональных действий представлена следующими характеристическими функциями:

$${}_k w^\ell \leftrightarrow {}_k(\tilde{\rho}, \tilde{\delta})^\ell \mid k \in {}_kL, {}_kL \leftrightarrow \{0, 0\pi, \pi\varepsilon, \varepsilon\}. \quad (2.9)$$

Функция выходов

$${}_k \tilde{\rho}^\ell \leftrightarrow {}_k \{\rho_t : C_t \times X_t \rightarrow Y_t \wedge t \in T\}^\ell. \quad (2.10)$$

Функция переходов

$${}_k \tilde{\delta}^\ell \leftrightarrow {}_k \{\delta_{t'} : C_t \times X_t \rightarrow C_{t'} \wedge t, t' \in T \wedge t' > t\}^\ell. \quad (2.11)$$

В [338, с. 480–487] предлагается формализация структуры элементов ℓ -го уровня с учетом структурных отношений γ_σ^ℓ , позволяющих оценивать эффективность структуры системы:

$$\tilde{\sigma}^\ell \leftrightarrow \{\sigma^{\ell\pm\tau}, \gamma_\sigma \mid \tau \in L\}. \quad (2.12)$$

Аналогично (2.6) $\sigma^{\ell\pm\tau}$ представляются в виде

$$\begin{aligned} \sigma^{\ell\pm\tau} &\leftrightarrow \{\tilde{\sigma}^{\ell\pm 0}, \tilde{\sigma}^{\ell\pm\tau}, \gamma_\sigma^\ell \mid \tau \in L, \tau \neq 0\} \leftrightarrow \\ &\leftrightarrow \{\{{}_{0\sigma}w_i \mid i \in I, {}_{0\sigma}\gamma_\sigma\}^{\ell\pm 0}, \{\sigma_\varepsilon w_i \mid i \in I, {}_\varepsilon\gamma_\sigma\}^{\ell\pm\tau}, \gamma_\sigma^\ell \mid \tau \in L, \tau \neq 0\}. \end{aligned} \quad (2.13)$$

Структурные отношения могут быть представлены элементами матрицы для внутривуровневых отношений:

$$[{}_0\gamma_\sigma^{\ell\pm 0}] \leftrightarrow \begin{cases} 1, |({}_0\sigma w_i, {}_0\sigma w_j) \in {}_0\gamma_\sigma^{\ell\pm 0}, \forall i, j \in I; \\ 0, |({}_0\sigma w_i, {}_0\sigma w_j) \notin {}_0\gamma_\sigma^{\ell\pm 0}, \forall i, j \in I, \end{cases} \quad (2.14)$$

и аналогично для межуровневых отношений:

$$[{}_\varepsilon\gamma_\sigma^{\ell\pm\tau}] \leftrightarrow \begin{cases} 1, |({}_\varepsilon\sigma w_i, {}_\varepsilon\sigma w_j) \in {}_\varepsilon\gamma_\sigma^{\ell\pm\tau}, \forall i, j \in I \wedge \tau \in L, \tau \neq 0; \\ 0, |({}_\varepsilon\sigma w_i, {}_\varepsilon\sigma w_j) \notin {}_\varepsilon\gamma_\sigma^{\ell\pm\tau}, \forall i, j \in I \wedge \tau \in L, \tau \neq 0. \end{cases} \quad (2.15)$$

Система $S^{\ell\pm 0}$ имеет выходы $Y^{\ell \rightarrow \ell \pm \tau}$, направленные на вышестоящий уровень – $Y^{\ell \rightarrow \ell + \tau}$ и на нижестоящий – $Y^{\ell \rightarrow \ell - \tau}$. Система $S^{\ell\pm 0}$ получает возмущения по входам $X^{\ell \leftarrow \ell \pm \tau}$: от вышестоящих уровней – по входам $X^{\ell \leftarrow \ell + \tau}$ и от нижестоящих уровней – по входам $X^{\ell \leftarrow \ell - \tau}$. Состояние C^ℓ системы ℓ -уровня определяется собственными входами и выходами:

$$C^\ell \leftrightarrow \{X^{\ell \leftarrow \ell}, Y^{\ell \rightarrow \ell}\}. \quad (2.16)$$

Благодаря системным связям w^ℓ любое функциональное действие w^ℓ может быть изменено другими элементами системы.

2.2 Идея метода поиска рациональных решений

В соответствии с представленной выше формализацией многоуровневой иерархической системы использование имитационной модели для поиска рациональных решений в ERP-системе управления промышленным предприятием осуществляется следующей последовательностью этапов [153, с. 15].

Этап 1. Формулировка целей управления. Определяются цели (множество целей), которые должны быть реализованы в процессе управления.

Этап 2. Определение объекта управления. Этот этап связан с выделением той части среды, состояние которой интересует исследователя в связи с реализацией сформулированных целей. При формальном описании среды, проводя декомпозицию и выделяя объекты, всегда можно проверить на модели, достигаются ли цели управления в данном объекте или нет. Если нет, то зачастую удастся оценить степень неуправляемости этого объекта. На стадии определения объекта экспертно синтезируют несколько вариантов объекта, затем оценивают по одному или нескольким критериям и выбирают лучший.

Этап 3. Структурный синтез модели. Этап является сложным и включает следующие шаги [231, с. 168–170].

Шаг 1. Определение входов и выходов объекта, т. е. синтез модели на уровне «черного ящика».

Шаг 2. Экспертное ранжирование важности входов и выходов объекта.

Шаг 3. Декомпозиция модели.

Шаг 4. Выбор структурных элементов модели.

Экспертный метод решения перечисленных задач чаще всего является основным.

Этап 4. Параметрический синтез модели связан с определением ее параметров при выбранной структуре. Для определения параметров модели необходимо иметь информацию о поведении неуправляемых, управляемых входов и состоянии объекта.

Этап 5. Синтез управления связан с принятием решения о том, како-во должно быть управление, чтобы в сложившейся ситуации достигнуть заданной цели управления в объекте. Это решение получают с помощью имеющейся модели объекта и преследуют заданную цель. При этом основываются также на полученной информации о состоянии среды и объекта при выделенных ресурсах управления, которые представляют собой ограничения, накладываемые на управление в связи со спецификой объекта и возможностями системы управления. Полученное управление должно быть оптимально с точки зрения целей управления и представлять собой программу изменения управляемых параметров во времени.

Этап 6. Реализация управления связана с процессом отработки объектом программы, полученной на предыдущем этапе. Такой процесс для неактивных элементов решается методом теории следящих систем, обрабатывающих заданную программу. Трудности анализа поведения активных элементов должны быть преодолены на стадии синтеза модели объекта управления.

Этап 7. Адаптация – коррекция всей системы управления, т. е. всех этапов управления. Адаптация системы управления как процесс приспособления ее к специфическим свойствам объекта и окружающей среды имеет несколько иерархических уровней, соответствующих различным этапам управления [153, с. 23] сложным объектом. Возможны четыре уровня адаптации системы управления, представленные на рисунке 2.3.

1 *Параметрическая адаптация*, связанная с коррекцией, подстройкой параметров в модели. Необходимость в такого рода адаптации возникает ввиду дрейфа характеристик управляемого объекта. Исходной информацией для нее является рассогласование откликов объекта и модели.

2 *Структурная адаптация*, возникающая при несовпадении структур модели и объекта, устраняется, например, процедурой перехода от одной альтернативной модели к другой. При этом альтернативы могут различаться числом и характером входов, выходов модели, вариантами декомпозиции и структурой элементов модели. Методически структурная адаптация модели использует алгоритмы структурного синтеза.

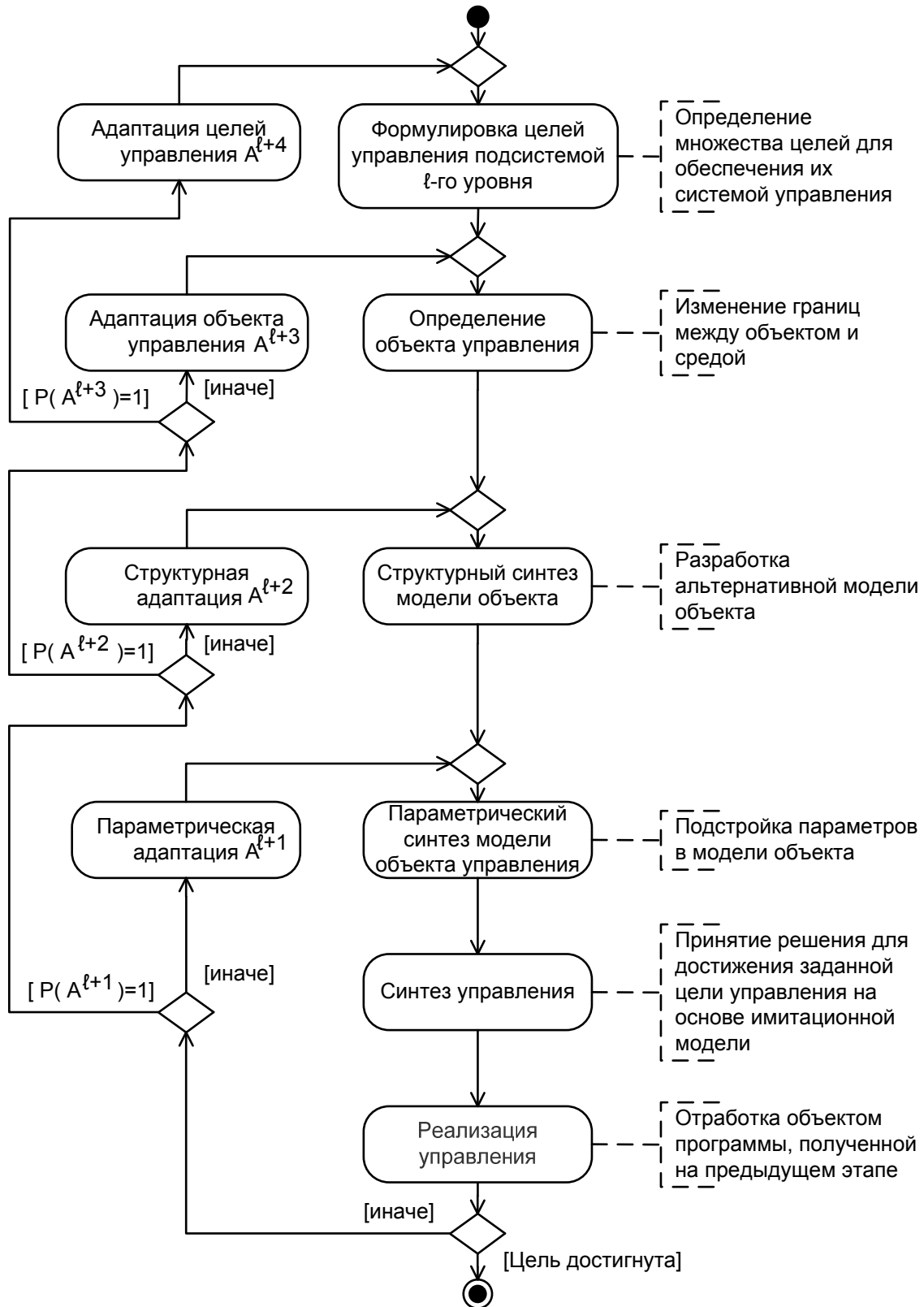


Рисунок 2.3 – Этапы использования имитационной модели в контуре управления сложной системой

3 *Адаптация объекта*, связанная с пересмотром границ, разделяющих объект и среду. Расширение объекта приводит, как правило, к повышению его управляемости, но требует дополнительных ресурсов для реализации управления. Разные варианты расширения объекта квалифицируются различным образом по управляемости и требуемому ресурсу управления. Выбор наилучшего варианта объекта в процессе управления им и составляет основу адаптации объекта.

4 *Адаптация целей управления*. Определяется новое множество целей, достижение которых обеспечивается созданной системой управления.

Все четыре уровня адаптации системы управления решают одну и ту же задачу – обеспечение достижения системой поставленных целей. Каждый последующий уровень адаптации имеет постоянно времени на несколько порядков выше, чем предыдущий, т. е. работает значительно медленнее. Это обстоятельство должно учитываться при создании системы адаптации: верхние уровни адаптации должны включаться лишь в том случае, если нижние не могут эффективно отследить изменение, произошедшее в объекте.

Процедура поиска рациональных решений основана на исследовании базовой имитационной модели с формированием показателей функционирования ERP-системы предприятия как многоуровневой иерархической системы. При этом иерархическая структура предприятия обладает следующими особенностями, которые учитываются при проведении исследований:

– для высших уровней иерархии системы имеет место лучшее понимание целей, назначения, функций, возможностей системы, но и одновременно высший уровень неполноты, неопределенности, неточности, противоречивости исходной информации для принятия решения;

– при последовательном переходе к нижним уровням иерархии уменьшается уровень неопределенности и повышается возможность детального конкретного описания структуры подсистем и конкретизация различных видов деятельности и практических задач [82, с. 306].

Специалистам промышленного предприятия в своей практической деятельности приходится принимать много технических решений, требующих научного обоснования. Сюда относятся, например, выбор структуры технологического процесса и оптимизация различных параметров технологических операций. При этом критериями оптимизации могут быть характеристики как самого технологического процесса (производительность, стабильность и др.), так и изготавливаемых изделий (себестоимость, точность, эксплуатационные показатели и т. д.) [88]. На этапе использования и исследования имитационной модели (объекта машиностроения, технологического процесса химического производства) возникает необходимость в изучении ее свойств, что выражается в нахождении значений параметров модели, обеспечивающих эти свойства.

При решении класса подобных задач часто используются методы оп-

тимизации, основанные на случайном поиске [153, с. 65–96; 287, с. 842–852; 298, с. 833–841; 308, с. 811–818]. Важными признаками данного класса алгоритмов являются универсальность и простота применения к конкретной задаче [27, с. 90–94; 267, с. 196–199].

Решение задач оптимизации сложных систем для достижения заданного уровня затруднено тем, что трудно выразить зависимость между входными и выходными параметрами модели в аналитическом виде. Это вынуждает использовать имитационные модели для описания подобных систем. Методы оптимизации, основанные на случайном поиске, обладают некоторыми недостатками (повышенные требования к вычислительным ресурсам по сравнению с методами, специально спроектированными для решения определенных проблем, отсутствие гарантии нахождения глобального оптимума, сложность реализации), однако часто являются основным способом решения задачи.

2.3 Формальное описание оптимизации объекта исследования на основе имитационной модели

Пусть имитационная модель M задана набором [267, с. 196]

$$M = \langle X_M, Y_M, A_M, Def_M \rangle, \quad (2.17)$$

где X_M – множество входных параметров модели,

$$X_M = \{x_{Mi} \mid x_{Mi} \in Def_M, i = 1, \dots, NX_M\}$$

и пространство значений входных параметров модели

$$SX_M = \{\overline{sx_M} \mid \overline{sx_M} = (x_{M1}, \dots, x_{Mn}), n = NX_M\}; \quad (2.18)$$

Y_M – множество откликов модели,

$$Y_M = \{y_{Mi} \mid y_{Mi} \in R, i = 1, \dots, NY_M\}$$

и пространство значений откликов модели

$$SY_M = \{\overline{sy_M}\}, \overline{sy_M} = (y_{M1}, \dots, y_{Mn}), n = NY_M; \quad (2.19)$$

A_M – алгоритм работы имитационной модели,

$$A_M : SX_M \rightarrow SY_M, \forall \overline{sx_M} \in SX_M \exists \overline{sy_M} \in SY_M \mid \overline{sy_M} = A_M(\overline{sx_M}). \quad (2.20)$$

В общем случае целевая функция задается набором

$$CF = \langle X_{CF}, Def_{CF}, F_{CF} \rangle, \quad (2.21)$$

где X_{CF} – множество параметров целевой функции,

$$X_{CF} = \{x_{CFi} \mid x_{CFi} \in Def_{CF}, i = 1, \dots, NX_{CF}\}$$

и пространство значений параметров целевой функции

$$SX_{CF} = \{\overline{sx_{CF}} \mid \overline{sx_{CF}} = (x_{CF1}, \dots, x_{CFn}), n = NX_{CF}\}. \quad (2.22)$$

Класс отображений целевой функции задается следующим образом:

$$CCF = \{F_{CF} \mid SX_{CF} \rightarrow R, \forall \overline{sx_{CF}} \in SX_{CF} \exists f \in R, f = F_{CF}(\overline{sx_{CF}})\}, \quad (2.23)$$

причем $\forall \overline{sx_{CF1}}, \overline{sx_{CF2}} \in SX_{CF}$, если решение $\overline{sx_{CF1}}$ лучше в понятиях предметной области решения $\overline{sx_{CF2}}$, то $F_{CF}(\overline{sx_{CF1}}) > F_{CF}(\overline{sx_{CF2}})$.

Пусть P_{AO} – множество параметров, используемых для настройки алгоритма оптимизации; X_{AO} – множество оптимизируемых параметров,

$$X_{AO} = \{x_{AOi}\}, x_{AOi} \in Def_{AO}, i = 1, \dots, NX_{AO}$$

и пространство поиска (решений) алгоритма

$$SX_{AO} = \{\overline{sx_{AO}}\}, \overline{sx_{AO}} = (x_{AO1}, \dots, x_{AOn}), n = NX_{AO}. \quad (2.24)$$

Тогда целевая функция алгоритма оптимизации задается набором

$$CF_{AO} = \langle X_{AO}, Def_{AO}, F_{AO} \rangle. \quad (2.25)$$

Таким образом, алгоритм оптимизации при заданном множестве параметров настройки P_{AO} и погрешности получаемого решения ε_{AO} задается как отображение:

$$A_{AO} : SX_{AO} \xrightarrow{P_{AO}} SX_{AO}^* \mid SX_{AO}^* \subseteq SX_{AO}, \forall \overline{sx_{AO}^*} \in SX_{AO}^*, \max_{SX_{AO}} (F_{AO} - F_{AO}(\overline{sx_{AO}^*})) \leq \varepsilon_{AO}. \quad (2.26)$$

Задача оптимизации задается следующим набором:

$$ZO = \langle M, X_{AO}, Def_{AO}, \overline{x_G}, CF_{ZO}, \varepsilon_{AO}, P_{AO} \rangle, \quad (2.27)$$

где $X_{AO} \subseteq X_M$, $\forall (x_M \stackrel{def}{=} x_{AO}) \in X_{AO}$, $x_{AO} \in Def_{AO}(x_{AO}) \subseteq Def_M(x_M)$;
 $\overline{x_G}$ – вектор условий, задающий значения постоянных параметров модели M ,

$$\begin{aligned} \overline{x_G} &= (x_{G1}, \dots, x_{Gn}), x_{Gi} \in X_M \mid X_{AO} \stackrel{def}{=} X_G, i = 1, \dots, n, \\ n &= NX_M - NX_{AO} \stackrel{def}{=} NX_G; \end{aligned} \quad (2.28)$$

CF_{ZO} – целевая функция задачи оптимизации,

$$CF_{ZO} = \langle X_{ZO}, Def_{ZO}, F_{ZO} \rangle, \quad (2.29)$$

где X_{ZO} – множество параметров целевой функции в задаче оптимизации,

$$X_{ZO} = \{x_{ZOi}\} \subseteq (X_M \cup Y_M), x_{ZOi} \in Def_{ZO} \subseteq (Def_M \cup Def(Y_M)), i = 1, \dots, NX_{ZO}.$$

Решение SX_M^* задачи ZO оптимизации модели M с использованием алгоритма оптимизации A_{AO} , настроенного в соответствии с множеством P_{AO} , задается в виде

$$\begin{aligned} SX_M^* &= A_{AO}(SX_{AO}) \mid SX_M^* \subseteq SX_M, \forall \overline{sx_M^*} \in SX_M^*, \\ \max_{SX_{ZO}} (F_{ZO} - \Phi_{ZO}^*(\overline{sx_M^*})) &\leq \varepsilon_{AO}; \end{aligned} \quad (2.30)$$

при оптимальном значении целевой функции

$$\Phi_{ZO}^*(\overline{sx_M^*}) = \left[A_M(\overline{sx_M^*}) = \overline{sy_M^*}, \overline{sx_{ZO}^*} = \underset{SX_{ZO}}{pr}(\overline{sx_M^*}, \overline{x_G}, \overline{sy_M^*}) \right] = F_{ZO}(\overline{sx_{ZO}^*}), \quad (2.31)$$

где $\underset{SX}{pr} \overline{sx}$ – проекция вектора \overline{sx} на пространство SX [267, с. 196–197].

Следует отметить, что для нахождения оптимального решения используется алгоритм оптимизации, в основу которого положен алгоритм случайного поиска.

2.4 Методика решения задачи рационального выбора алгоритма оптимизации систем

2.4.1 Формальное описание задачи рационального выбора структуры и параметров алгоритма оптимизации. Условно можно считать, что реализация алгоритма случайного поиска A_S состоит из структуры и параметров:

$$A_S = \langle St, C \rangle, \quad (2.32)$$

где St – структура реализации алгоритма A_S ;

C – параметры структуры St реализации алгоритма A_S ,
 $C = (c_1, c_2, \dots, c_k)$.

При проектировании реализации алгоритма случайного поиска выделены два этапа [29, с. 359]:

1) структурный синтез, целью которого является определение структуры St реализации алгоритма;

2) параметрический синтез, целью которого является определение параметров C реализации алгоритма.

Обычно алгоритмы случайного поиска не имеют строго фиксированной структуры, а постоянно претерпевают модификации с целью улучшения их работы. Структура реализации алгоритма случайного поиска определяется типом используемых операторов и отношений между ними. Структурную реализацию генетического алгоритма определяют, например, операторы репродукции, к которым относятся: селекция на основе рулетки, заданной шкалы, элитная, турнирная. Существует множество операторов кроссинговера: простой, двухточечный, упорядоченный, частично-соответствующий, циклический, универсальный. Мутацию реализуют двухточечный оператор и строительные блоки. Известны операторы транслокации, транспозиции, сегрегации, удаления, вставки, редукции, рекомбинации [51, с. 99–119]. Поэтому в ходе структурного синтеза исследователь получает не одну структуру реализации алгоритма, а некоторое их множество $\{St_i\}$. Проведя параметрический синтез для выделенных структур и определив множество параметров $\{C_i\}$, в соответствии с (2.32) получают некое множество A реализаций алгоритма случайного поиска A_S :

$$A = \{A_{S_i}\}, A_{S_i} = \langle St_i, C_i \rangle, \quad i = 1, \dots, NA, \quad (2.33)$$

где NA – число реализаций алгоритма A_S , выделенных при проектировании алгоритма.

Целевую функцию алгоритма случайного поиска можно задать следующим набором [29, с. 359]:

$$CF_{AS} = \langle X_{AS}, Def_{AS}, F_{AS} \in CFF \rangle, \quad (2.34)$$

где X_{AS} – множество оптимизируемых параметров,

$$X_{AS} = \{x_{ASi}\}, x_{ASi} \in Def_{AS}, i = 1, \dots, NX_{AS};$$

$F_{AS} \in CFF$ – отображение из класса, заданного следующим образом:

$$CFF = \{F_{AS} \mid SX_{AS} \rightarrow R, \forall \overline{sx_{AS}} \in SX_{AS} \exists f \in R, f = F_{AS}(\overline{sx_{AS}})\}, \quad (2.35)$$

где SX_{AS} – пространство значений оптимизируемых параметров,

$$SX_{AS} = \{\overline{sx_{AS}}\}, \overline{sx_{AS}} = (x_{AS1}, \dots, x_{ASn}), n = NX_{AS},$$

причем $\forall \overline{sx_{AS1}}, \overline{sx_{AS2}} \in SX_{AS}$, если решение $\overline{sx_{AS1}}$ лучше решения $\overline{sx_{AS2}}$ (в понятиях предметной области), то $F_{AS}(\overline{sx_{AS1}}) > F_{AS}(\overline{sx_{AS2}})$.

Тогда реализацию алгоритма A_S при заданной структуре St , множестве параметров C и погрешности получаемого решения ε_{AS} можно задать как отображение [29, с. 360]:

$$A_{AS} : SX_{AS} \xrightarrow{\langle St, C \rangle} SX_{AS}^* \mid SX_{AS}^* \subseteq SX_{AS}, \forall \overline{sx_{AS}^*} \in SX_{AS}^*, \\ \max_{SX_{AS}} (F_{AS} - F_{AS}(\overline{sx_{AS}^*})) \leq \varepsilon_{AS}. \quad (2.36)$$

Указанные выше информационные потоки более детально представлены в подразделе 6.1 при описании программной системы «Оптимизатор» для реализации метода поиска рациональных решений. При этом в качестве алгоритма оптимизации используется один из алгоритмов случайного поиска – генетический алгоритм.

2.4.2 Решение задачи рационального выбора структуры и параметров алгоритма оптимизации. Пусть в ходе анализа алгоритма A_S (см. выражение (2.36)) выделено множество целевых функций CF , которые удовлетворяют выражениям (2.34)–(2.35) и на которых необходимо проверить работоспособность алгоритма A_S :

$$CF = \{CF_{ASi}\}, i = 1, \dots, NCF. \quad (2.37)$$

Для каждой функции CF_{ASi} из множества CF проводится предварительное исследование RS_A^0 с целью определения такой реализации алгоритма A_S из множества A (см. выражение (2.33)), которая находит лучшее решение целевой функции:

$$(CF_{AS_i}, \{A_{S_i}^*\}) | A_{S_i}^* = \langle St_i^*, C_i^* \rangle, A_{S_i}^* \in A, \forall A_{S_j} \in A, \\ j \neq i \rightarrow CF_{AS_i}(SX_{AS_j}^*) < CF_{AS_i}(SX_{AS_i}^*), \quad (2.38)$$

где SX_{AS}^* – решения, найденные реализацией алгоритма в соответствии с выражением (2.36).

Используя статистические методы группировки, например кластерный анализ [229, с. 373], множество функций CF разбивается на классы функций [236, с. 204–205], для которых реализация алгоритма A_S^* дает наилучшее решение:

$$\Phi_1 = \langle (CF_{AS_1}, CF_{AS_2}, \dots, CF_{AS_{k_1}}), A_{S_1}^* \rangle; \\ \Phi_2 = \langle (CF_{AS_{k_1+1}}, CF_{AS_{k_1+2}}, \dots, CF_{AS_{k_2}}), A_{S_2}^* \rangle; \\ \dots \\ \Phi_{N\Phi} = \langle (CF_{AS_{k_{N\Phi-1}+1}}, CF_{AS_{k_{N\Phi-1}+2}}, \dots, CF_{AS_{k_{N\Phi}}}), A_{S_{N\Phi}}^* \rangle, \\ k_{N\Phi} = |CF| = NCF. \quad (2.39)$$

Пусть на вход алгоритма случайного поиска поступает очередная $(NCF+1)$ целевая функция $CF_{AS(NCF+1)}$ [26, с. 164–166]. Используя некоторый оператор R_X принадлежности, для произвольной функции определяется класс функций Φ_i , к которому она принадлежит, или новый $\Phi_{(N\Phi+1)}$, если эта функция не принадлежит ни одному из классов, т. е.

$$R_X(CF_{AS}) = \begin{cases} i | CF_{AS} \in \Phi_i; \\ N\Phi + 1 | CF_{AS} \notin \Phi_i, i = 1, \dots, N\Phi. \end{cases} \quad (2.40)$$

Данный оператор R_X может быть построен на основе квалификационных признаков, которые можно будет получить после анализа множества классов функций Φ_i . Также в основу оператора R_X может быть положена процедура исследования, подобная той, которая использовалась для получения пар (2.39).

Если оператор R_X находит класс функций Φ_i , для функции $CF_{AS(NCF+1)}$ используется соответствующая реализация алгоритма $A_{S_i}^*$ с уже известной структурой St_i^* и параметрами C_i^* [235, с. 156; 249, с. 332–334]. В противном случае для функции $CF_{AS(NCF+1)}$ необходимо провести предварительное исследование RS_A^0 с целью определения из множества A такой реализации алгоритма $A_{S(N\Phi+1)}^*$ (см. выражение (2.33)), которая находит лучшее решение целевой функции $CF_{AS(NCF+1)}$.

Полученная реализация алгоритма случайного поиска используется

для нахождения параметров исследуемой системы исходя из целевой функции $CF_{AS(NCF+1)}$.

В случае, когда оператор R_X (см. выражение (2.40)) не определил для новой функции $CF_{AS(NCF+1)}$ класс функций (см. выражение (2.39)), необходимо проводить исследование RS_{AS} . Возникновение новых задач, появление новых ситуаций – основные причины, вызывающие необходимость использовать адаптацию алгоритмов поиска, которая позволит автоматизировать исследование RS_{AS} .

Предлагается следующая последовательность шагов применения адаптации реализации алгоритма случайного поиска к решаемой задаче [28, с. 36–40].

Шаг 1. В ходе исследования алгоритма случайного поиска RS_A^0 определяется множество реализаций $\{A_{Si}^*\}$, удовлетворяющих (2.38). Для увеличения пространства возможных реализаций алгоритма, которые будут использоваться при адаптации, предлагается расширить данное множество до следующего:

$$\begin{aligned} (CF_{ASi}, \{A_{Si}^*\}) | A_{Si}^* = \langle St_i^*, C_i^* \rangle, A_{Si}^* \in A, \forall A_{Sj} \in A, \\ j \neq i \rightarrow |CF_{ASi}(SX_{ASj}^*) - CF_{ASi}(SX_{ASi}^*)| < \varepsilon_{AS^*}, \end{aligned} \quad (2.41)$$

где ε_{AS^*} – максимальное отклонение от лучшего решения, которое может быть найдено любой реализацией алгоритма из множества A .

Шаг 2. Полученные структуры $\{St_i^*\}$ являются исходными данными для *первого этапа адаптации*. На этом этапе параллельно запускаются реализации алгоритмов со структурами St_i^* . Различные структуры St_i^* используют различное количество ресурсов R для выполнения одной итерации поиска. Поэтому сравнение результатов их работы необходимо проводить при исчерпании одинакового количества ресурса R . В качестве такого ресурса выбрано число запусков целевой функции. Этот выбор обусловлен тем, что для вычисления целевой функции обычно используются результаты работы имитационной модели. После сравнения результатов работы различных реализаций алгоритма для дальнейшей адаптации выбирается k ($k < |\{St_i^*\}|$) реализаций.

Шаг 3. На *втором этапе* создаются алгоритмы адаптации для выделенных параметров на полученной для них области значений. Схема адаптации параметра в простейшем случае заключается в выборе шага или коэффициента изменения значения параметра, с которым параллельно запускается новая нить алгоритма.

Критерием останова адаптации алгоритма может быть достижение состояния системы, удовлетворяющего исследователя, или максимальное количество итераций адаптации [28, с. 40].

2.5 Целевая функция оценки эффективности взаимодействия компонентов предприятия

2.5.1 Конструирование интегральной целевой функции производственно-экономической деятельности предприятия. Задача исследования имитационной модели функционирования предприятия состоит в том, чтобы найти правильное (рациональное) решение, которое доставит выгоду (эффект) всей системе в целом, а не отдельным его подразделениям [260, с. 555–556]. При этом из-за наличия большого числа анализируемых показателей такую задачу относят к классу многокритериальных.

В формальном виде основные показатели функционирования предприятия с учетом введенных обозначений в таблице 2.1 и особенностями их расчетов представлены ниже.

Таблица 2.1 – Основные показатели деятельности промышленного предприятия

Наименование показателя	Формула расчета
1	2
Y₁: Общая оценка финансового состояния предприятия	
Y_{1.1}: Оценка неудовлетворительной структуры бухгалтерского баланса	
Коэффициент текущей ликвидности K ₁ на конец отчетного периода	$K_1 = \frac{\text{стр. 290}}{\text{стр. 790} - \text{стр. 720}} \text{ф. 1}$
Коэффициент обеспеченности собственными оборотными средствами K ₂ на конец отчетного периода	$K_2 = \frac{\text{стр. 590} + \text{стр. 690} - \text{стр. 190}}{\text{стр. 290}} \text{ф. 1}$
Y_{1.2}: Определение устойчивой неплатежеспособности предприятия	
Коэффициент обеспеченности финансовых обязательств активами	$K_3 = \frac{\text{стр. 790}}{\text{стр. 390}} \text{ф. 1}$
Y₂: Показатели оценки финансовой устойчивости организации	
Y_{2.1}: Показатели структуры капитала	
Коэффициент собственности	$K_C = \frac{\text{стр. 590} + \text{стр. 690}}{\text{стр. 890}} \text{ф. 1}$
Y_{2.2}: Показатели ликвидности	
Коэффициент абсолютной ликвидности	$K_{\text{АЛ}} = \frac{\text{стр. 260} + \text{стр. 270}}{\text{стр. 790} - \text{стр. 720}} \text{ф. 1}$

Продолжение таблицы 2.1

1	2
Коэффициент критической (быстрой) ликвидности	$K_{\text{БЛ}} = \frac{\text{стр. 290} - \text{стр. 210}}{\text{стр. 790} - \text{стр. 720}} \text{ф. 1}$
У3: Показатели роста и развития предприятия	
Темп роста продаж (нетто)	$T_{\text{РП}} = \frac{\text{стр. 030} \text{гр. 3, ф. 2}}{\text{стр. 030} \text{гр. 4, ф. 2}}$
Темп роста себестоимости	$T_{\text{РС}} = \frac{(\text{стр. 040} + \text{стр. 050} + \text{стр. 060}) \text{гр. 3, ф. 2}}{(\text{стр. 040} + \text{стр. 050} + \text{стр. 060}) \text{гр. 4, ф. 2}}$
Темп роста прибыли от реализации продукции	$T_{\text{РПР}} = \frac{\text{стр. 070} \text{гр. 3, ф. 2}}{\text{стр. 070} \text{гр. 4, ф. 2}}$
Темп роста чистой прибыли	$T_{\text{РЧП}} = \frac{\text{стр. 240} \text{гр. 3, ф. 2}}{\text{стр. 240} \text{гр. 4, ф. 2}}$
Темп роста стоимости активов	$T_{\text{РСА}} = \frac{\text{стр. 390} \text{гр. 3, ф. 1}}{\text{стр. 390} \text{гр. 4, ф. 1}}$
У4: Показатели рентабельности	
Рентабельность продукции по затратам	$R_{\text{ПЗ}} = \frac{\text{стр. 070}}{\text{стр. 040} + \text{стр. 050} + \text{стр. 060}} \text{гр. 3, ф. 2}$
Рентабельность выручки от продаж	$R_{\text{ВП}} = \frac{\text{стр. 070}}{\text{стр. 030}} \text{гр. 3, ф. 2}$
Рентабельность активов предприятия	$R_{\text{А}} = \frac{\text{стр. 200} \text{гр. 3, ф. 2}}{\frac{\text{стр. 390} \text{гр. 3} + \text{стр. 390} \text{гр. 4}}{2}} \text{ф. 1}$
Рентабельность основного капитала	$R_{\text{ОК}} = \frac{\text{стр. 200} \text{гр. 3, ф. 2}}{\frac{\text{стр. 110} \text{гр. 3} + \text{стр. 110} \text{гр. 4}}{2}} \text{ф. 1}$
Рентабельность собственного капитала	$R_{\text{СК}} = \frac{\text{стр. 240} \text{гр. 3, ф. 2}}{\frac{(\text{стр. 590} + \text{стр. 690}) \text{гр. 3} + (\text{стр. 590} + \text{стр. 690}) \text{гр. 4}}{2}} \text{ф. 1}$
У5: Показатели эффективности использования основных производственных фондов	
Коэффициент фондоотдачи	$K_{\text{ФО}} = \frac{\text{стр. 030} \text{гр. 3, ф. 2}}{\frac{\text{стр. 110} \text{гр. 3} + \text{стр. 110} \text{гр. 4}}{2}} \text{ф. 1}$
Период окупаемости собственного капитала	$K_{\text{ОСК}} = \frac{\text{стр. 590} + \text{стр. 690} \text{гр. 3} + \text{стр. 590} + \text{стр. 690} \text{гр. 4}}{2} \text{ф. 1}$ $K_{\text{ОСК}} = \frac{\text{стр. 240} \text{гр. 3, ф. 2}}{\text{стр. 240} \text{гр. 3, ф. 2}}$

Окончание таблицы 2.1

1	2
Y₆: Показатели эффективности использования оборотных активов	
Коэффициент оборачиваемости оборотных активов	$K_{\text{ООА}} = \frac{\text{стр. 030 гр. 3, ф. 2}}{\frac{\text{стр. 290 гр. 3 + стр. 290 гр. 4}}{2} \text{ф. 1}}$
Продолжительность оборота оборотных активов	$П_{\text{ООА}} = \frac{\text{стр. 290 гр. 3 + стр. 290 гр. 4}}{2} \text{ф. 1}}{\text{стр. 030 гр. 3, ф. 2}} \cdot 360$
Продолжительность оборота производственных запасов	$П_{\text{ОПЗ}} = \frac{\text{стр. 210 гр. 3 + стр. 210 гр. 4}}{2} \text{ф. 1}}{\text{стр. 030 гр. 3, ф. 2}} \cdot 360$
Продолжительность оборота незавершенного производства	$П_{\text{ОНП}} = \frac{\text{стр. 213 гр. 3 + стр. 213 гр. 4}}{2} \text{ф. 1}}{\text{стр. 030 гр. 3, ф. 2}} \cdot 360$
Продолжительность оборота готовой продукции	$П_{\text{ОГП}} = \frac{\text{стр. 230 гр. 3 + стр. 230 гр. 4}}{2} \text{ф. 1}}{\text{стр. 030 гр. 3, ф. 2}} \cdot 360$
Продолжительность оборота дебиторской задолженности	$П_{\text{ОДЗ}} = \frac{\text{стр. 250 гр. 3 + стр. 250 гр. 4}}{2} \text{ф. 1}}{\text{стр. 030 гр. 3, ф. 2}} \cdot 360$
Примечание – ф. – номер формы; стр. – номер строки; гр. – номер графы	

Общая оценка финансового состояния предприятия Y_1 и показатели оценки финансовой устойчивости организации Y_2 :

$$\begin{aligned} Y_1 &= Y_{1.1} \cup Y_{1.2} | Y_{1.1} = \{K_1, K_2\}; \quad Y_{1.2} = \{K_3\}; \\ Y_2 &= Y_{2.1} \cup Y_{2.2} | Y_{2.1} = \{K_C\}; \quad Y_{2.2} = \{K_{\text{АЛ}}, K_{\text{БЛ}}\}. \end{aligned} \quad (2.42)$$

Показатели роста и развития предприятия Y_3 , показатели рентабельности Y_4 :

$$\begin{aligned} Y_3 &= \{T_{\text{РП}}, T_{\text{РС}}, T_{\text{ТПР}}, T_{\text{РЧП}}, T_{\text{РСА}}\}; \\ Y_4 &= \{R_{\text{ПЗ}}, R_{\text{ВП}}, R_{\text{А}}, R_{\text{ОК}}, R_{\text{СК}}\}. \end{aligned} \quad (2.43)$$

Показатели эффективности использования основных производственных фондов Y_5 и оборотных активов Y_6 :

$$\begin{aligned}
 Y_5 &= \{K_{\text{ФО}}, K_{\text{ОСК}}\}; \\
 Y_6 &= \{K_{\text{ООА}}, \Pi_{\text{ООА}}, \Pi_{\text{ОПЗ}}, \Pi_{\text{ОНП}}, \Pi_{\text{ОГП}}, \Pi_{\text{ОДЗ}}\}.
 \end{aligned}
 \tag{2.44}$$

Целевая функция U производственно-экономической деятельности предприятия:

$$U = F(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6) \tag{2.45}$$

или с учетом обозначений, представленных в таблице 2.1:

$$\begin{aligned}
 U = F(<K_1, K_2>, <K_3>, <K_C>, <K_{\text{АЛ}}, K_{\text{БЛ}}>, <T_{\text{РП}}, T_{\text{РС}}, T_{\text{ТПР}}, T_{\text{РЧП}}, T_{\text{РСА}}>, <R_{\text{ПЗ}}, \\
 R_{\text{ВП}}, R_{\text{А}}, R_{\text{ОК}}, R_{\text{СК}}>, <K_{\text{ФО}}, K_{\text{ОСК}}>, <K_{\text{ООА}}, \Pi_{\text{ООА}}, \Pi_{\text{ОПЗ}}, \Pi_{\text{ОНП}}, \Pi_{\text{ОГП}}, \Pi_{\text{ОДЗ}}>.
 \end{aligned}$$

Наиболее популярным способом построения многокритериальной целевой функции в имитационном моделировании является алгебраическая модель [279, с. 642–649].

Интегральная целевая функция (ИЦФ) может быть представлена в виде линейной модели с весовыми коэффициентами:

$$\begin{aligned}
 U(Y_1, Y_2, \dots, Y_m) &= w_1 U_1(Y_1) + w_2 U_2(Y_2) + \dots + w_m U_m(Y_m) = \\
 &= \sum_{i=1}^m w_i U(Y_i),
 \end{aligned}
 \tag{2.46}$$

где w_1, w_2, \dots, w_m – весовые коэффициенты важности критериев ($i = 1, 2, \dots, m$) ИЦФ; выполняется требование для удобства вычислений и анализа:

$$\sum_{i=1}^m w_i = 1.$$

При этом требуется выполнение следующих условий [275, с. 1936–1937]:

– функция каждого отдельного критерия $w_1 U_1(Y_1), w_2 U_2(Y_2), \dots, w_m U_m(Y_m)$ может быть нелинейной, хотя для ИЦФ используется линейная форма суммы критериев;

– критерии ИЦФ должны быть взаимно независимыми, чтобы полезность одного критерия не оказывала влияние на полезность других критериев. При взаимной зависимости отмечаются ошибочные результаты по итогам их исследований;

– необходимо провести нормирование критериев, чтобы ИЦФ могла оценивать различные альтернативы.

Нормированные значения $U_i(Y_i)^*$ определяются путем деления абсо-

лутных значений $U_i(Y_i)$ на максимальное значение $U_i(Y_i)_{max}$ [105, с. 177–178].

В информационной теории используют понятие энтропии как меру неопределенности [96, с. 25]. Рассматривают энтропию событий с вероятностями p_1, p_2, \dots, p_n . Энтропия $S(p_1, p_2, \dots, p_n)$ принимает максимальное значение, когда неопределенность появления событий максимальна, т. е. $p_i = 1/n$.

Идея энтропии особенно полезна при исследовании контрастов между наборами данных. Эти наборы данных могут быть представлены как альтернативные решения в матрице решений (MP), где каждая альтернатива оценена в терминах набора критериев полезности решения [275, с. 1937].

Запись MP с l альтернативными решениями, оцениваемыми значениями m критериев, имеет вид:

$$MP = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1m} \\ y_{21} & y_{22} & \dots & y_{2m} \\ \dots & \dots & \dots & \dots \\ y_{l1} & y_{l2} & \dots & y_{lm} \end{bmatrix}$$

Альтернативные решения, по Р. Аль-Аомару (R. Al-Aomar, 2002) [275, с. 1938], могут быть представлены распределением вероятностей p_{kj} , где $k = 1, 2, \dots, l$ и $j = 1, 2, \dots, m$. Каждое значение p_{kj} содержит информацию, которая может быть измерена значением энтропии. При этом p_{kj} определяется по формуле

$$p_{kj} = y_{kj} / \sum_{k=1}^l y_k, \forall k, l,$$

где y_k – значение ИЦФ для k -го альтернативного решения.

Матрица вероятностей p_{kj} сформируется следующим образом:

$$\| p_{kj} \| = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1m} \\ p_{21} & p_{22} & \dots & p_{2m} \\ \dots & \dots & \dots & \dots \\ p_{l1} & p_{l2} & \dots & p_{lm} \end{bmatrix}$$

Энтропия S_j для набора значений критериев j в l альтернативных решениях определяется по формуле

$$S_j = -K \sum_{k=1}^l p_{kj} \ln p_{kj}, \forall j,$$

где K представлено постоянной величиной со значением $1/\ln l$ в S_{max} , которое гарантирует, что $0 \leq S_j \leq 1$.

При этом степень разнообразия d_j значений j -го критерия в l альтернативных решениях рассчитывается по формуле

$$d_j = 1 - S_j, \forall j.$$

Тогда вес важности каждого j -го критерия в ИЦФ определяется по формуле

$$w_j = d_j / \sum_{j=1}^m d_j, \forall j.$$

Если экспертом определены субъективные оценки $(\lambda_1, \lambda_2, \dots, \lambda_m)$ каждого из m критериев, то, используя оценки весов (w_1, w_2, \dots, w_m) , получают адаптированные веса по формуле

$$w_j^0 = w_j \lambda_j / \sum_{j=1}^m w_j \lambda_j, \forall j.$$

Таким образом, метод энтропии позволяет оценивать относительный вес значений критериев в линейной ИЦФ. При использовании генетического алгоритма [28, с. 36–40] в качестве метода оптимизации параметров в альтернативных решениях обеспечивается наличие обобщенного критерия для различения проектных альтернатив.

2.5.2 Координация задач в иерархической структуре предприятия. Деятельность промышленных предприятий М. Месарович и др. представляют многоуровневой иерархической структурой, в которой вводится понятие координации [119].

Координация подсистем означает такое воздействие на подсистемы, которое заставляет действовать их согласованно. Вводятся два понятия координируемости на примере двухуровневой системы. Первое – это координируемость по отношению к задаче вышестоящей системы, второе – координируемость по отношению к решаемой в настоящий момент глобальной задаче.

Пусть определен предикат [251, с. 132]

$$\forall(x, D) P(x, D) \equiv x \text{ есть решение } D,$$

где D – произвольная решаемая задача.

Следовательно, предикат $P(x, D)$ является истинным тогда, и только тогда, когда D – решаемая задача, а x – одно из ее решений.

Пусть D_0 – конкретная задача, решаемая вышестоящей системой, и каждый координирующий сигнал $g \in G$ уточняет задачу $D(g)$, которую будет решать координатор на ℓ -уровне. Задачи, решаемые нижестоящими элементами, координируемы тогда, и только тогда, когда истинным является следующий предикат:

$$\exists g \exists x P(x, D(g)) \wedge P(g, D_0),$$

т. е. координируемость требует, чтобы задача D_0 имела решение g и чтобы для этого координирующего воздействия множество $D(g)$ задач, решаемых нижестоящими элементами, также имело решение x .

Ни один из координаторов внутри иерархии не облечен специально полномочиями решать глобальную задачу и тем самым преследовать общую (глобальную) цель. Задачи, которые решаются нижестоящими решающими элементами, координируемы относительно глобальной задачи D^* , если справедливо следующее предложение [251, с. 132]:

$$\exists g \exists x P(x, D(g)) \wedge P(g, D^*).$$

М. Месарович ввел постулат совместимости [119, с. 122], в соответствии с которым для совместимости (согласования) решаемых задач, а тем самым и целей внутри двухуровневой системы, координация задач, решаемых нижестоящими элементами относительно задачи вышестоящего координатора, должна быть соответствующим образом связана с подлежащей решению глобальной задачей.

Высокий уровень абстракции введенного предложения не дает конкретных рекомендаций для применения его на практике. Были сделаны попытки разработки подобных рекомендаций. Например, А. Е. Алтуниным и др. на основе теории нечетких множеств разработаны рекомендации для иерархических технологических объектов [6]. Однако в целом задача осталась нерешенной.

2.5.3 Решение многокритериальных задач в управлении ресурсами предприятия. Постановка всякой задачи многокритериального выбора содержит три объекта: множество возможных решений, векторный критерий и отношение предпочтения лица, принимающего решение. Решить эту задачу – значит на основе векторного критерия и отношения предпочтения найти множество выбираемых решений.

В рассматриваемой модели принцип Эджворта-Парето (принцип Парето) формулируется в виде утверждения о том, что множество выбирае-

мых решений содержится в множестве Парето, т. е. каждое выбираемое решение является Парето-оптимальным [140].

Принцип Парето применяется не для всех многокритериальных задач. В. Д. Ногиним определен класс задач многокритериального выбора, для которого применение принципа Парето является обоснованным [129, с. 9].

Прежде всего должен быть задан набор решений X (вариантов), из которого следует осуществлять выбор. Выбор решения состоит в указании среди всех возможных такого решения, которое объявляется выбранным. Может быть сформировано множество выбираемых решений $Sel X$:

$$Sel X \subset X.$$

Имеется несколько числовых функций $f_1, f_2, \dots, f_m \mid m \geq 2$, заданных на множестве возможных решений X и именуемых целевыми функциями (критериями оптимальности, критериями эффективности, показателями или критериями качества). Числовые функции f_1, f_2, \dots, f_m образуют векторный критерий $f = (f_1, f_2, \dots, f_m)$, который принимает значения в пространстве m -мерных векторов R^m . Это пространство называют критериальным пространством, или пространством оценок, а всякое значение $f(x) = (f_1(x), f_2(x), \dots, f_m(x)) \in R^m$ векторного критерия f при определенном $x \in X$ именуется векторной оценкой возможного решения x . Все возможные векторные оценки образуют множество возможных оценок [129, с. 18]:

$$Y = f(X) = \{y \in R^m \mid y = f(x), x \in X\}.$$

Рассматривая $Sel X \subset X$, можно ввести множество выбираемых оценок:

$$Sel Y = f(Sel X) = \{y \in R^m \mid y = f(x), x \in Sel X\}.$$

Рассмотрим два возможных решения x' и x'' . Предположим, что координатор (лицо, принимающее решение (ЛПР)) выбирает (отдает предпочтение) первое из них, тогда

$$x' \succ_X x''.$$

Знак \succ_X называют отношением предпочтения [129, с. 20]. Отношение предпочтения обладает свойствами [128, с. 46]:

– антирефлексивности

$$\forall x \in X \neg \exists x \succ_X x;$$

– антисимметричности

$$\forall x', x'' \in X \quad x' \succ_X x'' \& x'' \succ_X x' \Rightarrow x' = x'';$$

– транзитивности

$$\forall x', x'', x''' \in X \quad x' \succ_X x'' \& x'' \succ_X x''' \Rightarrow x' \succ_X x'''.$$

Отношение предпочтения \succ_X , заданное на множестве возможных решений, естественным образом порождает отношение предпочтения \succ_Y на множестве возможных векторов Y :

$$f(x') \succ_Y f(x'') \Leftrightarrow x' \succ_X x'' \mid \forall x', x'' \in X.$$

Тем самым вектор $y' = f(x')$ предпочтительнее вектора $y'' = f(x'')$:

$$y' \succ_Y y''$$

тогда, и только тогда, когда решение x' предпочтительнее решения x'' :

$$x' \succ_X x''.$$

В. Д. Ногиным на основе аксиоматического подхода строго сформулирован принцип Парето и установлено, при выполнении каких требований *применение этого принципа оправдано* [129, с. 27].

Аксиома 1 (исключение доминируемых решений). Любое множество выбираемых решений не должно содержать ни одного такого решения, для которого может найтись более предпочтительное решение:

$$\forall x', x'' \in X \quad \exists x' \succ_X x'' \Rightarrow x'' \notin X.$$

В соответствии с аксиомой 1 любое доминируемое решение следует исключать из списка решений, претендующих на роль выбираемых. Множество недоминируемых решений обозначается $Ndom X$ и определяется равенством

$$Ndom X = \{x^* \in X \mid \neg \exists x \in X, x \succ_X x^*\}.$$

Тогда для любого непустого множества выбираемых решений $Sel X$, удовлетворяющих аксиоме 1, справедливо включение

$$Sel X \subset Ndom X,$$

которое устанавливает, что для класса задач, удовлетворяющих аксиоме 1, выбор решений следует производить только среди недоминируемых решений.

Аксиома 2 (продолжение отношения предпочтения). Существует продолжение \succ на все критериальное пространство R^m отношения \succ_Y , причем это продолжение \succ является антирефлексивным и транзитивным отношением.

Суть этого требования заключается в постулировании расширенных возможностей ЛПР сравнивать оценки по предпочтительности. В соответствии с ним выполняется одно, и только одно, из следующих трех соотношений:

- 1) $y' \succ y''$;
- 2) $y'' \succ y'$;
- 3) $\neg \exists y' \succ y'' \vee y'' \succ y'$.

В задаче многокритериального выбора отношение предпочтения и критерии оптимальности выражают интересы одного и того же ЛПР. Поэтому они должны быть согласованы друг с другом. Критерий f_i согласован с отношением предпочтения \succ , если для любых двух векторов $y', y'' \in R^m$, таких, что

$$y' = (y'_1, \dots, y'_{i-1}, y'_i, y'_{i+1}, \dots, y'_m), y'' = (y''_1, \dots, y''_{i-1}, y''_i, y''_{i+1}, \dots, y''_m), y' > y'' \Rightarrow y' \succ y''.$$

Взаимосвязь отношения предпочтения данного ЛПР с критериями оптимальности выражается следующим требованием.

Аксиома 3 (согласование критериев с отношением предпочтения). Каждый из критериев f_1, f_2, \dots, f_m согласован с отношением предпочтения \succ .

Заинтересованность ЛПР в получении по возможности больших значений всех компонент векторного критерия f выражается в терминах аксиомы Парето.

Аксиома Парето (в терминах решений). Для всех пар решений $x', x'' \in X$, для которых имеет место неравенство $f(x') \geq f(x'')$, выполняется соотношение $x' \succ_X x''$.

При этом запись $f(x') \geq f(x'')$ означает выполнение покомпонентных отношений $f_i(x') > f_i(x'') \vee f_i(x') = f_i(x'')$ для всех $i = 1, 2, \dots, m$, причем $f(x') \neq f(x'')$.

Если для некоторой пары возможных решений имеет место неравенство $f(x') \geq f(x'')$, то благодаря аксиоме Парето первое решение будет предпочтительнее второго, т. е. $x' \succ_X x''$. Тогда в соответствии с аксиомой 1

второе решение ни при каких обстоятельствах не может оказаться выбранным и его можно исключить из последующего учета в процессе принятия решений. Исключение всех подобного рода решений приводит к множеству Парето. Множество Парето-оптимальных решений обозначается $P_f(X)$ и определяется равенством

$$P_f(X) = \{x^* \in X \mid \neg \exists x \in X, f(x) > f(x^*)\}.$$

При выполнении аксиом 2 и 3 множество недоминируемых решений $\text{Ndom } X$ удовлетворяет включению $\text{Ndom } X \subset P_f(X)$.

В условиях выполнения аксиом 1–3 для любого непустого множества выбираемых решений $\text{Sel } X$ справедливо включение $\text{Sel } X \subset P_f(X)$. Это включение выражает собой принцип Парето, согласно которому, если ЛПР ведет себя в соответствии с аксиомами 1–3, то выбираемые им решения обязательно являются Парето-оптимальными.

Вектор $f(x^*)$ при Парето-оптимальном решении x^* называют Парето-оптимальным вектором решения x^* или просто Парето-оптимальным вектором. Для множества таких векторов используют обозначение $P(Y)$. Таким образом,

$$P(Y) = f(P_f(X)) = \{f(x^*) \in Y \mid \forall x^* \in P_f(X)\}.$$

В случае конечного множества возможных векторов Y (в частности, если множество решений X конечно) существует хотя бы одно Парето-оптимальное решение и, соответственно, хотя бы один Парето-оптимальный вектор.

При выполнении аксиомы Парето в терминах векторов для любой пары векторов $y', y'' \in R^m$, таких, что $y' > y''$, имеет место соотношение $y' \succ y''$, т. е.

$$y' > y'' \Rightarrow y' \succ y''.$$

Введенные ранее подмножества множества возможных решений связаны следующими включениями:

$$\text{Sel } X \subset \text{Ndom } X \subset P_f(X) \subset X.$$

В терминах векторов эти включения имеют вид:

$$\text{Sel } Y \subset \text{Ndom } Y \subset P(Y) \subset Y.$$

Для построения множества $P(Y)$ (и $P_f(X)$) в случае конечного множества возможных векторов Y можно применять алгоритм, представленный на рисунке 2.4 [129, с. 40].

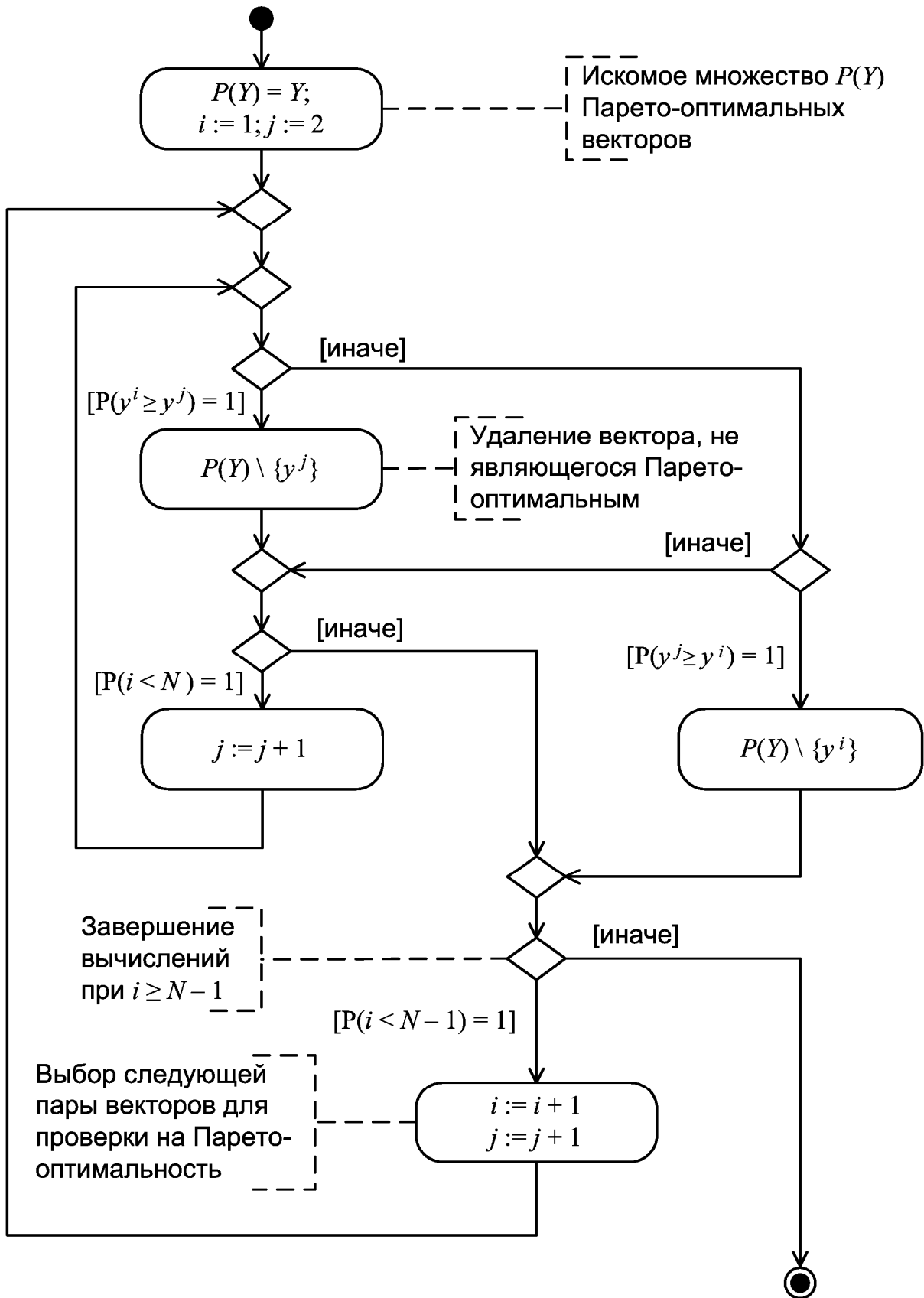


Рисунок 2.4 – Алгоритм построения множества Парето-оптимальных векторов $P(Y)$

Полученные результаты инвариантны относительно уровней иерархической структуры предприятия. В этом случае имитационная модель позволяет оценивать решение ЛПР на ℓ -уровне и вышестоящих уровнях, включая глобальную цель [233, с. 113]:

$$x' \succ_x x'' \Rightarrow f(x') \geq f(x'').$$

Вектор глобальной цели [251, с. 135]:

$$(x')^\ell \xrightarrow{A_M} (y')^0; (x'')^\ell \xrightarrow{A_M} (y'')^0,$$

где A_M – алгоритм модели.

Соответственно,

$$(x' \succ_x x'')^\ell \Rightarrow (y' \geq y'')^0.$$

Постулат совместимости М. Месаровича может быть представлен в виде

$$P[((x' \succ_\ell x'')^\ell \Rightarrow (y' \succ_\ell y'')^\ell) \wedge (x' \succ_\ell x'')^\ell \Rightarrow (y' \succ_0 y'')^0] = 1.$$

Получают два множества Парето-оптимальных векторов на ℓ -уровне $P(Y)^\ell$ и 0-уровне $P(Y)^0$ глобальной цели. Понятно, что этим множествам в общем случае соответствуют два множества Парето-оптимальных решений на ℓ -уровне: $P_f(X)_\ell$ и $P_f(X)_0$.

Тогда возможны следующие варианты [251, с. 135]:

– полная несовместимость

$$P_f(X)_\ell \cap P_f(X)_0 = \emptyset;$$

– частичная совместимость

$$P_f(X)_\ell \cap P_f(X)_0 \neq \emptyset;$$

– абсолютная совместимость при выполнении постулата совместимости

$$P_f(X)_\ell \cap P_f(X)_0 = P_f(X)_\ell = P_f(X)_0.$$

Таким образом, появляется возможность количественной оценки координирующих действий в иерархической системе.

С. С. Макаров предлагает фундаментальный подход к задачам моделирования на примере медицинских учреждений, отличающийся форму-

лировкой целей и задач моделирования, структурированием цели на основе системных методов с декомпозицией глобальной цели, позволяющий применять аналитические и эвристические методы для решения задач моделирования [112, с. 6]. В алгоритме проектирования информационно-управляющей системы отмечается, что выбранные варианты оцениваются экспертами, результаты согласовываются и узакониваются с заказчиком. Применяют модели достижения согласия, на основании чего предложения проектировщиков и запросы заказчиков анализируются совместно, что разрешает конфликты [112, с. 28].

2.5.4 Сужение множества Парето на основе информации об относительной важности критериев. Пусть I – множество номеров критериев:

$$I = \{1, 2, \dots, m\}.$$

Тогда i -й критерий важнее j -го критерия с заданными положительными параметрами w_i^* и w_j^* , если для всех векторов $y', y'' \in R^m$, для которых выполняется

$$\begin{aligned} y'_i > y''_i, y''_j > y'_j, y'_s = y''_s \quad \forall s \in I \setminus \{i, j\}; \\ y'_i - y''_i = w_i^*, y''_j - y'_j = w_j^* \Rightarrow y' \succ y''. \end{aligned} \quad (2.47)$$

Для ЛПР i -й критерий важнее j -го, если всякий раз при выборе из пары векторов ЛПР готово пожертвовать определенным количеством w_j^* по менее важному j -му критерию ради получения дополнительного количества w_i^* по более важному i -му критерию при условии сохранения всех остальных значений критериев. При этом соотношение между w_i^* и w_j^* позволяет количественно оценить указанную степень важности. Для указанной пары критериев коэффициентом относительной важности называют положительное число [233, с. 114]:

$$\theta_{ij} = \frac{w_j^*}{w_i^* + w_j^*}. \quad (2.48)$$

Коэффициент θ_{ij} показывает долю потери по менее важному критерию, на которую согласно пойти ЛПР, в сравнении с суммой потери и прибавки по более важному критерию.

Для построения содержательной теории относительной важности критериев принята следующая аксиома.

Аксиома 4 (инвариантность отношения предпочтения). Отношение предпочтения \succ является инвариантным относительно линейного положи-

тельного преобразования.

Признаком инвариантности отношения \succ является наличие у него свойства аддитивности

$$\forall y', y'' \in R^m \forall c \in R^m y' \succ y'' \Rightarrow (y' + c) \succ (y'' + c) \quad (2.49)$$

и однородности

$$\forall y', y'' \in R^m \forall \alpha > 0 y' \succ y'' \Rightarrow \alpha y' \succ \alpha y''. \quad (2.50)$$

Благодаря инвариантности отношения предпочтения \succ можно считать, что в определении (2.47) векторы y' и y'' фиксированы. В частности, в этом определении можно положить

$$y'_i = w_i^*, \quad y'_j = -w_j^*, \quad y'_s = 0 \quad \forall s \in I \setminus \{i, j\}, \quad y'' = 0_m,$$

или

$$y'_i = (1 - \theta_{ij}), \quad y'_j = -\theta_{ij}, \quad y'_s = 0 \quad \forall s \in I \setminus \{i, j\}, \quad y'' = 0_m.$$

Если отношение предпочтения \succ удовлетворяет аксиомам 1–4 и i -й критерий важнее j -го с коэффициентом относительной важности $\theta_{ij} \in (0, 1)$, тогда для любого непустого множества выбираемых оценок $\text{Sel } Y$ имеют место включения [129, с. 60]:

$$\text{Sel } Y \subset P(\hat{Y}) \subset P(Y), \quad (2.51)$$

где $P(\hat{Y})$ – множество Парето-оптимальных оценок в многокритериальной задаче с исходным множеством возможных решений X и новым векторным критерием $\hat{f} = (\hat{f}_1, \hat{f}_2, \dots, \hat{f}_m)$, т. е. $\hat{Y} = \hat{f}(X)$, компоненты которого вычисляются по формулам:

$$\hat{f}_j = \theta_{ij} f_i + (1 - \theta_{ij}) f_j, \quad (2.52)$$

$$\hat{f}_s = f_s \quad \forall s \in I \setminus \{j\}. \quad (2.53)$$

На основе информации о том, что один критерий важнее другого, множество Парето может быть сужено без потери выбираемых векторов. По формуле (2.52) векторный критерий \hat{f} получается из прежнего крите-

рия f заменой менее важного критерия f_j на выпуклую комбинацию критериев f_i и f_j с коэффициентом относительной важности θ_{ij} . Все остальные критерии f_s сохраняются. При подобном пересчете многие полезные с точки зрения оптимизации свойства критериев f_i и f_j сохраняются. Например, если f_i и f_j являются непрерывными, вогнутыми, выпуклыми или линейными, то критерий \hat{f} также будет обладать соответствующими свойствами [129, с. 65].

3 Метод имитационного моделирования ERP-системы управления ресурсами промышленного предприятия

3.1 Идея метода имитационного моделирования

Промышленное предприятие (ПП) представляется многоуровневой иерархической системой с координирующими элементами, принимающими решения на каждом из уровней в соответствии со своим положением в иерархии управления [257, с. 132]. Предлагаемый метод имитационного моделирования ERP-системы управления ПП в условиях неопределенности и риска выбора нерациональных ресурсов использует декомпозицию системы на подсистемы, процессы и активности, позволяет обеспечить важное сходство структуры модели и объекта исследования на основе процессного способа имитации.

В основу метода положены:

1) представление промышленного предприятия в виде *многоуровневой иерархической системы* с распределением компонентов исследуемой системы по уровням с активными элементами принятия решений на каждом из уровней иерархии [210, с. 76; 239, с. 75–76; 257, с. 132];

2) создание и использование *базовой имитационной модели* (БИМ) промышленного предприятия в ERP-системе управления [11, с. 11–16];

3) использование методологии *UML* и проведение функционального анализа на основе методологии *SADT* для построения вербальной и концептуальной модели объекта исследования [221, с. 112–118];

4) применение методологии *объектно-ориентированного анализа и проектирования* для составления программы имитационной модели на основе процессного способа имитации [221, с. 149–155];

5) использование технологии *ADO XML* для ввода данных в модель из комплексной информационной системы [261, с. 111–117];

6) выбор *варианта решения* на соответствующем уровне управления в условиях неопределенности и риска выбора нерациональных ресурсов предприятия [251, с. 150–153].

Реализация метода предполагает построение базовой имитационной модели промышленного предприятия с учетом его многоуровневой иерархической структуры и уточнением в модели объектов системы следующей последовательностью этапов (рисунок 3.1).

Этап 1. Построение вербальной модели исследуемого объекта системы с помощью UML-диаграмм и концепции SADT для анализа уровней. Формируется представление о многоуровневости и иерархичности системы.

Этап 2. Разработка концептуальной модели. На основе вербальной модели этот этап завершается построением концептуальной модели.

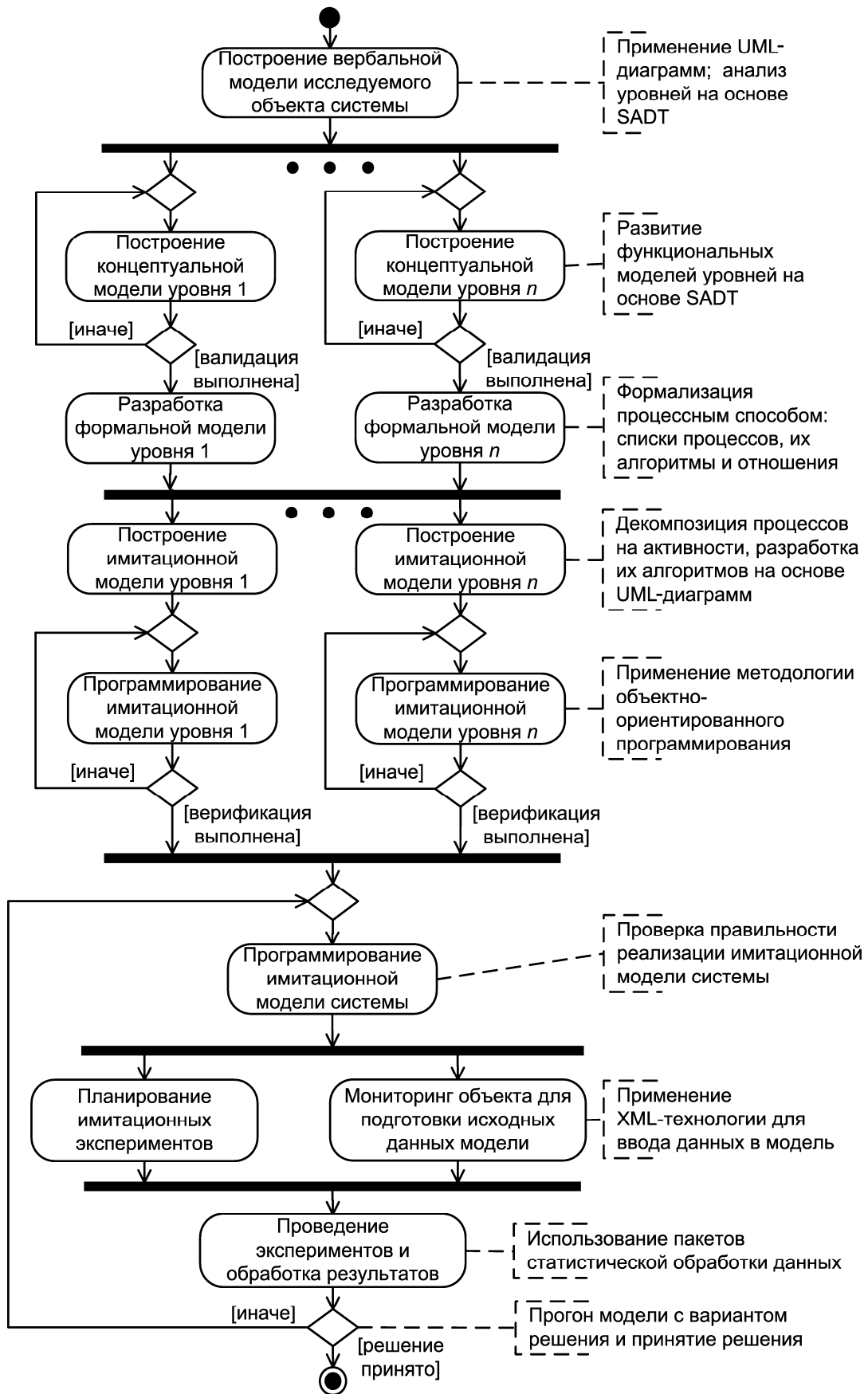


Рисунок 3.1 – Метод имитационного моделирования многоуровневой иерархической системы

Этап 3. Разработка формальной и построение имитационной модели системы. Использует методологию функционального моделирования *SADT*. Важной особенностью является постепенное введение все больших уровней детализации по мере создания диаграмм. Поэтому *SADT* в нотации *IDEF0* используется не только для формализации системы, но и для изучения и анализа выполняемых ею функций, для описания имитационной модели. При этом прорабатываются вопросы синхронизации процессов и их взаимодействия с управляющей программой моделирования (УПМ), задания начальных условий, организации сбора статистики и окончания имитации.

Этап 4. Программирование и отладка модели. Предназначен для создания программы имитационной модели на основе концепции объектно-ориентированного программирования, что позволяет создавать библиотеку моделей объектов системы, расширять их функциональные возможности, обеспечивая при этом обратную совместимость вследствие инкапсуляции деталей реализации.

Этап 5. Испытание и исследование модели в составе КИС. Требуется разработки программного обеспечения для интеграции ИМ с КИС в реальных условиях производства, что позволит решать вопросы подготовки актуальных исходных данных, ввода данных на основе XML-технологии, проведения экспериментов для решения поставленных задач.

Этап 6. Эксплуатация имитационной модели. Предполагает длительное использование модели в контуре управления промышленным предприятием и позволяет определить сочетание и место имитационных и аналитических моделей при исследовании производства. Имитация используется для уточнения теоретических аспектов в производстве, чего не позволяют аналитические методы. Возможна проверка и подтверждение адекватности аналитических моделей, отличающихся простотой эксплуатации в реальных условиях предприятия, для их практического использования [232, с. 69–80].

3.2 Вербальная модель структурных компонентов промышленного предприятия

В концепции диаграмм функционального моделирования *SADT* (Structured Analysis and Design Technique) разработан язык графического моделирования *IDEF0*, предназначенный для документирования процессов производства и отображения информации об использовании ресурсов на каждом из этапов проектирования систем [106, с. 24]. Нотация *IDEF0* реализована, например, в *CASE*-средстве BPwin AllFusion Process Modeler. Исходными элементами любой модели процесса в нотации *IDEF0* являются деятельность (activity), представленная на диаграммах блоком, и дуги (arrows) [310].

IDEF0 на этапе содержательного описания, или построения вербальной модели, позволяет эффективно представить описание системы, помогая рассуждать о предметной области одновременно эксперту и разработчикам модели, моделировать изменения в системе, информационные потоки и используемые ресурсы, логику сложных переходов в системе [306, с. 801–808]. *IDEF0*-диаграммы на последующих этапах, как отмечает Р. Р. Бартон (R. R. Barton, 2002), могут использоваться для уточнения целей эксперимента и откликов модели, определяемых целями эксперимента [283, с. 75].

Результатом применения *IDEF0* является функциональная модель. Место соединения дуги с блоком определяет тип интерфейса: управляющая информация входит в блок сверху; информация и объекты, которые подвергаются последующей обработке, – с левой стороны блока; результаты выполнения операции показаны выходящей дугой с правой стороны блока; механизмы (по терминологии *IDEF0*), представляющие собой средства выполнения функции, изображаются дугами с нижней стороны блока. При составлении содержательного описания системы отражение в модели механизмов не требуется [310].

Важной особенностью методологии *SADT*, обеспечивающей эффективность ее применения для составления вербальной модели, является постепенное введение дополнительных уровней детализации по мере создания диаграмм. Поэтому *SADT* может быть использована не только как средство эффективного представления содержательного описания системы, но и для изучения и анализа выполняемых ею функций. При этом исключается влияние организационной структуры системы на ее функциональную модель.

В контекстной диаграмме (диаграмма A0) функционирования предприятия на рисунке 3.2 в качестве функции верхнего уровня рассматривается деятельность всего предприятия. На вход поступают *ресурсы*, например, денежные средства отдельными платежами, характеризующимися суммой, датой и источником. Поток денежных средств определяется условиями заключенных контрактов на реализацию продукции, получением займов. Другие ресурсы включают материальные ресурсы (сырье, материалы), энергетические ресурсы (электроэнергия). Материальные ресурсы поступают дискретно, партиями определенного объема. Энергетические ресурсы поступают по мере использования и не запасаются. Результатом функционирования рассматриваемой системы являются *денежные средства и продукция*. Денежные средства выплачиваются в соответствии с заключенными контрактами на поставку ресурсов, видами и ставками налогов, установленным размером заработной платы, условиями получения кредитов. Продукция отгружается партиями в объеме, ассортименте и сроках в соответствии с условиями контрактов на реализацию продукции [221, с. 112–113].

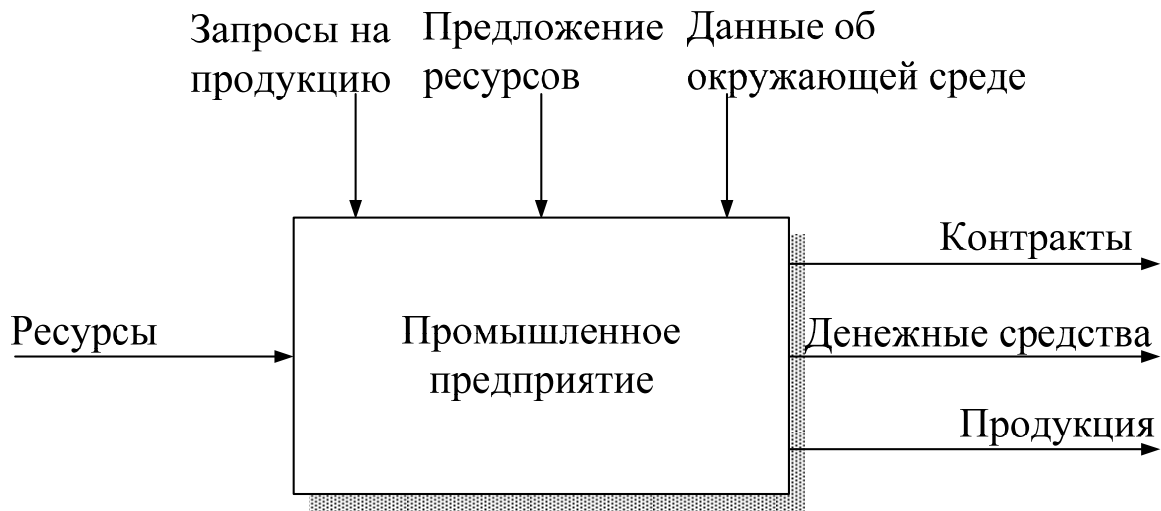


Рисунок 3.2 – IDEF0-диаграмма функционирования промышленного предприятия

В качестве управляющих воздействий выступают *запросы на продукцию, предложение ресурсов*. Запросы на продукцию характеризуются требуемыми ассортиментом и объемами продукции, сроками и формой отгрузки и оплаты. Предложение ресурсов определяет возможные цены, объемы, сроки поставки ресурсов и условия оплаты за них. Запросы на продукцию и предложение ресурсов являются основой для заключения контрактов.

Помимо вышеуказанных управляющих воздействий, используются также данные об *окружающей среде* предприятия: ставки, объекты и сроки уплаты налогов, процентная ставка кредитования, уровень инфляции, нормы амортизационных отчислений, минимальный размер заработной платы и др.

Декомпозиция контекстной диаграммы (диаграмма А1) представлена на рисунке 3.3. Выделены функции, соответствующие процессам планирования объема реализации продукции, планирования производства, заключения и выполнения контрактов, управления финансами, ведения бухгалтерского учета, ведения складского хозяйства, производства продукции.

Планирование объема реализации продукции включает в себя планирование спроса (процесс прогнозирования спроса на определенный период) и планирование продаж и производства (процесс преобразования бизнес-плана и плана спроса в планы продаж основных видов продукции). При этом производственные мощности могут не учитываться или учитываться обобщенно. План носит среднесрочный характер. В дальнейшем план продаж по видам продукции преобразуется в объемный или объемно-календарный план производства видов продукции (семейств однородной продукции). В этом плане в качестве планово-учетных единиц выступают изделия, но представления о них носят усредненный характер. Исходными данными для планирования объема реализации продукции являются объемы реализации продукции за прошедшие периоды, объемы текущих заказов [221, с. 113–115].

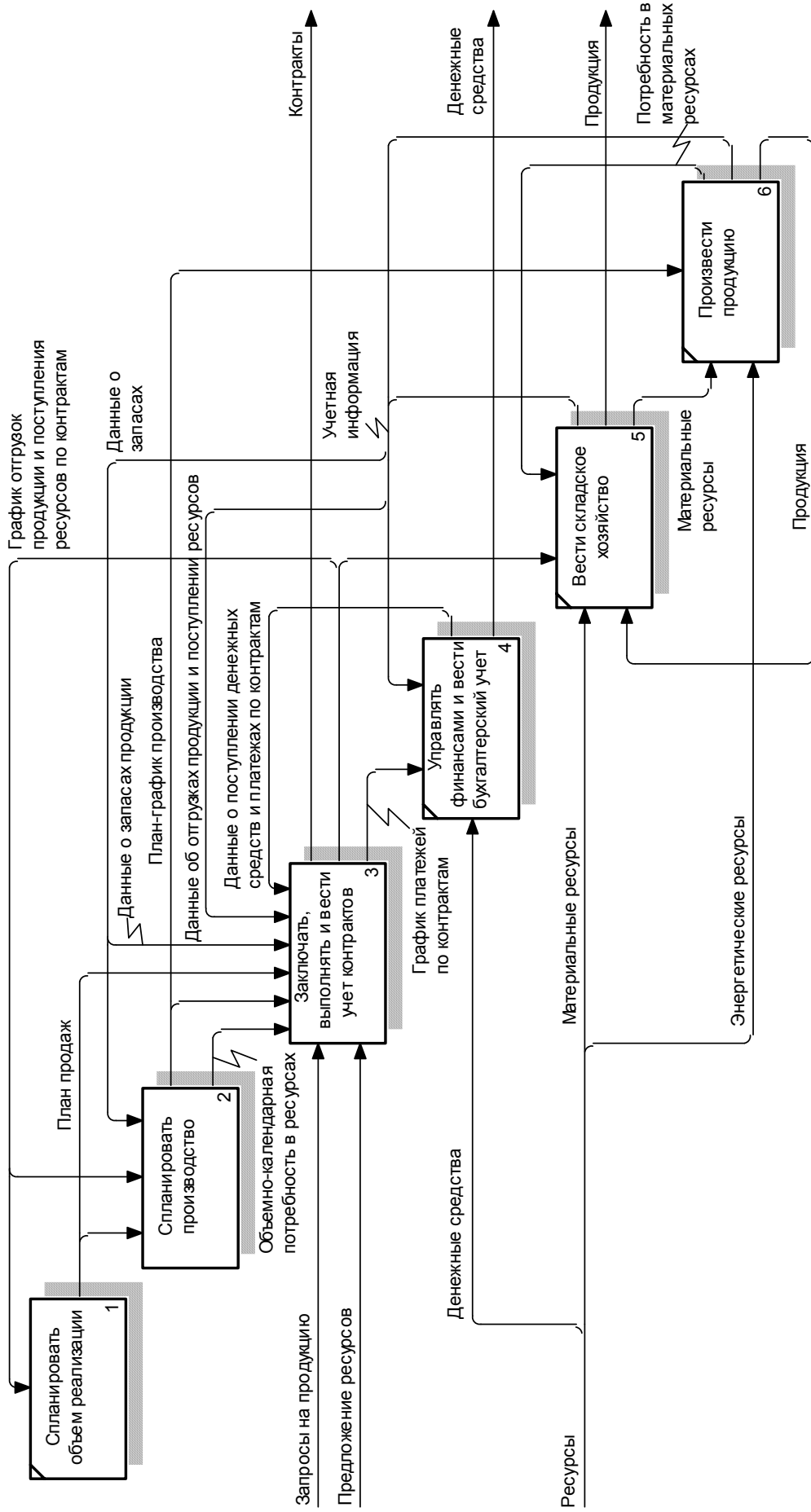


Рисунок 3.3 – Функциональная схема деятельности промышленного предприятия. Диаграмма А1

Процесс планирования производства состоит из трех основных этапов: составление плана производства, планирование потребностей в ресурсах и мощностях (рисунок 3.4). На этапе составления плана производства укрупненный план производства преобразуется в график изготовления продукции – план-график производства. Это среднесрочный объемно-календарный план, задающий объемы и сроки изготовления конкретных изделий. Исходными данными выступают план продаж (объемно-календарный план для семейств однородной продукции), объемы запасов готовой продукции на складе, объемно-календарный график отгрузок продукции по текущим контрактам, сведения об обеспеченности ресурсами и мощностями, поступающие в виде обратной связи с этапов планирования потребностей в ресурсах и мощностях.

Планирование потребностей в материальных ресурсах – определение, в количественном выражении и по срокам, потребностей в материальных ресурсах, необходимых для обеспечения графика производства продукции. Входными данными для планирования потребностей в материалах являются спецификации изделий (состав и количественные характеристики комплектующих конкретного изделия), размер текущих материальных запасов, а также график поступления ресурсов по контрактам (в виде обратной связи от процесса заключения контрактов на поставку ресурсов).

Планирование потребностей в мощностях – расчет располагаемых и потребных производственных мощностей и других видов производственных ресурсов, способных повлиять на пропускную способность предприятия. Входными данными при планировании производственных мощностей является план-график производства, а также маршрутизация выпускаемых изделий [221, с. 115].

Функциональная модель процесса «Заключать, выполнять и вести учет контрактов» приведена на рисунке 3.5.

На первом этапе (блок А31) поступающие запросы клиентов сопоставляются с планами выпуска продукции с целью определения возможности их выполнения. Для этого используется информация о плане продаж, запасах готовой продукции на складе, план-график производства продукции, график отгрузок продукции по заключенным контрактам.

Проработанные запросы с указанием возможных сроков их выполнения являются основой для заключения контрактов на отгрузку продукции (блок А32). Здесь же производится оценка предложения ресурсов, и на основании объемно-календарной потребности в ресурсах согласовываются условия и заключаются контракты на поставку материальных и энергетических ресурсов. Блок А33 представляет процесс контроля выполнения контрактов: на основе множества контрактов, данных об отгрузках продукции, поступлений ресурсов и денежных средств, платежей вырабатываются управляющие воздействия в виде графика необходимых платежей и отгрузок продукции, графика ожидаемых поступлений ресурсов и денежных средств [221, с. 118].

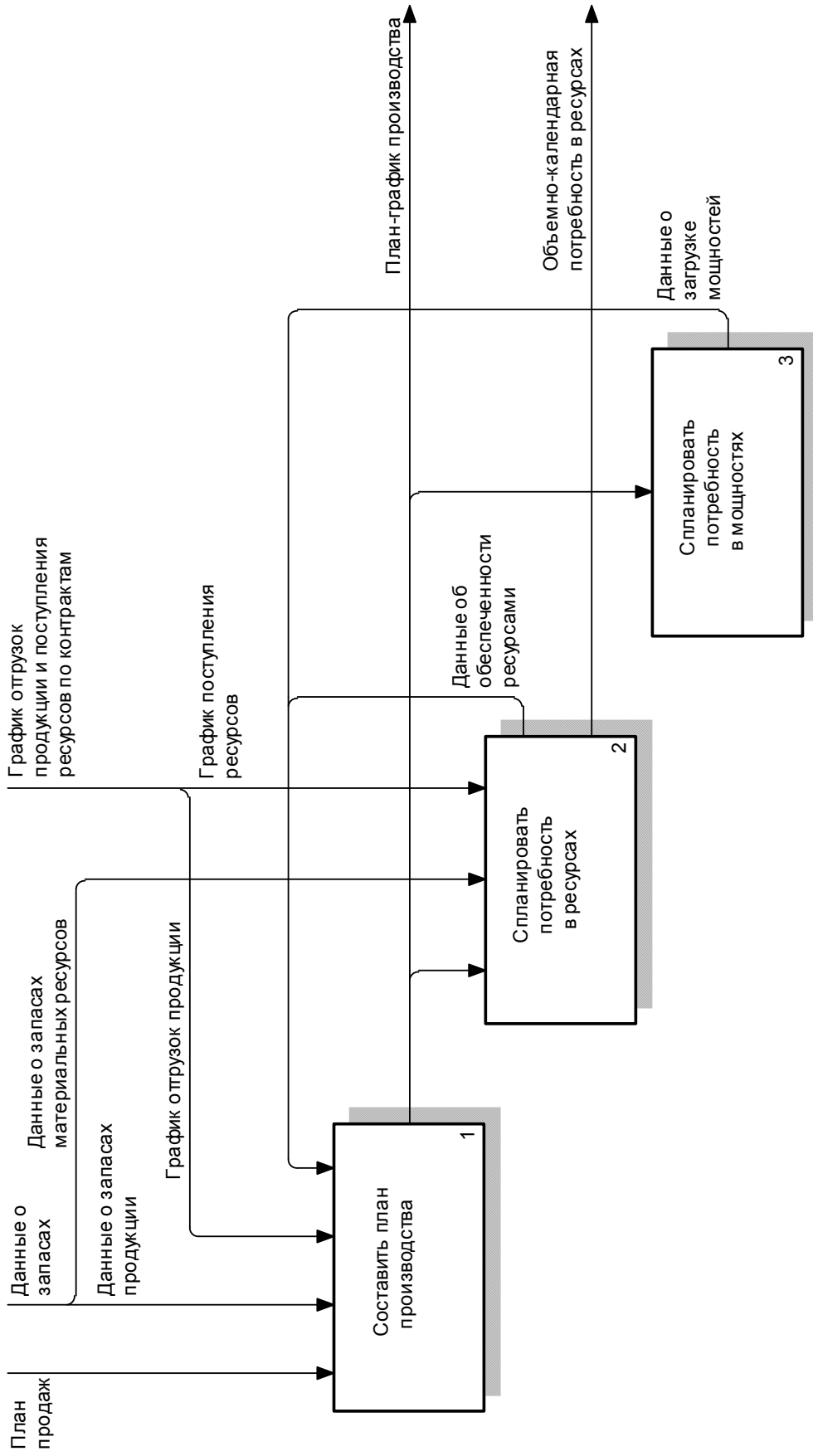


Рисунок 3.4 – Функциональная схема деятельности промышленного предприятия. Диаграмма А2 – Спланировать производство

Управление финансами и ведение бухгалтерского учета (см. рисунок 3.3) заключается в осуществлении необходимых платежей по контрактам, формировании бухгалтерской отчетности, проведении расчетов по налогам, заработной плате, а также в формировании запросов на получение кредитов и займов в случае нехватки денежных средств для покрытия текущих расходов.

3.3 Концептуальные модели аспектов функционирования промышленного предприятия

Рассматриваемые процессы снабжения, производства, сбыта продукции и управления финансами предприятия, его окружающую среду можно охарактеризовать следующими группами параметров: характеристики производственного процесса; текущее финансово-экономическое состояние предприятия; алгоритм функционирования системы управления предприятием; структура персонала и параметры системы оплаты труда; характеристики производственных фондов; характеристики финансовых ресурсов; параметры контрактов; параметры системы налогообложения; параметры денежно-кредитной системы государства; характеристики спроса на продукцию; характеристики предложения материальных и энергетических ресурсов.

Параметры производственного процесса включают: тип и метод организации производства; длительность технологического цикла; потребность в сырье, материалах и энергоресурсах в процессе изготовления продукции; трудоемкость изготовления единицы продукции; производительность оборудования; структуру затрат [221, с. 120].

Множество параметров, характеризующих *текущее финансово-экономическое состояние* предприятия, соответствует набору коэффициентов, рассчитываемых для определения показателей, входящих в состав целевой функции:

- показателями общей оценки финансового состояния являются коэффициенты текущей ликвидности K_1 , обеспеченности собственными оборотными средствами K_2 , обеспеченности финансовых обязательств активами K_3 ;

- показатели оценки финансовой устойчивости – коэффициенты абсолютной ликвидности $K_{АЛ}$, критической (быстрой) ликвидности $K_{БЛ}$, собственности K_C ;

- показатели роста и развития предприятия – темпы роста продаж (нетто) $T_{РП}$, себестоимости $T_{РС}$, прибыли от реализации продукции $T_{РПР}$, чистой прибыли $T_{РЧП}$, стоимости активов $T_{РСА}$;

- показатели эффективности использования оборотных активов – коэффициент оборачиваемости $K_{ООА}$, продолжительности оборота производственных запасов $П_{ОПЗ}$, готовой продукции $П_{ОГП}$, дебиторской задолженности $П_{ОДЗ}$;

– показатели рентабельности – рентабельность продукции по затратам $R_{ПЗ}$, выручки от продаж $R_{ВП}$, активов предприятия R_A , основного капитала $R_{ОК}$, собственного капитала $R_{СК}$.

Алгоритм функционирования системы управления предприятием определяет способ управления запасом сырья и способ формирования производственной программы.

К параметрам *трудовых ресурсов* и системы оплаты труда относятся:

- показатели структуры трудовых ресурсов: численность основных и вспомогательных рабочих, специалистов, служащих, руководителей;
- показатели производительности труда [244, с. 136–138];
- используемая форма оплаты труда;
- размер заработной платы: тарифные ставки для рабочих и должностные оклады для служащих; размер дополнительной заработной платы и других выплат.

В состав *характеристик производственных фондов* входят:

- видовой состав основных и оборотных производственных фондов;
- первоначальная, восстановительная, остаточная стоимость основных производственных фондов;
- изношенность, используемые виды и нормы амортизации основных производственных фондов;
- производственная мощность предприятия.

Параметры *финансовых ресурсов*:

- структура финансовых ресурсов по источникам формирования: размер собственного капитала, долгосрочных и краткосрочных обязательств (кредиты, займы, кредиторская задолженность);
- структура оборотных средств.

Характеристики контракта определяются в момент его заключения и в дальнейшем остаются неизменными. К ним относятся данные о видах продукции и объемах их отгрузки, ценах, формах и сроках оплаты, сроке действия контракта.

Параметры *налоговой системы* представляют собой множество видов налогов, для каждого элемента которого определена ставка и объект налогообложения, а также источник, из которого предприятие может его оплачивать [101]. Параметры *денежно-кредитной системы* государства включают возможные объемы, сроки и процентные ставки кредитов и уровень инфляции.

В состав *характеристик спроса* входят параметры потока заказов на продукцию (интенсивность поступления заказов), параметры поступающих заказов (запрашиваемый ассортимент, объемы и сроки поставки продукции, формы и сроки оплаты).

Предложение материальных и энергетических ресурсов определяется объемом, сроками поставки, ценой, формой оплаты по каждому виду сырья, материалов, энергоресурсов. Параметры денежно-кредитной систе-

мы, характеристики спроса на продукцию и предложения материальных и энергетических ресурсов характеризуют внешнюю среду предприятия, являются неуправляемыми и определяются измерениями или оценками. Параметры контрактов относятся к управляемым параметрам. Остальные параметры можно считать неизменными в ходе моделирования. Представляется важным учитывать следующие основные аспекты деятельности промышленного предприятия, характеризующие функционирование и взаимодействие соответствующих реально существующих подсистем: планирование производства; заключение и контроль выполнения контрактов; управление финансами; бухгалтерский учет; ведение складского хозяйства; производство продукции.

Планирование производства рассматривается на таких двух уровнях детализации, как обобщенное планирование объема продаж и составление плана-графика производства.

Обобщенное планирование объема продаж включает две составляющие: планирование спроса и планирование объемов продаж основных видов продукции.

Для описания *планирования спроса* используются переменные:

D_{C_i} – текущий спрос в денежном выражении на i -й вид продукции;

τ_D – период планирования спроса;

$D_{C\tau_D i}$ – планируемый суммарный объем спроса на период τ_D в денежном выражении по i -му виду продукции.

Планирование объема продаж основных видов продукции представлено с помощью следующего набора переменных:

τ_{SV} – период планирования объема продаж основных видов (групп) продукции, $\tau_{SV} \leq \tau_D$;

$R_{ПН}$ – нормативная рентабельность предприятия;

FC – сумма постоянных затрат предприятия;

(VC_{1i}) – переменные затраты на единицу i -го вида продукции;

(PC_i) – производственная мощность предприятия по i -му виду продукции;

$SV_{C\tau_{SV} i}, i = \overline{1, n_{A\Sigma}}$ – планируемый объем продаж в стоимостном выражении i -й группы продукции за период τ_{SV} ;

$SV_{\tau_{SV} i}, i = \overline{1, n_{A\Sigma}}$ – планируемый объем продаж в количественном выражении i -й группы продукции за период τ_{SV} ;

$n_{A\Sigma}$ – общее количество основных видов (групп) продукции.

Концептуальная модель обобщенного планирования объема продаж показана на рисунке 3.6.

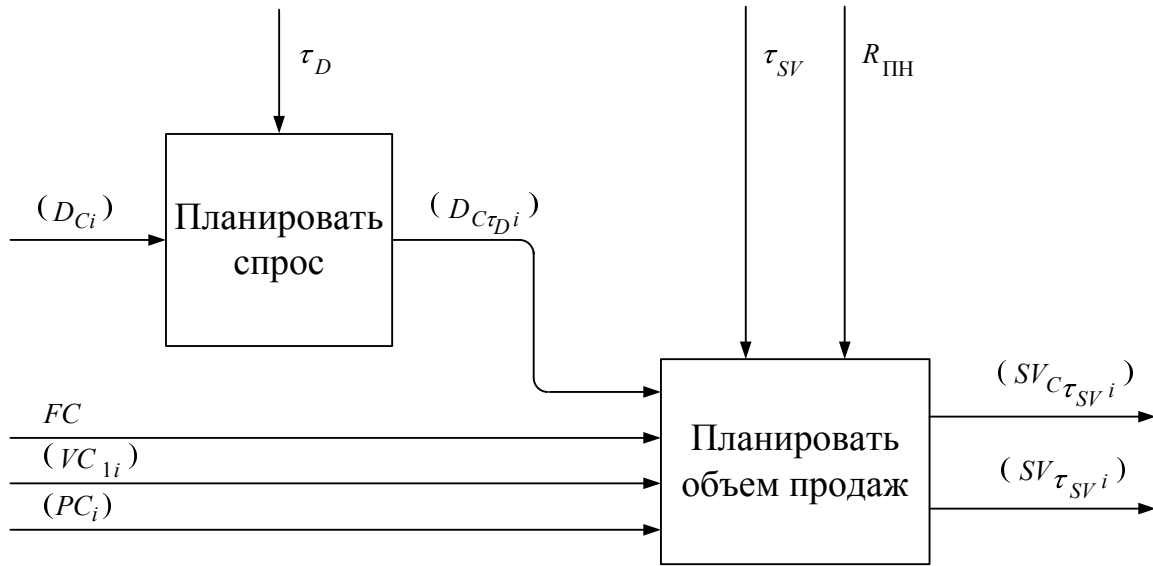


Рисунок 3.6 – Концептуальная модель обобщенного планирования объема продаж

Деятельность служб предприятия по составлению плана-графика производства включает: непосредственно составление плана-графика производства; планирование потребности в ресурсах; планирование потребности в мощностях.

Составление плана-графика производства можно представить следующим набором переменных [221, с. 124]:

τ_{MPS} – период, на который составляется основной план-график производства;

τ_{MPSU} – интервал планирования основного плана-графика производства;

$\|PV_{ij}\|$ – матрица объемов производства i -го наименования продукции в j -м интервале планирования, $i = \overline{1, n_A}$, $j = \overline{1, \tau_{MPS}/\tau_{MPSU}}$.

Для описания планирования потребности в ресурсах определены следующие переменные:

$\|RC_{ijk}\|$ – матрица временной потребности в j -м ресурсе для производства единицы i -го наименования продукции, $i = \overline{1, n_A}$, $j = \overline{1, n_R}$;

$\|RV_{ij}\|$ – матрица потребностей в i -м ресурсе в j -м интервале планирования для выполнения плана-графика производства продукции, $i = \overline{1, n_R}$, $j = \overline{1, \tau_{MPS}/\tau_{MPSU}}$ (выполнение объемно-календарной потребности в ресурсах);

$\|\Delta RV_{ij}\|$ – матрица обеспеченности ресурсами. Элементы представляют собой разность наличного и потребного количества i -го ресурса в j -м интервале планирования для выполнения плана-графика производства, $i = \overline{1, n_R}$, $j = \overline{1, \tau_{MPS}/\tau_{MPSU}}$.

Планирование потребности в мощностях заключается в формировании матриц использования производственных ресурсов:

$\|CL_{ij}\|$ – загрузки i -го оборудования в j -м интервале планирования;

$\|LT_{ij}\|$ – затрат рабочего времени i -го вида труда в j -м интервале планирования.

Концептуальная модель составления плана-графика производства приведена на рисунке 3.7.

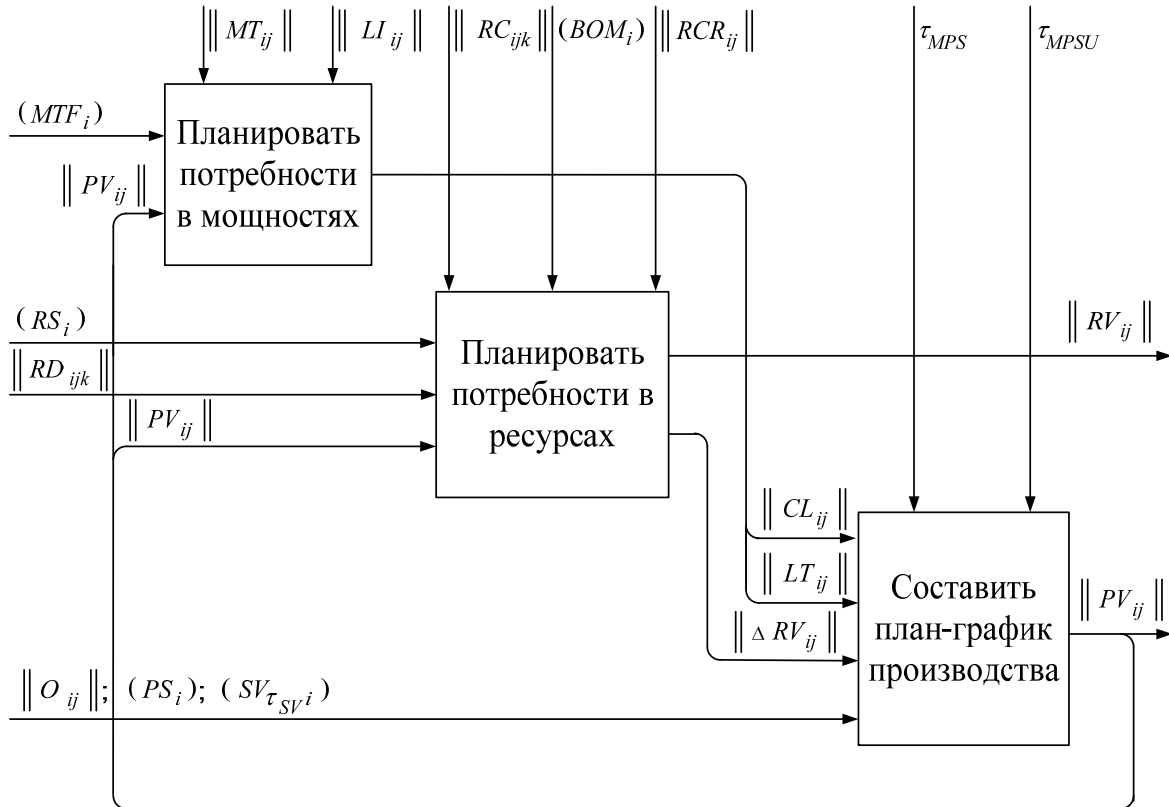


Рисунок 3.7 – Концептуальная модель составления плана-графика производства

Процесс заключения и выполнения контрактов представлен следующим набором переменных [221, с. 126–127]:

$\{PR_i\}$ – множество заявок на поставку продукции;

$\{PR_{\tau i}\}$ – множество проработанных заявок на поставку продукции,

подлежащих выполнению в периоде времени τ ;

$\{CS_i\}$ – условия заключения контрактов;

$\{C_i\}$ – множество заключенных контрактов;

$\|O_{ijk}\|$ – матрица объемов отгрузок по k -м контрактам i -го вида продукции в j -м интервале планирования, $i = \overline{1, n_A}$, $j = \overline{1, \tau_{MPS} / \tau_{MPSU}}$;

$\|RD_{ijk}\|$ – матрица объемов поставок по k -м контрактам i -го вида ре-

сурса в j -м интервале планирования, $i = \overline{1, n_R}$, $j = \overline{1, \tau_{MPS} / \tau_{MPSU}}$;

RC_i – цена i -го ресурса, $i = \overline{1, n_R}$;

$\|PI_{jk}\|$, $\|PO_{jk}\|$ – матрицы входящих и исходящих платежей по контрактам в j -м интервале планирования, $j = \overline{1, \tau_{MPS} / \tau_{MPSU}}$;

$\|O_{ijk}\|_A$ – матрица, элементами которой являются произведенные отгрузки i -го вида продукции в j -м периоде по k -му контракту;

$\|RD_{ijk}\|_A$ – матрица фактических объемов поставок по k -м контрактам i -го вида ресурса в j -м периоде;

$\|PI_{ij}\|_A$, $\|PO_{ij}\|_A$ – матрицы поступивших денежных средств и произведенных платежей по контрактам в j -м периоде, i – номер контракта.

Концептуальная модель заключения и выполнения контрактов показана на рисунке 3.8.

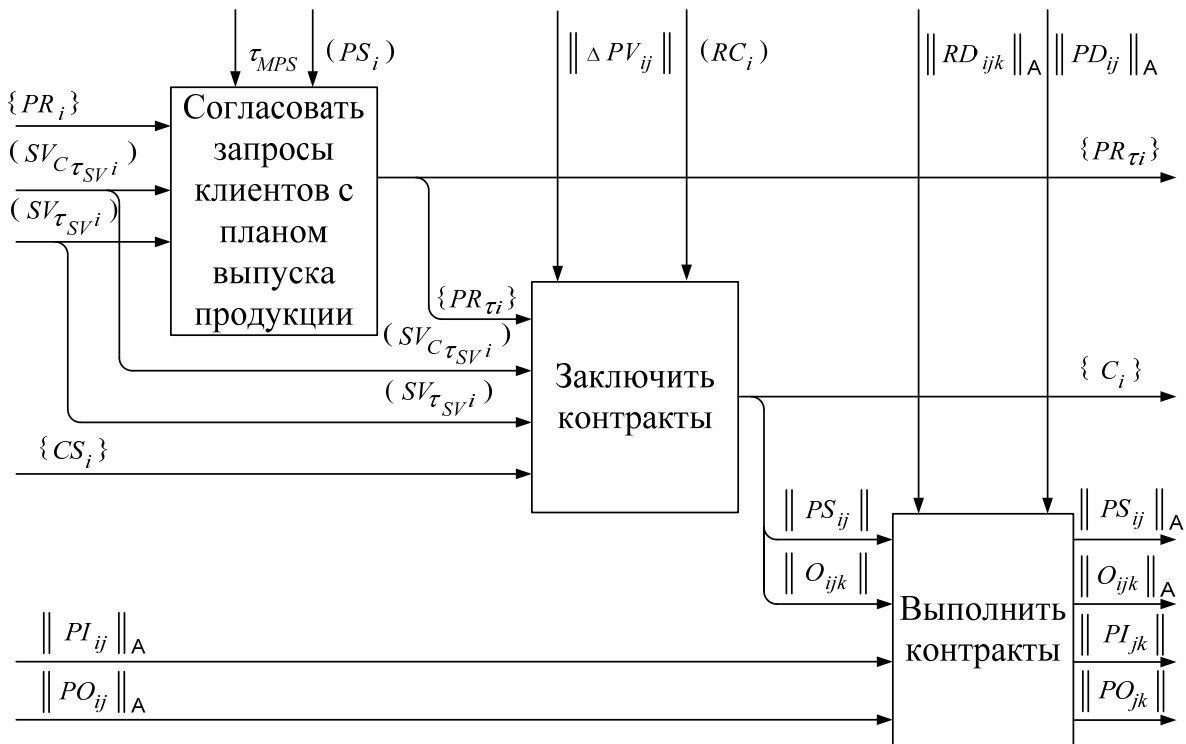


Рисунок 3.8 – Концептуальная модель заключения и выполнения контрактов

Для описания подсистемы управления финансами и бухгалтерского учета используются следующие переменные [221, с. 127–128]:

(B_i) – текущий бухгалтерский баланс, i – номер статьи баланса;

$(IS_i)_\tau$ – отчет о прибылях и убытках за период времени τ , i – номер статьи отчета;

$\|(S_v; S_c)_i\|_\tau$ – матрица объемов реализации продукции в количест-

венном S_v и стоимостном S_c выражении за период времени τ , i – вид продукции;

$\|BPAC_{io}; BPAC_{ir}; BPAC_{id}\|$ – первоначальная, восстановительная, остаточная стоимость основных производственных фондов i -го вида;

DR_i – норма амортизации основных производственных фондов i -го вида;

$\{L_i\}$ – множество кредитов банков; $L_i = (S_i; T_i; k_{\%i})$, где S_i – сумма, T_i – срок, $k_{\%i}$ – процентная ставка i -го кредита;

$\|PI_{ij}\|$, $\|PO_{ij}\|$ – матрицы поступлений денежных средств и произведенных платежей по контрактам в j -м периоде, i – номер контракта.

Концептуальная модель складского хозяйства и производства продукции в графическом виде представлена на рисунке 3.9.

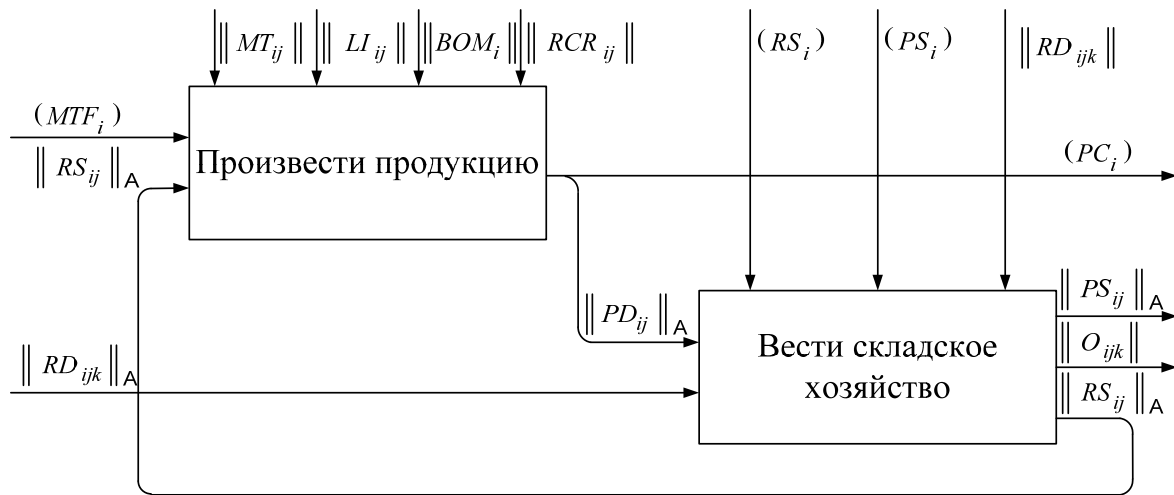


Рисунок 3.9 – Концептуальная модель складского хозяйства и производства продукции

Работа складского хозяйства представлена с помощью следующих переменных [221, с. 128–129]:

$\|RD_{ijk}\|_A$ – матрица поступлений на склад i -го вида материальных ресурсов в j -м периоде по k -м контрактам;

$\|O_{ijk}\|_A$ – матрица произведенных отгрузок продукции i -го вида в j -м периоде по k -му контракту;

$\|PD_{ij}\|_A$ – матрица поступивших на склад объемов продукции i -го вида в j -м периоде;

$\|RS_{ij}\|_A$ – матрица объемов отпущенных со склада материальных ресурсов i -го вида в j -м периоде;

$\|PS_{ij}\|_A$ – матрица отгрузок со склада i -го вида продукции в j -м периоде;

$\|O_{ijk}\|$ – матрица объемов отгрузок по k -м контрактам i -го вида продукции в j -м интервале планирования;

$\|RD_{ijk}\|$ – матрица объемов поставок по k -м контрактам i -го вида ресурса в j -м интервале планирования;

(PS_i) – запасы продукции на складе, $i = \overline{1, n_A}$;

(RS_i) – запасы материальных ресурсов на складе, $i = \overline{1, n_R}$.

Для характеристики производственного процесса определены следующие переменные:

MTF_i – фонд машинного времени i -го вида;

BOM_i – граф-спецификация i -го изделия (определяет состав и количество комплектующих i -го изделия);

RCR_{ij} – норма расхода i -го ресурса на изготовление единицы j -го вида продукции;

MT_{ij} – норма расхода фонда машинного времени i -го вида на изготовление единицы j -го вида продукции;

LI_{ij} – норма затрат (рабочего времени) i -го вида труда (по профессиям) на изготовление единицы j -го вида продукции;

(RC_i) – потребность в ресурсах в текущем периоде, $i = \overline{1, n_R}$.

Внешняя среда рассматривается состоящей из таких основных элементов, как потребители продукции предприятия, поставщики материальных и энергетических ресурсов, денежно-кредитная и налоговая подсистемы государства.

Потребители продукции предприятия формируют поток заказов на продукцию предприятия со следующими характеристиками [221, с. 130]:

V_Z – интенсивность поступления заказов;

$P_{Ai}, i = \overline{1, n_A}$ – вероятности запроса i -го наименования продукции;
 n_A – общее количество наименований продукции;

$\xi_{O_i}, \xi_{T_i}, i = \overline{1, n_A}$ – случайные величины, характеризующие запрашиваемые объемы и сроки поставки i -го наименования продукции;

Характеристики потока заказов определяются путем расчетов на основе измерений реального потока заказов или соответствующих оценок.

Поставщики материальных и энергетических ресурсов определяют предложение материальных и энергетических ресурсов, для описания которого используется следующий набор переменных:

$T_{Ri}, i = \overline{1, n_R}$ – минимальный срок поставки i -го ресурса;

$C_{Ri}, i = \overline{1, n_R}$ – цена i -го ресурса;

n_R – общее количество используемых видов ресурсов.

В элементе «государство» выделены следующие две основные подсистемы, с которыми взаимодействует предприятие: денежно-кредитная система, налоговая система.

Денежно-кредитная система представлена следующим набором переменных:

$k_{\%}$ – уровень процентной ставки за кредит для рассматриваемого предприятия;

$\tau_{\%}$ – период уплаты процентов за кредит;

k_{α} – уровень инфляции.

Уровень процентной ставки определяется следующим образом [40]:

$$k_{\%} = k_{\%}^* + ПИ + ПКР + ПЛ + ПРСП, \quad (3.1)$$

где $k_{\%}^*$ – реальная свободная от риска процентная ставка;

ПИ – премия за инфляцию;

ПКР – премия за кредитный риск;

ПЛ – премия за ликвидность;

ПРСП – премия за риск, связанный со сроком погашения.

Значения уровней процентной ставки и инфляции, закон их изменения задаются перед началом моделирования на основе экспертных оценок развития ситуации на финансовых рынках.

Налоговая система представлена множеством налогов:

$$N = \{(k_{Ni}; o_{Ni}; s_{Ni}; t_{Ni})\}, i = \overline{1, n_N}, \quad (3.2)$$

где k_{Ni} – ставка i -го налога;

$o_{Ni} \in O_N$ – объект i -го налога из множества возможных объектов налогообложения;

$s_{Ni} \in S_N$ – источник, из которого предприятие может оплачивать i -й налог, из множества возможных источников уплаты налогов;

t_{Ni} – срок оплаты i -го налога;

n_N – общее количество налогов.

Виды налогов, их ставки, объекты налогообложения, источники оплаты определяются действующим законодательством.

3.4 Имитационная модель промышленного предприятия

3.4.1 Процессный способ имитации динамики взаимодействия компонентов промышленного предприятия. На промышленном предприятии, как правило, все функциональные действия $w^{\ell\pm\tau}$ подсистем $S^{\ell\pm\tau}$ на $\ell\pm\tau$ -уровнях системы различны. Функционирование подсистем представляет собой последовательность $w^{\ell\pm\tau}$, которая выполняется на некотором временном интервале t^ℓ . В результате выполнения $w^{\ell\pm\tau}$ система переходит в состояние ${}_k C^\ell$. Каждое из состояний ${}_k C^\ell$ связано с соответствующим компонентом $K_i^{\ell\pm\tau} \in S^{\ell\pm\tau}$. При построении ИМ $w^{\ell\pm\tau}$ аппроксимируются упрощенными функциональными действиями $w^{\ell\pm\tau*}$. В ИМ каждое $w^{\ell\pm\tau*}$ описывается в общем случае некоторым алгоритмом $AL^{\ell\pm\tau}$ и выполняется за время Δt . Пару $(AL^{\ell\pm\tau}, \Delta t)$ обычно называют активностью ИМ и обозначают $AK^{\ell\pm\tau}$ [114, с. 10].

Условия появления состояний, приводящих к выполнению $w^{\ell\pm\tau}$, также индивидуальны. У каждого компонента существует определенная последовательность выполнения $w^{\ell\pm\tau}$. В любой момент времени в данном компоненте может выполняться только одно $w^{\ell\pm\tau}$.

При процессном способе организации квазипараллелизма декомпозиция системы определяется следующим. Во-первых, каждому компоненту $K_i^{\ell\pm\tau} \in S^{\ell\pm\tau}$ соответствует свой процесс. Таким способом достигается соответствие структуры ИМ и реальной системы. Во-вторых, все $w^{\ell\pm\tau}$ аппроксимируются соответствующими активностями исходя из величины ошибки аппроксимации. Если ошибка аппроксимации некоторых $w^{\ell\pm\tau}$ на интервале Δt велика, то этот интервал разбивается на несколько более мелких интервалов, на каждом из которых часть $w^{\ell\pm\tau}$ аппроксимируется своей активностью. Таким образом, компоненты модели делят на последовательность активностей, для которых ошибки аппроксимации $w^{\ell\pm\tau}$ находятся в допустимых пределах для данного исследования. Важным для процессного способа имитации является то, что условия свершения событий индивидуальны для каждого компонента реальной системы $K_i^{\ell\pm\tau} \in S^{\ell\pm\tau}$ и активности $AK^{\ell\pm\tau}$ тесно взаимосвязаны между собой.

Процессным способом можно организовать имитацию любых сложных систем. Однако процессный способ наиболее эффективен в тех случаях, когда требуется высокий уровень детализации выполнения $w^{\ell\pm\tau}$ при их аппроксимации с помощью $AL^{\ell\pm\tau}$ и сама ИМ используется для поиска узких мест в системе. При такой постановке задачи на моделирование очень важно соблюдение структуры модели и объекта моделирования, что обеспечивается процессным способом моделирования. Процессный способ имитации обладает хорошими изобразительными возможностями при осуществлении многоуровневого и модульного подхода к моделированию систем. Это преимущество возрастает по мере роста размеров модели.

Всю имитационную модель можно представить в виде набора опи-

саний процессов, каждый из которых соответствует одному определенному классу. Между компонентами и отдельными алгоритмами их функционирования могут быть установлены информационные и управляющие связи. Алгоритм функционирования ИМ представляется последовательным взаимодействием процессов и УПМ. Причем в процессы объединяются связанные между собой активности, которые определяют функционирование одного и того же компонента модели. Таким образом, достигается полное соответствие компонентов реальной системы и ее ИМ, когда каждому компоненту объекта моделирования соответствует свой процесс [11, с. 11–16].

3.4.2 Декомпозиция функционирования компонентов системы.

Декомпозиция системы на компоненты проведена согласно функциональной схеме деятельности промышленного предприятия. Функциональные действия объединены по принадлежности к одному реальному компоненту системы. В результате получен следующий список компонентов: K_1 – планирование производства; K_2 – заключение и выполнение контрактов; K_3 – финансы и бухгалтерский учет; K_4 – производство продукции. Фактически отдельный компонент модели соответствует каждому блоку диаграммы верхнего уровня функциональной схемы (см. рисунок 3.3), за исключением функциональных действий по планированию объема реализации и формированию плана-графика производства, которые отнесены к одному компоненту, и ведению складского хозяйства, для которых отдельный компонент не выделяется.

Во внешней среде выделены следующие компоненты: потребители продукции (формируют поток заказов на продукцию предприятия); поставщики материальных и энергоресурсов; государство (денежно-кредитная и налоговая подсистемы).

Функциональные действия, соответствующие процессу выполнения контрактов внешними контрагентами, отнесены к компоненту K_2 . Это позволяет моделировать все функциональные действия по выполнению контрактов с помощью одного компонента ИМ. Полный перечень компонентов ИМ приведен в таблице 3.1. Все компоненты взаимодействуют с другими посредством информационных и управляющих связей, которые соответствуют связям между блоками функций на функциональной схеме предприятия.

Каждому компоненту объекта моделирования соответствует свой процесс. Функциональные действия отдельных компонентов модели могут выполняться одновременно, поэтому имитацию таких функциональных действий необходимо выполнять отдельными процессами. При этом каждый реальный компонент будет представлен одним или несколькими процессами.

Конкретный состав процессов, соответствующих компоненту K_4 , зависит от типа производственной структуры (предметный, технологический

и предметно-технологический) и особенностей технологического процесса на конкретном предприятии. В случае предметной производственной структуры состав процессов компонента K_4 соответствует специализации подразделений предприятия на изготовление определенного изделия или группы однотипных изделий.

Таблица 3.1 – Перечень процессов ИМ промышленного предприятия

Компонент	Обозначение	Наименование
K ₁	ПК ₁₁ : АК ₁₁	<i>Планирование производства:</i> составление плана-графика производства
	ПК ₂₁ : АК ₂₁	<i>Управление поставками ресурсов:</i> размещение заявок на поставку ресурсов
K ₂	ПК ₂₂ : АК _{22.1} АК _{22.2} АК _{22.3}	<i>Поставка ресурсов:</i> предоплата поставки выполнение поставки оплата поставки
	ПК ₂₃ : АК ₂₃	<i>Поступление заявок на отгрузку продукции:</i> размещение заявок на отгрузку продукции
	ПК ₂₄ : АК _{24.1} АК _{24.2} АК _{24.3} АК _{24.4}	<i>Реализация продукции:</i> генерация отгрузки отгрузка продукции оплата продукции предоплата продукции
	ПК ₃₁ : АК ₃₁	<i>Выплата заработной платы (ЗП):</i> расчет и выплата ЗП производственным рабочим
K ₃	ПК ₃₂ : АК ₃₂	<i>Уплата налогов:</i> расчет и уплата налогов
	ПК ₃₃ : АК ₃₃	<i>Оплата постоянных затрат и издержек:</i> расчет и оплата постоянных затрат и издержек
	ПК ₃₄ : АК ₃₄	<i>Получение кредитов:</i> получение кредита
	ПК ₃₅ : АК _{35.1} АК _{35.2}	<i>Обслуживание кредита:</i> поступление денежных средств по кредиту выплата процентов и погашение кредита
	ПК ₃₆ : АК ₃₆	<i>Бухгалтерский учет:</i> формирование статей бухгалтерского учета
	K ₄	ПК ₄₁ : АК ₄₁
ПК ₄₂ : АК _{42.1} АК _{42.2}		<i>Технологический процесс:</i> запуск партии продукции в производство завершение производства и передача готовой продукции на склад
K ₅	ПК ₅₁ : АК ₂₃	<i>Инфляция:</i> расчет индекса инфляции и корректировка цен
Примечание – ПК – процесс компонента; АК – активность процесса		

Если предприятие имеет технологическую структуру, то процессы выделяются в соответствии со специализацией цехов предприятия на выполнение определенной части технологического процесса или отдельной стадии производственного процесса. В первом случае отдельный процесс используется для моделирования производства изделия определенного вида, во втором – определенной части технологического процесса.

При предметной производственной структуре связи между процессами определяются спецификацией выпускаемых изделий, при технологической структуре производства – конкретным технологическим процессом. Если предприятие имеет смешанную производственную структуру, то часть процессов используется для моделирования производства отдельных видов продукции, а часть – для моделирования определенных технологических операций. Уровень детализации моделирования производства определяется выбранными показателями качества [221, с. 137–139].

3.4.3 Алгоритмизация компонентов имитационной модели. Отдельные процессы, соответствующие основным компонентам имитационной модели, рассмотрены более подробно.

Процесс «Планирование производства» отражает функциональные действия по составлению плана-графика производства. Планирование осуществляется на период τ_{MPS} вперед по интервалам τ_{MPSU} . Поскольку история планирования сохраняется, вначале выполняется подготовка структур данных плана производства: добавляется новая ячейка и очищаются предыдущие данные за период $(\tau_{MPS} - \tau_{MPSU})$. Далее формируется новый план производства каждого вида продукции исходя из объемов запланированных отгрузок в каждом интервале планирования. Полученный план производства корректируется в меньшую сторону на величину запасов готовой продукции на складе, и определяется потребность в ресурсах для его выполнения в каждом интервале τ_{MPSU} . Последним выполняется оператор синхронизации $WAIT(\tau_{MPSU})$, по которому процессу назначается момент следующей активизации по окончании ожидания длительностью τ_{MPSU} , соответствующей текущему интервалу планирования [221, с. 139].

Схема алгоритма процесса «Планирование производства» приведена на рисунке 3.10.

Процесс «Управление производством» (рисунок 3.11) моделирует действия по определению времени и объемов начала производства по каждому виду продукции. На основе плана производства и длительности технологического цикла с учетом объемов незавершенного производства определяется количество продукции, изготовление которой необходимо начать в текущем периоде. В случае ненулевого значения этой величины выполняется оператор $CREATE_PROCESS (AK_{42}, 0)$, посредством которого

осуществляется обращение к УПМ с целью создания процесса «Технологический процесс» [241, с. 54–55].

В результате вызова оператора CREATE_PROCESS создается новый процесс с начальной активностью АК₄₂ – «Запуск партии продукции в производство» и назначается момент его первой активизации, равный значению локального времени текущего процесса. Описанная выше последовательность действий выполняется для всех видов продукции, после чего вызывается оператор синхронизации процесса WAIT(τ_{MPSU}) [221, с. 139–142].

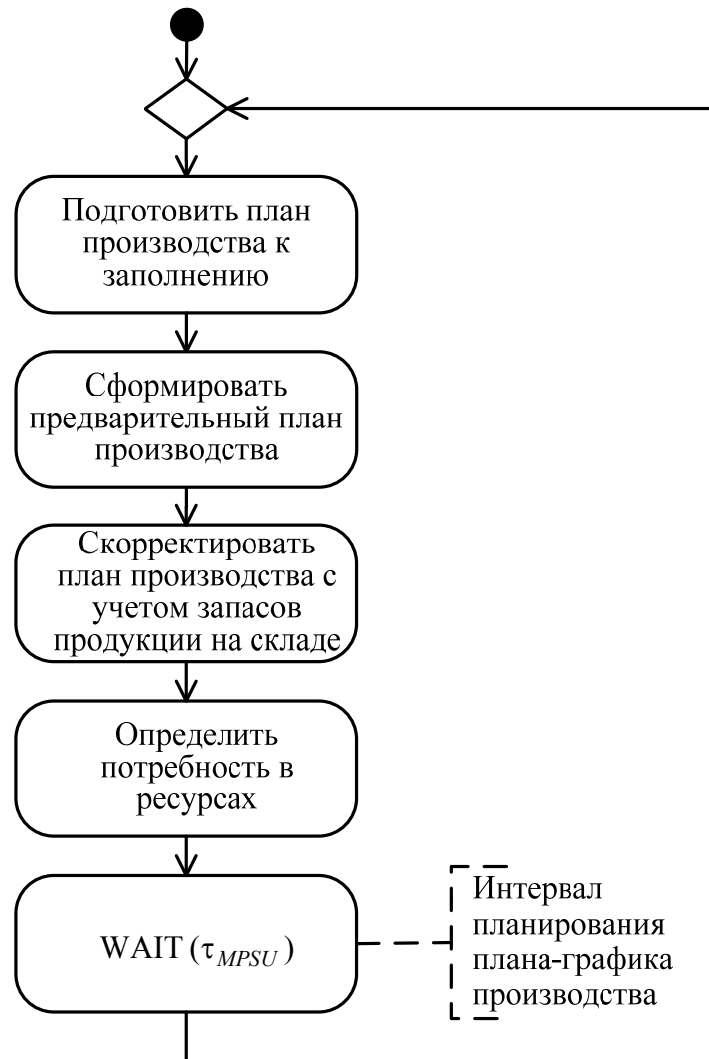


Рисунок 3.10 – Схема алгоритма процесса «Планирование производства»

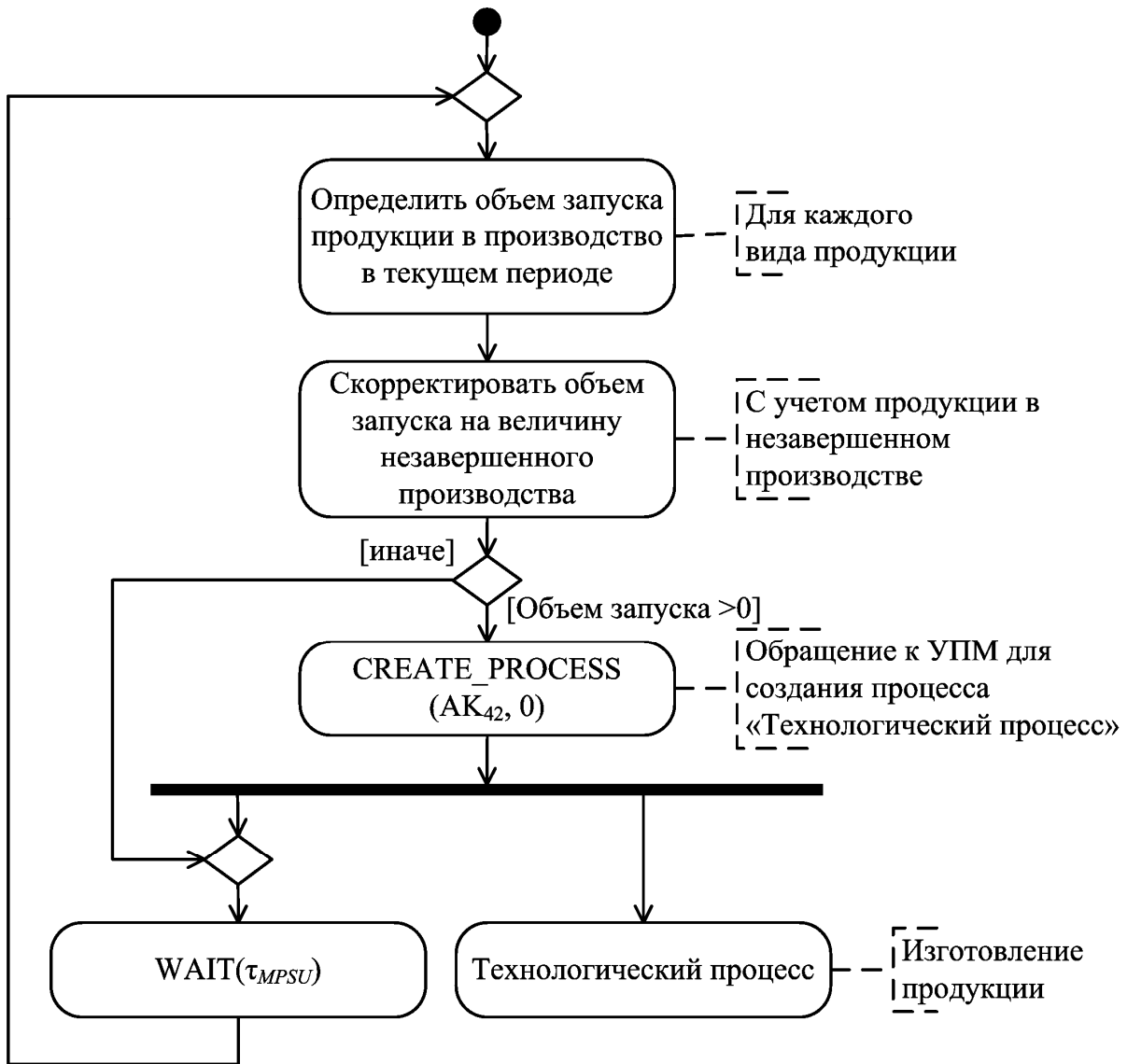


Рисунок 3.11 – Схема алгоритма процесса «Управление производством»

Процесс ПК₄₂ «Технологический процесс» имитирует выполнение технологических операций для производства заданной партии продукции (рисунок 3.12). Первым оператором процесса является проверка наличия на складе необходимых для производства ресурсов. В случае их недостачи процесс переводится в состояние ожидания до тех пор, пока не будет выполнено условие

$$RS_i \geq O_j \cdot RCR_{ij}, i = \overline{1, n_R}, \quad (3.3)$$

т. е. пока запасы ресурсов будут не меньше количества, необходимого для производства партии продукции объемом O_j исходя из нормы RCR_{ij} расхода i -го ресурса на изготовление единицы продукции j -го вида.

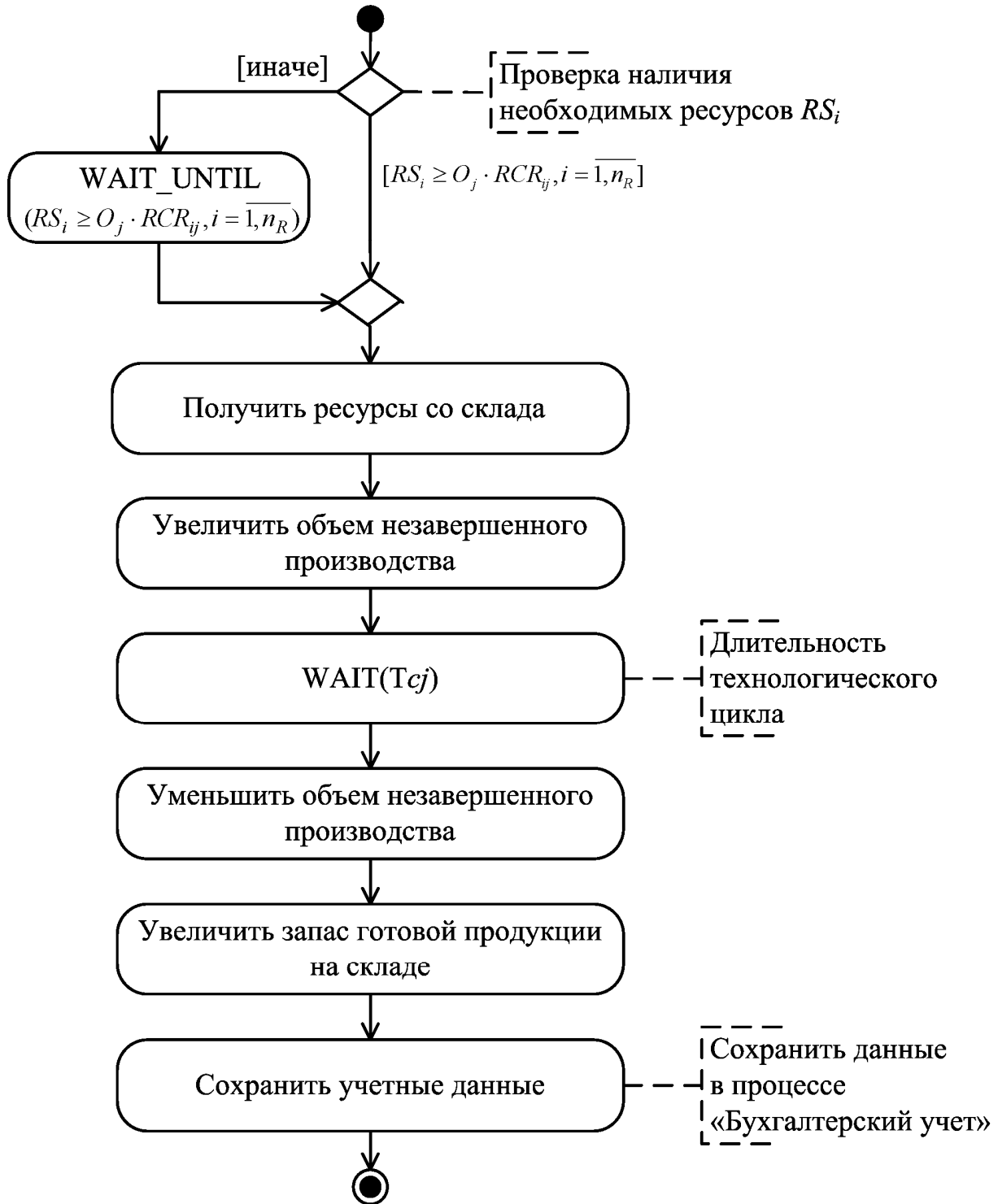


Рисунок 3.12 – Схема алгоритма процесса «Технологический процесс»

Для этого вызывается оператор синхронизации `WAIT_UNTIL` с соответствующим условием в качестве параметра. Затем размер запасов ресурсов на складе уменьшается на величину $(O_j \cdot RCR_{ij}), i = \overline{1, n_R}$, а объем незавершенного производства увеличивается на величину O_j . Вызовом оператора `WAIT(TCj)` процесс переводится в состояние ожидания на время T_{Cj} ,

равное длительности технологического процесса для j -го вида продукции. При следующей активизации процесса выполняются операторы, уменьшающие объем незавершенного производства, увеличивающие запас готовой продукции на складе и сохраняющие необходимые данные для бухгалтерского учета. После этого процесс завершается вызовом оператора END, который удаляет его из списка процессов УПМ [221, с. 142–143].

Алгоритм процесса «Управление поставками ресурсов» имеет вид, аналогичный алгоритму процесса «Управление производством» (см. рисунок 3.11). В данном случае цикл выполняется для всех типов ресурсов. Исходя из графика потребностей в ресурсах, который формируется процессом «Планирование производства», и сроков поставки, а также учитывая размеры складских запасов и уже размещенных заказов, определяется объем заказа, который необходимо сделать в текущем периоде. Если этот требуемый объем заказа больше нуля, то размещается заказ. Для этого посредством вызова оператора CREATE_PROCESS(AK₂₂, 0) создается процесс «Поставка ресурсов». Затем цикл повторяется для следующего типа ресурса. По завершении обработки всех ресурсов вызывается оператор синхронизации WAIT(τ_{MPSU}).

Алгоритм процесса «Поставка ресурсов» (рисунок 3.13) описывает функциональные действия при выполнении контракта на поставку ресурсов. При этом можно выделить три основных блока: предварительная оплата, поставка ресурсов и оплата по факту, отсрочка платежа.

В случае, если контрактом предусматривается предварительная оплата, осуществляется проверка наличия необходимых денежных средств на расчетном счете предприятия ($S_{пред} > S_{p/c}$). Если необходимая сумма отсутствует, фиксируется недостаток денежных средств в размере ($S_{пред} - S_{p/c}$) и выполнение процесса приостанавливается посредством оператора WAIT_WHILE ($S_{пред} > S_{p/c}$). При следующей активизации процесса необходимая сумма снимается с расчетного счета и управление вновь передается УПМ вызовом оператора WAIT ($T_{пред}$), где $T_{пред}$ – срок предварительной оплаты. Если же контракт не предусматривает предварительную оплату, то выполняется оператор WAIT ($T_{пост}$), где $T_{пост}$ – период времени с момента подачи заявки до поставки ресурсов. Далее следуют операторы, модифицирующие величину складских запасов, что соответствует поставке ресурсов. Группы операторов, реализующие оплату по факту и отсрочку платежа с синхронизацией по объему наличных денежных средств, соответствуют группе операторов для представления предварительной оплаты. Последним следует оператор END, который завершает данный процесс и удаляет его из списка процессов.

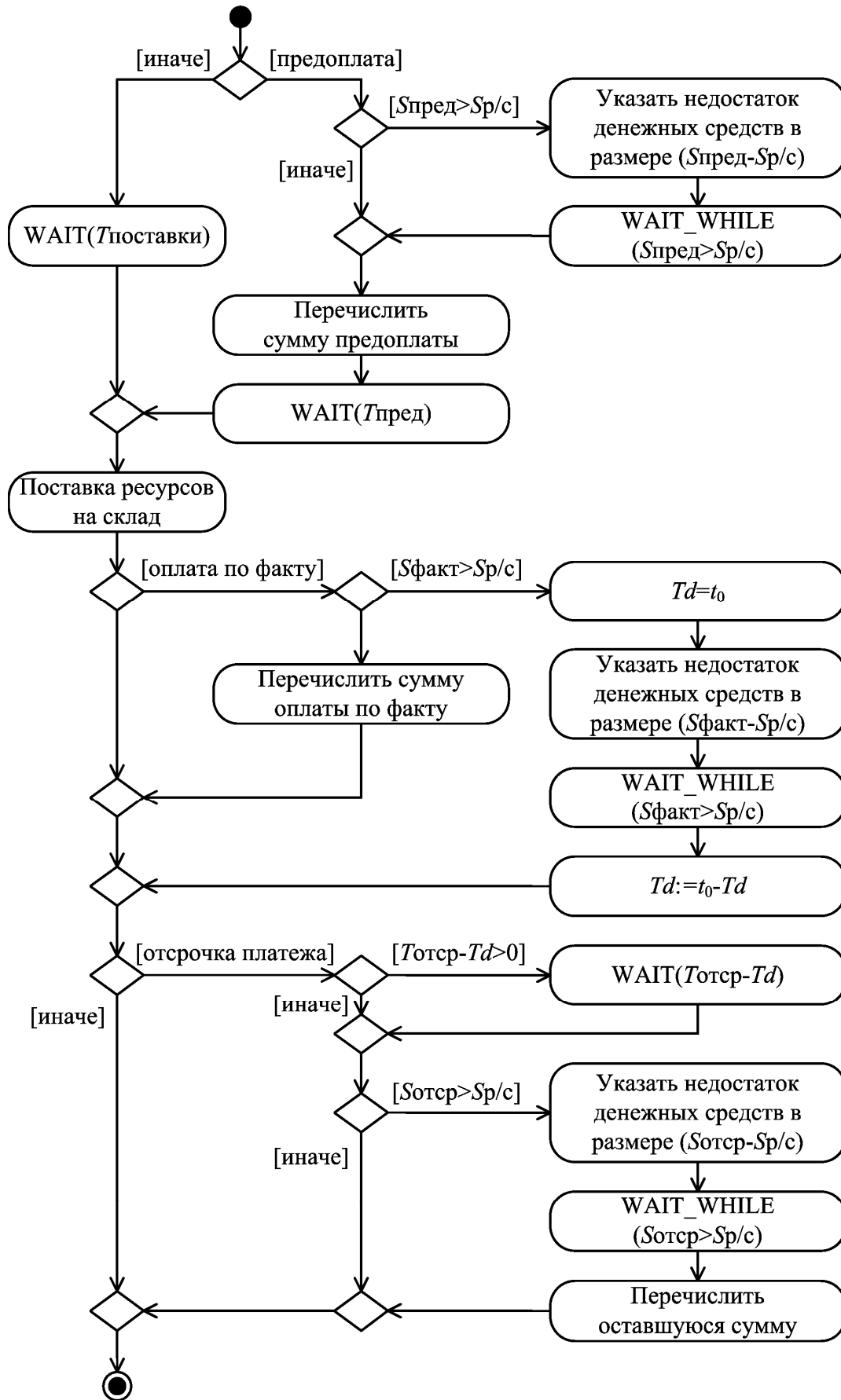


Рисунок 3.13 – Схема алгоритма процесса «Поставка ресурсов»

Алгоритм процесса «Реализация продукции» (рисунок 3.14) в целом аналогичен алгоритму процесса «Поставка ресурсов». Он описывает функциональные действия при выполнении контракта на реализацию продукции. Основное отличие заключается в сделанном предположении платежеспособности заказчика и, как следствие, отсутствии соответствующих операторов синхронизации. Кроме того, осуществляется проверка наличия необходимого количества отгружаемой продукции, а не выплачиваемой суммы денежных средств.

Процессы «Уплата налогов», «Выплата заработной платы», «Оплата постоянных затрат и издержек» представлены схожими алгоритмами (рисунок 3.15). Первыми следуют операторы, с помощью которых рассчитывается сумма денежных средств, подлежащая к уплате: сумма налоговых отчислений, заработной платы производственным рабочим и постоянных затрат. Далее следуют операторы, осуществляющие проверку наличия достаточной суммы денежных средств на расчетном счете.

В случае отсутствия достаточных денежных средств выплачивается имеющаяся сумма, указывается недостаток денежных средств, а остаток сохраняется (переносится на следующий период). Затем осуществляется передача управления УПМ посредством оператора $WAIT(T_v)$, где T_v – периодичность выплат, $T_v = 1$ мес. Для процесса «Уплата налогов» при недостатке денежных средств периодичность $T_v = 1$ дн., так как штрафные санкции начисляются за каждый день просрочки платежа и необходимая сумма выплачивается при первой возможности [221, с. 146–147].

Процесс «Получение кредитов» активизируется только в случае недостатка в денежных средствах, поэтому алгоритм его реализации (рисунок 3.16) начинается с вызова оператора синхронизации $WAIT_WHILE(S_{\text{деф}} = 0)$, где $S_{\text{деф}}$ – величина, определяющая текущий дефицит наличных денежных средств. Затем группа операторов реализует оформление заявки на получение кредита в размере $S_{\text{деф}}$. Процесс завершается вызовом оператора $CREATE_PROCESS(АК_{35}, T_k)$. В результате создается новый процесс «Обслуживание кредита» с начальной активностью $АК_{35}$ и активизацией через время T_k получения кредита. Следующий оператор обнуляет величину $S_{\text{деф}}$ потребности в денежных средствах, и управление снова передается на начало алгоритма.

Алгоритм процесса «Обслуживание кредита» (рисунок 3.17) реализует функциональные действия, соответствующие получению, выплате процентов и возврату кредита. Первый оператор переводит сумму денежных средств по кредиту на расчетный счет предприятия (получение кредита). Последующие операторы реализуют обслуживание полученного кредита – периодическую выплату денежных средств в соответствии с условиями кредита. Первым следует вызов оператора $WAIT(T_{k1})$, где T_{k1} – периодичность выплат денежных средств. По прошествии времени T_{k1} выполняется расчет суммы требуемого платежа.

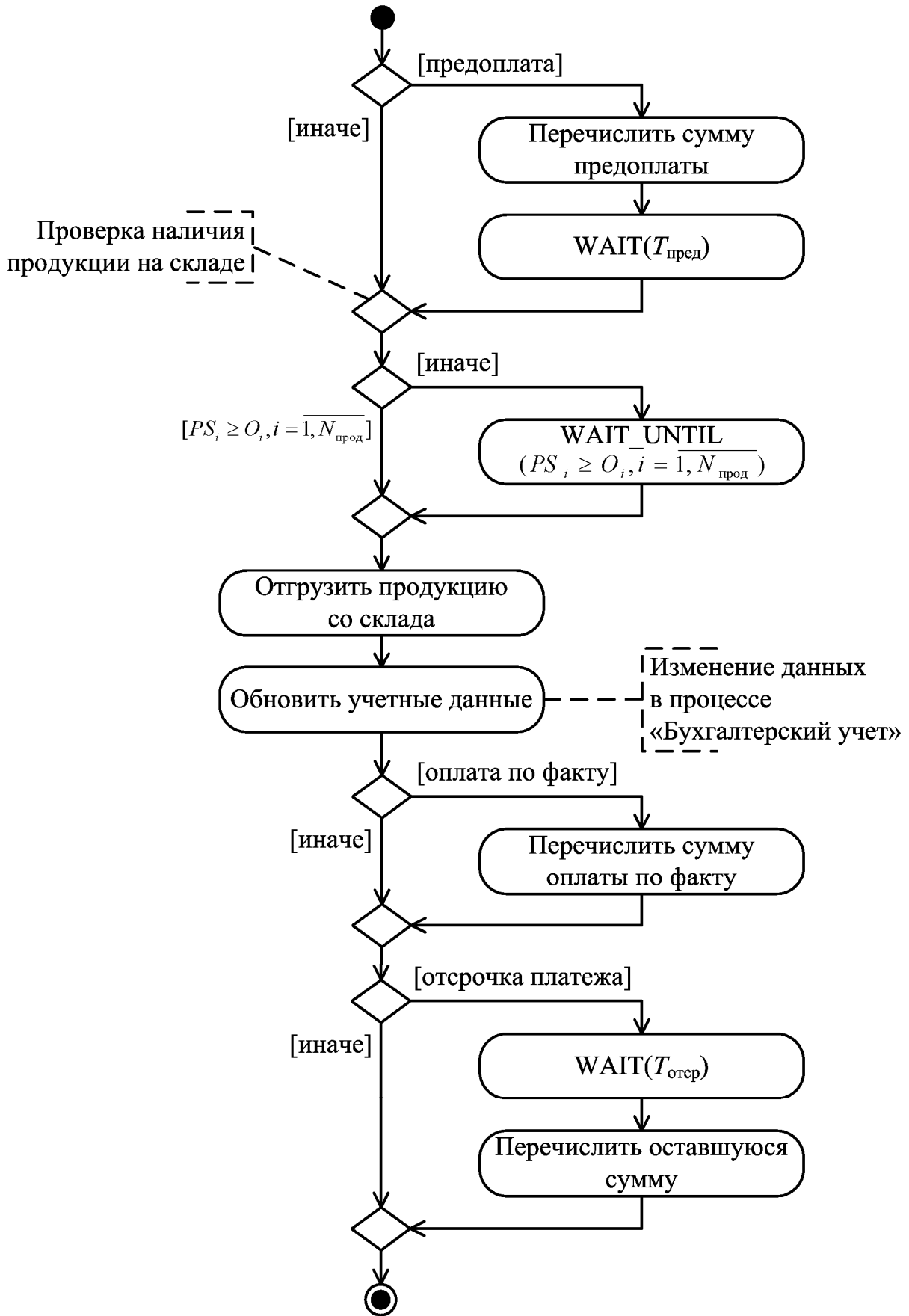


Рисунок 3.14 – Схема алгоритма процесса «Реализация продукции»

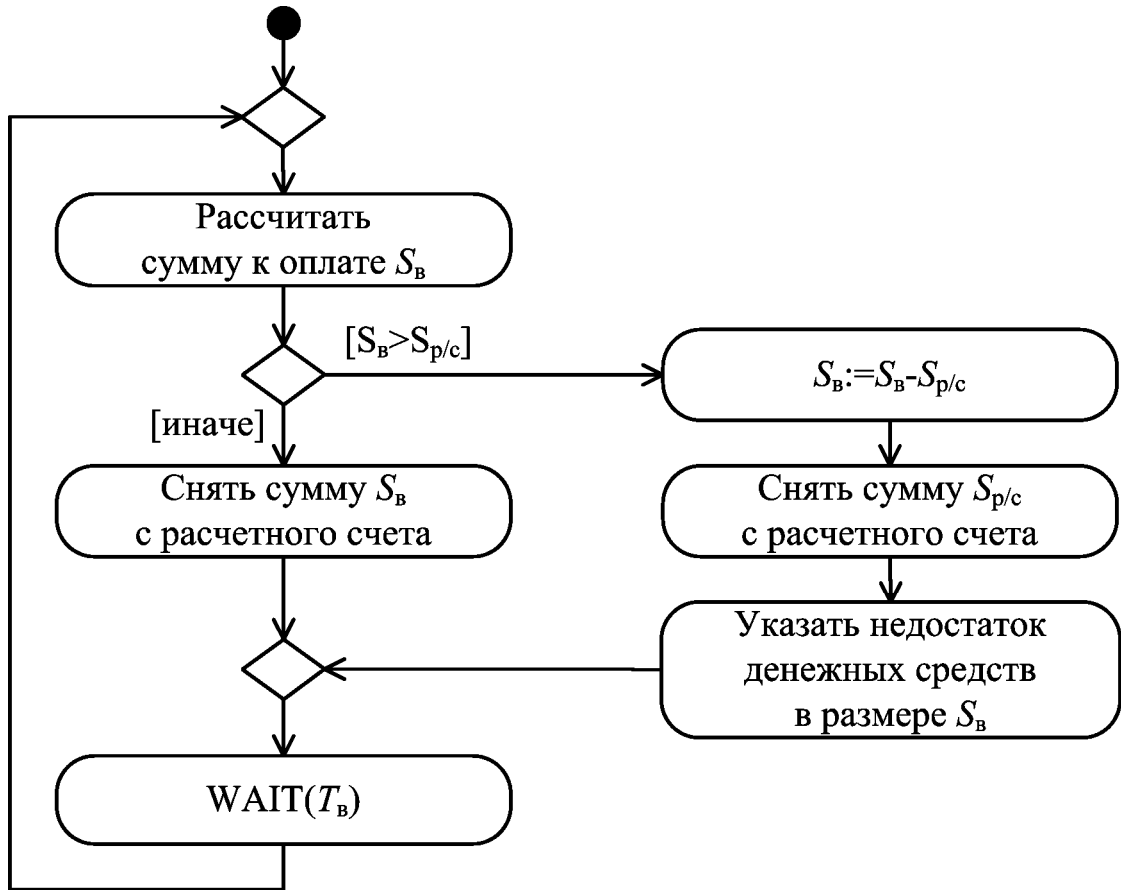


Рисунок 3.15 – Общая схема алгоритмов процессов «Уплата налогов», «Выплата заработной платы» и «Оплата постоянных затрат и издержек»

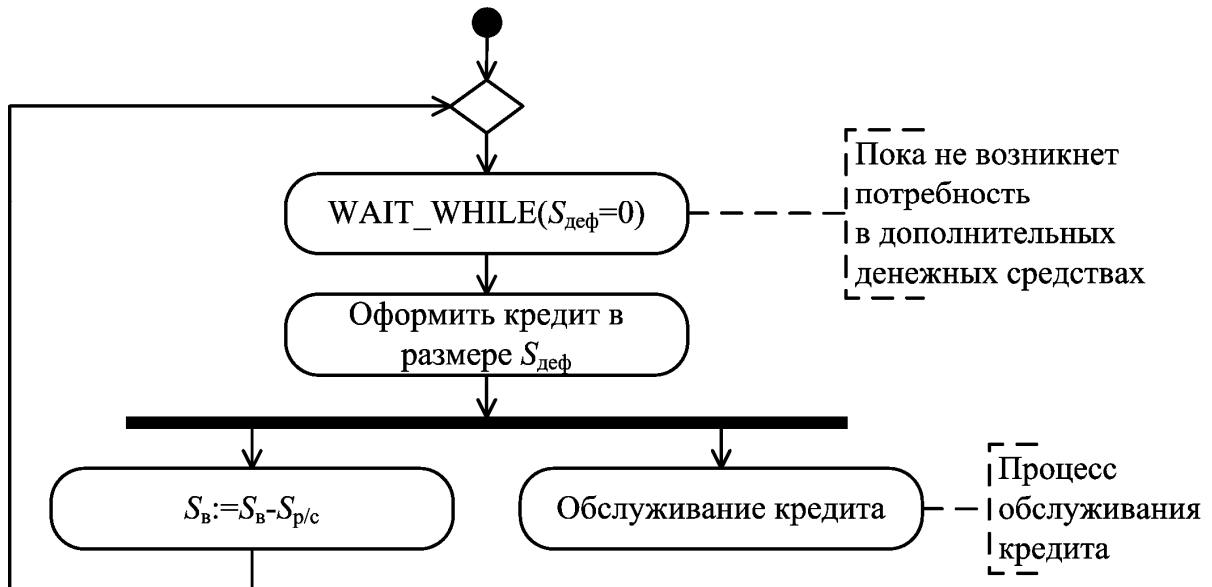


Рисунок 3.16 – Схема алгоритма процесса «Получение кредитов»

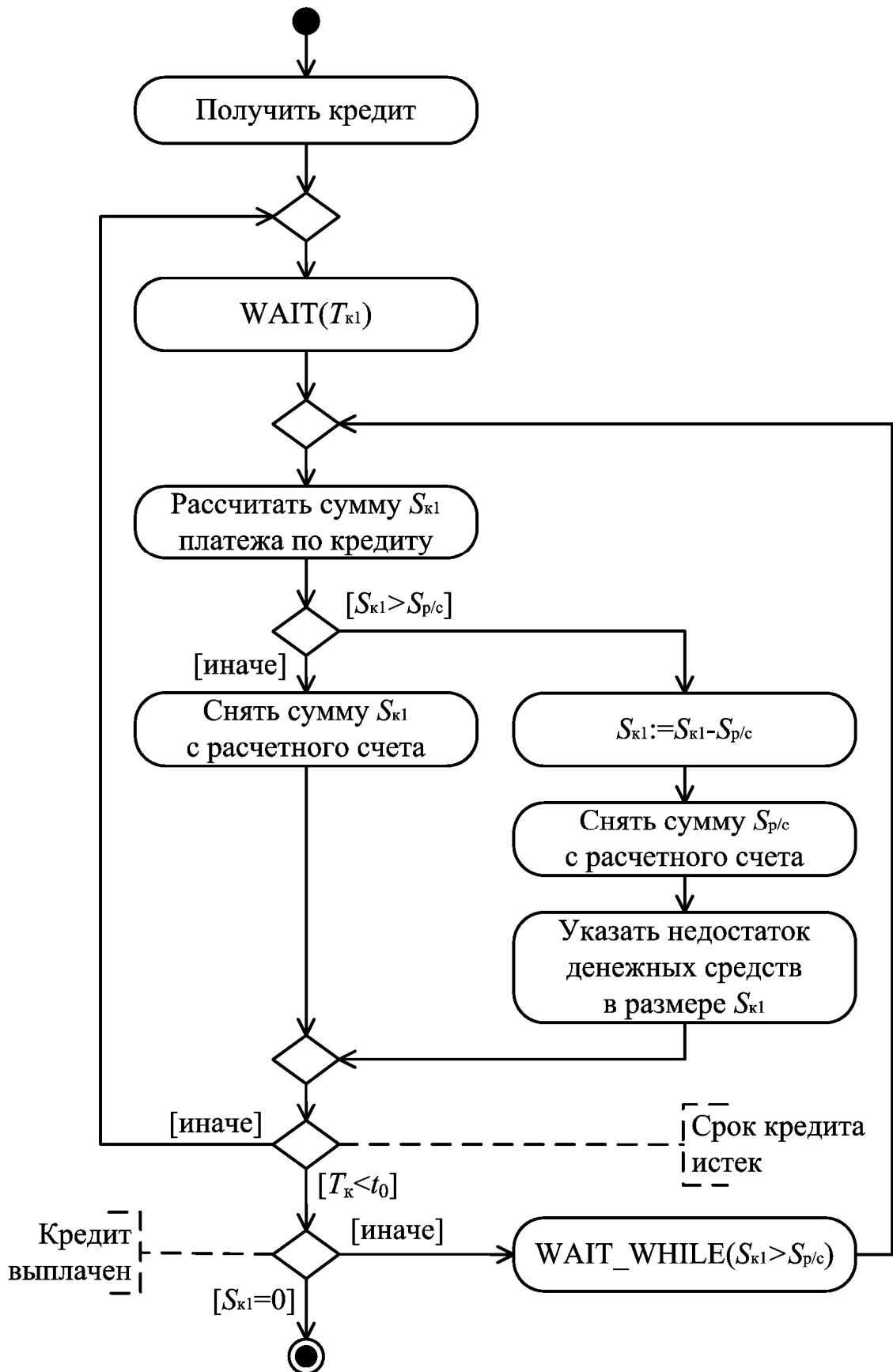


Рисунок 3.17 – Схема алгоритма процесса «Обслуживание кредита»

Следующий оператор сравнивает сумму к уплате с суммой денежных средств на расчетном счете. В случае их недостатка выплачивается имеющаяся сумма и указывается задолженность денежных средств. Невыплаченная часть переносится на следующий период.

Процесс периодических выплат продолжается до момента окончания срока кредита T_k , после чего выплачивается оставшаяся часть кредита. При этом $WAIT_WHILE(S_{k1} > S_{p/c})$ реализует задержку выполнения процесса до выполнения условия – наличие необходимой суммы денежных средств. Оператор END завершает процесс [221, с. 147–149].

3.4.4 Модификация состава и структуры имитационной модели промышленного предприятия. При эксплуатации модели для ее коррекции в условиях неопределенности и риска деятельности промышленного предприятия требуется изменение структуры введением новых компонентов или процессов.

Для создания процессов используется оператор $ADD_PROCESS(Ai, [t], [p])$, где Ai – первая активность процесса, t – время первой активизации (необязательный параметр), p – приоритет процесса (необязательный параметр). Для синхронизации процессов используется оператор $WAIT(t, [Ai])$, где t – время ожидания, Ai – следующая активность процесса (необязательный параметр). Для завершения текущего процесса и удаления его из списка процессов УПМ используется оператор END .

Для отладки ИМ в текст программы каждой активности включаются операторы $WRITE$. Система моделирования (СМ) обеспечивает запись информации об активности (наименование, значение локального времени, глобальные данные модели) в XML -файл, пригодный для дальнейшей обработки.

По окончании моделирования можно провести анализ полученных результатов и обработать их, используя средства автоматизации. Для упрощения процедуры верификации ИМ при прогоне модели имеется возможность зафиксировать значения генератора случайных чисел, что позволяет заменить стохастические значения переменных детерминированными.

3.4.5 Формализация взаимодействия компонентов имитационной модели предприятия с внешней средой. Исходные данные для ввода в ИМ делятся на две группы: управляющие параметры и входные характеристики. В качестве управляющих параметров используются данные ИМ, регламентированные нормативными документами (например, удельные нормы расхода сырья и материалов). Управляющие параметры импортируются в файл данных модели без предварительной обработки. Это обусловлено тем, что актуальность и достоверность информации обеспечивается компанией, занимающейся сопровождением КИС на предприятии, либо операторами, которые вносят данные из соответствующих норматив-

ных документов в КИС. Поэтому задача получения управляющих параметров ИМ сводится к автоматизации импорта массивов требуемых данных из КИС в ИМ.

Входные характеристики вносятся в файл данных ИМ после предварительной обработки средствами пакета *STATISTICA*. Эти данные хранятся в КИС предприятия в виде статистической информации, полученной вводом показаний контрольно-измерительных приборов, внесенной из товарно-транспортных накладных, из документов-контрактов и иной документации, имеющейся на предприятии.

В КИС могут присутствовать недостоверные данные $D_{нд}$, наличие которых обусловлено ошибками измерения $D_{ои}$, ошибками снятия показаний контрольно-измерительных приборов $D_{оп}$, ошибками операторов при внесении информации в базу данных КИС $D_{оо}$:

$$D_{нд} = D_{ои} + D_{оп} + D_{оо}. \quad (3.4)$$

Для исключения недостоверной статистической информации при ее использовании, например, для получения аналитических зависимостей, которые могут быть использованы в процессе моделирования, выполняется следующая процедура:

1) осуществляется импорт данных из таблиц базы данных КИС в пакет *STATISTICA* через табличный процессор MS Excel или непосредственно, в зависимости от формата базы данных КИС (выгрузка данных в *MS Excel* поддерживается большинством современных КИС);

2) выполняется обработка полученной статистической информации для исключения недостоверных данных с привлечением экспертов. Для автоматизации исключения недостоверных данных знания экспертов могут быть сохранены в системе *STATISTICA* посредством реализации специального макроса;

3) получают регрессионные зависимости, которые могут использоваться в процессах ИМ. Для каждой из входных характеристик, которые могут изменяться случайным образом, определяется закон распределения, среднее значение и стандартное отклонение.

Обработанные данные затем импортируются в файл данных модели [252, с. 90–92; 268, с. 459].

3.4.6 Исследование ИМ подсистем промышленного предприятия.

Исследование свойств ИМ включает оценку погрешности моделирования, анализ длины переходного процесса и устойчивости результатов моделирования, анализ чувствительности откликов к изменению входных параметров. При этом следует отметить, что до настоящего времени, как отмечает Дж. С. Карсон (J. S. Carson II, 2003), отсутствуют общие правила для определения длины прогона и числа повторений опытов. В каждом случае

все определяется моделью. Число повторений определяется точностью исследуемой величины, задаваемой длительностью доверительного интервала [289, с. 12].

Для обработки результатов имитационных экспериментов по исследованию свойств ИМ используются пакет *STATISTICA* и специальные макросы, входящие в состав ПТКИ *BelSim*. Если некоторые процессы ИМ, выполняющие одинаковые функции, реализованы на основе различных алгоритмов или имеют различный состав и структуру, то необходимо исследовать свойства ИМ для каждого варианта состава и структуры ИМ, что является *особенностью исследования ИМ* для промышленного предприятия.

Например, для ИМ существует несколько алгоритмов реализации процессов планирования продаж и производства или управления запасами ресурсов. Выбор различных алгоритмов обуславливает различный состав процессов ИМ и, следовательно, различные свойства ИМ.

При оценке *погрешности моделирования* известно, что любая ИМ неточна с дисперсией $D_{им}$ из-за наличия ошибок при аппроксимации реальных процессов $D_{ап}$ и их реализацией в ИМ. К ошибкам аппроксимации добавляются также ошибки, связанные с розыгрышем случайных значений $D_{сз}$ некоторых параметров ИМ. Эти два типа ошибок трудно разделить и приходится оценивать суммарную погрешность моделирования:

$$D_{им} = D_{ап} + D_{сз}. \quad (3.5)$$

В общем случае Б. В. Шмейсер (B. W. Schmeiser, 2001) определяет пять источников ошибок при исследовании модели: ошибки программирования (иначе их называют ошибкой верификации; для моделирования – ошибкой валидации); вычислительные ошибки, связанные с ограничениями вычислений на компьютере; ошибки случайных чисел (чем быстрее работают компьютеры, чем больше размер памяти, тем более повышается чувствительность к ошибкам случайных чисел); ошибка формирования закона распределения случайной величины; ошибка статистической выборки (метод Монте-Карло является, по существу, статистическим методом, поэтому статистические ошибки неизбежны) [366, с. 40].

Стационарность режима моделирования характеризуется некоторым установившимся равновесием процессов в модели, когда при дальнейшей имитации новой информации о системе не получают [114, с. 47].

Под *устойчивостью* результатов имитации, как правило, понимают сходимость контролируемого параметра моделирования к определенной величине при увеличении времени моделирования варианта сложной системы [114, с. 47]. Обычно устойчивость результатов имитации оценивают дисперсией отклика. Если эта дисперсия при увеличении времени моделирования T_m не увеличивается, то результаты моделирования устойчивы [105, с. 216].

Для анализа длины переходного процесса и устойчивости результатов моделирования сбор статистики в ИМ запускается в начальный момент модельного времени t_0 и статистика имитации собирается для каждого отклика $\{Y_c\}$ через заданные промежутки времени Δt_i . В *XML*-файл с результатами моделирования значения откликов ИМ записываются через каждые промежутки времени Δt_i . Далее выполняется анализ полученной статистики моделирования средствами математического пакета *STATISTICA*.

Для анализа длины переходного процесса строятся графики зависимостей $\{Y_c(t_i)\}$ и определяется время τ_i перехода в стационарный режим отклика, изменяющегося медленнее всех остальных. Это время определяет длину переходного процесса. Для исключения статистики моделирования, собранной в период разгона модели, при дальнейшем исследовании ИМ сбор статистики запускается в момент окончания переходного процесса [250, с. 116–121].

При исследовании нескольких бизнес-процессов с различными вариантами их алгоритмов длиной переходного процесса ИМ считается *максимальное* из полученных значений. Это обусловлено необходимостью одновременного начала сбора статистики для последующего сравнительного анализа и выбора рационального алгоритма организации бизнес-процессов.

При изменении состава и структуры ИМ на соответствующем уровне анализ длины переходного процесса выполняется каждый раз заново.

Для оценки устойчивости результатов моделирования строятся зависимости *средних значений* каждого отклика ИМ от времени $\{\hat{Y}_c(t_i)\}$. Если амплитуды полученных значений откликов существенно не увеличиваются с ростом модельного времени в несколько раз, то результаты имитации можно считать устойчивыми [113, с. 38].

При исследовании *чувствительности имитационной модели* для каждой входной переменной модели (параметра модели $\{X_c\}$ и внешней среды $\{Z_c\}$) указываются три значения: в центральной точке и с отклонениями от нее в большую и меньшую сторону на выбранную длину интервала изменения (уровни 0, +1 и –1 соответственно). Далее проводится серия из $2 \cdot n$ экспериментов, где n – количество входных параметров. Для каждого i -го параметра проводится пара экспериментов для уровней +1 и –1 соответственно, при этом значения остальных параметров устанавливаются на уровне 0 и вычисляются соответствующие значения вектора откликов модели $Y^{i(1)}$ и $Y^{i(-1)}$. Результаты экспериментов записываются в *XML*-файл и обрабатываются средствами пакета *STATISTICA*. В результате описанной процедуры оценивается влияние каждого входного параметра на отклики ИМ и минимизируется множество откликов, чувствительных к входным параметрам [113, с. 38–40].

Для ИМ промышленного предприятия зачастую *проверка адекватности* заменяется *верификацией* модели объекта, поскольку эта процедура является весьма затратным мероприятием. Оценка адекватности ИМ ре-

альному объекту проводится в тех случаях, когда для моделируемой системы можно провести натурные эксперименты, в ходе которых можно определить отклики системы при известных значениях входных характеристик и управляющих переменных.

3.4.7 Задачи, решаемые на основе имитационной модели промышленного предприятия. Предложенный метод имитационного моделирования промышленного предприятия обеспечивает решение следующих задач.

1 *Оценка влияния алгоритмов организации бизнес-процессов в информационной системе (ИС) предприятия и их параметров на финансово-экономические показатели предприятия* для поиска путей реинжиниринга бизнес-процессов. В процессе комплексного анализа необходимо не просто найти рациональный вариант состава и структуры ИС на основе суперпозиции интегральных показателей качества функционирования отдельных подсистем ИС, но и учесть ограничения, накладываемые на каждую исследуемую подсистему другими подсистемами сложной системы. Для этого в ИМ подсистем включаются процессы, реализующие функции ИМ каждой подсистемы, которая передает входные характеристики и управляющие параметры исследуемой подсистеме. Алгоритмы бизнес-процессов реализуют функции подсистемы на более высоком уровне детализации, чем ИМ других подсистем, с помощью аналитических зависимостей и операторов ветвления. Выходные характеристики процесса влияют на отклики модели непосредственно или через промежуточные переменные. Для комплексного анализа систем в этом случае можно использовать метод многокритериальной оптимизации. Интегральный показатель качества функционирования сложной системы определяется на основе суперпозиции показателей качества отдельных компонентов [13, с. 64–67; 246, с. 299–301].

2 *Рациональный выбор состава, структуры и параметров системы управления производственными процессами для поиска путей снижения себестоимости продукции.* Для рационального выбора параметров ИС на уровне производственного процесса принятие решений осуществляется на основе показателей его эффективности: производственная мощность, длительность производственного цикла, коэффициенты загрузки оборудования и себестоимость продукции. При этом может использоваться как метод главного критерия (которым является, как правило, себестоимость продукции), так и метод «свертки» компонентов вектора откликов для учета других показателей эффективности системы управления производственным процессом [247, с. 36–39].

3 *Оценка выбора альтернативных вариантов* – целесообразность запуска отдельного производства на краткосрочный период при повышенном спросе на продукцию или равномерный выпуск продукции с хранением ее на складе. Модель планирования производственной нагрузки с жест-

кой координацией основана на том, что выпуск продукции, т. е. производственная нагрузка меняется в зависимости от поступающих заказов на уровне производственного объединения. Производственная нагрузка не может превышать некоторого максимального значения $ВП_{\max i}$ и не может быть ниже $ВП_{\min i}$, обусловленного особенностями технологического процесса. Продукция производится непрерывно, загружается в тару и перемещается на склад. Если заказы отсутствуют, производство останавливается и требуются дополнительные затраты на его запуск. Модель с интегральной координацией основана на постоянной производственной нагрузке, когда за определенный длительный период времени требуется выпустить заданное количество продукции. Если спрос на продукцию отсутствует, она поступает на склад. С другой стороны, при отсутствии сырья требуется брать кредиты для его закупки. Финансовые средства могут отсутствовать, когда продукция не реализуется. При этом следует определить максимально допустимые цены на закупку сырья и учитывать дополнительные затраты на хранение готовой продукции на складе [217, с. 239–245].

4 Оценка целесообразности закупки сырья в период низких цен с последующим хранением на складе или ритмичная закупка сырья в течение заданного периода. При определении максимально допустимых закупочных цен себестоимость продукции и время перехода на другой ассортимент используются в качестве ограничений в ходе оптимизации. Возможны следующие варианты постановки данной задачи:

– необходимо определить максимальное значение закупочной цены основного сырья. В этом случае задача сводится к максимизации значения закупочной цены для заданного вида основного сырья при однозначно заданном плане выпуска продукции, себестоимости продукции и времени перехода на другой ассортимент, не превышающем заданные значения. В качестве ограничения для каждого цеха предприятия задается также минимальная и максимальная производственная нагрузка на выпуск продукции;

– необходимо установить максимальное значение закупочной цены $Ц_i$ для N видов основного сырья ($i = 1, \dots, N$). В этом случае при поиске решения может быть использована процедура, аналогичная процедуре метода последовательных уступок: для выбранного одного вида сырья определяется максимальное значение закупочной цены $\max Ц_1$ при фиксированных регламентированных значениях цен на другие виды основного сырья. Для найденного максимального значения назначается некоторая уступка $\Delta Ц_1$. Задавая в качестве ограничения $\max(Ц_1 - \Delta Ц_1 \leq Ц_1) \leq \max Ц_1$, определяется максимальное значение для второго вида основного сырья $\max Ц_2$. Далее назначается уступка для максимального значения цены второго вида основного сырья $\Delta Ц_2$ и т. д. [150, с. 20].

5 Оценка эффективности заключения контрактов на поставку ресурсов, реализацию готовой продукции с учетом экономических условий контракта. Одна из главных составляющих маркетинговой деятельности

предприятий – ценовая политика. При этом подразумевается использование ориентированной на потребителя системы обоснования цен. Различные условия контракта оказывают влияние на финансовое состояние предприятия. Контракт с отсрочкой платежа при реализации продукции приводит к потерям предприятия из-за невовлечения денежных средств в производственный оборот. Один из недостатков отсрочки платежа для продавца – сложность определения потерь от отсутствия оборотных средств и темпов инфляции и, соответственно, связанного с этим риска. Возможны следующие варианты решения задач оценки эффективности контрактов:

- анализ различных показателей финансовой деятельности предприятия (чистой прибыли, рентабельности и т. д.) при заключении контрактов купли, продажи и дилерского обслуживания и сравнение их с теми же показателями без заключения контрактов;

- оценка влияния процента и срока предоплаты, а также срока оплаты поставки в контракте купли с фиксированными ценами при одних и тех же исходных данных о состоянии предприятия на показатели финансовой деятельности предприятия; поиск наилучшего способа оплаты поставок по критерию, например, чистой прибыли;

- оценка влияния сроков поставок ресурсов в контракте купли с фиксированными ценами при одних и тех же исходных данных о состоянии предприятия на показатели финансовой деятельности предприятия;

- оценка влияния интервалов между изменениями цен и коэффициентов изменения цен на различные показатели финансовой деятельности предприятия при одинаковых условиях заключения контракта купли с фиксированными ценами;

- оценка влияния процента и срока предоплаты, а также срока оплаты отгрузки в контракте продажи при одних и тех же исходных данных о состоянии предприятия на показатели финансовой деятельности предприятия; поиск наилучшего способа оплаты отгрузок по критерию, например, чистой прибыли;

- оценка влияния сроков отгрузок продукции в контракте продажи при одних и тех же исходных данных о состоянии предприятия на показатели финансовой деятельности предприятия;

- оценка влияния интервалов между изменениями цен и коэффициентов изменения цен на различные показатели финансовой деятельности предприятия при одинаковых условиях заключения контрактов продажи;

- оценка влияния интервалов между отгрузками в контракте на дилерское обслуживание на показатели финансовой деятельности предприятия при фиксированных исходных данных о состоянии предприятия;

- оценка влияния объемов продукции в отгрузках по контракту на дилерское обслуживание на показатели финансовой деятельности предприятия при фиксированных исходных данных о состоянии предприятия;

– определение вида продукции, для которого при одинаковых исходных данных наиболее выгодно заключать контракт на дилерское обслуживание [150, с. 20–22].

6 Минимизация длительности переходного процесса при смене ассортимента выпускаемой продукции, сокращение потерь при переходных процессах, снижение себестоимости выпускаемой продукции. Данные по основному сырью, вспомогательным материалам, энергетическим и постоянным затратам хранятся в КИС. На основе обработки статистических данных определяются зависимости расхода энергоресурсов от выработки. Полагают, что расход энергоресурсов не зависит от вида выпускаемого продукта. В качестве ограничения используется суточный план выпуска продукции, для которого по результатам моделирования задаются оптимальные значения интенсивности и продолжительности подачи технологической добавки [150, с. 22].

7 Оптимальное распределение производственной нагрузки по цехам промышленного предприятия. На основе обработки статистических данных, хранящихся в базах данных КИС, выявляются зависимости расхода энергоресурсов, основного сырья, вспомогательных материалов и постоянных затрат от выработки продукции. Выявленные зависимости с течением времени могут изменяться в результате проведения организационно-технологических мероприятий. При известных значениях минимальной и максимальной суточной производительности каждого цеха, планируемом суточном плане выпуска требуется распределить производственную нагрузку между цехами для минимизации себестоимости продукции за счет уменьшения постоянных и энергетических затрат на единицу вида продукции [243, с. 45–46].

4 Программно-технологический комплекс имитации сложных систем BELSIM

4.1 Состав и структура комплекса

Программно-технологический комплекс имитации (ПТКИ) BelSim (рисунок 4.1) реализует метод построения имитационной модели деятельности промышленного предприятия, представленный в разделе 3. ПТКИ BelSim построен в соответствии с системными принципами функциональной ортогональности и рациональности, многоцелевого назначения, процедурной открытости и рационального дополнения [82, с. 85–86], состоит из следующих компонентов: BelSim IDE (*Integrated Development Environment*); BelSim Optimizer (*Оптимизатор*) для решения оптимизационных задач; BelSim Simulator Core (*Ядро системы моделирования*); BelSim Experimenter (*Экспериментатор*); StatSoft STATISTICA; BelSim Data Integrator (*Интегратор*) для интеграции с КИС [387, с. 204–207].

Интегрированная среда разработки программного обеспечения (ПО) BelSim IDE представляет собой сочетание коммерческих программных систем: Microsoft Visio Drawing Control и Microsoft Visual Studio .NET для построения функциональной модели системы на основе IDEF0 и реализации приложений на языке C++. Программное обеспечение для построения функциональной модели исследуемой системы на основе методологии IDEF0 используется на этапах составления содержательного описания и концептуальной модели объекта моделирования. С увеличением сложности анализируемой системы применение специализированных CASE-систем функционального моделирования (например, AllFusion Process Modeler, прежнее наименование PLATINUM BWin, для построения IDEF0-диаграмм) позволяет обеспечить строгое следование методологии IDEF0, документирование процесса, автоматическую проверку корректности модели [69, с. 64–67].

Интегрированная среда Microsoft Visual Studio .NET разработки приложений на языке C++ и ядро BelSim Simulator Core, представленное системой моделирования (СМ) PSTL (Process Simulation Template Library), являются важной и неотъемлемой частью комплекса для имитационного моделирования на основе процессного способа имитации. СМ PSTL является расширением стандартных средств языка ANSI C++. В результате отсутствуют какие-либо ограничения на среду разработки и применяемую вычислительную систему. Разработчик волен выбирать для реализации ИМ любую систему программирования на основе языка C++.

Однако при разработке сложных моделей рекомендуется использовать среду разработки, обладающую развитыми инструментальными средствами написания и отладки программ (например, Microsoft Visual C .NET) [262, с. 3–4].

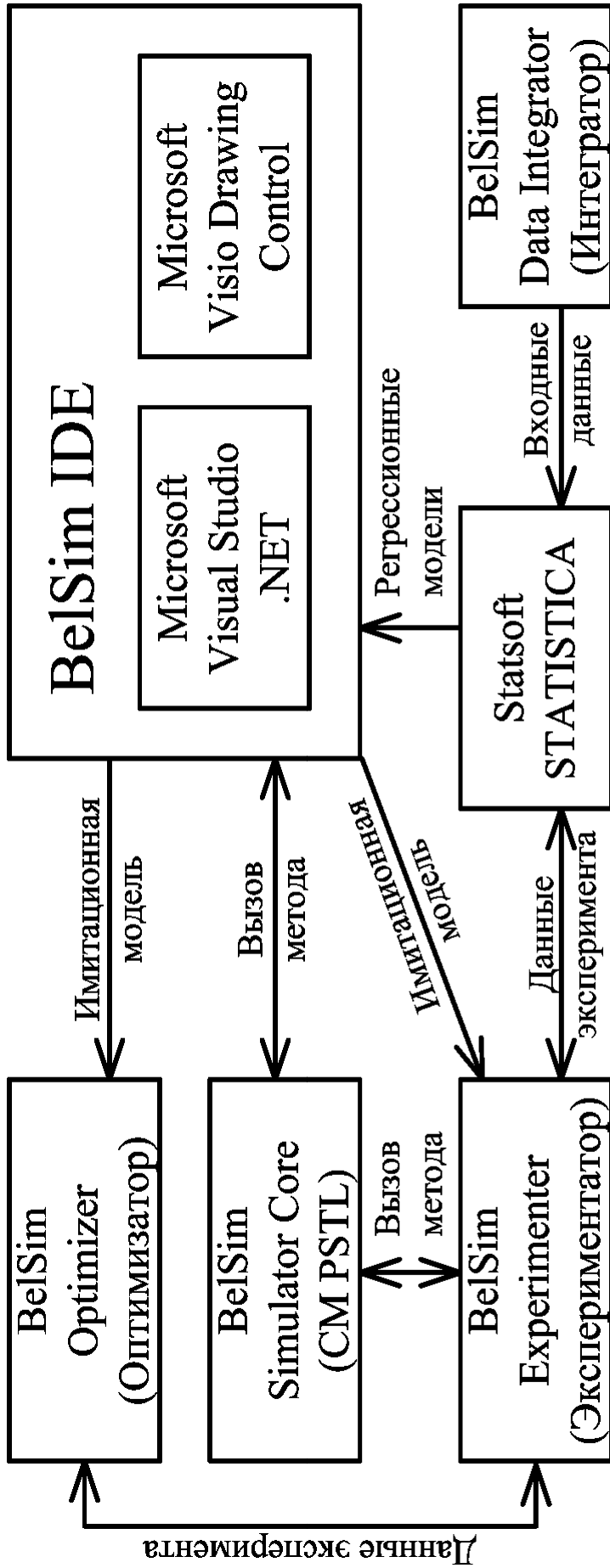


Рисунок 4.1 – Структура ПТКИ BelSim

Преимущества платформы .NET Framework: единая программная модель, упрощенная модель программирования; отсутствие проблем с версиями; упрощенная разработка; работа на нескольких платформах; интеграция языков программирования; упрощенное повторное использование кода; автоматическое управление памятью (сбор мусора); проверка безопасности типов; развитая поддержка отладки; единый принцип обработки сбоев; безопасность; взаимодействие с существующим кодом [155, с. 22–24]. Многие разработчики систем имитационного моделирования указывают на достоинства Microsoft .NET Framework: развитие систем взаимодействия между различными имитационными моделями, удобство сотрудничества через Интернет, объектно-ориентированное программирование [315, с. 629–633].

BelSim Experimenter представляет собой отдельную подсистему для планирования, проведения и обработки результатов имитационных экспериментов (ИЭ). В подсистеме BelSim Experimenter ПО для планирования ИЭ является приложением *ExperimentDesigner* на основе Microsoft .NET Framework 2.0. Дополнительно для получения плана при дробном факторном эксперименте используется модуль *Experimental Design (DOE)* пакета статистического анализа *STATISTICA* фирмы StatSoft Inc и макрос *DesignOfExperiment* для данного приложения, который сохраняет результат работы данного модуля. ПО для проведения имитационных экспериментов является программным приложением *Experimenter*. ПО для обработки результатов ИЭ представляет собой макрос *ExperimentData* для приложения *STATISTICA*. Для анализа данных при статистической обработке и наглядного представления результатов имитации возможно использование любого пакета статистического анализа, позволяющего осуществлять импорт данных из внешних источников аналогично, например, пакету *STATISTICA* фирмы StatSoft Inc [221, с. 72].

4.2 Система имитационного моделирования PSTL

На этапе программирования для эффективной реализации ИМ необходим аппарат моделирования (отсутствующий в существующих универсальных системах программирования), который должен предоставлять способы организации данных, обеспечивающие простое и эффективное моделирование, а также удобные средства формализации и воспроизведения динамических свойств моделируемой системы. Использование существующих специализированных средств имитации процессным способом затруднено или невозможно вследствие использования ими специфического аппаратного (тип ЭВМ) и программного (базовый язык программирования) обеспечения. Одна из немногих доступных современных СМ MICIC4 [105, с. 25], разработанная в Гомельском государственном университете им. Ф. Скорины под руководством профессора И. В. Максимея, не позволяет обеспечить требуемую для рассматриваемой предметной области краткость и точность выражения понятий вследствие закрытости данной СМ и ограничений базо-

вого языка программирования. Возможности создания библиотеки дополнительных стандартных компонентов, которые могут использоваться при построении ИМ, также ограничены [221, с. 74].

На этом этапе используется СМ PSTL (рисунок 4.2), которая позволяет учесть вышеуказанные требования и преодолеть недостатки существующих систем имитационного моделирования. Использование концепций ООП и языка С++ позволяет устранить ограничения на описание модели и создать набор абстрактных типов данных (АТД) [141, с. 107] для описания модели на языке, максимально соответствующем предметной области. При проектировании СМ PSTL в основу положен принцип открытости, что в результате позволяет без каких-либо ограничений расширять ее функциональные возможности, обеспечивая при этом обратную совместимость вследствие инкапсуляции деталей реализации. Диаграмма компонентов СМ PSTL представлена на рисунке 4.2. Каждый компонент содержит описание классов, отражающих соответствующий аспект СМ. Все компоненты используют стандартную библиотеку шаблонов С++ STL.

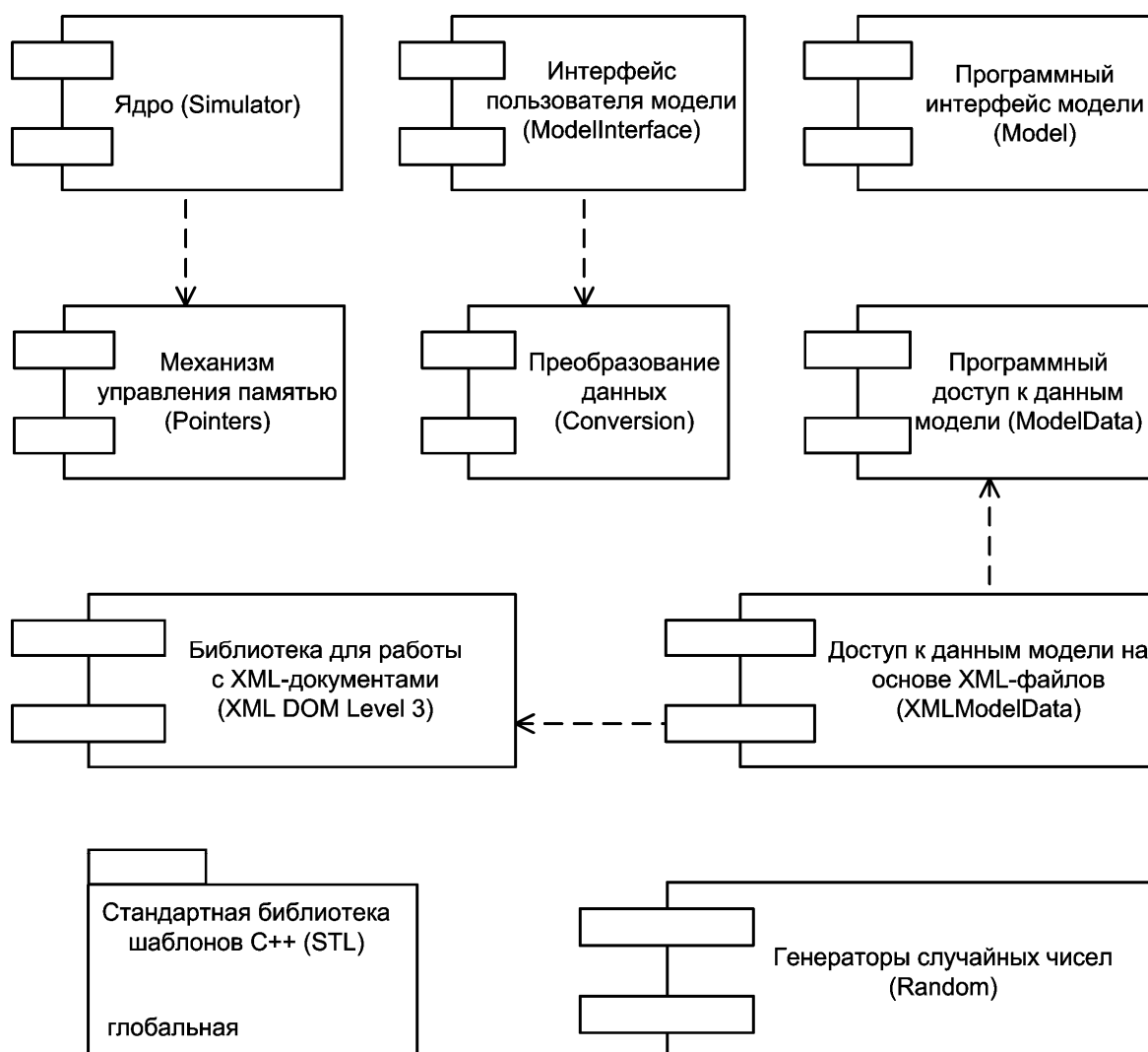


Рисунок 4.2 – Диаграмма компонентов СМ PSTL

Программный интерфейс модели в СМ PSTL используется для представления модели как единого целого и обеспечения доступа к ней посредством унифицированного набора методов, позволяющего задействовать средства автоматизации моделирования.

Доступ к данным модели (XMLModelData) обеспечивается на основе файлов в формате XML [197], работа с которыми осуществляется посредством библиотеки для работы с XML-документами (интерфейс XML DOM Level 3).

Генераторы случайных чисел (Random) (см. рисунок 4.2) построены с использованием стандартных функций языка C++ `int rand(void)`, `void srand(unsigned seed)` библиотеки `<stdlib.h>`.

Основным элементом СМ является управляющая программа моделирования (УПМ). Семантика и отношения между классами УПМ представлены на диаграмме классов (рисунок 4.3). Функции УПМ реализует параметризованный класс `CSimulator`, который включает в себя следующие вложенные классы [221, с. 76–77]:

- `CActivity` – абстрактный базовый класс для представления абстракции *Активность*;
- `TProcess` – структура, содержащая данные о процессе: идентификатор, приоритет, сведения о моменте следующей активизации процесса (экземпляр структуры `TNextActivity`);
- `CProcesses` – класс-коллекция, содержащая список экземпляров структур `TProcess` для работающих процессов;
- `TNextActivity` – структура данных о моменте активизации процесса: время и выполняемая активность.

Функция `CSimulator::run` запускает основной цикл моделирования. Схема алгоритма этой функции приведена на рисунке 4.4, работа заключается в последовательной активизации процессов, смене модельного времени и проверке условий окончания моделирования (прошествие заданного периода модельного времени, завершение всех процессов, вызов оператора останова активностью какого-либо процесса).

Важной частью языка моделирования является механизм *Управление памятью*. В его основу положен метод на основе подсчета ссылок. Основная идея заключается в отслеживании количества указателей, ссылающихся на объект – когда счетчик становится равным нулю, объект удаляется. Подсчет ссылок обладает рядом существенных ограничений, которые снижают его практическую ценность и объясняют малую распространенность. Во-первых, данный метод слишком полагается на соблюдение всех правил программистом (каждый раз при получении или копировании адреса объекта необходимо увеличивать, а при завершении использования адреса уменьшать счетчик ссылок). Во-вторых, возникает проблема зацикливания, когда какая-либо группа объектов содержит ссылки друг на друга, но внешние ссылки на объекты этой группы отсутствуют.

Для устранения недостатков метода управления памятью на основе подсчета ссылок создан специальный класс-шаблон – «умный указатель»

$SP<Type>$, заменяющий стандартный тип указателя. Для этого перегружены операторы присваивания (*operator =*), разыменования (*operator **), селекторы членов (*operator ->*), приведения типа к стандартному указателю на объект (*operator Type**).

Экземпляр класса $SP<Type>$ обеспечивает автоматический подсчет ссылок, при необходимости увеличивая или уменьшая счетчик в конструкторах, деструкторе и операторе присваивания.

Если значение счетчика уменьшается до нуля, указываемый объект удаляется. Чтобы исключить возможность создания объектов в обход данного механизма, все объекты создаются с помощью производящих функций, которые являются статическими членами соответствующего класса и возвращают «умный указатель» на объект. При этом конструкторы объектов объявлены закрытыми. «Несанкционированное» (без увеличения счетчика ссылок) копирование запрещено. Для этого доступ к конструктору копии и оператору присваивания также закрыт.

Счетчики ссылок представлены ассоциативной коллекцией, в которой каждому значению указателя соответствует целое число – количество ссылок на данный адрес памяти. Проблема снижения быстродействия решена путем использования бинарного дерева в качестве внутренней структуры данных, что обеспечивает логарифмическую зависимость времени поиска требуемых элементов от их общего количества [221, с. 78].

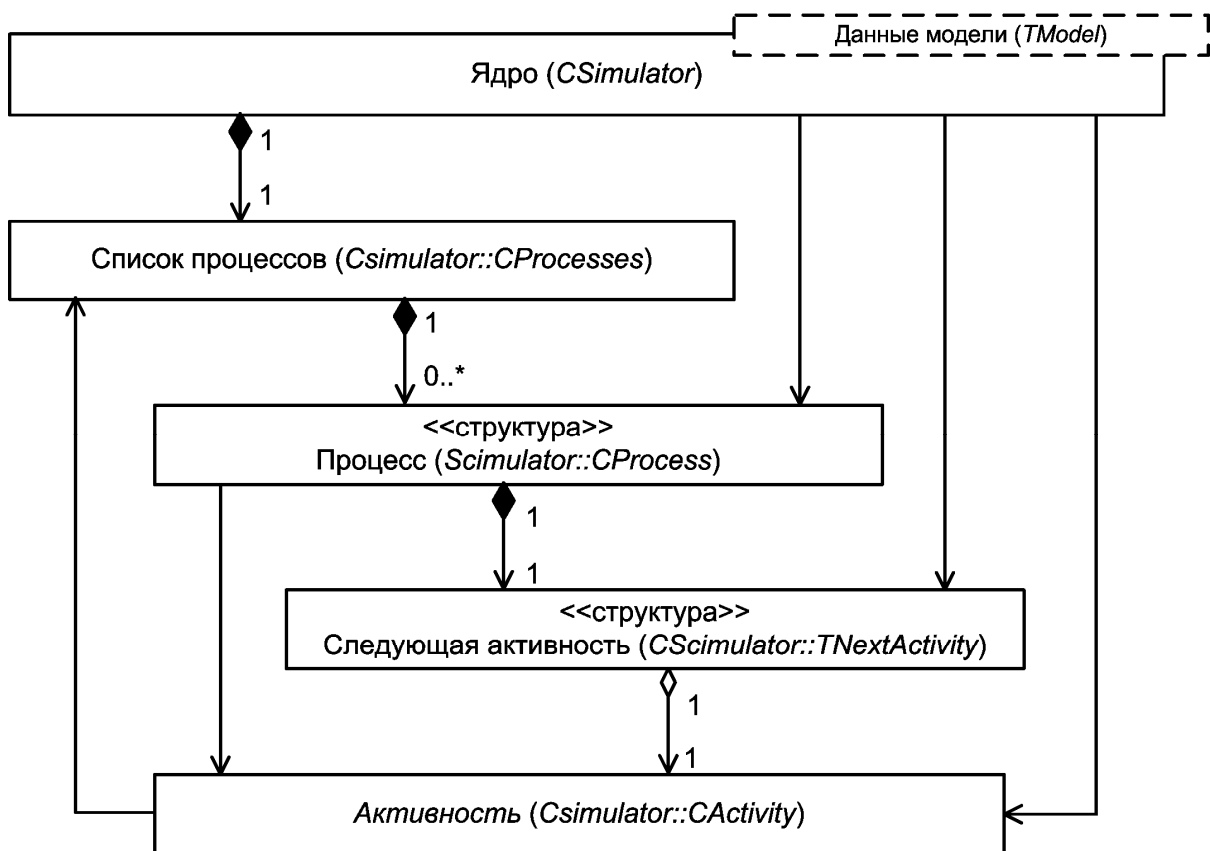


Рисунок 4.3 – Диаграмма классов УПМ

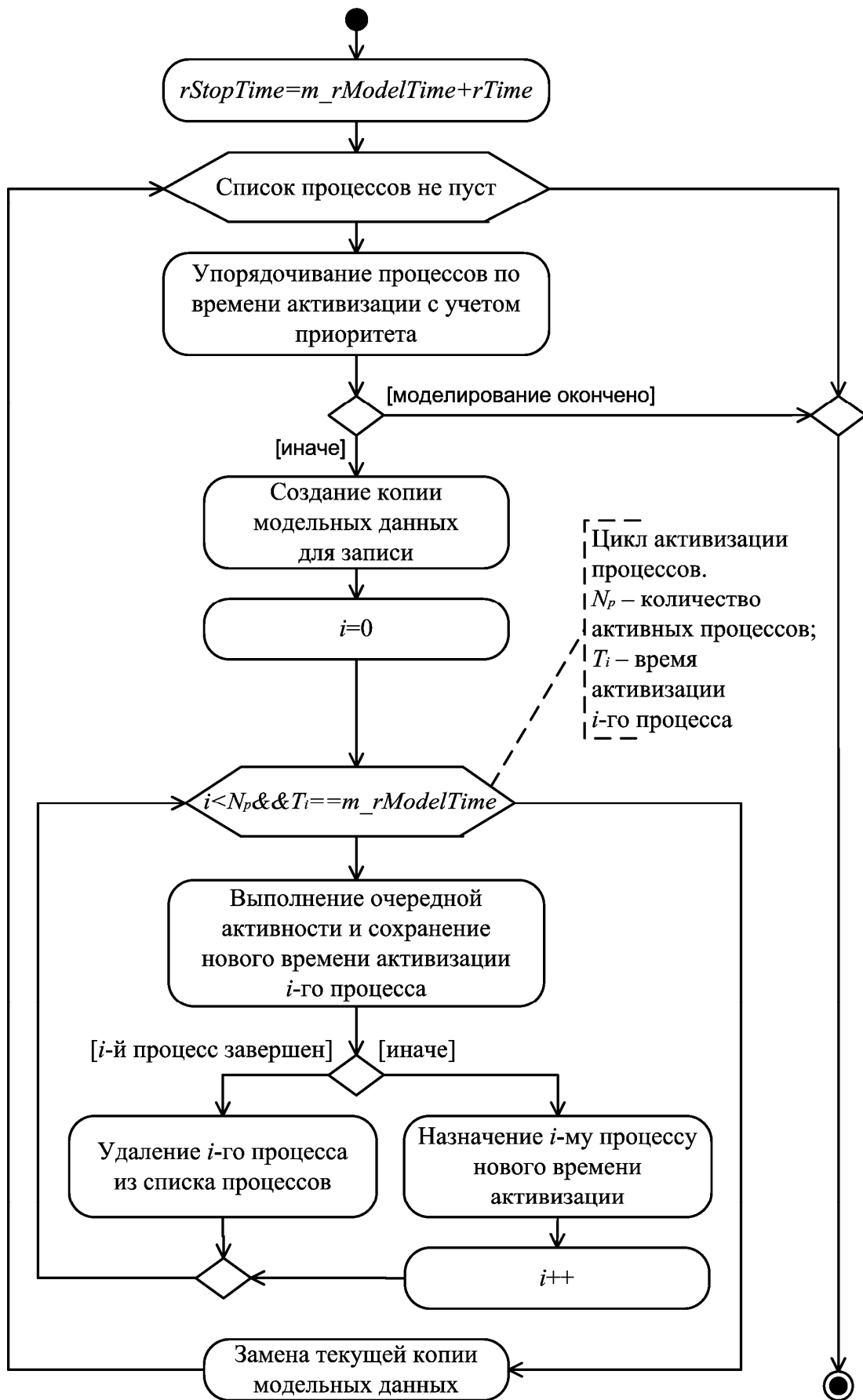


Рисунок 4.4 – Схема алгоритма цикла моделирования

4.3 Технология построения имитационной модели

Основные этапы технологии построения ИМ представлены последовательностью следующих действий.

Создание процессов. Оператор $CREATE_PROCESS(t, A_i, p)$, где t – время активизации, A_i – начальная активность процесса, p – приоритет активности, создает новый процесс с начальной активностью A_i , приоритетом p и назначает момент его первой активизации t .

Оператор $RUN(T)$, где T – модельное время имитации, запускает цикл моделирования длительностью T [221, с. 55].

Синхронизация процессов и их взаимодействие с УПМ. На этапе построения имитационной модели формальное описание системы, полученное на предыдущем этапе, преобразуется в описание имитационной модели. При этом прорабатываются вопросы синхронизации процессов и их взаимодействия с управляющей программой моделирования (УПМ), задания начальных условий, организации сбора статистики и окончания имитации.

Синхронизация процессов включает в себя реализацию двух функций:

1) внутренняя синхронизация процессов по времени путем добавления в алгоритмы их функционирования операторов изменения временной координаты;

2) синхронизация межпроцессных взаимодействий (информационного обмена) исходя из функциональной модели, построенной на этапе формализации.

Синхронизация и взаимодействие процессов осуществляются посредством обращения к УПМ с помощью набора специальных операторов, представленных ниже.

Операторы синхронизации организуют возврат на УПМ и модификацию временной координаты процесса, разбивая алгоритм его функционирования на кванты – активности:

1) $WAIT(t, A_i)$, где t – время ожидания процесса в модельном времени, A_i – следующая активность процесса. Оператор изменяет временную координату и состояние процесса. В результате выполнения оператора процессу назначается момент следующей активизации по окончании ожидания длительностью t , и при последующей активизации процесса управление будет передано на выполнение активности A_i . Возможен сокращенный вариант данного оператора – $WAIT(t)$, когда указывается только время ожидания. В этом случае состояние процесса (текущая активность) не изменяется;

2) $WAIT_WHILE(E, A_i)$, где E – логическое выражение, A_i – следующая активность процесса. Оператор изменяет состояние процесса, однако момент следующей активизации процесса не назначается до тех пор, пока выполняется условие E . УПМ проверяет условие E после каждого измене-

ния модельного времени t_i и в случае его невыполнения ставит процесс в очередь для активизации в момент времени, равный текущему значению t_i . Если изменение состояния процесса не требуется, второй параметр оператора не указывается;

3) END. Оператор завершает текущий процесс и удаляет его номер из списка процессов в УПМ [221, с. 54].

В начале очередного цикла алгоритма функционирования УПМ после процедуры изменения модельного времени и перед активизацией процессов запоминается текущее состояние модели путем создания копии структуры данных со значениями всех параметров. Далее каждому процессу при его активизации передается два набора (экземпляра структуры) параметров отдельно для чтения и для записи. Таким образом, информационная стыковка процессов, активизируемых в один и тот же момент модельного времени, обеспечивается тем, что все процессы используют идентичные значения входных параметров, зафиксированные в момент изменения модельного времени. Если процессу требуется доступ к выходным данным других процессов, то вызывается оператор WAIT(0), при этом изменения модельного времени не происходит, а требуемые параметры передаются процессу при последующей его активизации как входные, доступные для чтения. Помимо этого, используется механизм приоритетов процессов: каждому процессу назначается приоритет и активизация процессов в один и тот же момент модельного времени осуществляется в порядке убывания приоритетов. Это позволяет синхронизировать процессы-источники и процессы-потребители информации.

Задание начальных условий. Вторым шагом построения имитационной модели является разработка алгоритма задания начальных условий, который выполняется один раз перед началом моделирования и содержит следующие блоки операторов: инициализация глобальных структур данных модели, общих для всех процессов; инициализация локальных данных процессов; последовательность вызовов оператора CREATE_PROCESS для формирования списка процессов в УПМ; вызов оператора RUN для запуска процесса имитации.

Сбор статистики. Следующим шагом является организация сбора статистики. Для этой цели следует предусмотреть один или несколько процессов. Алгоритмы функционирования этих процессов должны обеспечивать их активизацию с заданным интервалом времени или при выполнении определенных условий – вычисление и запись статистики моделирования. Исходной информацией для этого являются состояние модели в момент активации процесса сбора статистики и стандартная статистика, формируемая УПМ: моменты создания первой и последней активизации, завершения процессов.

Окончание моделирования. Завершающим шагом этапа построения имитационной модели является задание условий окончания моделирования.

Командой завершения имитации является вызов оператора STOP. Проверку условия окончания моделирования можно организовать с помощью отдельного процесса или добавить необходимые операторы в алгоритм сбора статистики. По умолчанию цикл имитации УПМ завершается по прошествии периода T модельного времени, указанного при вызове оператора RUN [221, с. 55–56].

4.4 Технология программирования имитационной модели

ИМ в СМ PSTL представляет собой один или несколько программных модулей на базовом языке программирования C++, содержащих описание и реализацию структур данных модели, классов активностей процессов, УПМ, интерфейса модели, а также любых других необходимых классов, структур данных и функций [181]. Типовой программный модуль ИМ включает блоки, содержащие описания директив включения заголовочных файлов СМ, структуры (класса) глобальных данных модели (общих для всех процессов), описание и реализацию классов активностей процессов, описание и реализацию класса программного интерфейса модели, реализацию внешнего интерфейса модели. Список заголовочных файлов и их назначение приведены в таблице 4.1.

Таблица 4.1 – Заголовочные файлы СМ PSTL

Файл	Тип данных	Назначение
<i>simulator.h</i>	<i>CSimulator, TProcess, CProcesses, CActivity, TNextActivity</i>	Описание базовых параметризованных классов ядра СМ
<i>model.h</i>	<i>CModel</i>	Программный интерфейс модели
<i>ModelData.h</i>	<i>CModelData</i>	Программный интерфейс для доступа к данным модели
<i>XMLModelData.h</i>	<i>CXMLModelData</i>	Описание класса, реализующего интерфейс для доступа к данным модели, представленным в виде XML-файла
<i>pointers.h</i>	<i>SP</i>	Реализация механизма управления памятью
<i>random.h</i>	<i>CRandom, CRandomUniform, CRandomNormal, CRandomExponential, CGammaFunction</i>	Описание генераторов случайных чисел
<i>ModelInterface.h</i>	–	Библиотека шаблонов функций и макросов для реализации интерфейса пользователя модели
<i>conversion.h</i>	–	Описание функций преобразования данных

Обязательным является включение файла *simulator.h*, содержащего описание классов, составляющих ядро СМ – УПМ для организации имита-

ции процессным способом. Функции УПМ реализует параметризованный класс *CSimulator*. Параметром его выступает тип данных *TModel*, описывающий глобальные переменные модели. В состав глобальных данных модели включаются входные и выходные переменные модели, а также переменные, к которым должны иметь доступ одновременно несколько процессов. Глобальные данные оформляются в виде структуры (класса), которая может содержать вспомогательные элементы-функции, облегчающие доступ к элементам-данным. Эта структура указывается в качестве параметра класса, а ее экземпляр передается как параметр конструктора объекта УПМ. Никаких ограничений на типы данных модели не накладывается [221, с. 103–105].

Активности процессов представляют собой классы-функции, производные от базового класса *CSimulator<TModel>::CActivity*, где *TModel* – наименование типа, представляющего глобальные данные модели. Класс *CActivity* является абстрактным базовым классом и содержит чистую виртуальную функцию *operator()*, которая вызывается УПМ при очередной активизации процесса и должна быть определена для представления алгоритма активности. Параметрами функции *operator()* являются модельное время; указатель на константную копию экземпляра структуры данных модели на момент модельного времени, указанный в первом параметре до вызова активностей других одновременно активизируемых процессов; указатель на текущий экземпляр структуры данных модели; указатель на коллекцию процессов УПМ, обеспечивающий доступ к локальным данным других процессов и создание новых. Вызовы операторов синхронизации реализуются посредством оператора *return* языка С с указанием соответствующего возвращаемого значения типа *TNextActivity*, содержащего время ожидания до очередной активизации и указатель на следующую активность процесса. Дополнительно активности могут содержать любые данные и функции, необходимые для реализации алгоритма процесса. Инициализация данных возможна посредством передачи соответствующих параметров в конструктор активности. Порядок размещения описаний классов активностей и их реализаций произвольный, однако вследствие многочисленных взаимных обращений в алгоритмах описания классов следует выделять в отдельный блок, размещаемый перед блоком их реализаций [221, с. 105].

Программный интерфейс проектируемой имитационной модели обеспечивает доступ к ней посредством унифицированного набора методов и позволяет использовать средства автоматизации моделирования. Для реализации интерфейса создается класс, наследующий интерфейс *CModel*. *CModel* содержит две функции: *getModelData* и *run*.

Функция *CModel::getModelData* вызывается для получения сведений о модели. Параметром функции является ссылка на интерфейс *CModelData* объекта-приемника информации о модели. Посредством вызова методов *setName*, *setVersion*, *setRevision*, *setCopyright*, *addParameter*, *addParameterArray*, *addResponse* и *addResponseArray* указываются сведения о наименовании, вер-

сии, авторских правах, параметрах и откликах модели соответственно.

Функция *CModel::run* вызывается для проведения имитационного эксперимента. Параметрами функции являются: ссылка на интерфейс *CModelData* для чтения входных и записи выходных значений параметров и откликов соответственно; целое число для инициализации генераторов псевдослучайных чисел, что обеспечивает возможность точного повторного воспроизведения результатов экспериментов [221, с. 106].

В текст функции включают следующие блоки операторов, в совокупности представляющие собой блок задания начальных условий моделирования.

1 Блок инициализации глобальных данных модели. Вначале создается экземпляр структуры данных модели:

$$SP<TModel> spModel(new TModel),$$

где *TModel* – имя типа (структуры или класса), представляющего глобальные данные модели;

spModel – указатель на созданный экземпляр структуры данных.

Затем производится инициализация его элементов. Чтение исходных данных осуществляется посредством вызова метода *getParameterValue* интерфейса *CModelData*.

2 Объявление объекта УПМ:

$$CSimulator<TModel> simulator(spModel),$$

где *simulator* – имя создаваемого объекта УПМ.

3 Блок инициализации процессов, включающий последовательность операторов для инициализации структур данных процессов и вызовов метода *CSimulator::addProcess* для формирования списка процессов УПМ.

4 Блок инициализации генераторов псевдослучайных чисел на основе значения второго параметра функции.

5 Вызов метода *run* объекта УПМ для запуска процесса имитации.

Завершающим элементом является реализация внешнего интерфейса имитационной модели. Будучи стандартным для всех имитационных моделей, построенных по настоящей методике, он позволяет задействовать средства автоматизации проведения экспериментов и анализа их результатов. Для реализации интерфейса используется макрос

$$MODEL_MAIN(Model, ModelData),$$

где *Model* – имя класса, реализующего программный интерфейс модели;

ModelData – имя класса для доступа к данным модели.

Указанный макрос помещает в текст программного модуля имитаци-

онной модели функцию *main*, обрабатывающую командную строку для организации интерфейса пользователя [221, с. 106–107].

Объектно-ориентированный подход и открытость СМ находят выражение в гибкости при создании программы ИМ. Для большинства задач существуют стандартные рекомендуемые средства, однако разработчик не обязан использовать их, заменяя их собственными, наилучшим образом подходящими для данной конкретной ситуации. Более того, все средства построения ИМ легко доступны для внесения изменений с использованием механизмов наследования, делегирования и полиморфизма.

4.5 Программная реализация базовой имитационной модели промышленного предприятия

Анализ функциональной модели показывает, что необходимо обеспечить синхронизацию следующих процессов источников и потребителей информации (см. таблицу 3.1): ПК₃₆ (бухгалтерский учет), ПК₂₃ (поступление заявок на отгрузку продукции), ПК₁₁ (планирование производства). С учетом выбранного способа аппроксимации функциональных действий компонент, когда выполнение функционального действия предшествует изменению временной координаты, процессам назначаются следующие приоритеты: ПК₃₆ – Highest, ПК₂₃ – Normal+2, ПК₁₁ – Normal+1, где Normal – обычный приоритет, Highest – наивысший приоритет.

В блоке задания начальных условий моделирования необходимо обеспечить инициализацию структур данных модели на основе значений входных переменных, которые однозначно определяют состояние модели.

Для организации сбора статистики необходимо фиксировать состояние модели через заданные интервалы модельного времени. Это удобно реализовать в виде отдельного процесса. Так как основой для формирования статистики моделирования являются данные «Бухгалтерская отчетность», требуется синхронизация с процессом ПК₃₆ (бухгалтерский учет). Для этого процессу сбора статистики следует назначить приоритет Highest-1. Условием окончания моделирования является истечение заданного периода модельного времени [221, с. 149].

Имитационная модель физически представлена в виде одного модуля *Enterprise3.cpp*, который содержит описание и реализацию структур данных модели, классов активностей процессов, класса программного интерфейса модели, внешнего интерфейса модели.

Статическая структура классов, представляющих данные модели, приведена на рисунке 4.5. Глобальные данные представлены структурой *TEnterprise* (таблица 4.2), которая включает ряд вспомогательных классов: *TDistribution* (параметры случайной величины), *TAccount* (счет бухгалтерского учета), *TWarehouseItem* (учетная единица на складе), *TBatch* (партия продукции или сырья). Для представления абстракций контракта и кредита используются классы *TContract* и *TCredit* соответственно.

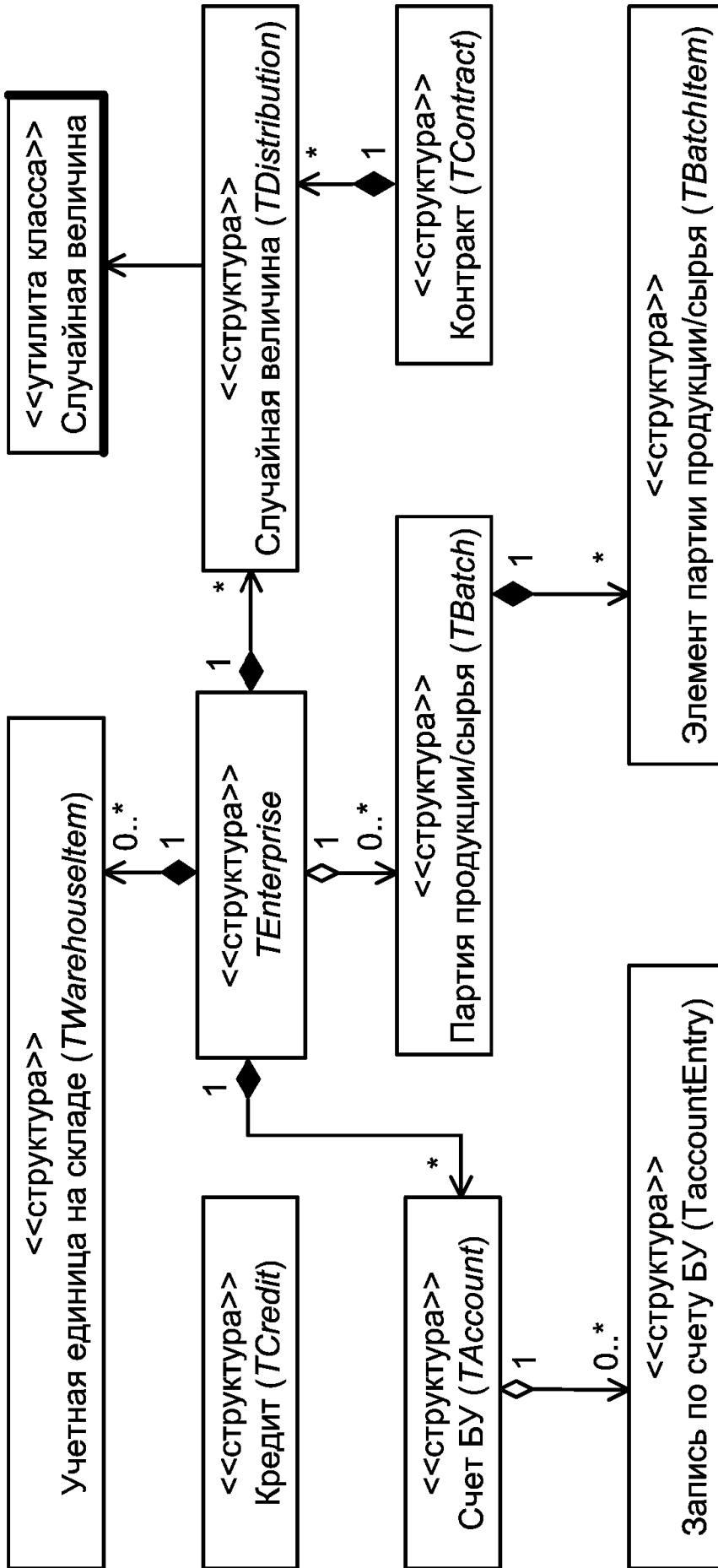


Рисунок 4.5 – Диаграмма структур данных модели

Таблица 4.2 – Структура TEnterprise глобальных данных модели

Ответ- ственность	Представление глобальных данных модели	
	Наименование	Назначение
1	2	3
Атрибуты	<i>numberOfProductTypes</i> <i>numberOfResourceTypes</i> <i>MPSTimeUnit</i> <i>MPSPlanningHorizon</i> <i>MPS</i> <i>launchingSchedule</i> <i>resourceSchedule</i> <i>productionCycle</i> <i>specificResourceConsumption</i> <i>goodsInProgress</i> <i>products</i> <i>productActualCosts</i> <i>resourcesOrdered</i> <i>resources</i> <i>shipmentIntervalDistribution</i> <i>shipmentVolumeDistributions</i> <i>shipmentPaymentPeriod</i> <i>Distribution</i> <i>productPrices</i> <i>productPriceChangeMultiplier</i> <i>Distribution</i> <i>productPriceChangeInterval</i> <i>Distribution</i> <i>deliveryPaymentPeriods</i> <i>resourcePriceChangeMultiplier</i> <i>Distribution</i> <i>resourcePriceChangeInterval</i> <i>Distribution</i> <i>resourcePrices</i> <i>shipments</i> <i>currentAccount</i> <i>creditRequirements</i> <i>loanInterestRate</i> <i>loanApplicationInterval</i> <i>loanPeriod</i> <i>cashShortageCoverage</i> <i>fixedCosts</i>	Количество видов продукции Количество видов ресурсов Интервал планирования основного плана-графика производства Количество интервалов в основном плане-графике производства Основной план-график производства График запуска в производство График потребности в ресурсах Длительность технологического цикла Нормы расхода ресурсов на единицу продукции Незавершенное производство Запасы продукции на складе Фактическая себестоимость продукции на складе Объем заказанных ресурсов Запасы ресурсов на складе Параметры распределения интервала между отгрузками продукции Параметры распределения объемов продукции в отгрузке Параметры распределения срока оплаты за отгруженную продукцию Цены на продукцию Параметры распределения коэффициента изменения цен продукции Параметры распределения интервала между изменениями цен продукции Сроки оплаты за полученные ресурсы Параметры распределения коэффициента изменения цен ресурсов Параметры распределения интервала между изменениями цен ресурсов Цены ресурсов Список отгрузок Сумма денежных средств на расчетном счете Потребность в кредите (сумма нехватки денежных средств) Процентная ставка займа Периодичность обращения за кредитом Срок кредита Коэффициент покрытия кредитом текущей нехватки денежных средств Постоянные затраты предприятия

Продолжение таблицы 4.2

1	2	3
	<i>wageShareInFixedCosts</i>	Доля заработной платы в постоянных затратах предприятия
	<i>otherSpecificVariableCosts</i>	Переменные затраты на единицу продукции без учета стоимости ресурсов
	<i>wageShareInVariableCosts</i>	Доля заработной платы в переменных затратах
	<i>costsChangeMultiplierDistribution</i>	Параметры распределения коэффициента изменения затрат
	<i>costsChangeIntervalDistribution</i>	Параметры распределения интервала между изменениями затрат
	<i>fixedAssetsReplacementCost</i>	Восстановительная стоимость основных средств
	<i>fixedAssetsWearFactor</i>	Коэффициент износа основных средств
	<i>fixedAssetsDepreciationRate</i>	Норма амортизации основных средств
	<i>fixedAssetsReappraisalMultiplier</i>	Коэффициент переоценки основных средств
	<i>taxableFixedAssetsCost</i>	Стоимость основных средств, с которой уплачивается налог на недвижимость
	<i>VATRate</i>	Ставка НДС
	<i>receiptsTaxRate</i>	Ставка налогов с выручки
	<i>profitTaxRate</i>	Ставка налогов с прибыли
	<i>profitInDisposalTaxRate</i>	Ставка налогов с прибыли, остающейся в распоряжении предприятия
	<i>socialInsuranceTaxRate</i>	Ставка отчислений в Фонд социальной защиты
	<i>wageTaxRate</i>	Ставка налогов с заработной платы
	<i>fixedAssetsTaxRate</i>	Ставка налога на недвижимость
	<i>ecologicalTaxRate</i>	Сумма экологического налога на единицу продукции
	<i>landTax</i>	Сумма налога на землю
	<i>VAT</i>	Сумма НДС в реализованной продукции
	<i>quantityShipped</i>	Количество отгруженной продукции
	<i>quantityProduced</i>	Количество произведенной продукции
	<i>resourceCosts</i>	Стоимость ресурсов в произведенной продукции
	<i>resourcesTaken</i>	Стоимость ресурсов, взятых со склада
	<i>clearReceipts</i>	Выручка (за минусом налогов)
	<i>costOfSales</i>	Себестоимость реализации продукции
	<i>profit</i>	Прибыль
	<i>profitTax</i>	Налог на прибыль и иные обязательные платежи
	<i>dispositionOfProfits</i>	Использовано прибыли
	<i>accounts</i>	Счета бухгалтерского учета

Окончание таблицы 4.2

1	2	3
Операции	<i>getResourceStockLevel</i> <i>getResourceCost</i> <i>takeProduct</i> <i>takeResource</i> <i>calculateProductPrice</i> <i>increaseCreditRequirements</i> <i>decreaseCreditRequirements</i>	Объем ресурса на складе Стоимость ресурса на складе Получение вида продукции со склада (возвращает суммарные затраты ресурсов) Получение ресурса со склада (возвращает суммарную стоимость) Расчет цены продукции Увеличение суммы требуемого кредита Уменьшение суммы требуемого кредита

Перечень классов-функций, представляющих активности процессов, приведен в таблице 4.3.

Таблица 4.3 – Перечень классов активностей

Имя класса	Активность
<i>CShipmentGenerating</i>	Формирование входного потока заказов в виде требуемых отгрузок продукции
<i>CShipmentPrepaying</i>	Предварительная оплата за продукцию
<i>CShipmentShipping</i>	Отгрузка продукции
<i>CShipmentPaying</i>	Оплата за продукцию (с отсрочкой)
<i>CMasterProductionScheduling</i>	Планирование производства
<i>CResourceOrdering</i>	Управление поставками ресурсов
<i>CDeliveryPrepaying</i>	Предварительная оплата за ресурсы
<i>CDeliveryDelivering</i>	Поставка ресурсов
<i>CDeliveryPaying</i>	Оплата за ресурсы (с отсрочкой)
<i>CSalesContractRunning</i>	Выполнение контракта на реализацию продукции
<i>CProductionLaunching</i>	Запуск продукции в производство
<i>CProductionStartup</i>	Начало производства партии продукции
<i>CProductionCompletion</i>	Завершение производства партии продукции
<i>CCostChanging</i>	Изменение стоимости во времени
<i>CProductPriceChanging</i>	Изменение цены на один вид продукции
<i>CResourcePriceChanging</i>	Изменение цены ресурса
<i>CFixedCostsChanging</i>	Изменение постоянных затрат предприятия
<i>COtherVariableCostsChanging</i>	Изменение переменных затрат предприятия
<i>CAccounting</i>	Бухгалтерский учет
<i>CDebtPaying</i>	Оплата задолженности, базовый класс
<i>CWagePaying</i>	Оплата задолженности по заработной плате
<i>CTaxPaying</i>	Оплата задолженности по налогам
<i>CSocialInsuranceTaxPaying</i>	Оплата задолженности социальному страхованию
<i>CCreditPaying</i>	Оплата задолженности по кредиту (по выплате процентов)
<i>CBillPaying</i>	Оплата задолженности перед поставщиками и подрядчиками
<i>CApplyingForCredit</i>	Обращение за кредитом
<i>CCreditReceiving</i>	Получение кредита
<i>CCreditRepaying</i>	Выплата процентов и возврат кредита
<i>CStatisticsGathering</i>	Сбор статистики

Программный интерфейс модели представлен классом *CEnterprise*, который обеспечивает получение перечня входных и выходных переменных, инициализацию и прогон модели посредством реализации интерфейса *CModel* [221, с. 155].

Внешний интерфейс модели организован с использованием стандартных средств, предоставляемых системой моделирования. Для этого в текст модуля добавлена реализация функции `main()`

```
MODEL_MAIN(CEnterprise, CXMLModelData),
```

организующая обмен данными на основе XML-файлов. В готовом виде имитационная модель представлена программным модулем *Enterprise3.EXE*.

Представленная выше базовая ИМ, входящая в ПТКИ *BelSim*, позволяет с минимальными затратами и в кратчайшие сроки реализовать ИМ для решения практических задач в составе ERP-системы управления предприятием благодаря высокому уровню автоматизации основных этапов разработки и использования ИМ.

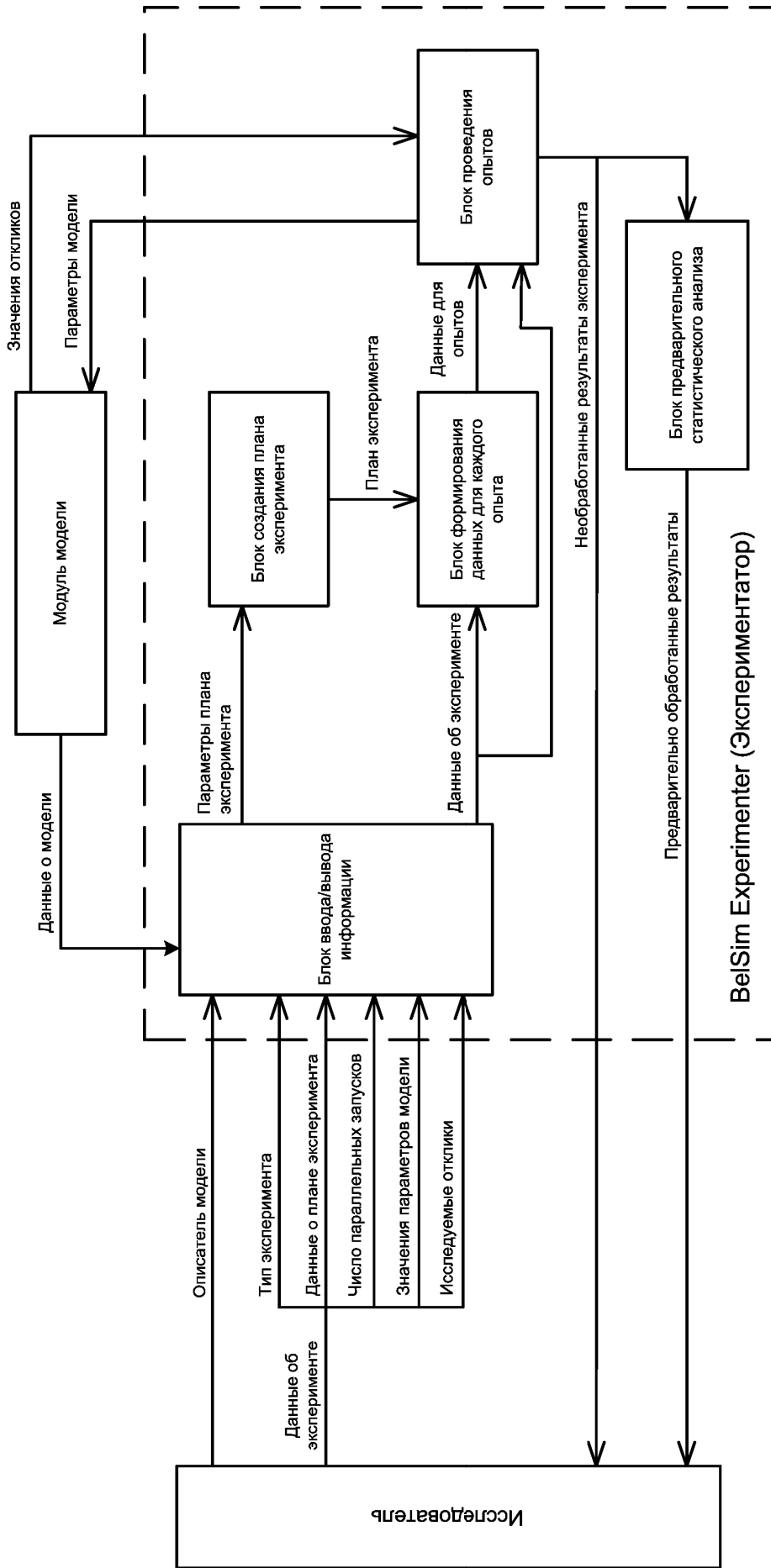
4.6 Программные средства автоматизации планирования, проведения и обработки результатов имитационных экспериментов

Эксперименты с имитационными моделями сложных систем требуют значительных временных затрат. При этом важно исключить или значительно уменьшить ошибки при построении и реализации плана эксперимента, что становится непростой задачей при большом количестве исследуемых параметров и откликов модели, свойственных сложным системам. Автоматизация процедуры проведения эксперимента является необходимой задачей, для решения которой разработана программная подсистема *BelSim Experimenter* (см. рисунок 4.1).

Программная система состоит из нескольких блоков: блок ввода/вывода информации, блок создания плана эксперимента, блок подготовки данных для проведения опытов, блок проведения опытов и блок предварительного статистического анализа. Схема информационных потоков между блоками системы представлена на рисунке 4.6 [221, с. 79–81].

Блок ввода/вывода информации направлен на поддержку интерфейса, обеспечивающего удобный ввод необходимой информации. Реализован в виде приложения *ExperimentDesigner*, представляющего собой набор форм с использованием Microsoft Visual C++ .NET, Microsoft .NET Framework 2.0.

Блок создания плана эксперимента формирует его в зависимости от указанного типа, количества параметров и опытов на основе общепринятых методик [114]. Указанный блок использует внутренний модуль *Experimental Design (DOE)* приложения STATISTICA 6.0.

Рисунок 4.6 – Схема информационных потоков в подсистеме *BelSim Experimenter*

Блок формирования данных для каждого опыта производит сопоставление уровней из плана эксперимента и имен параметров модели. Реализован в виде макроса в приложении статистического анализа STATISTICA 6.0.

Блок проведения экспериментов по представленному плану эксперимента проводит опыты над моделью: осуществляет взаимодействие с моделью, сохраняет результаты опытов, для удобства пользователя обеспечивает вывод информации о ходе проведения эксперимента. Блок проведения экспериментов, механизмы взаимодействия с моделью и потоками данных реализованы в программе *Experimenter*.

Блок предварительного статистического анализа выводит результаты в удобной форме и проводит необходимые действия для последующего анализа результатов. Блок реализован в виде макроса *ExperimentData* в приложении STATISTICA 6.0.

Технология использования подсистемы *BelSim Experimenter* при проведении экспериментов различного типа представляет собой последовательность действий, изображенную на диаграмме (рисунок 4.7) [211, с. 3–9], выполненной с использованием унифицированного языка моделирования UML [106].

Подсистема проводит следующие типы экспериментов (поток «*Тип эксперимента*»): оценка погрешности моделирования; анализ длины переходного процесса и устойчивости результатов моделирования; анализ чувствительности откликов к изменению входных параметров; полный факторный эксперимент; дробный факторный эксперимент [211, с. 5].

ПО для планирования, проведения и обработки результатов имитационных экспериментов оперирует потоками данных, организованными в виде XML-файлов, доступ к которым реализован через стандартный интерфейс XML DOM Level 3 с использованием библиотеки MSXML 4.0 [374]. Для обмена данными используются XML-файлы двух типов: файлы данных модели [268, с. 459] и файлы данных эксперимента, – структура которых приведена на рисунках 4.8 и 4.9 соответственно.

Проектирование блока проведения опытов – программы *Experimenter* выполнено с использованием объектно-ориентированной технологии, в основе которой лежит объектная модель. Данная модель направлена на описание ключевых абстракций и механизмов, которые формируют предметную область и архитектуру блока. Для описания ролей и обязанностей объектов использована диаграмма классов, для выражения решений о поведении блока – диаграммы состояний/активностей и сценарии. Все наименования в модуле удовлетворяют правилам построения идентификаторов C++, так как модель экспортируется в код C++. Семантика отношений между классами модуля *Experimenter* представлена на рисунке 4.10.

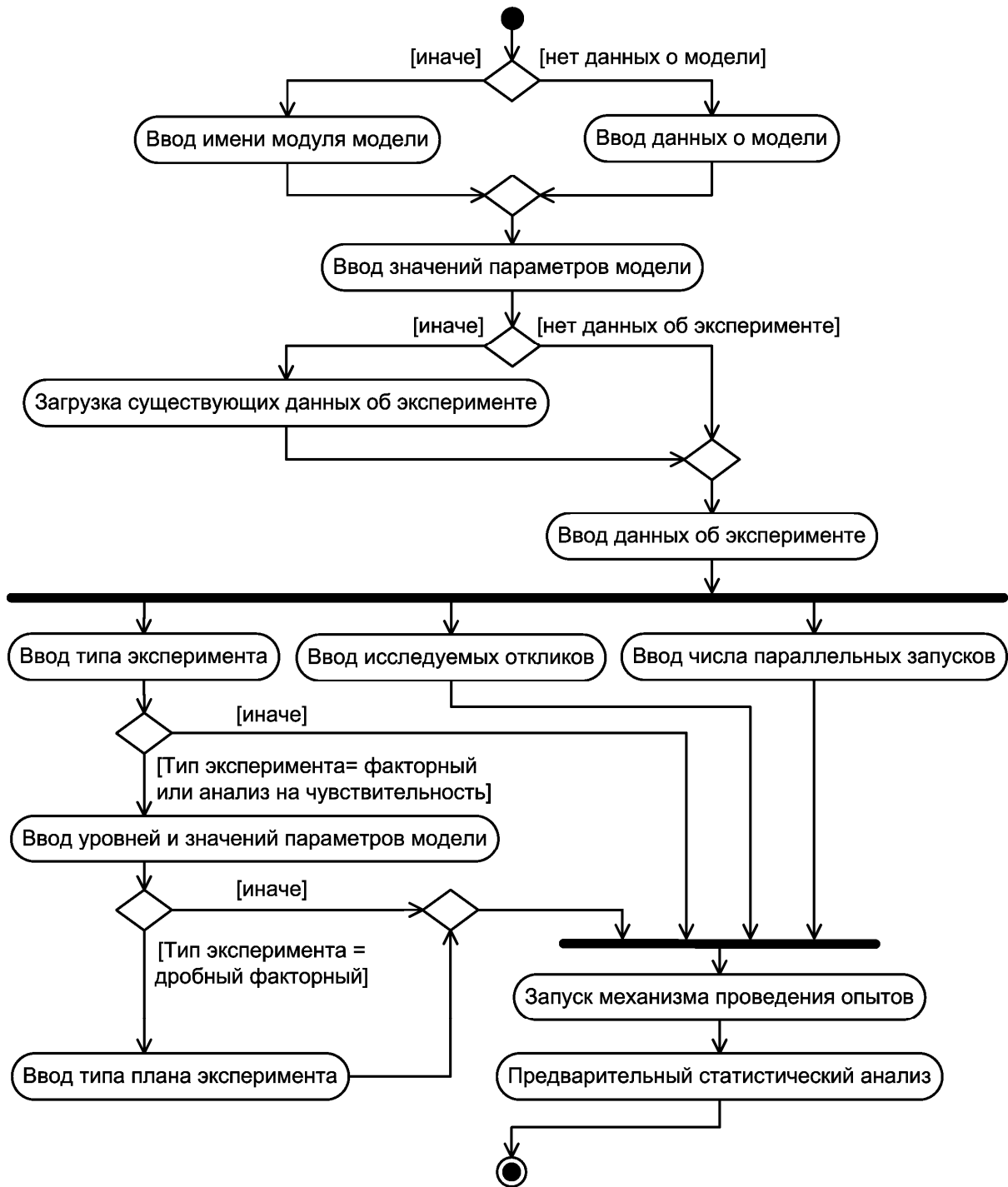


Рисунок 4.7 – Технология использования подсистемы *BelSim Experimenter* для проведения экспериментов различного типа с моделями сложных систем

Корневой узел

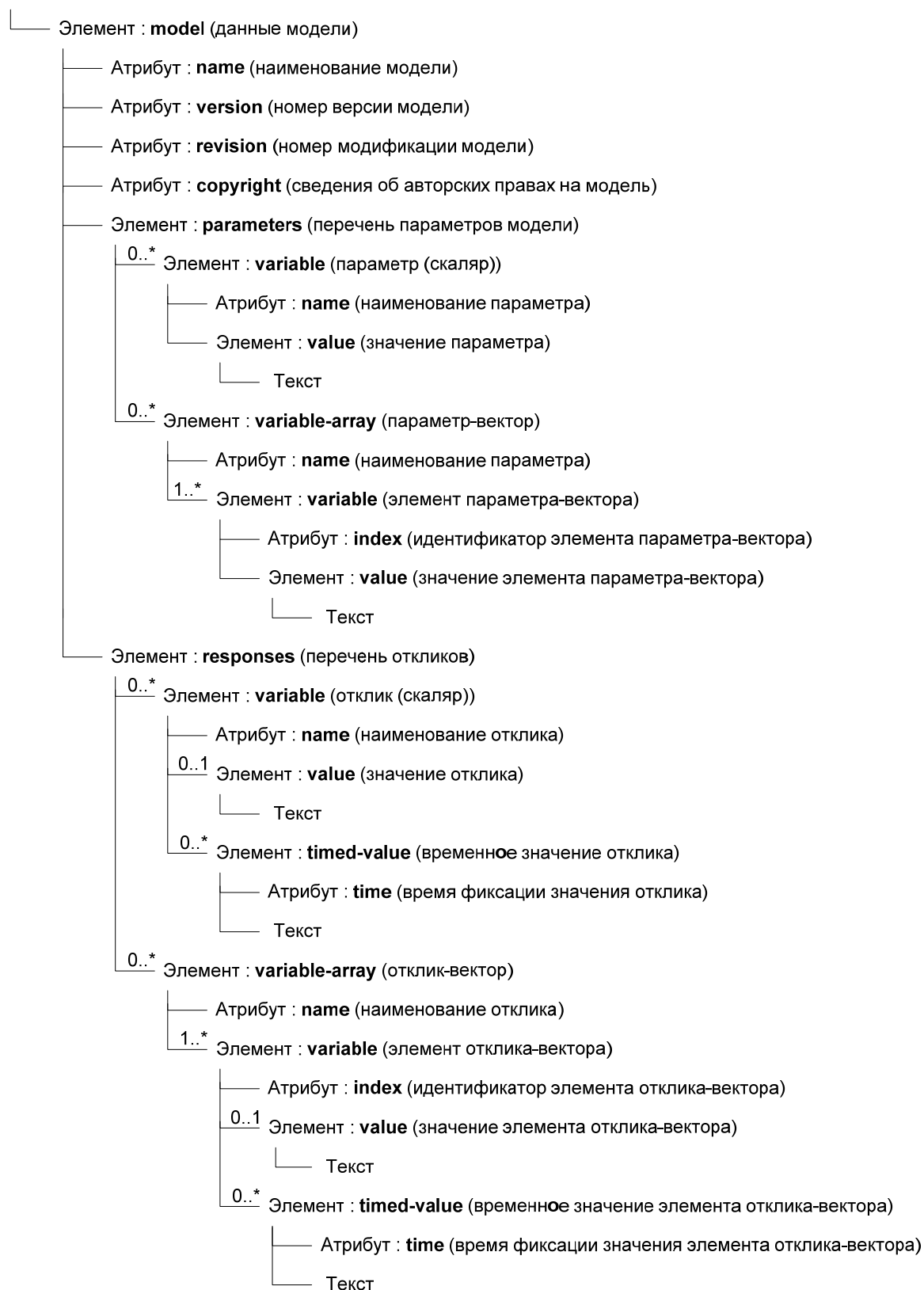


Рисунок 4.8 – Структура файла данных модели

Корневой узел

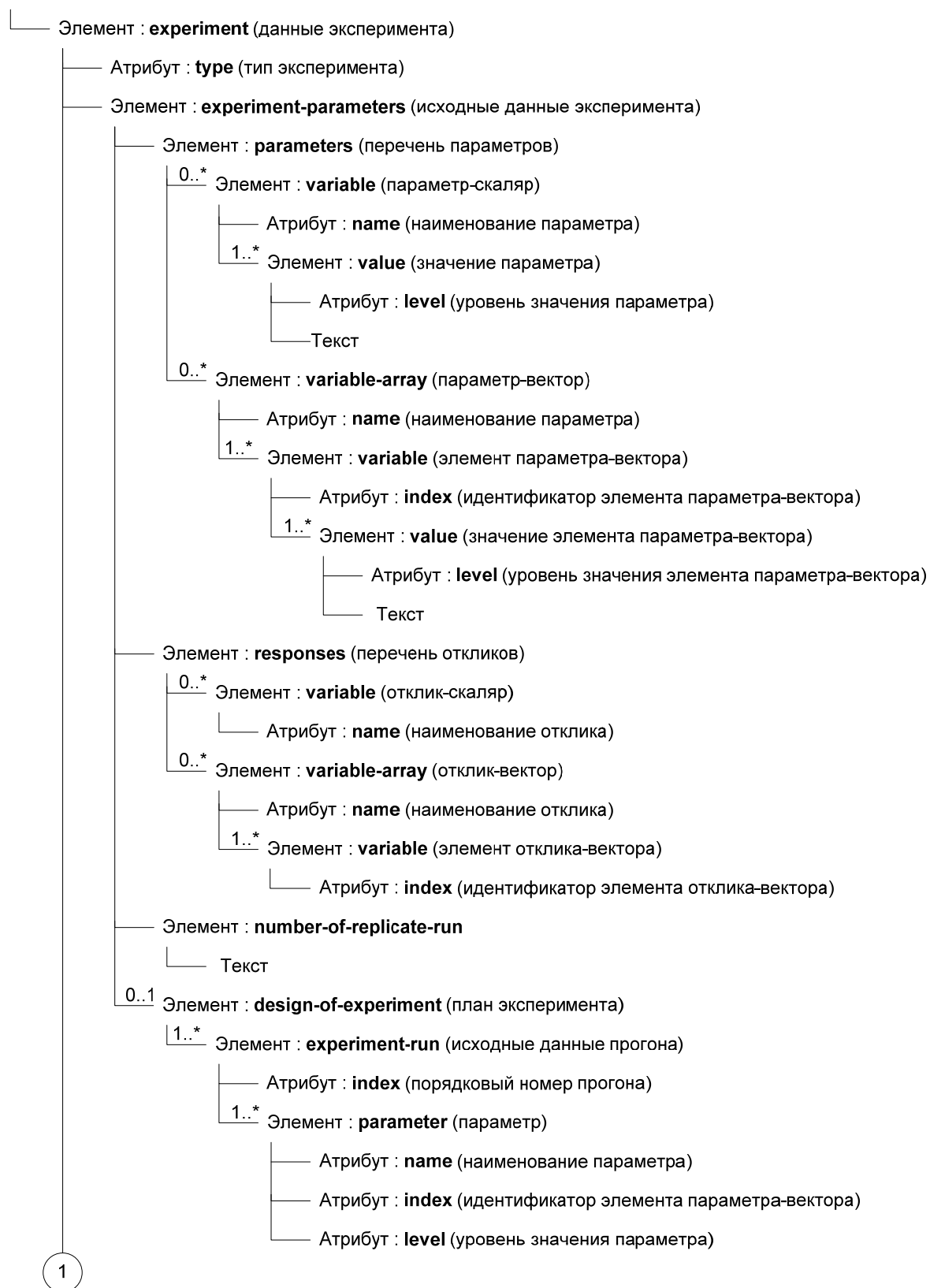
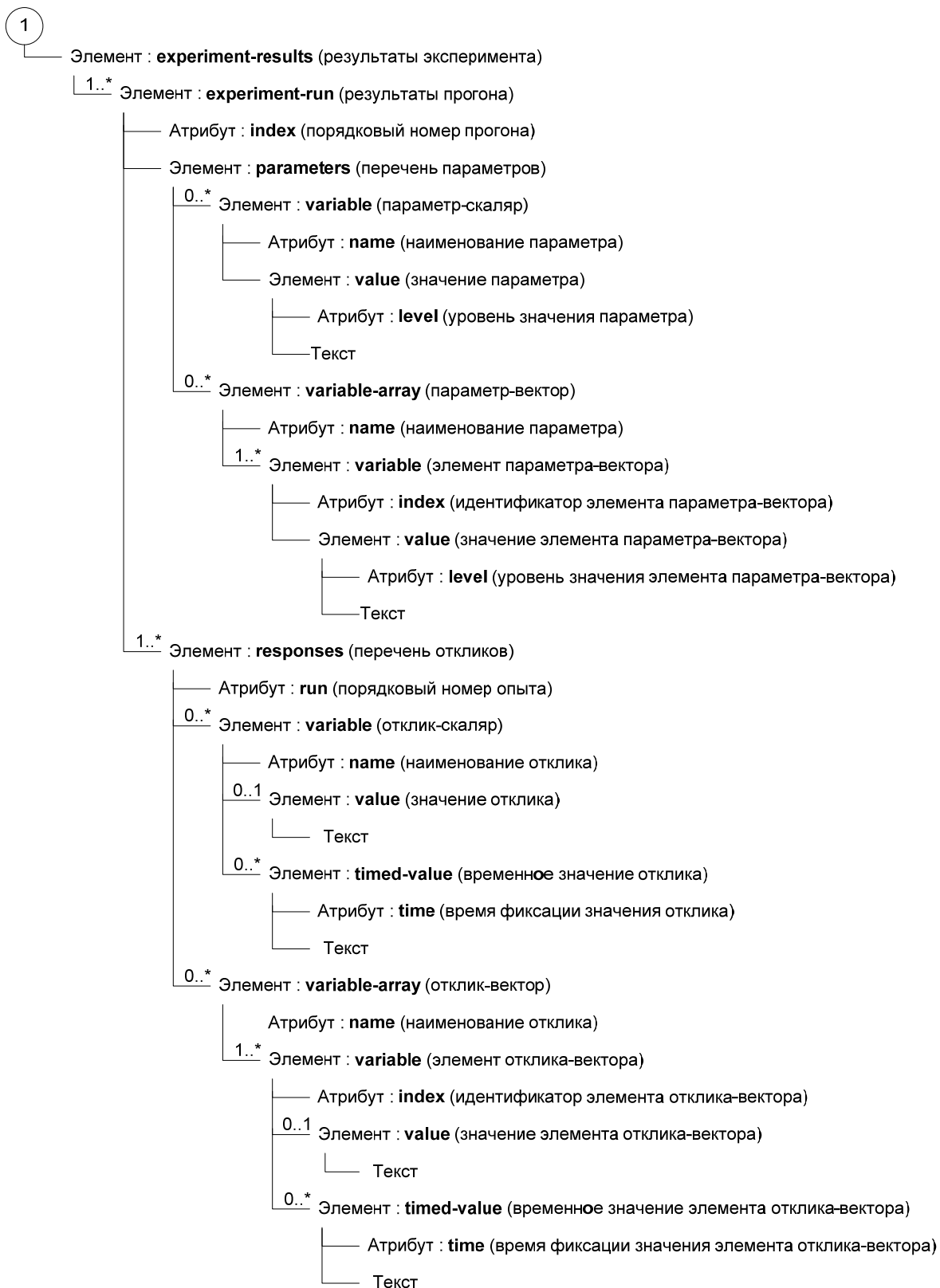


Рисунок 4.9 – Структура файла данных эксперимента



Окончание рисунка 4.9

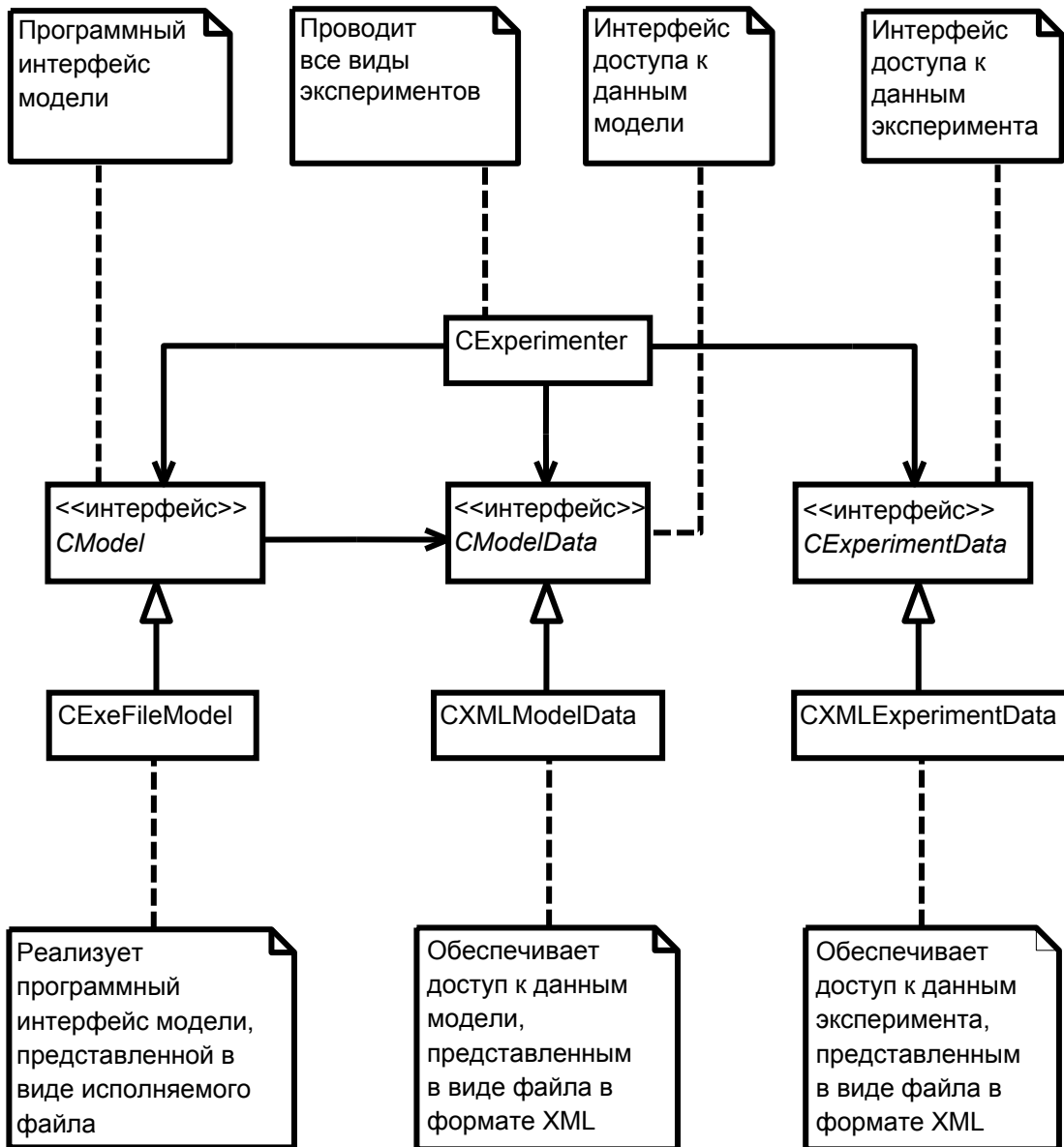


Рисунок 4.10 – Диаграмма классов программы *Experimenter*

Основная действующая абстракция (механизм проведения опытов) – *CExperimenter*. Ответственность – организация всех видов экспериментов. Жизненный цикл объектов класса заключается в выполнении операции *run()*:

```
void run(CModel &model, CModelData &modelData, CExperimentData &experimentData),
```

которая осуществляет проведение эксперимента на основе данных *ExperimentData* над моделью с интерфейсом *model*, доступ к данным которой осуществляется через интерфейс *modelData*.

Посредством операций *getModelData()* и *run()* интерфейса *CModel* осуществляется получение данных и прогон модели. Один из вариантов

взаимодействия модели в виде исполняемого модуля и блока проведения опытов реализован в классе *CExeFileModel*. Объекты данного класса имеют жизненный цикл, представленный на рисунке 4.11. При инициализации создается экземпляр класса, соответствующий модулю с именем *sModelFileName*. Из состояния ожидания он переводится сообщениями *run* и *getModelData* в состояния, из которых производится вызов модуля модели соответствующими параметрами:

$$/I \mid /R \langle \text{файл} \rangle [/RS: \langle \text{число} \rangle] [/IO: \langle \text{файл} \rangle] [/F],$$

где */I* – команда создания файла данных модели со списком входных и выходных переменных;

/R – команда запуска модели;

<файл> – имя файла данных модели;

/RS: <число> – начальное число для инициализации генераторов псевдослучайных чисел;

/IO: <файл> – файл для вывода промежуточных результатов моделирования;

[/F] – команда отказа от выполнения дополнительных операций (форматирования выходных данных и т. д.) с целью ускорения работы модели [221, с. 88–89].

Интерфейсы *CModelData* и *CExperimentData* и их реализация в виде классов *CXMLModelData* и *CXMLExperimentData* используются для организации доступа к данным модели и эксперимента. Объекты этих классов имеют жизненный цикл, представленный на рисунке 4.12. После создания объект переходит в состояние «Создание», в котором формируются основные теги документа. Затем осуществляется переход в состояние «Динамическое изменение документа». Содержимое документа можно изменить группой операций *setMethods* в состоянии «Редактирование», прочитать группой операций *getMethods* в состоянии «Чтение данных», загрузить новыми данными из существующего файла *sFileName* операцией *load*, сохранить в файле *sFileName* операцией *save*.

Сценарий взаимодействия объектов в ходе проведения экспериментов следующий. Исследователем создается объект *model*, привязанный к модулю модели *sModelFileName*. Загружается файл данных модели сообщением *load* объекту *modelData: CXMLModelData*. С помощью этого объекта выводится информация о модели. Загружается файл данных эксперимента сообщением *load* объекту *experimentData: CXMLExperimentData*, при помощи которого считывается информация об эксперименте и выводится на экран. Затем сообщением *run* экземпляру класса *CExperimenter* запускается механизм проведения опытов (рисунок 4.13). Результаты эксперимента сохраняются выдачей сообщения *save* объекту *experimentData* [221, с. 89].



Рисунок 4.11 – Схема алгоритма блока проведения опытов

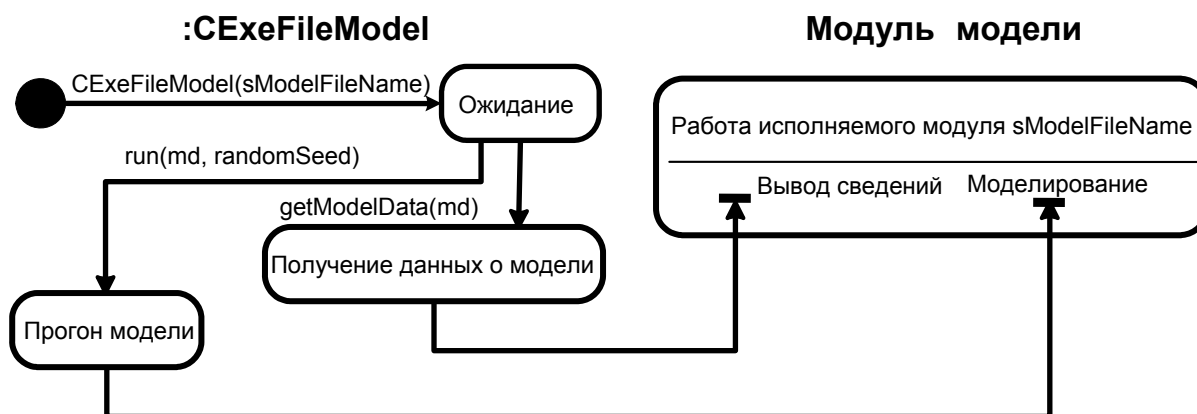


Рисунок 4.12 – Жизненный цикл экземпляров класса CExecFileModel

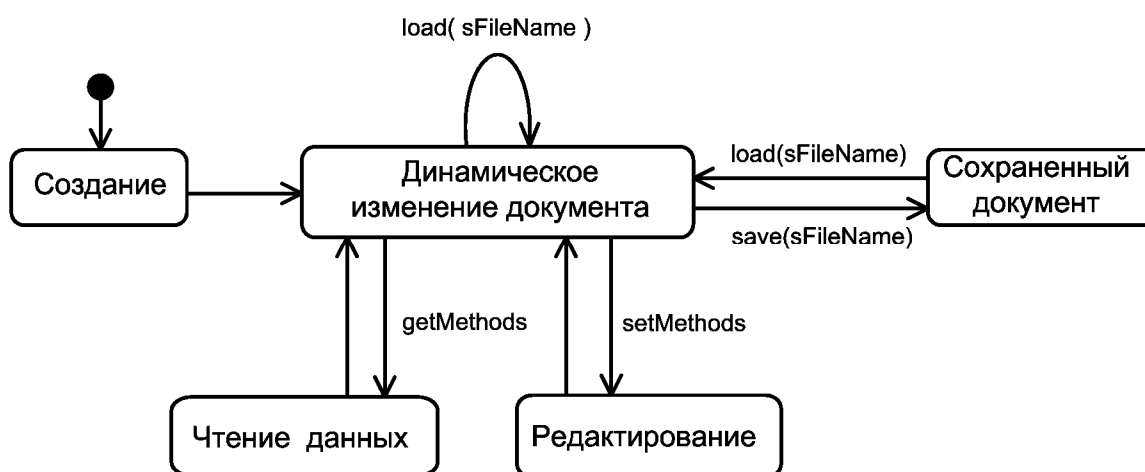


Рисунок 4.13 – Жизненный цикл экземпляров классов, реализующих интерфейсы CModelData и CExperimentData

Взаимодействие с компонентами ПТКИ в процессе планирования, проведения и обработки результатов экспериментов показано на рисунке 4.14. Исследователь подает на вход системы необходимые данные о планируемом эксперименте. В соответствии с ними ПТКИ проводит эксперимент с моделью.

Взаимодействие с моделью заключается в передаче модели исходных данных для ее работы. Модель после окончания работы выдает значения множества откликов, которые затем группируются системой и предоставляются пользователю в виде выходной информации. Эта информация может быть проанализирована как в представленной системе, так и в другой. Моделью также выдаются сведения о параметрах и откликах. Используется структура информационных потоков, представленная на рисунке 4.6.

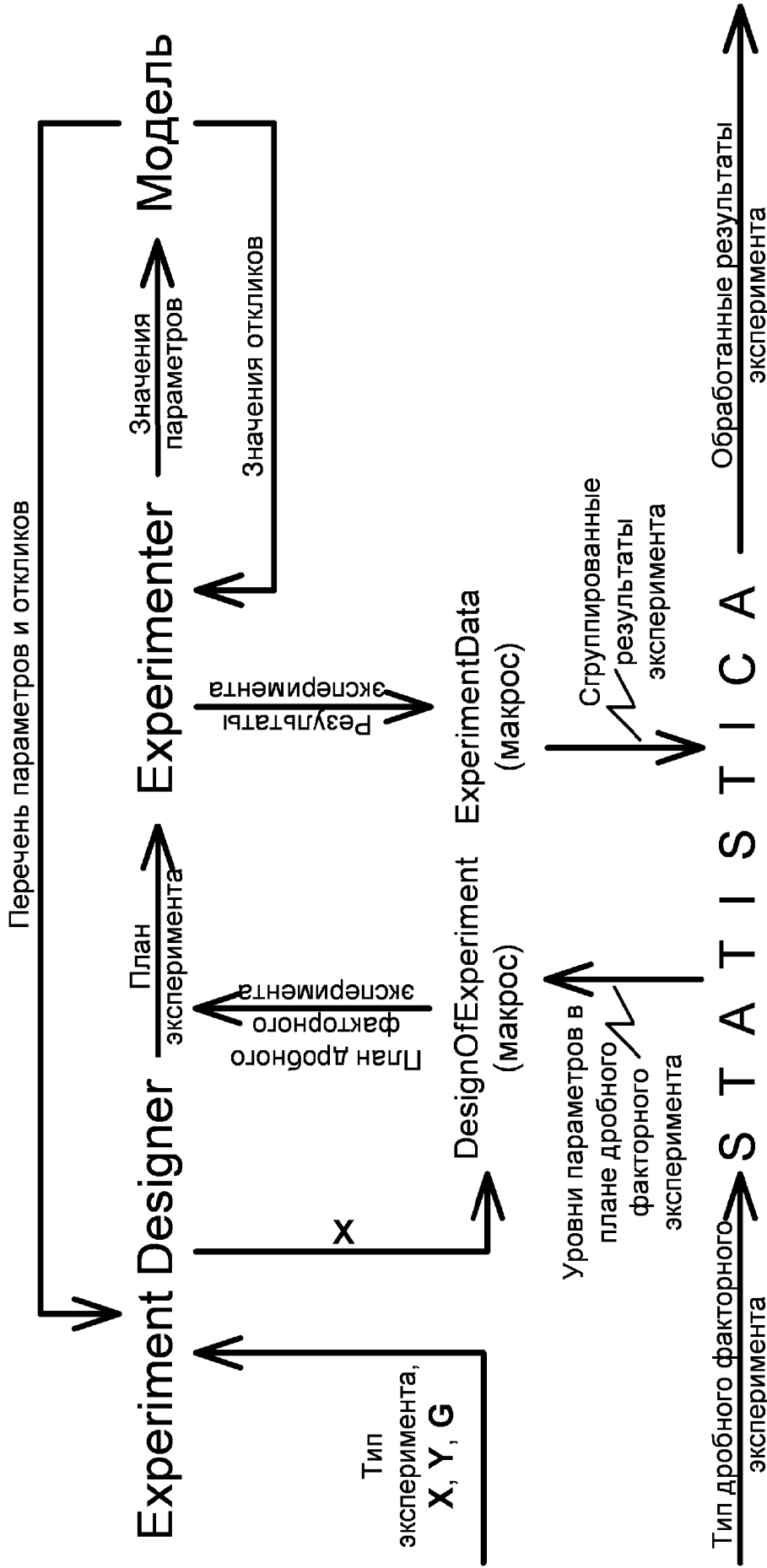


Рисунок 4.14 – Схема постановки имитационных экспериментов

Поток «*Описатель модели*» может содержать следующие значения:

- имя модуля модели, тогда блок ввода/вывода информации сам запросит по данному имени поток «*Данные о модели*»;
- поток «*Данные о модели*».

Поток «*Данные об эксперименте*» имеет сложную структуру. При этом поток «*Тип эксперимента*» может иметь следующие возможные значения: оценку погрешности моделирования; анализ длины переходного процесса и устойчивости результатов моделирования; анализ чувствительности откликов к изменению входных параметров; полный факторный эксперимент; дробный факторный эксперимент [221, с. 108].

В зависимости от содержимого потока «*Тип эксперимента*» поток «*Данные о плане эксперимента*» представляет собой оценку погрешности моделирования или анализ длины переходного процесса и устойчивости результатов моделирования – множество двоек <имя параметра, значение>; анализ чувствительности откликов к изменению входных параметров – множество структур (имя параметра, {<уровень, значение>}), причем уровень принимает значения из множества {–1, 0, 1}; полный факторный эксперимент – множество структур (имя параметра, {<уровень, значение >}), причем уровень в пределах от –1 до 1; дробный факторный эксперимент – множество структур (имя параметра, {<уровень, значение >}), причем уровень в пределах от –1 до +1; тип плана эксперимента.

Поток «*Значения параметров модели*» содержит информацию (множество пар <имя параметра, значение>) о параметрах модели, не участвующих в эксперименте.

Поток «*Исследуемые отклики*» – список имен исследуемых откликов.

Поток «*Данные о модели*» представляет собой множество имен параметров и откликов модели. Поток «*Значения откликов*» – это множество структур (имя отклика, список значений). Поток «*Параметры модели*» является множеством пар <имя параметра, значение>.

Поток «*Необработанные результаты эксперимента*» – это множество опытов. Опыт описывается списком исходных параметров (имя, значение, уровень) и списком запусков. Запуск описывается номером, списком откликов (имя, список значений).

Поток «*Параметры плана эксперимента*» формируется, когда поток «*Тип эксперимента*» содержит дробный факторный эксперимент. Он включает в себя информацию о количестве параметров, участвующих в эксперименте, о количестве опытов, типе плана эксперимента.

Поток «*План эксперимента*» содержит план эксперимента. Поток «*Данные для опытов*» – множество комбинаций уровней параметров модели <имя, уровень> для каждого опыта в соответствии с планом эксперимента. Эти потоки используются лишь для дробного факторного эксперимента.

Последовательность выполнения операций при проведении экспериментов представлена на рисунке 4.15. Пользователю предоставляются развитые средства формализации, построения моделей, постановки экспериментов, анализа их результатов [221, с. 108–109].

4.7 Программная поддержка интеграции ПТКИ BelSim с комплексной информационной системой

Для интеграции ПТКИ *BelSim* с КИС выполняется последовательность действий, определяемая по аналогии с механизмом интеграции программных информационных систем, представленным в [230, с. 60], следующими этапами (рисунок 4.16) [261, с. 113].

Этап 1. Настройка параметров подключения к базе данных.

Шаг 1. Выбор пути к базе данных, из которой необходимо получить данные для исследуемой модели.

Шаг 2. Выбор файла, в котором хранится структура XML-файла данных модели.

Шаг 3. Формирование запроса к базе данных с указанием названий таблиц и полей, в которых находятся требуемые данные.

Этап 2. Подключение к базе данных предприятия или КИС (например, 1С: Предприятие) и выбор требуемых данных с применением технологии *ADO*, используя библиотеку *msado15.dll*, входящую в дистрибутив операционной системы *Windows XP Professional*.

Этап 3. Запись полученной информации в XML-файл данных модели (после выбора данных из КИС предприятия их следует представить в виде XML-файла, который ПТКИ *BelSim* может использовать для проведения имитационных экспериментов). Интерфейсом доступа к содержимому XML-документов является объектная модель документа *DOM (Document Object Model)*, представляющая внутреннюю структуру документа в виде совокупности объектов [252, с. 91–92].

Программное приложение *BelSim Data Integrator* (рисунок 4.17) считывает параметры подключения к базе данных из конфигурационного файла *Settings.cfg*, выполняет подключение к базам данных и выбор требуемых данных из таблиц, используя объекты классов библиотеки *msado15.dll*. При помощи *DOM* и файла со структурой будущего документа составляется конечный XML-файл, который передается в *Experiment Designer* ПТКИ *BelSim* (на рисунке 4.14 XML-файл условно не показан).

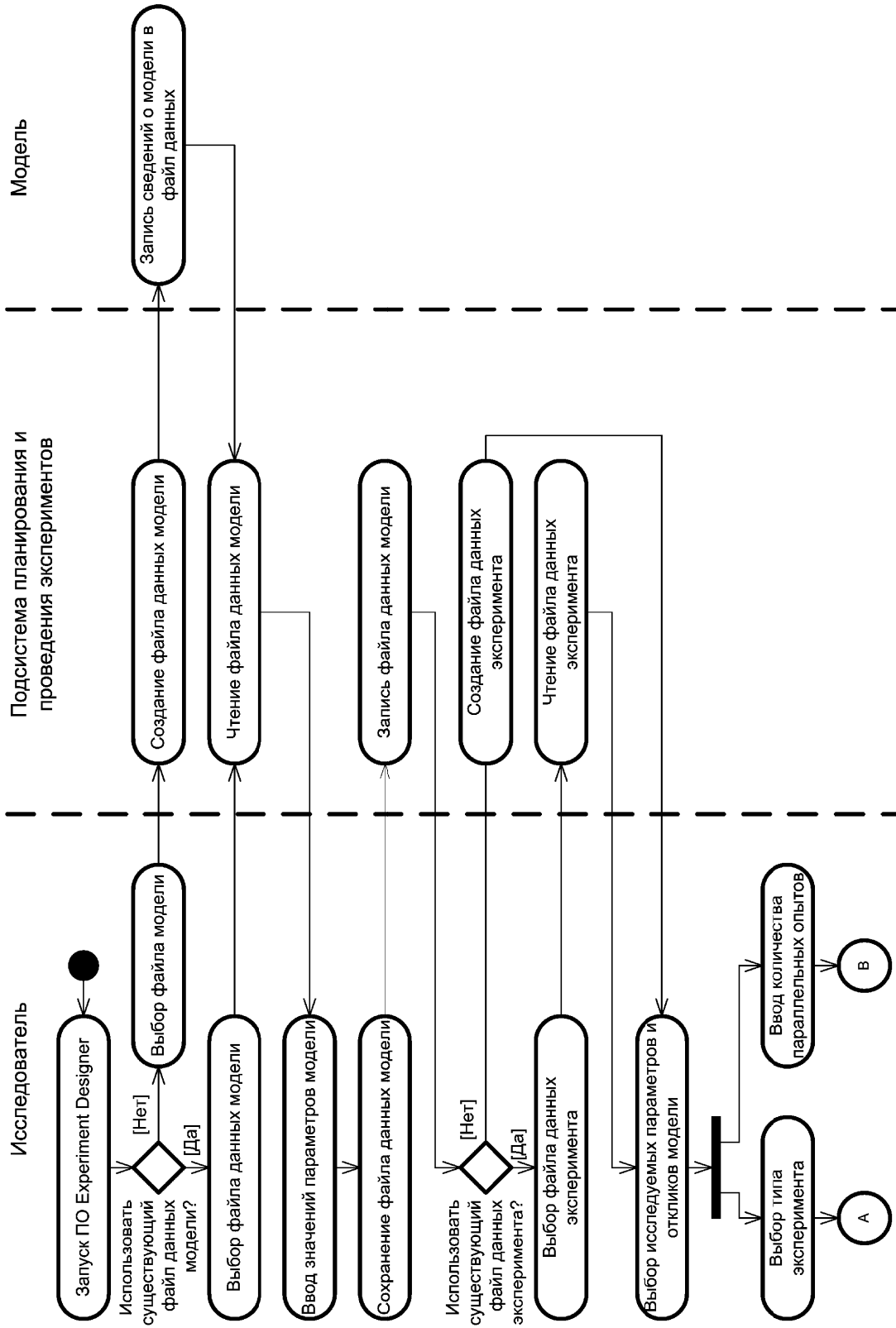
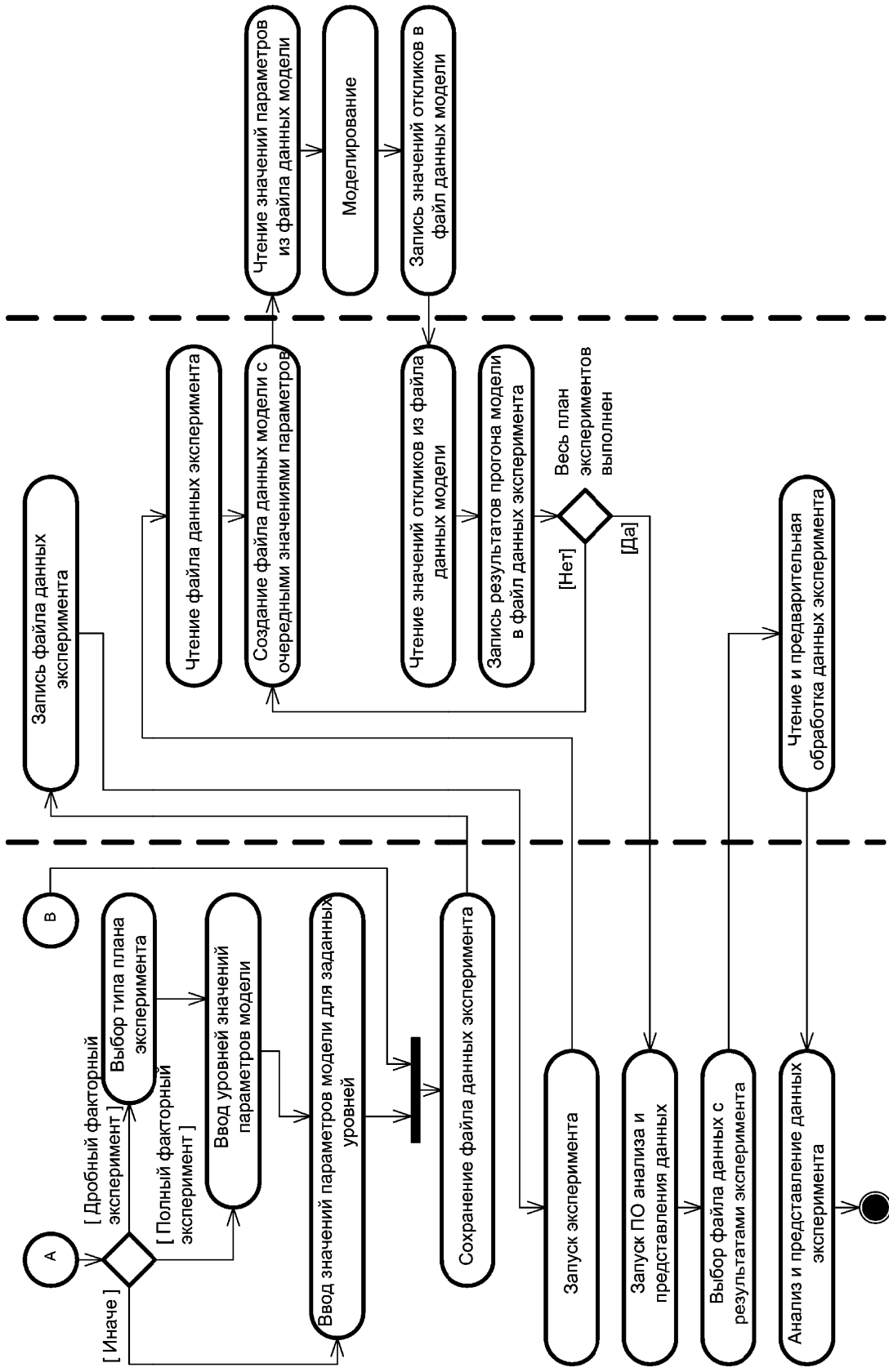


Рисунок 4.15 – Диаграмма выполнения операций при планировании, проведении и обработке результатов имитационных экспериментов



Окончание рисунка 4.15

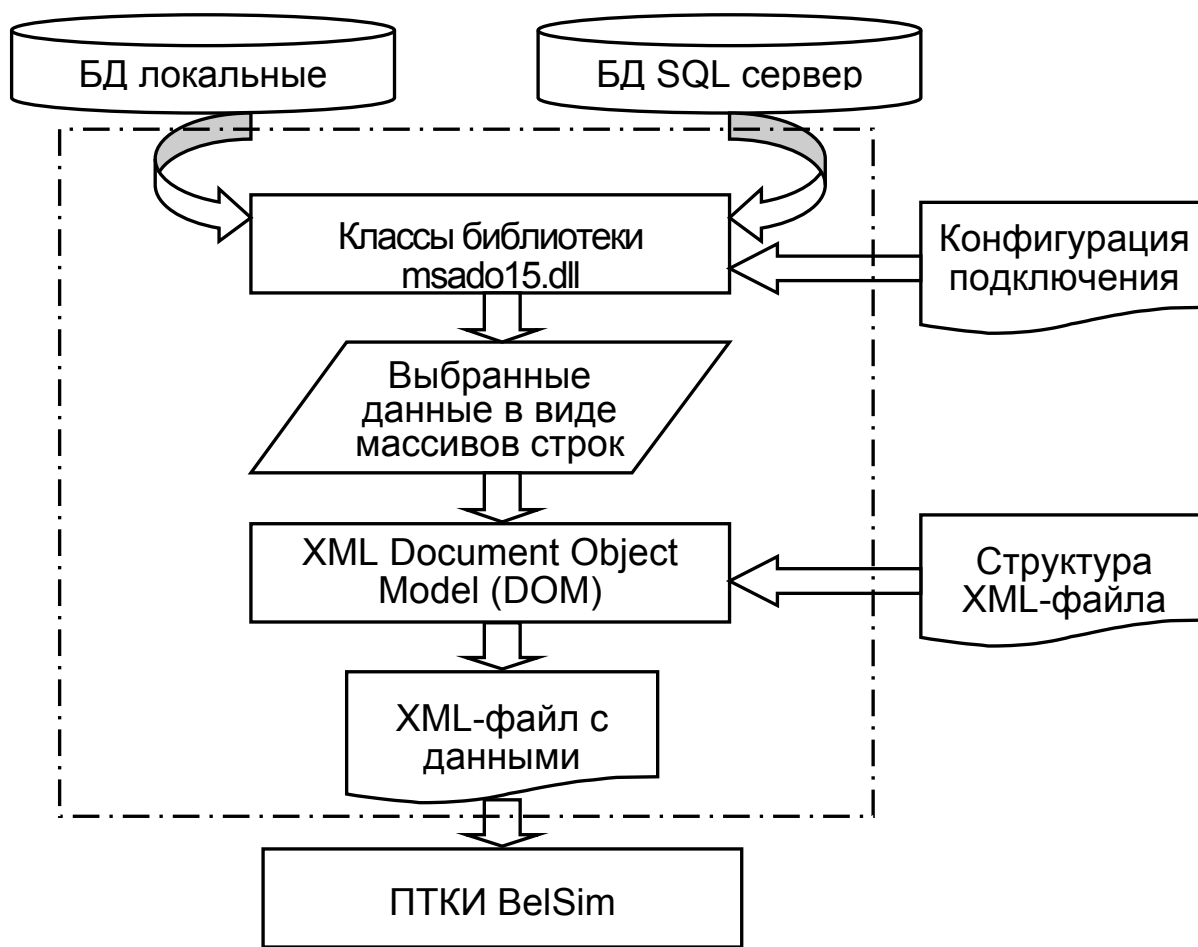


Рисунок 4.16 – Логическая структура интеграции ПТКИ BelSim в КИС

Основным классом, ответственным за сбор и конвертирование данных в приложении *BelSim Data Integrator*, является *DBF2XML* со следующими атрибутами: *NumAllParameters* – задает общее число параметров файла *Settings.cfg*; *BelsimPath* – определяет путь к директории, в которой установлен ПТКИ *BelSim*; *OnecPath* – путь к директории, в которой хранятся базы данных; *ModelStructureXML* – имя *XML*-файла, загружаемого в качестве шаблона выходного документа; *TablesAndColumns* – указатель на динамический массив, содержащий названия таблиц и полей, необходимых для импорта данных; *SQLServer* – имя сервера, на котором хранятся базы данных (если для хранения данных используется СУБД *SQL Server*, см. рисунок 4.16).

Ядром приложения *BelSim Data Integrator* является метод *convert()* класса *DBF2XML*, обеспечивающий сбор и конвертирование данных по следующему алгоритму: создать объект класса *CRealXMLMData*; загрузить структуру *XML*-файла; создать объекты *Connection* и *Recordset*, используя *COM*; установить соединение с базой данных; для каждой таблицы и колонки сформировать и выполнить запрос к базе данных; создать динамический массив строк; преобразовать к типу *CString* и копировать записи из запроса в массив строк; копировать имена и значения параметров из массива строк в *XML*-файл данных модели; закрыть соединение с базой данных.



Рисунок 4.17 – Диаграмма классов приложения *BelSim Data Integrator*

Используемый в реализации алгоритма класс *Connection* предназначен для хранения такой информации о подключении к источнику данных, как имя, местоположение данных, имя пользователя и пароль, имя провайдера *OLE DB* и т. д. Основные функции класса *Connection* реализуют открытие источника данных, выполнение запроса к базе данных и возврат результата, закрытие соединения с источником данных [261, с. 113].

Класс *Recordset* содержит реализацию методов, позволяющих пользователю обновлять данные в базе данных и перемещать указатель текущей записи по результирующему набору. Также этот объект содержит поля, хранящие число записей в наборе, тип курсора, тип блокировки и т. д. Функции класса *Recordset*: *MoveFirst* – передвигает курсор на первую запись; *MoveNext* – передвигает курсор на следующую запись; *RecordCount* – возвращает количество записей; *adoEOF* – возвращает true, если курсор находится на первой записи; *GetCollect* – считывает текущее значение записи.

Класс *CRealXMLModelData* (см. рисунок 4.17) обеспечивает доступ к классам библиотеки *MSXML* на основе COM-технологии с использованием директивы `#import <msxml4.dll>`. При этом используются функции класса *IXMLDOMDoc* и функции класса *IXMLDOMElement*. Функции класса *IXMLDOMDoc*: *AppendChild* – добавляет новый дочерний узел как послед-

ний (в конец списка) дочерний узел данного узла; *CreateElement* – создает узел с заданным именем; *Load* – загружает XML-документ по указанному местоположению; *Save* – сохраняет XML-документ по заданному местоположению. Основные функции класса *IXMLDOMElement*: *SetAttribute* – присваивает значение указанному атрибуту; *findVariable* – возвращает первый найденный узел переменной; *setAttributeNode* – добавляет или изменяет указанный атрибут; *findVariableArray* – возвращает первый найденный узел переменной массива.

Функции классов *CDialog*, *CBSMenuOptDlg*, *Cdbf2xmlDlg*, *COneCmenuOptDlg* (см. рисунок 4.17) обеспечивают реализацию пользовательского интерфейса [261, с. 111–117].

4.8 Технология модернизации ПТКИ BelSim для организации распределенных вычислений

Современная концепция имитационного моделирования хорошо согласуется с моделью параллельных вычислений *MPMD* (*Multiple Program – Multiple Data*) [203, с. 9]. Во-первых, сложные системы состоят из параллельно функционирующих компонентов на разных уровнях иерархической структуры, что приводит к использованию в ИМ множества модулей, одновременно реализующих поведение элементов и подсистем объекта; во-вторых, разнородность элементов и подсистем порождает разнородность применяемых математических схем и разнородность программной реализации модулей ИМ [132, с. 210].

Вычислительные эксперименты с моделями функционирования промышленного предприятия требуют значительных временных затрат, для сокращения которых в ПТКИ реализована возможность прогона модели с использованием ресурсов локальной вычислительной сети. Для этого используется библиотека MPI-функций обмена данными между процессами, реализованными для языка C++. Программным средством реализации MPI является MPICH, обеспечивающая выполнение всех функций MPI в исполнительной среде [203, с. 28]. Для реализации MPI-версии в программе *Experimenter* необходимо внести следующие изменения [248, с. 138].

Шаг 1. Добавляются новые переменные для получения информации об активном процессе и о программе *Experimenter* в целом: идентификатор номера процесса – *MyId*; количество выполняемых процессов – *NumProcs*; для определения времени работы программы – *Start*.

Шаг 2. Для слияния XML-файлов с результатами эксперимента добавлена функция *mergeXML*:

```
void mergeXML(wstring fileNameDist, wstring fileNameSrc, int madeMerge),
```

где *fileNameDist* – путь к файлу, в который будет добавлен файл-источник;

fileNameSrc – путь к файлу-источнику;

madeMerge – количество вызовов функции mergeXML.

Алгоритм работы программы *Experimenter* с дополнительными возможностями распределения вычислений представлен на рисунке 4.18.

В MPI-программе после строк определения переменных следуют три обязательные строки:

```
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&NumProcs);
MPI_Comm_rank(MPI_COMM_WORLD,&MyId).
```

Обращение к MPI_Init должно быть первым обращением, т. к. оно инициализирует MPI.

Коммуникатор MPI_COMM_WORLD описывает состав процессов и связи между ними. Вызов MPI_Comm_size возвращает в NumProcs число процессов (прогонов модели), задаваемых пользователем в программе. Вызывая MPI_Comm_rank, каждый процесс выясняет свой номер (rank) в группе, связанной с коммуникатором [203, с. 25].

Для синхронизации процессов используется функция MPI_Barrier, которая блокирует работу вызвавшего ее процесса до тех пор, пока все другие процессы группы также не вызовут эту функцию. Завершение работы этой функции возможно только всеми процессами одновременно [42, с. 111].

На рисунке 4.19 представлен фрагмент программы *Experimenter* с дополнениями для организации распределенных вычислений.

Результаты прогонов модели записываются в свой отдельный XML-файл. Для слияния результатов эксперимента из отдельных временных XML-файлов в одном целевом файле создана функция mergeXML.

Одному из процессов присваивается нулевой идентификатор, и он создает целевой файл. Другие процессы создают временные файлы. Функция mergeXML создает структуру для XML-документов docSrc и docDist. Результаты из docSrc добавляются в docDist, который сохраняется как целевой файл с результатами эксперимента [248, с. 137–142].

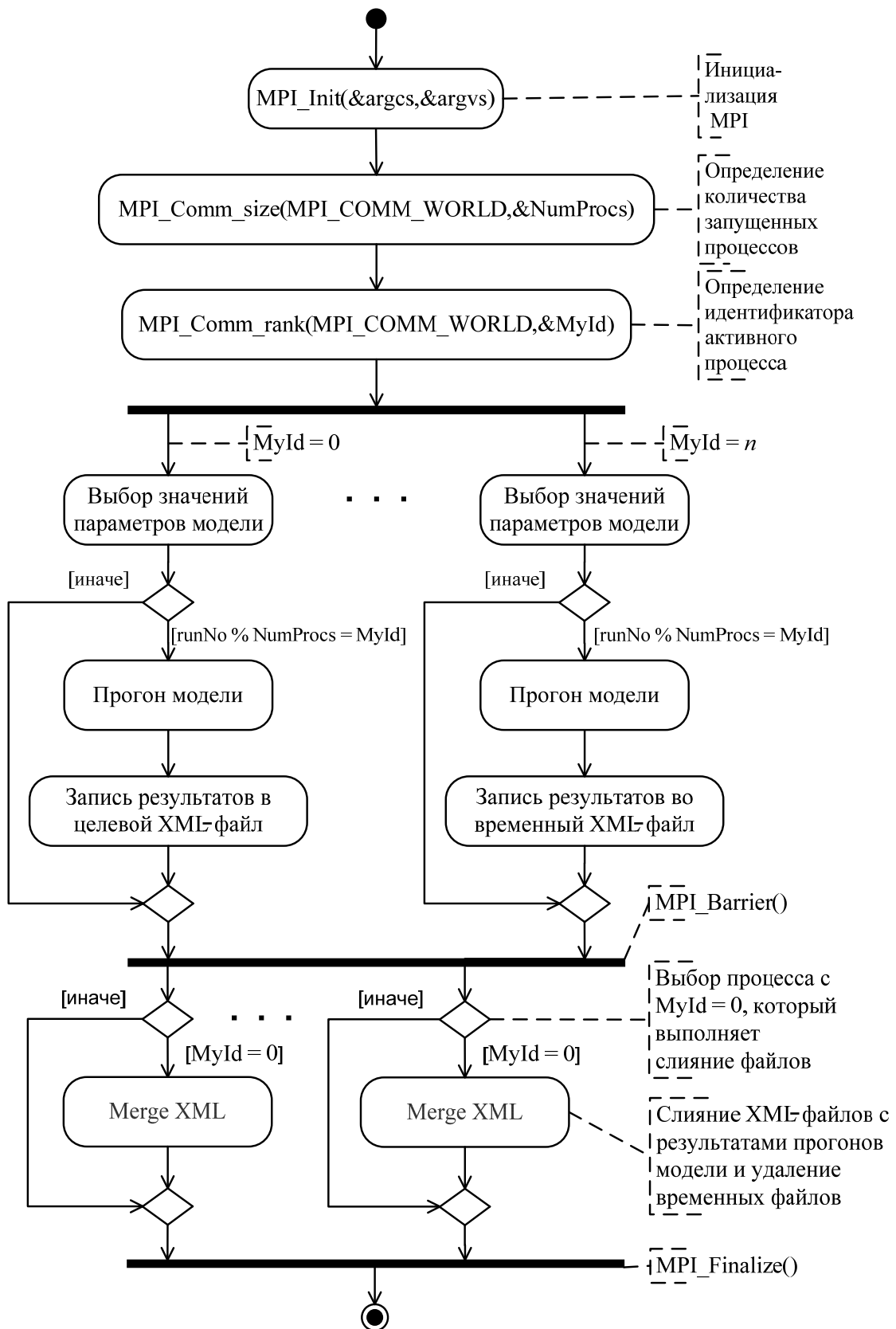


Рисунок 4.18 – UML-диаграмма работы программы *Experimenter* с распределением вычислений

```

int wmain(int argc, wchar_t *argv[])
clock_t start = clock(); // отмечают время запуска эксперимента
argvs[1] = "experimenter.exe";
int arges = 1, MyId/*id текущего процесса*/, NumProcs/*количество процессов*/;
MPI_Init(&argc,&argvs); // инициализация MPI
MPI_Comm_size(MPI_COMM_WORLD,&NumProcs); // получают количество процессов
MPI_Comm_rank(MPI_COMM_WORLD,&MyId); // получают Id текущего процесса
if(MyId == 0) // чтобы не выводить на экран несколько раз одну и ту же информацию
    cout << endl << g_szTitle << endl;
if(MyId == 0) // чтобы не выводить на экран несколько раз одну и ту же информацию
    cout << endl;
    if(MyId == 0) {
        cout << endl << L"Тип эксперимента:" << endl;
        switch (spExperimentData->getExperimentType())
        {
        case CExperimentData::AccuracyEvaluation:
            cout << L" оценка погрешности" << endl;
            break;
        case CExperimentData::TransientAndStabilityAnalysis:
            cout << L" анализ переходного процесса и устойчивости" << endl;
            break;
        case CExperimentData::SensitivityAnalysis:
            cout << L" анализ чувствительности" << endl;
            break;
        case CExperimentData::FullFactorial:
            cout << L" полный факторный эксперимент" << endl;
            break;
        case CExperimentData::FractionalFactorial:
            cout << L" дробный факторный эксперимент" << endl;
        }
    }
if(MyId == 0) // первый поток записывает свои результаты в целевой файл
    spExperimentData->save(argv[3]);
else // остальные записывают результаты во временный файл
    spExperimentData->save(argv[3] + uint2wstring(myid) + wstring(L".xml"));
cout << L" ОК" << endl;
MPI_Barrier(MPI_COMM_WORLD); // дожидаются, пока все потоки запишут свои результаты, чтобы можно было начать их собирать
if(MyId == 0) { // собирает все файлы только первый поток
    for(int i = 1; i < numprocs; i++) {
        CExperimenter::mergeXML(argv[3], argv[3] + uint2wstring(i) + wstring(L".xml"), i);
        // записывают результаты потока с номером i
        std::string s = wstring2string(argv[3] + uint2wstring(i) + wstring(L".xml"));
        remove(s.c_str()); // удаляют временный файл
    }
}
float duration = (float) (clock() - start) / CLK_TCK;
cout<<"Time - "<<<duration<<"sec"<<endl; // выводят время работы программы
void CExperimenter::mergeXML(wstring fileNameDist, wstring fileNameSrc, int madeMerge)
// копирует результаты из файла с именем fileNameSrc в файл fileNameDist
void CExperimenter::run(SP<CModel> model, SP<CModelData> modelData,
    if(runNo % NumProcs == MyId) { // запускают только те эксперименты, которые должен запустить данный процесс

```

Рисунок 4.19 – Фрагмент программы *Experimenter* с дополнениями для организации распределенных вычислений

5 Апробация метода, средств и технологии имитационного моделирования ERP-системы управления

5.1 Состав задач, решаемых при апробации, и задание исходной информации для моделирования

Для демонстрации эффективности предложенного метода имитационного моделирования ERP-системы управления промышленным предприятием, оценки работоспособности ПТКИ BelSim и технологии проектирования имитационных моделей с его использованием выбраны задачи, наиболее часто требующиеся в практике управления.

Основной задачей является исследование свойств базовой имитационной модели функционирования промышленного предприятия. При этом должны быть исследованы точностные характеристики модели: оценка погрешности имитации, длины переходного процесса в модели, устойчивости результатов моделирования, чувствительности откликов ИМ к изменениям входных параметров модели [221, с. 159–178].

Состав задач, выбранных на апробацию, можно было бы продолжить ввиду универсальности метода и программных средств для его реализации. Например, можно исследовать свойства библиотеки имитационных моделей информационной системы предприятия при стратифицированном представлении на уровне бизнес-процессов, на уровне производственных процессов и на уровне технических средств информационно-вычислительной сети предприятия. Этот список может быть значительно расширен, т. к. метод используется, например, при выполнении выигранного по тендеру хозяйственного договора на разработку динамической производственно-экономической модели завода органического синтеза, при выполнении научной работы по разработке имитационной модели информационной системы для оптимального управления производственно-торговым предприятием.

Для оценки степени доверия к базовой имитационной модели предприятия запланированы и проведены следующие исследования:

- оценка погрешности имитации базовой имитационной модели функционирования промышленного предприятия в среде BelSim с исследованием генератора псевдослучайных чисел в базовой модели;
- верификация базовой имитационной модели промышленного предприятия в среде BelSim с исследованием детерминированных характеристик модели, построением графа переходов, UML-диаграммы компонентов модели, графа переходов между активностями модели;
- исследование длины переходного процесса в базовой имитационной модели промышленного предприятия при детерминированных параметрах с изменением рабочей и внешней нагрузки модели;
- исследование длины переходного процесса в базовой имитацион-

ной модели промышленного предприятия в среде BelSim с применением визуального метода при изменении рабочей и внешней нагрузки модели;

- исследование устойчивости результатов моделирования промышленного предприятия на основе базовой имитационной модели в среде BelSim;

- исследование чувствительности откликов базовой имитационной модели промышленного предприятия к изменениям входных параметров в среде BelSim.

Планирование имитационного эксперимента позволяет определить [188, с. 249]:

- входные данные для каждого эксперимента и количество прогонов имитационной модели;

- длительность одного прогона модели и переходного процесса моделирования;

- стратегию сбора данных во время каждого прогона модели;

- методы оценивания точности выходных данных и построение доверительных интервалов;

- чувствительность модели к входным данным, разным видам распределения случайных величин, сценариям поведения моделируемой системы;

- условия и сценарии проведения эксперимента;

- условия генерирования потоков случайных чисел в системе моделирования и вероятностей входных данных;

- стратегию достижения цели эксперимента, например, сравнение альтернативных вариантов системы или оптимизация целевой функции.

Во время экспериментов используются факторные планы и строятся полиномы различного порядка, которые аппроксимируют поверхность отклика.

Один из важнейших видов эксперимента, проводимых с моделью, – *структурная оптимизация*, под которой понимают поиск наилучшей структуры моделируемой системы. В этом случае анализируется несколько моделей, которые могут различаться структурой, параметрами и алгоритмами функционирования, а для выбора используют простой перебор вариантов.

Проблемой проведения экспериментов с имитационной моделью также является определение начальных условий проведения эксперимента, которые влияют на длительность переходного процесса. Наиболее сложным моментом является установление достижения стационарного состояния системы. До сих пор не существует надежного метода определения этого момента. Однако эта проблема может быть решена с помощью диалоговых или интеллектуальных систем моделирования, которые дают возможность контролировать и графически отображать ход моделирования. Наилучший способ определения начала стационарного режима – это слежение за графиками смены переходного процесса во времени [354, с. 763–770].

Другой проблемой исследования является останов процесса моделирования. Правила останова процесса моделирования позволяют определить длительность прогона имитационной модели, а от этого зависит точность результатов [289, с. 7–13].

Динамика функционирования промышленного предприятия определяется множеством задаваемых переменных производства и управляющих параметров (таблица 5.1). Внешняя среда представлена множеством параметров, характеризующих поведение рынка. Вероятностный характер реализации запросов определяет наличие воздействия множества случайных факторов. Компоненты отклика представлены статьями бухгалтерского баланса и показателями эффективности деятельности предприятия. Их структура определена действующими положениями [84, с. 7–22; 85; 144].

Таблица 5.1 – Параметры внешней и рабочей нагрузки модели

Параметры	Значение
1	2
Количество видов продукции, ед.	1
Количество видов ресурсов, ед.	1
План-график производства: Интервал планирования, дн.	5
План-график производства: Количество интервалов планирования, ед.	20
Производство: Длительность цикла [0], дн.	30
Производство: Расход ресурсов [0;0], ед.рес./ед.прод.	1
Запасы: Готовая продукция [0], ед.	500
Запасы: Себестоимость готовой продукции [0], р./ед.	1,5
Запасы: Материальные ресурсы [0], ед.	1000
Запасы: Цена материальных ресурсов [0], р./ед.	0,75
Реализация: Интервал между отгрузками: Среднее, дн.	5
Реализация: Интервал между отгрузками: Стандартное отклонение, дн.	5
Реализация: Интервал между отгрузками: Вид функции плотности распределения (0; 1; 2)	2
Реализация: Объем отгрузки: Среднее [0], ед.	500
Реализация: Объем отгрузки: Стандартное отклонение [0], ед.	500
Реализация: Объем отгрузки: Вид функции плотности распределения [0] (0; 1; 2)	2
Реализация: Срок платежа: Среднее, дн.	15
Реализация: Срок платежа: Стандартное отклонение, дн.	15
Реализация: Срок платежа: Вид функции плотности распределения (0; 1; 2)	1
Реализация: Цены продукции [0], р./ед.	2,5
Реализация: Коэффициент изменения цен, отн.ед./мес.	1,025
Реализация: Коэффициент изменения цен: Стандартное отклонение, отн.ед.	0,01
Реализация: Коэффициент изменения цен: Вид функции плотности распределения (0; 1; 2)	0
Реализация: Интервал между изменениями цен: Среднее, дн.	30
Реализация: Интервал между изменениями цен: Стандартное отклонение, дн.	30
Реализация: Интервал между изменениями цен: Вид функции плотности распределения (0; 1; 2)	2
Снабжение: Срок платежа: Среднее, дн.	15
Снабжение: Срок платежа: Стандартное отклонение, дн.	15

Продолжение таблицы 5.1

1	2
Снабжение: Срок платежа: Вид функции плотности распределения (0; 1; 2)	1
Снабжение: Цены ресурсов [0], р./ед.	0,75
Снабжение: Коэффициент изменения цен, отн.ед./мес.	1,025
Снабжение: Коэффициент изменения цен: Стандартное отклонение, отн.ед.	0,01
Снабжение: Коэффициент изменения цен: Вид функции плотности распределения (0; 1; 2)	0
Снабжение: Интервал между изменениями цен: Среднее, дн.	30
Снабжение: Интервал между изменениями цен: Стандартное отклонение, дн.	30
Снабжение: Интервал между изменениями цен: Вид функции плотности распределения (0; 1; 2)	2
Финансы: Сумма на расчетном счету, р.	5000
Кредит: Использование для покрытия нехватки денежных средств (0; 1)	1
Кредит: Процентная ставка, отн.ед./год	0,5
Кредит: Интервал между обращениями, дн.	5
Кредит: Срок, дн.	30
Кредит: Коэффициент покрытия нехватки денежных средств, отн.ед.	1,25
Затраты: Постоянные, р./мес.	1000
Затраты: Доля заработной платы в постоянных затратах, отн.ед.	0,25
Затраты: Прочие переменные [0], р./ед.прод.	0,5
Затраты: Доля заработной платы в переменных затратах [0], отн.ед.	0,5
Затраты: Коэффициент изменения, отн.ед./мес.	1,025
Затраты: Коэффициент изменения: Стандартное отклонение, отн.ед.	0,01
Затраты: Коэффициент изменения: Вид функции плотности распределения (0; 1; 2)	0
Затраты: Интервал между изменениями: Среднее, дн.	30
Затраты: Интервал между изменениями: Стандартное отклонение, дн.	30
Затраты: Интервал между изменениями: Вид функции плотности распределения (0; 1; 2)	2
Основные средства: Восстановительная стоимость, р.	20000
Основные средства: Коэффициент износа, отн.ед.	0,5
Основные средства: Норма амортизации, отн.ед./год	0,05
Основные средства: Коэффициент переоценки, отн.ед./год	1,35
Налоги: НДС, отн.ед.	0,2
Налоги: С выручки, отн.ед.	0,045
Налоги: С прибыли, отн.ед.	0,25
Налоги: С прибыли, остающейся в распоряжении предприятия, отн.ед.	0,05
Налоги: В Фонд социальной защиты, отн.ед.	0,35
Налоги: С заработной платы, отн.ед.	0,05
Налоги: На недвижимость, отн.ед./год	0,015
Налоги: Экологический, р./ед.прод.	0,01
Налоги: На землю, р./мес.	100
Контракт: Заключение (0; 1)	1
Контракт: Дата подписания, дн.	360
Контракт: Срок действия, дн.	360
Контракт: Срок платежа, дн.	0
Контракт: Объем продукции [0], ед.	25000

Окончание таблицы 5.1

1	2
Контракт: Использование фиксированных цен (0; 1)	1
Контракт: Рентабельность продукции [0], отн.ед.	0,2
Контракт: Объем отгрузки: Среднее [0], ед.	1000
Контракт: Объем отгрузки: Стандартное отклонение [0], ед.	1000
Контракт: Объем отгрузки: Вид функции плотности распределения [0] (0; 1; 2)	2
Контракт: Интервал между отгрузками: Среднее, дн.	30
Контракт: Интервал между отгрузками: Стандартное отклонение, дн.	30
Контракт: Интервал между отгрузками: Вид функции плотности распределения (0; 1; 2)	2

5.2 Верификация имитационной модели

5.2.1 Верификация модели методом прогона с выводом промежуточных данных. Следует отметить, что, по данным зарубежных источников, **верификация** имеет дело с преобразовательной точностью, **валидация** имеет дело с поведенческой точностью или точностью представления, **аккредитация** определяется как официальное удостоверение, что модель или блок моделей приемлемы к использованию для специфической цели [281, с. 653].

Для выполнения верификации исследуются свойства ИМ, при этом прогон модели из текущего каталога выполняется по следующей команде:

```
experimenter.exe enterprise3.exe ModelData.xml ExperimentData.xml /f
```

где *experimenter.exe* – программа запуска модели;
enterprise3.exe – исполняемый файл модели;
ModelData.xml – файл данных модели;
ExperimentData.xml – файл данных эксперимента модели;
/f – опция максимальной скорости работы за счет отказа от дополнительных операций по форматированию данных.

Текст программы ИМ дополнен для вывода информации о ходе моделирования следующим фрагментом, представленным на рисунке 5.1.

Верификация имитационной модели заключается в проверке соответствия алгоритмов активностей замыслу моделирования и в анализе работы модели на тестовой нагрузке. Для этого в каждую активность добавлены операторы *WRITE* вывода промежуточных данных. Система моделирования обеспечивает запись этих данных в файл, пригодный для дальнейшей обработки. Для каждой активности выдается информация о ее наименовании, текущем значении локального времени, используемых глобальных данных модели, которые определяют содержание функционального действия, моделируемого активностью. После окончания моделирования проводится его анализ. Так как файл соответствует требованиям

спецификации *XML*, то на этом этапе возможно использование средств автоматизации для получения статистики моделирования [268, с. 458–459].

```

//вывод информации о ходе моделирования
class CProgress : public CSimulator<TEnterprise>::CActivity
{
public:
    CProgress(double totalTime) : m_totalTime(totalTime) {}
    virtual CSimulator<TEnterprise>::TNextActivity operator()(double time, SP<const
TEnterprise> in, SP<TEnterprise> out, SP<CSimulator<TEnterprise>::CProcesses>
processes)
    {
        //вывод информации о ходе моделирования
        cerr << char(0x0D) << "time: " << time << " of " << m_totalTime << ";
number of processes: " << processes->count();
        return CSimulator<TEnterprise>::TNextActivity(1, this);
    }
private:
    double m_totalTime;
};

```

Рисунок 5.1 – Общая информация о ходе моделирования

Включение оператора *WRITE* показано на примере активности «Отгрузить продукцию», представленном на рисунке 5.2. Описание данных активности «Отгрузить продукцию» представлено в структуре *TBatch* на рисунке 5.3.

Для вывода данных в *XML*-файл о партии товара в активности «Отгрузить продукцию» создана процедура *WRITEBATCH*, представленная на рисунке 5.4. В сформированном *XML*-файле информация о партии отгруженного товара представлена в виде, показанном на рисунке 5.5. Для прогона с формированием промежуточных результатов используется команда

$$Enterprise3.exe /r_ModelData.xml /io:DataVer.xml,$$

где */r* – опция прогона исполняемого файла модели *Enterprise3.exe* с файлом данных модели *ModelData.xml*;

/io: – опция записи промежуточных результатов прогона модели в файл *DataVer.xml*.

Для вывода входных и выходных параметров модели используется команда

$$Enterprise3.exe /i_InOutData.xml,$$

где */i* – опция прогона исполняемого файла модели *Enterprise3.exe* с созданием файла входных и выходных параметров модели *InOutData.xml*.

```

/** Сгенерировать отгрузку
    SP<TBatch> shipment(new TBatch);
    //срок оплаты
    shipment->paymentPeriod = (*spRndShipmentPaymentPeriod)();
    //время
    shipment->time = shipment->paymentPeriod < 0 ? time - shipment->paymentPeriod : time;
    //продукция
    shipment->items.resize(out->numberOfProductTypes);
    for(unsigned int i = 0; i < out->numberOfProductTypes; ++i)
    {
        SP<CRandom> rnd = createRandomizer(out->shipmentVolumeDistributions[i]);
        shipment->items[i].volume = floor(getNextPositiveRandomNumber(rnd));
        shipment->items[i].price = out->productPrices[i];
    }
    WRITE(L" <shipment>");
    WRITEBATCH(shipment, L"  ");
    WRITE(L" </shipment>");

```

Рисунок 5.2 – Листинг активности «Отгрузить продукцию» с операторами WRITE

```

struct TBatch
{
    struct TItem
    {
        double volume;
        double price;
        TItem(double _volume = 0.0, double _price = 0.0) : volume(_volume),
price(_price) {}
    };
    double time;
    double paymentPeriod;
    vector<TItem> items;
    double sum()
    {
        double sum = 0.0;
        vector<TItem>::const_iterator it;
        for (it = items.begin(); it != items.end(); it++) sum += it->volume * it->price;
        return sum;
    }
    double volume()
    {
        double volume = 0.0;
        vector<TItem>::const_iterator it;
        for (it = items.begin(); it != items.end(); it++) volume += it->volume;
        return volume;
    }
};

```

Рисунок 5.3 – Листинг структуры TBatch с описанием данных активности «Отгрузить продукцию»

// Вывод информации о партии товара

```
void WRITEBATCH(SP<TBatch> batch, const wchar_t* indention)
{
    WRITE(indention << "<time>" << batch->time << "</time>");
    WRITE(indention << "<paymentPeriod>" << batch->paymentPeriod <<
"</paymentPeriod>");
    WRITE(indention << "<sum>" << batch->sum() << "</sum>");
    WRITE(indention << "<items>");
    for(unsigned int i = 0; i < batch->items.size(); ++i)
    {
        WRITE(indention << " <item type=\"\" << i << "\"\"
        << " volume=\"\" << batch->items[i].volume << "\"\"
        << " price=\"\" << batch->items[i].price << "\"/>");
    }
    WRITE(indention << "</items>");
}
}
```

Рисунок 5.4 – Листинг процедуры WRITEBATCH для вывода результатов имитации отгрузки партии товара

```
<shipment>
<time>5.25052</time>
<paymentPeriod>7.42802</paymentPeriod>
<sum>2737.5</sum>
<items>
<item type="0" volume="1095" price="2.5"/>
</items>
</shipment>
```

Рисунок 5.5 – Листинг фрагмента тестового XML-файла с результатами активности «Отгрузить продукцию»

5.2.2 Верификация модели методом оценки трудоемкости алгоритма. Д. А. Садовски (D. A. Sadowski, 2004) [360, с. 61–66] отмечает, что к провалу проектирования имитационной модели приводит недооценка трех основных задач: сбора данных, проверки модели, анализа прогонов модели.

К. М. Оверстрит (C. M. Overstreet, 2002) обращает внимание на общую проблему тестирования – обнаружение непредвиденных взаимодействий между компонентами [342, с. 642]. В еще большей степени это относится и к ИМ, поскольку функционирование модели обычно состоит из последовательности выполнения компонентов, чей порядок определен разработчиком модели, программной реализацией имитационной модели и случайными числами.

До настоящего времени не существует метода для сбора и представления взаимодействия элементов в ходе имитационного моделирования. Т. Ф. Брэди (T. F. Brady, 2005) предлагает проводить интроспективный

анализ [286, с. 285–289]. При этом исследуются объекты и ресурсы модели, заданные ключевыми словами, с помощью программного анализатора частоты, который определяет распределение частоты ключевых слов и корреляцию между ними.

Для решения задачи верификации и валидации (V&V) модели предлагается метод на основе оценки трудоемкости алгоритма компонентов модели. Реализация метода представлена на примере компонента модели «Планирование производства».

Под трудоемкостью алгоритма $F_A(N)$ для данного конкретного входа N понимают количество «элементарных» операций, совершаемых алгоритмом для решения конкретной проблемы в данной формальной системе [191].

Пусть D_A – множество конкретных проблем данной задачи, заданное в формальной системе. В общем случае существует собственное подмножество D_N множества D_A , включающее все конкретные проблемы, имеющие мощность N [190, с. 61]:

$$D_N = \{D \in D_A \mid |D| = N\}.$$

Тогда данный алгоритм, решая различные задачи размерности N , будет выполнять в каком-то случае наибольшее количество операций, а в каком-то случае наименьшее количество операций. Введены следующие обозначения:

– $\hat{F}_A(N)$ – худший случай на D_N – наибольшее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N :

$$\hat{F}_A(N) = \max_{D \in D_N} (F_A(D)); \quad (5.1)$$

– $\check{F}_A(N)$ – лучший случай на D_N – наименьшее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N :

$$\check{F}_A(N) = \min_{D \in D_N} (F_A(D)); \quad (5.2)$$

– $\overline{F}_A(N)$ – средний случай на D_N – среднее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N :

$$\bar{F}_A(N) = \frac{1}{|D_N|} \sum_{D \in D_N} F_A(D). \quad (5.3)$$

Для расчета вычислительной сложности алгоритма составлены функции трудоемкости [191, с. 19]. Элементарные операции, которые выделены при составлении функций трудоемкости:

- присваивание (=);
- логические операции (&, |, !, ==);
- арифметические операции (+, -, *, /);
- индексация массивов ([]);
- доступ к элементам класса (->).

Определение трудоемкости процесса «Планирование производства» обеспечивается расстановкой по тексту программы модели счетчика элементарных операций *elcounter*. Для последующего статистического анализа создан отклик «Количество элементарных операций», параметрами которого являются *elcounter* и *time* (время вызова активности), а также «*Cnucок отгрузок*» с параметром *shipments.size()*.

Функция трудоемкости алгоритма активности *CMasterProductionScheduling* процесса «Планирование производства» для лучшего случая представлена в виде

$$F_A = 13M^2N^2 + 4M^2N + 75MN + 7M + 8CN + 42N + 9S + 42, \quad (5.4)$$

где M – количество интервалов планирования в основном плане-графике производства, $M \equiv MPSPlanningHorizon$;

N – количество видов продукции, $N \equiv numberOfResourceTypes$;

C – длительность производственного цикла, $C = productionCycle / MPSUnit$ ($productionCycle / MPSUnit$ – интервал планирования основного плана-графика производства);

S – список отгрузок [215], $S \equiv shipments[]$.

По формуле (5.4) построен график теоретической трудоемкости алгоритма модели (рисунок 5.6), показывающий рост трудоемкости алгоритма от параметра «Список отгрузок». Результаты экспериментальных исследований (рисунок 5.7) подтверждают правильность программы процесса «Планирование производства». Отклонения между экспериментальными и теоретическими результатами оценки трудоемкости алгоритма не превышают 5 %.

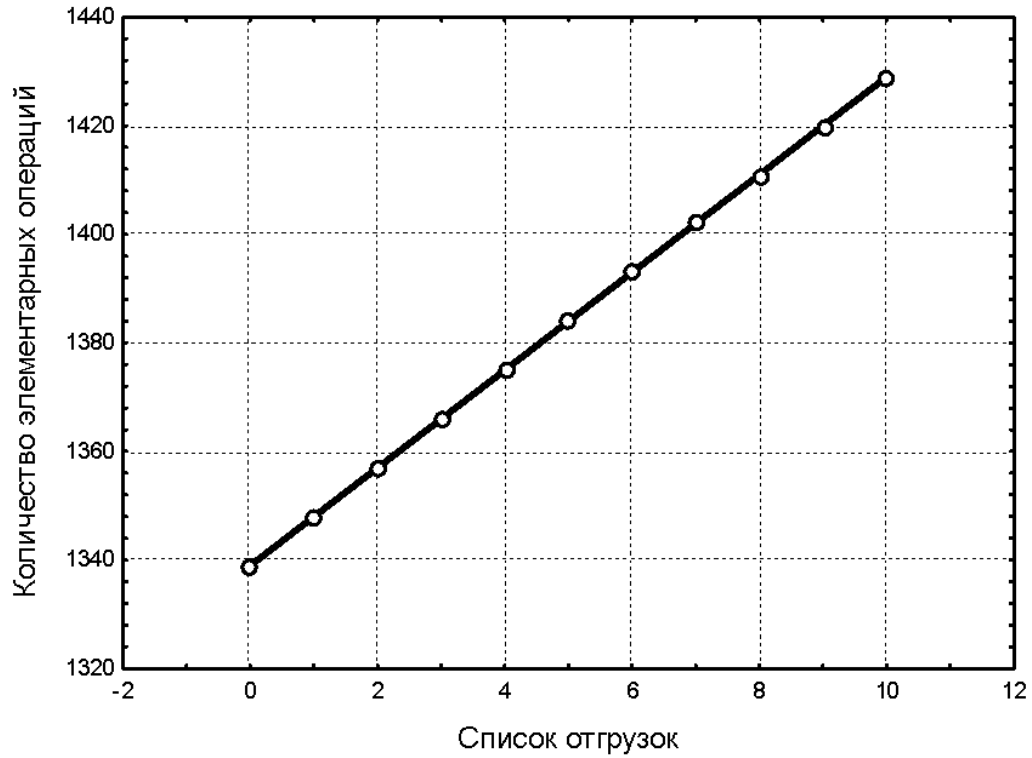


Рисунок 5.6 – Зависимость теоретической трудоемкости алгоритма от параметра «Список отгрузок» при $M = 1$, $N = 6$, $C = 2$

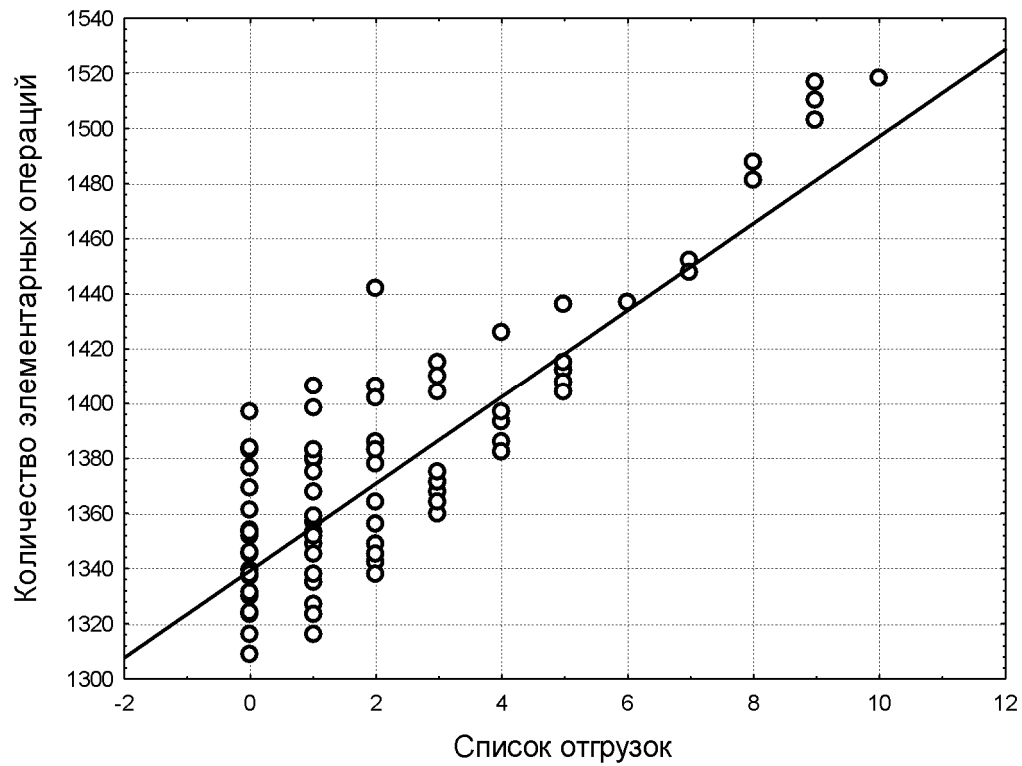


Рисунок 5.7 – Зависимость экспериментальной трудоемкости алгоритма от параметра «Список отгрузок» при $M = 1$, $N = 6$, $C = 2$

5.3 Исследование погрешности имитации

Любая система имитационного моделирования содержит несколько генераторов псевдослучайных чисел, и существует общеизвестное правило, что применение единого генератора в ходе имитации может быстро привести процесс моделирования к вырождению из-за выхода псевдослучайной последовательности за пределы отрезка апериодичности [113, с. 35; 114, с. 134–135]. В то же время некоторые авторы, например Б. В. Шмейсер (B. W. Schmeiser, 2001), именуют мифом использование отдельного генератора случайных чисел для каждого вероятностного процесса. Мифом является необходимость наличия отдельных потоков случайных чисел для обеспечения независимости случайных параметров в модели. Если генератор качественный, то независимость обеспечивается автоматически при использовании только одного потока [366, с. 42].

Для проверки качества генераторов и погрешности имитации используют следующую процедуру [113, с. 35]. В серединной точке области изменения управляющих параметров организуется несколько прогонов ($k > 10$) с одними и теми же значениями X и задаваемыми параметрами G , но с разными начальными значениями базового генератора ξ_{0k} (k – номер прогона). Для этого прогон реализуется по команде

Enterprise3.exe /r _ModelData.xml /rs:<число>DataError.xml,

где */rs:<число>* – начальное число для инициализации генератора псевдослучайных чисел.

Для каждого k -го прогона ИМ фиксируются значения n -й компоненты вектора откликов Y_{nk} . В результате получают выборки значений откликов $\{Y_{nk}\}$, $k = 1, \dots, 10$. По этим выборкам определяются оценки математического ожидания и дисперсии, рассчитывается доверительный интервал нахождения истинного значения математического ожидания n -й компоненты вектора отклика Y_{mi} . При этом допускают нормальность распределения отклонения Y_{nk} от Y_{mi} . Поскольку объемы выборки малы ($k < 30$), то для нахождения доверительных интервалов используется t -статистика [37, с. 23].

Параметры файла данных модели *ModelData.xml* для исследования погрешности имитации заданы следующими значениями:

- период времени равен 12 мес.;
- интервал сбора статистики равен 1 мес.;
- начальное значение датчика случайных чисел равно 1.

Параметры внешней и рабочей нагрузки модели даны в таблице 5.2.

В качестве параметра, изменяемого в ходе эксперимента, выбран «Реализация: Объем отгрузки» со средним значением, равным 500, стандартным отклонением, равным 100. Вид функции плотности распределения задан опцией, равной 0 (равномерный).

Таблица 5.2 – Параметры внешней и рабочей нагрузки модели при исследовании погрешности моделирования

Параметры	Значение
1	2
Количество видов продукции, ед.	1
Количество видов ресурсов, ед.	1
План-график производства: Интервал планирования, дн.	5
План-график производства: Количество интервалов планирования, ед.	52
Производство: Длительность цикла [0], дн.	30
Производство: Расход ресурсов [0; 0], ед.рес./ед.прод.	1
Запасы: Готовая продукция [0], ед.	500
Запасы: Себестоимость готовой продукции [0], р./ед.	1,5
Запасы: Материальные ресурсы [0], ед.	1000
Запасы: Цена материальных ресурсов [0], р./ед.	0,75
Реализация: Интервал между отгрузками: Среднее, дн.	5
Реализация: Интервал между отгрузками: Стандартное отклонение, дн.	0
Реализация: Интервал между отгрузками: Вид функции плотности распределения (0; 1; 2)	0
Реализация: Объем отгрузки: Среднее [0], ед.	500
Реализация: Объем отгрузки: Стандартное отклонение [0], ед.	100
Реализация: Объем отгрузки: Вид функции плотности распределения [0] (0; 1; 2)	0
Реализация: Срок платежа: Среднее, дн.	15
Реализация: Срок платежа: Стандартное отклонение, дн.	0
Реализация: Срок платежа: Вид функции плотности распределения (0; 1; 2)	0
Реализация: Цены продукции [0], р./ед.	2,5
Реализация: Коэффициент изменения цен, отн.ед./мес.	1,025
Реализация: Коэффициент изменения цен: Стандартное отклонение, отн.ед.	0
Реализация: Коэффициент изменения цен: Вид функции плотности распределения (0; 1; 2)	0
Реализация: Интервал между изменениями цен: Среднее, дн.	30
Реализация: Интервал между изменениями цен: Стандартное отклонение, дн.	0
Реализация: Интервал между изменениями цен: Вид функции плотности распределения (0; 1; 2)	0
Снабжение: Срок платежа: Среднее, дн.	15
Снабжение: Срок платежа: Стандартное отклонение, дн.	0
Снабжение: Срок платежа: Вид функции плотности распределения (0; 1; 2)	0
Снабжение: Цены ресурсов [0], р./ед.	0,75
Снабжение: Коэффициент изменения цен, отн.ед./мес.	1,025
Снабжение: Коэффициент изменения цен: Стандартное отклонение, отн.ед.	0
Снабжение: Коэффициент изменения цен: Вид функции плотности распределения (0; 1; 2)	0
Снабжение: Интервал между изменениями цен: Среднее, дн.	30
Снабжение: Интервал между изменениями цен: Стандартное отклонение, дн.	0
Снабжение: Интервал между изменениями цен: Вид функции плотности распределения (0; 1; 2)	0
Финансы: Сумма на расчетном счете, р.	5000
Кредит: Использование для покрытия нехватки денежных средств (0; 1)	1
Кредит: Процентная ставка, отн.ед./год	0,5
Кредит: Интервал между обращениями, дн.	5

Окончание таблицы 5.2

1	2
Кредит: Срок, дн.	30
Кредит: Коэффициент покрытия нехватки денежных средств, отн.ед.	1,25
Затраты: Постоянные, р./мес.	1000
Затраты: Доля заработной платы в постоянных затратах, отн.ед.	0,25
Затраты: Прочие переменные [0], р./ед.прод.	0,5
Затраты: Доля заработной платы в переменных затратах [0], отн.ед.	0,5
Затраты: Коэффициент изменения, отн.ед./мес.	1,025
Затраты: Коэффициент изменения: Стандартное отклонение, отн.ед.	0
Затраты: Коэффициент изменения: Вид функции плотности распределения (0; 1; 2)	0
Затраты: Интервал между изменениями: Среднее, дн.	30
Затраты: Интервал между изменениями: Стандартное отклонение, дн.	0
Затраты: Интервал между изменениями: Вид функции плотности распределения (0; 1; 2)	0
Основные средства: Восстановительная стоимость, р.	20000
Основные средства: Коэффициент износа, отн.ед.	0,5
Основные средства: Норма амортизации, отн.ед./год	0,05
Основные средства: Коэффициент переоценки, отн.ед./год	1,35
Налоги: НДС, отн.ед.	0,2
Налоги: С выручки, отн.ед.	0,045
Налоги: С прибыли, отн.ед.	0,25
Налоги: С прибыли, остающейся в распоряжении предприятия, отн.ед.	0,05
Налоги: В Фонд социальной защиты, отн.ед.	0,35
Налоги: С заработной платы, отн.ед.	0,05
Налоги: На недвижимость, отн.ед./год	0,015
Налоги: Экологический, р./ед.прод.	0,01
Налоги: На землю, р./мес.	100
Контракт: Заключение (0; 1)	1
Контракт: Дата подписания, дн.	360
Контракт: Срок действия, дн.	360
Контракт: Срок платежа, дн.	30
Контракт: Объем продукции [0], ед.	25000
Контракт: Использование фиксированных цен (0; 1)	1
Контракт: Рентабельность продукции [0], отн.ед.	0,2
Контракт: Объем отгрузки: Среднее [0], ед.	1000
Контракт: Объем отгрузки: Стандартное отклонение [0], ед.	0
Контракт: Объем отгрузки: Вид функции плотности распределения [0] (0;1;2)	0
Контракт: Интервал между отгрузками: Среднее, дн.	30
Контракт: Интервал между отгрузками: Стандартное отклонение, дн.	0
Контракт: Интервал между отгрузками: Вид функции плотности распределения (0; 1; 2)	0

В качестве отклика выбраны показатели, зависящие от изменяемого параметра:

- Деловая активность: Выручка от реализации (Y1);
- Деловая активность: Чистая прибыль (Y2);

– Рентабельность продукции (Y3).

В имитационной модели применяется генератор случайных чисел (ГСЧ), построенный с использованием стандартных функций языка C++ `int rand(void)`, `void srand(unsigned seed)` библиотеки `<stdlib.h>`. Иницилирующее число (ИЧ) ГСЧ выбрано в качестве изменяемого входного параметра с 21 уровнем изменения. Значение нулевого уровня принято равным единице. Значения последующих уровней вычислены по формуле

$$M_n = 1 + n \cdot 1638,$$

где M_n – значение иницилирующего числа ГСЧ;

n – номер уровня;

1638 – результат целочисленного деления максимально допустимого значения иницилирующего числа на количество уровней (без учета нулевого), т. е. $32768 / 20$.

Для получения опорных значений откликов модель запускается с нулевыми значениями стандартных отклонений входных параметров с вероятностными характеристиками. При этом получены следующие значения:

$$Y1_0 = 81037 \text{ р.}; \quad Y2_0 = 7335 \text{ р.}; \quad Y3_0 = 0,1271 \text{ отн. ед.}$$

Обработка результатов исследования модели с ненулевым стандартным отклонением входных параметров с вероятностными характеристиками проведена с использованием приложения статистического анализа STATISTICA 6.0 (рисунки 5.8–5.10). Между значением иницилирующего числа ГСЧ и величиной откликов выявлена слабая корреляционная зависимость. Результаты корреляционного анализа приведены на рисунках 5.11–5.13.

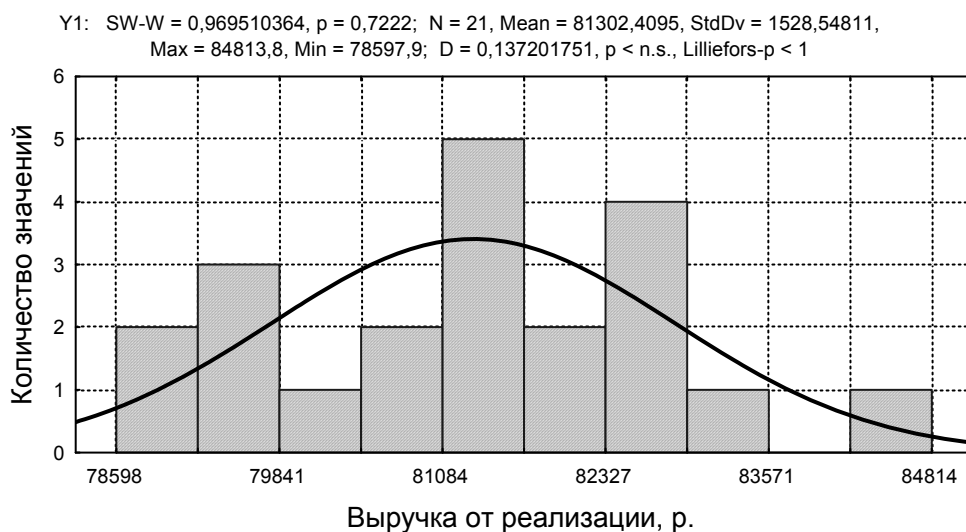


Рисунок 5.8 – Распределение отклика Y1

Y2: SW-W = 0,96062496, p = 0,5288; N = 21, Mean = 7432,99, StdDv = 308,550394,
Max = 8169,82, Min = 6930,31; D = 0,11822664, p < n.s., Lilliefors-p < 1



Рисунок 5.9 – Распределение отклика Y2

Y3: SW-W = 0,955363886, p = 0,4281; N = 21, Mean = 0,12826781, StdDv = 0,00292645992,
Max = 0,1352, Min = 0,123758; D = 0,134662113, p < n.s., Lilliefors-p < 1

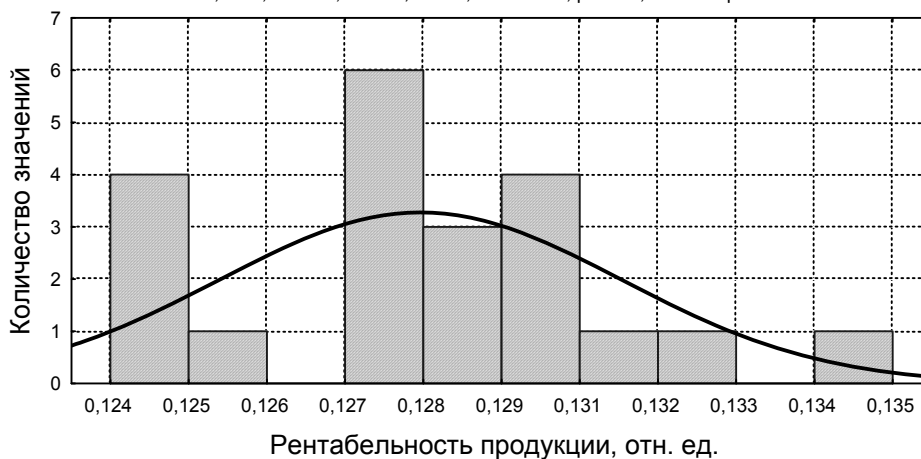


Рисунок 5.10 – Распределение отклика Y3

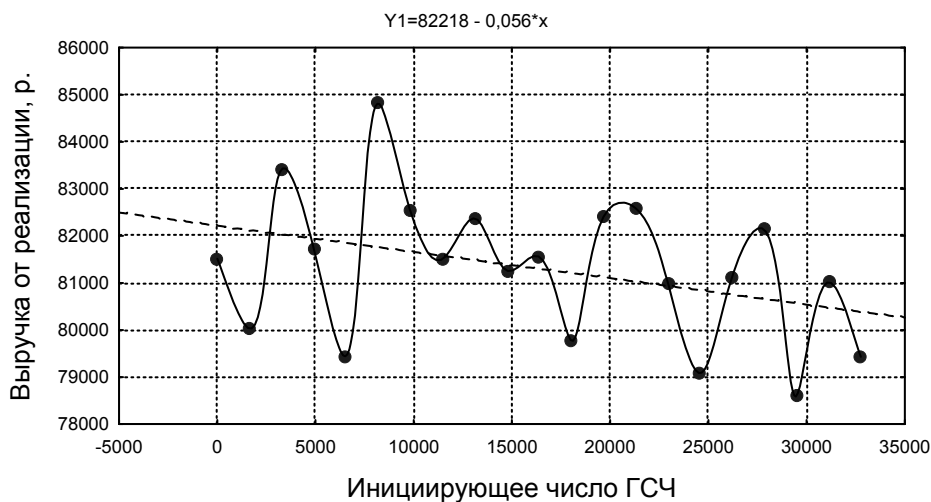


Рисунок 5.11 – Диаграмма корреляционной зависимости Y1 от иницирующего числа ГСЧ

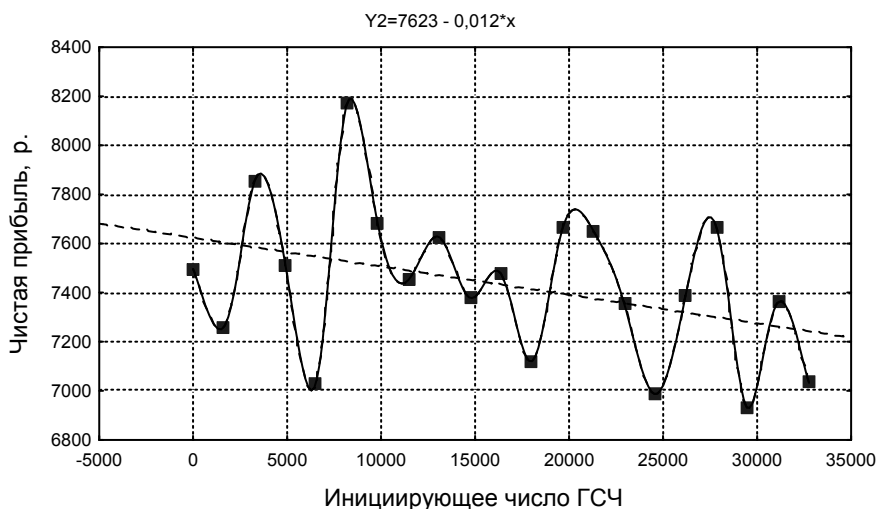


Рисунок 5.12 – Диаграмма корреляционной зависимости Y_2 от иницирующего числа ГСЧ

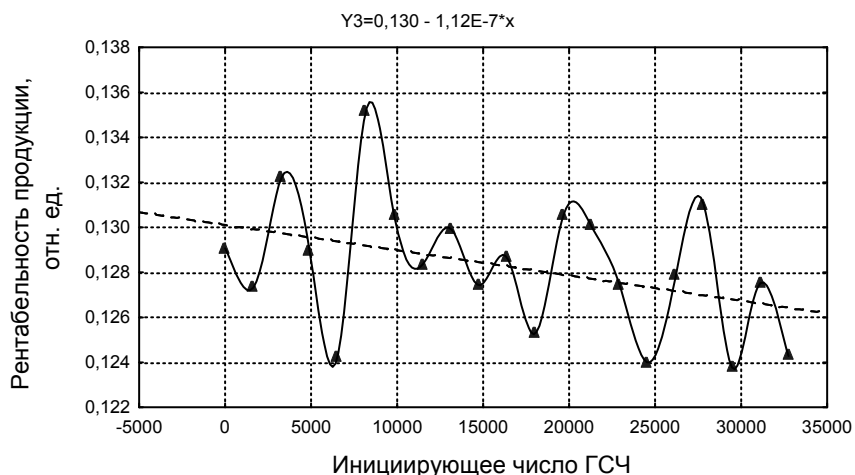


Рисунок 5.13 – Диаграмма корреляционной зависимости Y_3 от иницирующего числа ГСЧ

Сравнительный анализ значений откликов Y_1 , Y_2 , Y_3 позволяет сделать вывод о наличии погрешности при использовании разных иницирующих чисел ГСЧ. Эта погрешность определяется по формуле

$$\varepsilon_Y = \frac{|\bar{Y} - Y_0|}{Y_0} 100 \%,$$

где ε_Y – погрешность среднего значения отклика;

\bar{Y} – среднее значение отклика;

Y_0 – истинное (опорное) значение отклика.

Тогда

$$\varepsilon_{Y1} = \frac{|81302,41 - 81037,0|}{81037,0} 100 \% = 0,33 \%;$$

$$\varepsilon_{Y2} = \frac{|7432,99 - 7335,62|}{7335,62} 100 \% = 1,33 \%;$$

$$\varepsilon_{Y3} = \frac{|0,12827 - 0,12705|}{0,12705} 100 \% = 0,96 \%.$$

Задавшись величиной допустимого отклонения (смещения) для значений откликов можно судить о приемлемости использования ГСЧ.

Экономическим исследованиям вполне удовлетворяет отклонение 5 % (или уровень значимости 0,05). Полученные значения погрешностей свидетельствуют о допустимости использования имеющегося в нашем распоряжении ГСЧ для имитации случайного изменения входных параметров модели производственно-экономической деятельности промышленного предприятия.

Аналогично проведены исследования погрешности модели в точках, отличных от середины, при разных значениях реализации: среднее значение, стандартное отклонение, вид функции плотности распределения интервала между отгрузками (равномерная, нормальная, экспоненциальная), коэффициент изменения цен. Оценка погрешности имитации показывает, что отклики имеют различную относительную погрешность [221, с. 159].

В значительной мере результаты использования BelSim зависят от четкого понимания соответствия процессов экономической деятельности предприятия алгоритмам, лежащим в основе построения имитационной модели. Как показали результаты исследования, весьма важным для принятия окончательного решения руководителем является правильное обоснование выбора закона распределения входных параметров в модели. Формирование входных параметров модели как аналогов непредсказуемых процессов реальности должно осуществляться с учетом условий, обуславливающих эту непредсказуемость (дефициты, спрос, сезонность, перебои в поставках сырья и т. п.). Неверный выбор функции плотности распределения изменения даже одного из нескольких входных параметров может повлечь принятие ошибочного решения руководителем и, как следствие, цепь экономических промахов в деятельности предприятия на длительный период.

5.4 Исследование длительности переходного процесса в модели

5.4.1 Технология оценки длительности переходного процесса имитационной модели с трендом отклика в стационарном режиме. Исследованию методов определения периода разгона (warm-up period) и длины прогона (run-length) модели уделяется до настоящего времени значительное внимание, делается попытка автоматизации определения указанных параметров моделирования [354, с. 763–770; 372, с. 781–789].

С. Робинсон (S. Robinson, 2002) отмечает 26 подходов к определению переходного процесса модели, распределяя их по группам: графические методы, эвристические подходы, статистические методы, тестирование со смещением параметров инициализации, гибридные методы [353, с. 440].

Поведение откликов имитационной модели (ИМ) функционирования промышленного предприятия характеризуется наличием тренда в стационарном режиме. Например, стоимость запасов на складе постоянно увеличивается с учетом индекса инфляции.

В практике исследования ИМ, как правило, наблюдают отклики, которые в стационарном режиме приходят к установившемуся значению [105, с. 207]. Использование общепринятой методики [114, с. 137] приводит к погрешностям определения длительности переходного процесса в имитационных моделях с трендом в стационарном режиме [221, с. 170].

Значение отклика в стационарном режиме с учетом тренда на интервале $[T_i, T_{i+1}]$:

$$Y_{ict} = Y_{ict}(T_i) + c_i t \mid \forall t \in [T_i, T_{i+1}] \& \exists t = T_i \ c_i t = 0.$$

Для определения переходного процесса (рисунок 5.14) принимают во внимание значение тренда $Y_{ict}(T_i)$ в момент T_i , значение тренда $Y_{ict}(T_{i+1})$ – в момент T_{i+1} . Оценивают максимальное $Y_{imax}(T_i)$ и минимальное $Y_{imin}(T_{i+1})$ отклонения отклика относительно тренда. Если отклонения отклика Y_i не превышают допустимых значений на величину до 5 %, то середину интервала $[T_i, T_{i+1}]$ принимают за оценку длительности T_{nn} переходного процесса ИМ.

Методика определения длительности переходного процесса ИМ реализуется следующей последовательностью шагов исследования [250, с. 116–121].

Шаг 1. Для локального максимума отклонения отклика $Y_{imax}(T_i)$ фиксируют момент времени T_i и находят отклонение $\Delta Y_i(T_i)$ от значения тренда $Y_{ict}(T_i)$:

$$Y_{imax}(T_i), Y_{ict}(T_i) \Rightarrow \Delta Y_i(T_i) = Y_{imax}(T_i) - Y_{ict}(T_i).$$

Шаг 2. Для соседнего локального минимума отклонения отклика $Y_{imin}(T_{i+1})$ фиксируют момент времени T_{i+1} и находят отклонение $\Delta Y_i(T_{i+1})$ от

значения тренда $Y_{ict}(T_{i+1})$:

$$Y_{imin}(T_{i+1}), Y_{ict}(T_i + 1) \Rightarrow \Delta Y_i(T_{i+1}) = Y_{ict}(T_{i+1}) - Y_{imin}(T_i + 1).$$

Шаг 3. Находят отклонение отклика ΔY_{iTr} относительно тренда:

$$\Delta Y_{iTr} = (\Delta Y_i(T_i) + \Delta Y_i(T_{i+1})) \cdot \cos \alpha,$$

где α – угол наклона линии тренда в стационарном режиме относительно оси абсцисс.

Шаг 4. Шаги 1–3 выполняют, пока не будет выполнено условие характеристического предиката

$$P\left(\frac{\Delta Y_{iTr}}{\Delta Y_{icm}(T_i)} \cdot 100 \% \leq \varepsilon, \%\right) = 1,$$

где $\varepsilon, \%$ – допустимая погрешность отклонения отклика от линии тренда в стационарном режиме.

Шаг 5. Принимают за оценку длительности T_{nn} переходного процесса ИМ середину интервала $[T_i, T_{i+1}]$:

$$T_{nn} = \frac{T_i + T_{i+1}}{2}.$$

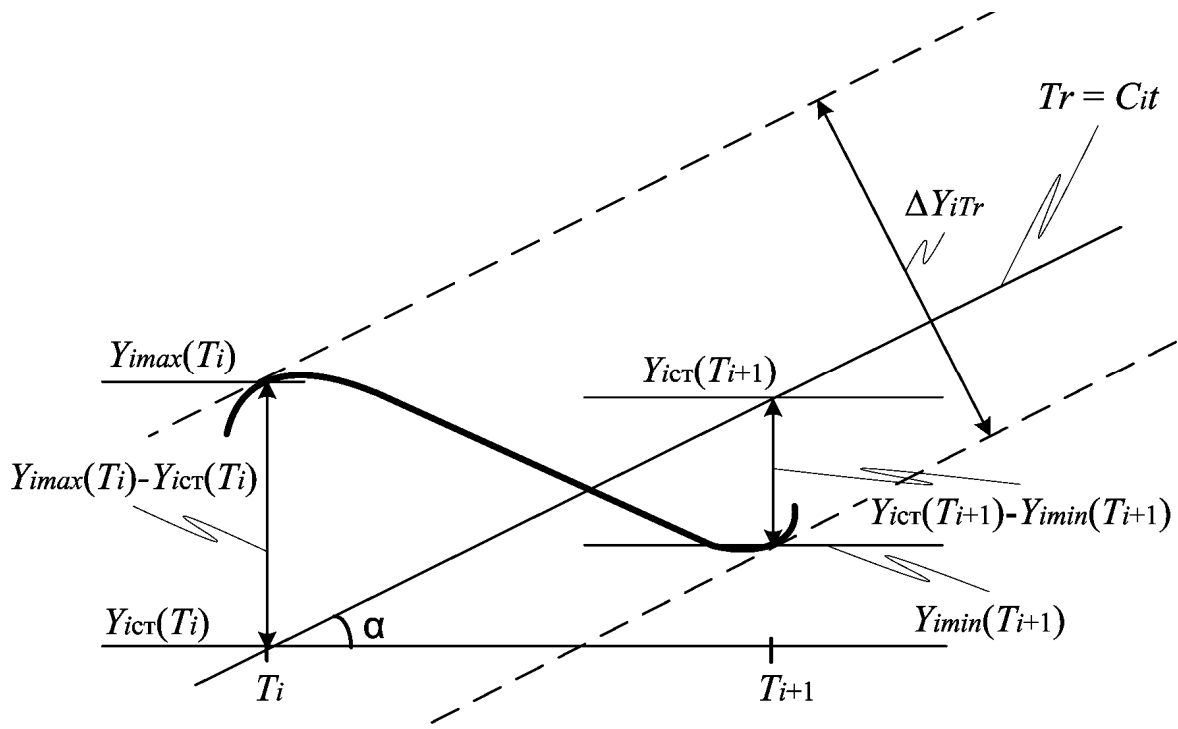


Рисунок 5.14 – Графическая модель оценки длительности переходного процесса с учетом тренда в стационарном режиме

Полученное значение T_{mn} учитывают при оценке устойчивости ИМ [254, с. 45].

Для описания динамических рядов и выделения тренда целесообразно использовать уравнение, предложенное В. В. Нешитым [127, с. 22–23]:

$$y = y_0(1 + \alpha ut)^{\frac{1}{u}} = (A + Bt)^{\frac{1}{u}}, \quad (5.5)$$

где $A = y_0^u$; $B = \alpha u y_0^u$.

Для нахождения оценок параметров уравнения (5.5) его приводят к виду

$$y^u = A + Bt. \quad (5.6)$$

Обозначив y^u как Y , записывают (5.6) в виде

$$Y = A + BT$$

и оценки параметров A и B находят по методу наименьших квадратов:

$$B = \frac{n \sum_i T_i Y_i - \sum_i T_i \sum_i Y_i}{n \sum_i T_i^2 - \left(\sum_i T_i \right)^2};$$

$$A = \frac{1}{n} \left(\sum_i Y_i - B \sum_i T_i \right).$$

Значения параметра u задаются с некоторым шагом $\Delta u = 0,1$ или $\Delta u = 0,001$, или некоторым другим в зависимости от требуемой точности аппроксимации. При каждом заданном значении параметра u и вычисленных оценках параметров A , B необходимо вычислять коэффициент корреляции и индекс корреляции. При достижении наибольшего значения коэффициента корреляции (или индекса корреляции) вычислительный процесс останавливают.

5.4.2 Исследование длительности переходного процесса в модели с детерминированными параметрами. Особенность исследуемой базовой ИМ – наличие разнородных по интенсивности переходных процессов: краткосрочных контрактов [226, с. 139–141], переоценки основных средств [237, с. 38–42] и т. д.

Исследование переходного процесса проведено по известным техно-

логиям [114, с. 137–139] и технологии с учетом тренда [250, с. 116–121]. При этом использованы средства мониторинга BelSim с применением пакета STATISTICA, позволяющие в процессе имитации осуществлять сбор статистики по основным откликам ИМ (таблица 5.3) и строить диаграммы по собранной статистике. Диаграммы построены по всем откликам, и по каждому графику определен момент наступления стационарного режима, максимальный из которых принят за момент окончания переходного режима (рисунки 5.15–5.17).

Значения параметров модели для проведения эксперимента: период времени равен 72 мес.; интервал сбора статистики равен 3 мес. Особенность проведенных исследований заключается в том, что параметры внешней и рабочей нагрузки (см. таблицу 5.1) с вероятностными характеристиками приняты детерминированными. Модель имеет переходный режим длительностью 12 мес. (модельного времени) [221, с. 170].

Таблица 5.3 – Обозначения откликов базовой имитационной модели

Обозначение	Наименование
1	2
Y1	Баланс 110: Основные средства
Y2	Баланс 120: Нематериальные активы
Y3	Баланс 130: Доходные вложения в материальные ценности
Y4	Баланс 140: Вложения во внеоборотные активы
Y5	Баланс 150: Прочие внеоборотные активы
Y6	Баланс 190: ИТОГО ВНЕОБОРОТНЫЕ АКТИВЫ
Y7	Баланс 210: Запасы и затраты
Y8	Баланс 211: сырье, материалы и другие ценности
Y9	Баланс 212: животные на выращивании и откорме
Y10	Баланс 213: незавершенное производство (издержки обращения)
Y11	Баланс 214: прочие запасы и затраты
Y12	Баланс 220: Налоги по приобретенным ценностям
Y13	Баланс 230: Готовая продукция и товары
Y14	Баланс 240: Товары отгруженные, выполненные работы, оказанные услуги
Y15	Баланс 250: Дебиторская задолженность
Y16	Баланс 251: расчеты с покупателями и заказчиками
Y17	Баланс 252: расчеты с учредителями по вкладам в уставный фонд
Y18	Баланс 253: расчеты с разными дебиторами и кредиторами
Y19	Баланс 253: прочая дебиторская задолженность
Y20	Баланс 260: Финансовые вложения
Y21	Баланс 270: Денежные средства
Y22	Баланс 280: Прочие оборотные активы
Y23	Баланс 290: ИТОГО ОБОРОТНЫЕ АКТИВЫ
Y24	Баланс 390: Баланс АКТИВОВ
Y25	Баланс 510: Уставный фонд (капитал)
Y26	Баланс 515: Собственные акции (доли), выкупленные у акционеров
Y27	Баланс 520: Резервный фонд
Y28	Баланс 530: Добавочный фонд

Продолжение таблицы 5.3

1	2
Y29	Баланс 540: Нераспределенная прибыль
Y30	Баланс 550: Непокрытый убыток
Y31	Баланс 560: Целевое финансирование
Y32	Баланс 590: ИТОГО ИСТОЧНИКИ СОБСТВЕННЫХ СРЕДСТВ
Y33	Баланс 610: Резервы предстоящих расходов
Y34	Баланс 620: Расходы будущих периодов
Y35	Баланс 630: Доходы будущих периодов
Y36	Баланс 640: Прибыль отчетного года
Y37	Баланс 650: Убыток отчетного года
Y38	Баланс 660: Прочие доходы и расходы
Y39	Баланс 690: ИТОГО РАСХОДЫ И ДОХОДЫ
Y40	Баланс 710: Краткосрочные кредиты и займы
Y41	Баланс 720: Долгосрочные кредиты и займы
Y42	Баланс 730: Кредиторская задолженность
Y43	Баланс 731: расчеты с поставщиками и подрядчиками
Y44	Баланс 732: расчеты по оплате труда
Y45	Баланс 733: расчеты по прочим операциям с персоналом
Y46	Баланс 734: расчеты по налогам и сборам
Y47	Баланс 735: расчеты по социальному страхованию и обеспечению
Y48	Баланс 736: расчеты с акционерами (учредителями) по выплате доходов
Y49	Баланс 737: расчеты с разными дебиторами и кредиторами
Y50	Баланс 740: Прочие виды обязательств
Y51	Баланс 790: ИТОГО РАСЧЕТЫ
Y52	Баланс 890: Баланс ПАССИВОВ
Y53	Отчет о прибылях и убытках 030: Выручка (за минусом налогов)
Y54	Отчет о прибылях и убытках 040: Себестоимость реализованных товаров
Y55	Отчет о прибылях и убытках 050: Управленческие расходы
Y56	Отчет о прибылях и убытках 060: Расходы на реализацию
Y57	Отчет о прибылях и убытках 070: Прибыль (убыток) от реализации
Y58	Отчет о прибылях и убытках (210, 220, 230): Налоги на прибыль и иные обязательные платежи
Y59	Отчет о прибылях и убытках 240: ПРИБЫЛЬ (УБЫТОК) К РАСПРЕДЕЛЕНИЮ
Y60	Коэффициент текущей ликвидности К1
Y61	Коэффициент обеспеченности собственными оборотными средствами К2
Y62	Коэффициент обеспеченности финансовых обязательств активами К3
Y63	Коэффициент собственности КС
Y64	Коэффициент абсолютной ликвидности КАЛ
Y65	Коэффициент критической (быстрой) ликвидности КБЛ
Y66	Темп роста продаж (нетто) ТРП
Y67	Темп роста себестоимости ТРС
Y68	Темп роста прибыли от реализации продукции ТРПР
Y69	Темп роста чистой прибыли ТРЧП
Y70	Темп роста стоимости активов ТРСА
Y71	Рентабельность продукции по затратам РПЗ
Y72	Рентабельность выручки от продаж РВП
Y73	Рентабельность активов предприятия РА

Окончание таблицы 5.3

1	2
Y74	Рентабельность основного капитала ROK
Y75	Рентабельность собственного капитала RCK
Y76	Коэффициент фондоотдачи КФО
Y77	Период окупаемости собственного капитала ПОСК
Y78	Коэффициент оборачиваемости оборотных активов КООА
Y79	Продолжительность оборота оборотных активов ПООА
Y80	Продолжительность оборота производственных запасов ПОПЗ
Y81	Продолжительность оборота незавершенного производства ПОНП
Y82	Продолжительность оборота готовой продукции ПОГП
Y83	Продолжительность оборота дебиторской задолженности ПОДЗ

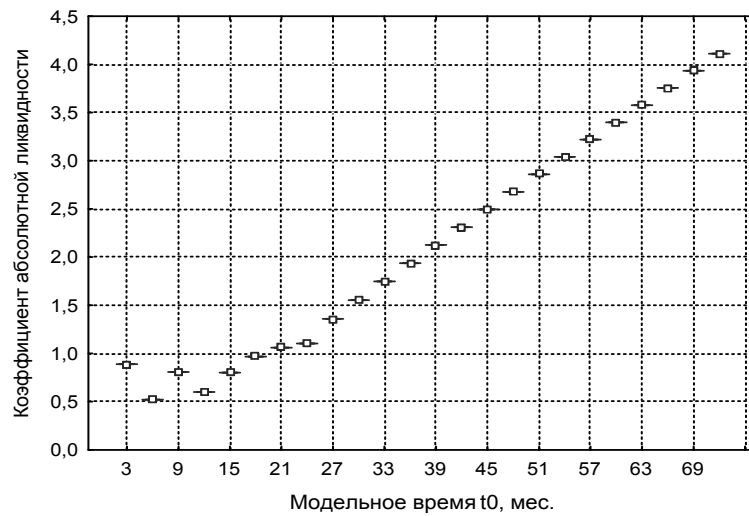


Рисунок 5.15 – Переходный процесс отклика Y64 при детерминированных исходных данных

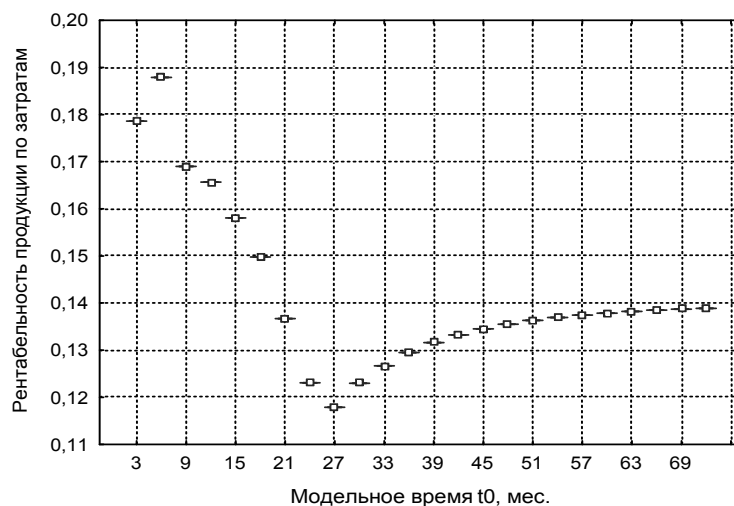


Рисунок 5.16 – Переходный процесс отклика Y71 при детерминированных исходных данных

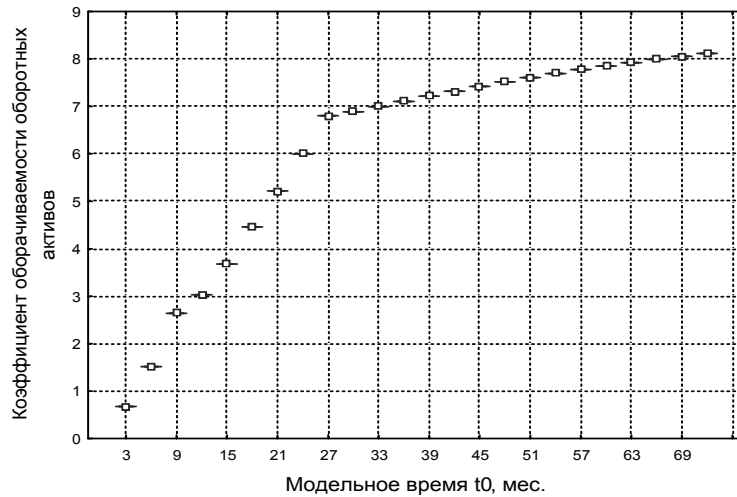


Рисунок 5.17 – Переходный процесс отклика Y_{78} при детерминированных исходных данных

5.5 Исследование чувствительности имитационной модели

Для оценки чувствительности ИМ необходимо установить диапазон изменения отклика модели Y_j в зависимости от изменения каждой компоненты вектора параметров X_i . В соответствии с диапазоном изменения откликов Y_j определяется стратегия планирования экспериментов на ИМ. Если при значительной амплитуде изменения некоторой компоненты вектора параметров X_i отклик Y_j меняется незначительно, то это означает, что точность представления этой компоненты в ИМ не играет существенной роли. Кроме того, в планировании имитационных экспериментов эта компонента не будет использоваться как основная. Если же отклик Y_j окажется высокочувствительным к изменению некоторой компоненты вектора X_i , то это служит прямым указанием на необходимость представления её в модели с максимально возможной точностью [113, с. 38–40].

Вторым не менее важным моментом исследования чувствительности модели является проверка зависимости отклика модели Y_j от изменений параметров внешней среды G_k . Меняя характеристики G_k в обе стороны на некоторую величину ΔG_k , оценивают диапазон вектора откликов модели δY_j . Если δY_j незначителен, то требования к точности задания модели внешней среды могут быть несущественными. В противном случае изменение характеристик G_k^* , по которым строится модель внешней среды, и сами способы стабилизации значения G_k в модели должны быть высокоточными. Анализ поведения приращения отклика модели δY_j при колебаниях X_i и G_k иногда позволяет скорректировать алгоритмы ИМ в сторону их упрощения.

Исследование чувствительности имитационной модели осуществляется средствами ПТКИ следующим образом [221, с. 170–178]. Для каждой входной переменной модели (параметра модели и внешней среды) указы-

ваются три значения: в центральной точке $X_i^{(0)}$ и с отклонениями от нее в большую $X_i^{(+1)}$ и меньшую $X_i^{(-1)}$ сторону на выбранную длину интервала изменения (уровни 0, +1 и -1 соответственно). Далее проводится серия из $2n$ экспериментов, где n – количество входных параметров. Для каждого i -го параметра проводится пара экспериментов для уровней +1 ($X_i^{(+1)}$) и -1 ($X_i^{(-1)}$) соответственно, при этом значения остальных параметров устанавливаются на уровне 0 ($X_i^{(0)}$) и вычисляются соответствующие значения вектора откликов модели $Y_j^{i(+1)}$ и $Y_j^{i(-1)}$. Приращение i -го параметра рассчитывается по формуле

$$\Delta X_i = \frac{|X_i^{(+1)} - X_i^{(-1)}|}{X_i^{(0)}}, \quad (5.7)$$

где $X_i^{(0)}$ определяют по формуле

$$X_i^{(0)} = \frac{|X_i^{(+1)}| + |X_i^{(-1)}|}{2}. \quad (5.8)$$

Аналогично рассчитываются приращения каждого j -го элемента вектора откликов:

$$\Delta Y_j^i = \frac{|Y_j^{i(+1)} - Y_j^{i(-1)}|}{Y_j^{i(0)}}, \quad (5.9)$$

где $Y_j^{i(0)}$ определяют по формуле

$$Y_j^{i(0)} = \frac{|Y_j^{i(+1)}| + |Y_j^{i(-1)}|}{2}. \quad (5.10)$$

Чувствительность имитационной модели к изменению i -го параметра определяется соотношением

$$\delta X_i = \frac{\max_j \Delta Y_j^i}{\Delta X_i}. \quad (5.11)$$

Обозначения откликов имитационной модели приведены в таблице 5.3. С целью повышения достоверности результатов число параллельных

опытов увеличено до 500. При этом в расчетах принимаются средние значения откликов. Проверка гипотезы о значимости изменения средних значений откликов осуществляется по t -критерию с уровнем доверительной вероятности 0,01. Результаты расчетов представлены ранжированными по значению чувствительности вектора откликов (рисунок 5.18) [221, с. 176].

5.6 Проверка адекватности базовой имитационной модели по тесту Тьюринга

Проверка адекватности имитационной модели подтверждается, если установлено, что ее выходные данные идентичны выходным данным, которых можно ожидать от реальной системы. В том случае, когда два набора данных оказываются идентичными, модель существующей системы считается адекватной. Абсолютно точного метода определения адекватности модели исследуемой системы нет [90, с. 333].

Для сравнения модельных и системных выходных данных можно использовать тест Тьюринга [90, с. 334]. Данный тест заключается в том, что специалистам, хорошо знакомым с системой, предоставляется возможность изучить одну или несколько групп системных данных, а также одну или несколько групп модельных данных без указания, какие из них к чему относятся. При этом все группы данных должны быть представлены в одном формате. Если специалисты смогут отличить системные данные от модельных и объяснить, как они это сделали, их разъяснения помогут найти пути усовершенствования модели.

Для подтверждения доверия к модели проведены исследования на ряде промышленных предприятий с демонстрацией возможностей базовой имитационной модели предприятия.

Отмечена важность создания модели производственно-экономической системы, которая подтверждается возможностью «подсказать» какие-нибудь усовершенствования для исследуемой системы, и необходимость модели для более точного сравнения существующей и предлагаемой систем, поскольку существующая система предоставляет ограниченную информацию [90, с. 334].

Подтверждением доверия к модели являются заключенные договоры о научно-производственном сотрудничестве, выигранный тендер на разработку динамической производственно-экономической модели завода органического синтеза [150].

6 Программные средства для реализации метода поиска рациональных решений

6.1 Система проведения оптимизации «Оптимизатор»

В решении задачи оптимизации задействованы модули модели, алгоритма оптимизации, целевой функции. Их взаимодействие обеспечивает служебный модуль *Optimizer*. Схема потоков данных между указанными модулями дана на рисунке 6.1 [267, с. 197–198].

Соответствие модулей, содержимого потоков математической модели следующее. Модули: Модель – $A_M \in M$; Алгоритм оптимизации – A_{AO} ; Целевая функция – F_{ZO} ; *Optimizer* – Φ_{ZO} . Потоки: Оптимизируемые параметры – X_{AO} , Def_{AO} ; Значения постоянных параметров – $\overline{x_G} \in X_G$; Погрешность решения – ε_{AO} ; Значения параметров решения – $\overline{sx_M^*} \in SX_M^*$; Значения параметров модели – $\overline{sx_M} \in SX_M$; Значения откликов модели – $\overline{sy_M} \in SY_M$; Данные о модели – $\langle X_M, Y_M, Def_M \rangle \in M$; Значения параметров – $\overline{sx_{AO}} \in SX_{AO}$; Значение целевой функции – $F_{AO}(\overline{sx_{AO}}) = F_{ZO}(\overline{sx_{ZO}})$; Параметры алгоритма – P_{AO} ; Значения параметров целевой функции – $\overline{sx_{ZO}} \in SX_{ZO}$; Вид целевой функции – $F_{ZO} \in CF_{ZO}$; Параметры целевой функции – $X_{ZO} \in CF_{ZO}$.

Модулям (см. рисунок 6.1) соответствуют отдельные классы, формирующие структуру системы (рисунок 6.2). Класс *CCallBackFunction* представляет собой интерфейс вызова целевой функции задачи алгоритмом оптимизации.

Соответствие методов классов математической модели приведено в таблице 6.1 [267, с. 198–199].

При рассмотрении реализации алгоритма случайного поиска – генетического алгоритма (ГА), используемого для решения задач оптимизации в ПТКИ *BelSim*, выделены следующие основные абстракции сущностей: *Популяция*, *Индивидуум*, *Генотип*, которые можно выделить в отдельные классы *CPopulation*, *CIndividual* и *CGenotype* соответственно. Так как размер генотипа является величиной постоянной для любого индивидуума и, по сути, определяющей эту абстракцию, то класс *CGenotype* будет параметризованным с размером генотипа в качестве параметра [221, с. 96].

Описание класса *CPopulation*, представляющего абстракцию *Популяция*, приведено в таблице 6.2. Экземпляр этого класса имеет динамический цикл жизни, который отражен на диаграмме состояний и переходов на рисунке 6.3.

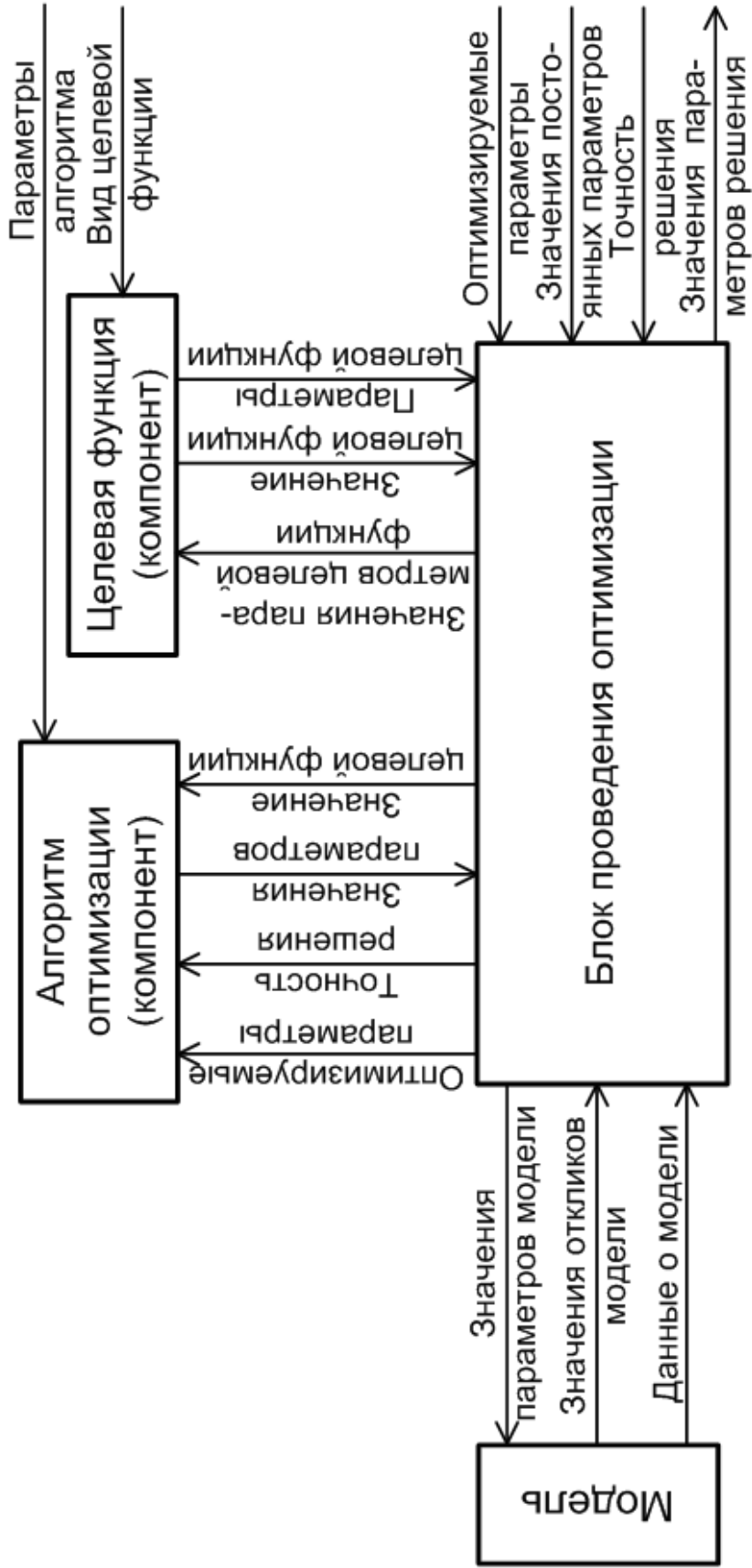


Рисунок 6.1 – Схема потоков данных в системе «Оптимизатор»

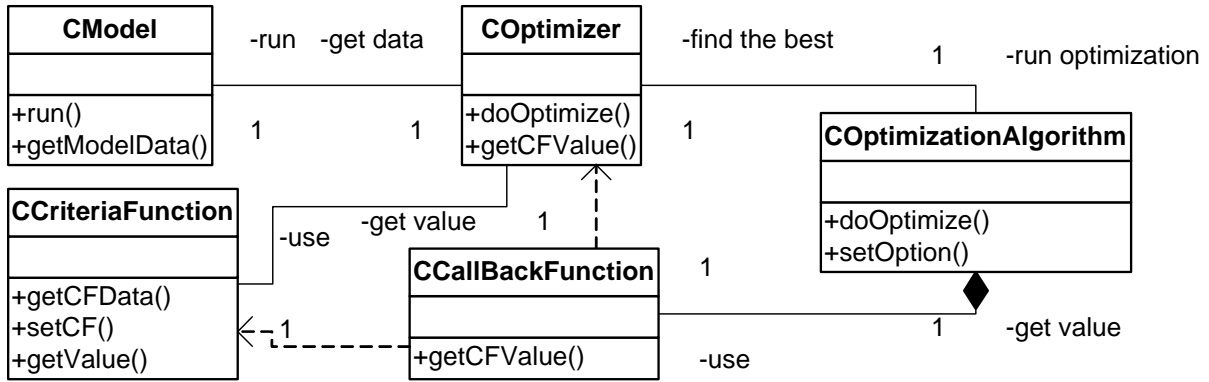


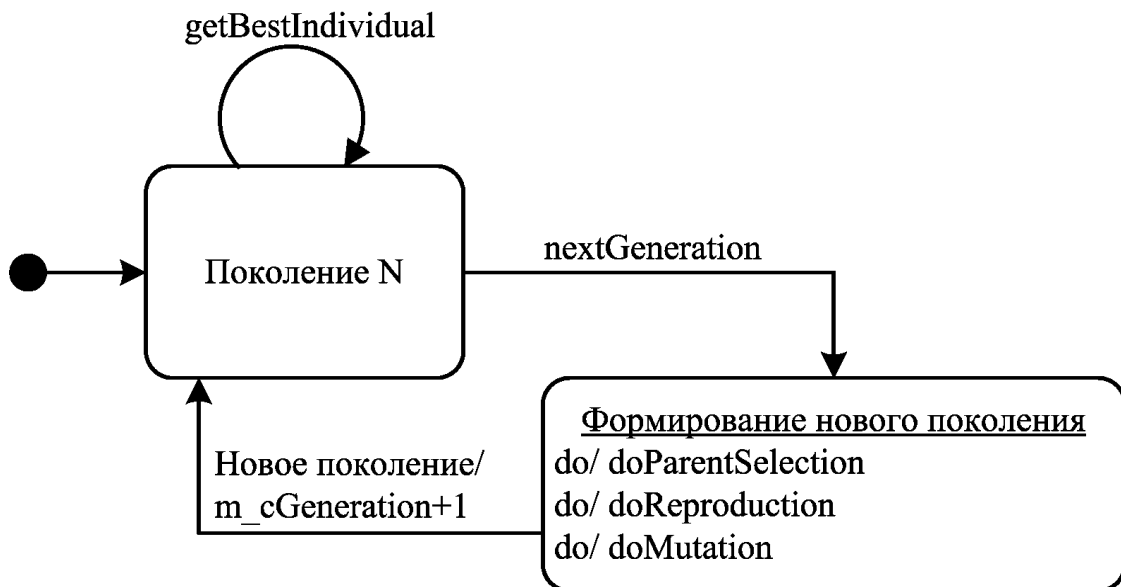
Рисунок 6.2 – Диаграмма классов системы «Оптимизатор»

Таблица 6.1 – Соответствие методов классов системы «Оптимизатор»

Метод	Параметры	Возвращаемое значение	Назначение
<i>CModel</i> (Модель)			
<i>getModelData</i>	–	$\langle X_M, Y_M, Def_M \rangle \in M$	Вывод информации о модели
<i>run</i>	$\overline{sx}_M \in SX_M$	$\overline{sy}_M \in SY_M$	Запуск модели
<i>CCriteriaFunction</i> (Целевая функция)			
<i>getCFData</i>	–	$X_{ZO} \in CF_{ZO}$	Вывод информации о параметрах целевой функции
<i>setCF</i>	$F_{ZO} \in CF_{ZO}$	–	Формирование целевой функции
<i>getValue</i>	$\overline{sx}_{ZO} \in SX_{ZO}$	$F_{ZO}(\overline{sx}_{ZO})$	Вычисление значения целевой функции
<i>COptimizer</i> (Блок проведения оптимизации)			
<i>doOptimize</i>	ZO	$\overline{sx}_M^* \in SX_M^*$	Запуск задачи оптимизации
<i>getCFValue</i>	$\overline{sx}_{AO} \in SX_{AO}$	$F_{AO}(\overline{sx}_{AO}) = F_{ZO}(\overline{sx}_{ZO})$	Обращение к модели для получения значений параметров целевой функции, ее вызов и передача рассчитанного значения функции алгоритму оптимизации
<i>COptimizationAlgorithm</i> (Алгоритм оптимизации)			
<i>setOption</i>	$p_{AO} \in P_{AO}$	–	Установка параметра алгоритма
<i>doOptimize</i>	$X_{AO}, Def_{AO}, \epsilon_{AO}, F_{AO} \in CF_{AO}$	$\overline{sx}_{AO}^* \in SX_{AO}^*$	Запуск алгоритма
<i>CCallbackFunction</i> (Интерфейс вызова целевой функции задачи алгоритмом оптимизации)			
<i>getCFValue</i>	$\overline{sx}_{AO} \in SX_{AO}$	$F_{AO}(\overline{sx}_{AO})$	Вызывает метод <i>Optimizer::getCFValue</i> и возвращает его значение

Таблица 6.2 – Представление абстракции *Популяция*

Наименование	<i>CPopulation</i>	
Ответственность	Формирование очередного поколения	
Операция	<i>nextGeneration</i>	Формирование нового поколения
	<i>doParentSelection</i>	Отбор родителей
	<i>doReproduction</i>	Организация скрещивания родителей
	<i>doMutation</i>	Организация мутации полученной популяции
	<i>getBestIndividual</i>	Определение лучшего индивидуума
	<i>setCrossoverProbability</i>	Установка вероятности кроссовера пары
	<i>setMutationProbability</i>	Установка вероятности мутации индивидуума
	<i>operator[]</i>	Получение некоторого индивидуума
	<i>getGenerationNumber</i>	Возвращение номера поколения
Атрибут	<i>m_size</i>	Размер популяции
	<i>m_vIndividuals</i>	Набор индивидуумов
	<i>m_cGeneration</i>	Номер поколения
	<i>m_crossoverProbability</i>	Вероятность кроссовера
	<i>m_mutationProbability</i>	Вероятность мутации индивидуума

Рисунок 6.3 – Жизненный цикл класса *CPopulation*

При инициализации экземпляра класса происходит создание начальной популяции и безусловный переход в состояние существования поколения. В данном состоянии можно получить информацию о лучшем индивидууме (операция *getBestIndividual*) и изменить параметры алгоритма, касающиеся воспроизведения (операции *setCrossoverProbability* и *setMutationProbability*). Передача сообщения *nextGeneration* о формировании нового поколения переводит его в соответствующее состояние, внутри которого происходит выполнение внутренних операций объекта (*doParentSelection*, *doReproduction*,

doMutation). После окончания операций объект возвращается в состояние жизни, но уже нового поколения [221, с. 96].

Описание класса *CIndividual*, представляющего абстракцию *Индивидуум*, приведено в таблице 6.3. Динамический цикл жизни экземпляра этого класса отражен на диаграмме состояний и переходов (рисунок 6.4). Первым обязательным состоянием всякого объекта является *Рождение*, из которого после завершения одной из операций (*make* или *makeClone*) осуществляется безусловный переход в состояние существования. Сообщение *getQuality*, не изменяя состояния объекта, возвращает приспособляемость.

При истинности условия свершения скрещивания (*Crossover::doReproduction(aParents)[rand() < m_crossoverProbability]*) операция *doCrossover* переводит объект в состояние скрещивания. В нем формируется потомок путем последовательного перехода для каждого выделенного фрагмента при выполнении условия ($\text{rand}() < 0.5$) в состояние рекомбинации. После этого идет возврат в прежнее состояние, но с новым содержанием. Схожим является состояние мутации (переход – операция *doMutation*), где идет генная мутация (условие $\text{rand}() < m_mutationProbability$).

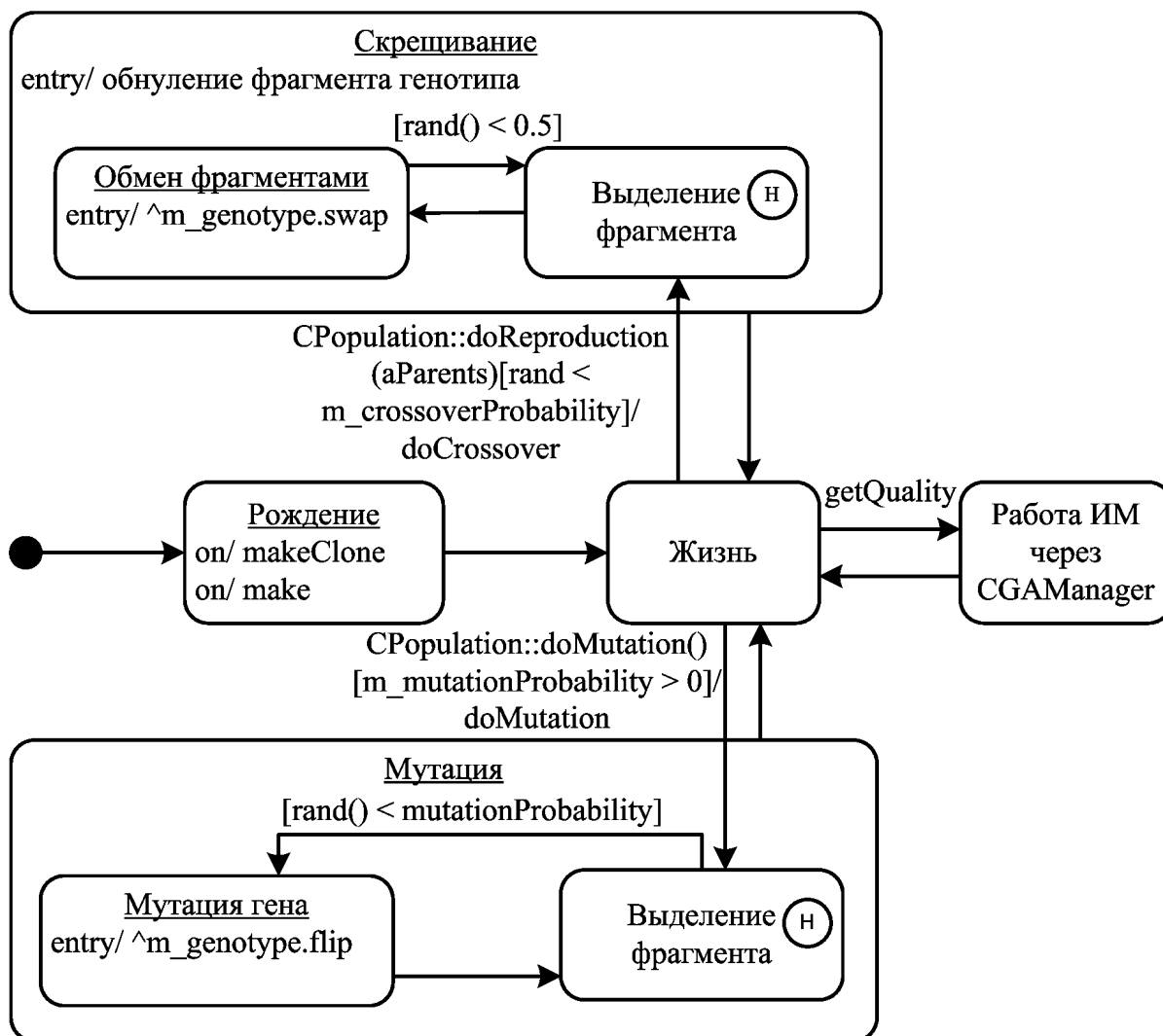
Представление абстракции *Генотип* приведено в таблице 6.4 [221, с. 99].

При рассмотрении информационной модели системы отдельно выделен интерфейс, обеспечивающий взаимодействие генетического алгоритма с самим приложением. Данный интерфейс, представляющий собой абстракцию виртуальной машины *CGAManager*, является внешним объектом как для алгоритма (по сути, блок оптимизации), так и для имитационной модели (экономическая часть задачи). Он занимает промежуточное положение и должен рассматриваться в обеих частях. Поэтому данная абстракция (таблица 6.5) включена в рассматриваемую объектную модель.

Динамический цикл жизни экземпляра этого класса *CGAManager* отражен на диаграмме состояний и переходов (рисунок 6.5). При появлении объект класса переходит в холостое состояние. Операции определения параметров целевой функции (*addParameter*, *delParameter*) и связи с ней (*setQualityFunction*) наполняют содержимое объекта [221, с. 100–101].

Таблица 6.3 – Представление абстракции *Индивидуум*

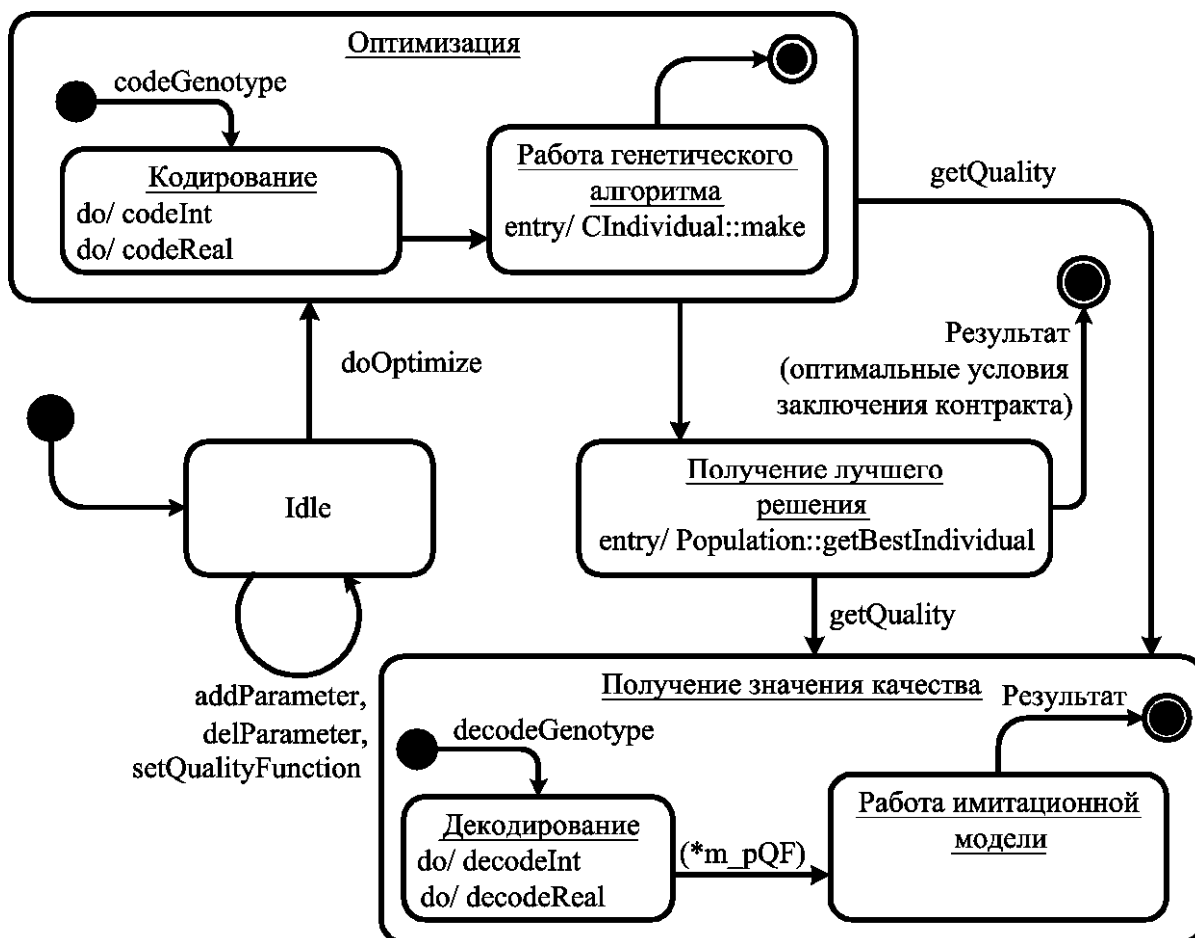
Наименование	<i>CIndividual</i>	
Ответственность	Отражение механизма существования носителя генетического кода	
Операция	<i>make</i>	Создание индивидуума с определенной структурой генотипа
	<i>makeClone</i>	Создание себе подобного индивидуума
	<i>doMutation</i> <i>doCrossover</i>	Мутация генотипа Скрещивание с другим индивидуумом
	<i>getQuality</i>	Получение значения приспособленности
Атрибут	<i>m_genotype</i>	Генотип индивидуума
	<i>m_pGAManager</i>	Указатель на интерфейс взаимодействия с «внешней средой» алгоритма

Рисунок 6.4 – Жизненный цикл класса `CIndividual`Таблица 6.4 – Представление абстракции *Генотип*

Наименование	<i>CGenotype</i>	
Ответственность	Хранение информации о генетическом коде	
Операция	<i>clear</i>	Очистка кода
	<i>assign</i>	Слияние с другим кодом в один
	<i>increase</i>	Увеличение кода на некоторое количество генов
	<i>swap</i>	Рекомбинация с другим кодом
	<i>operator[]</i>	Получение значения некоторого гена
Атрибут	<i>m_genome</i>	Генетический код

Таблица 6.5 – Представление абстракции виртуальной машины оптимизации

Наименование	<i>CGAManager</i>	
Ответственность	Обеспечение взаимодействия с генетическим алгоритмом	
Операция	<i>addParameter</i>	Добавление управляемого параметра
	<i>delParameter</i>	Исключение параметра
	<i>setQualityFunction</i>	Установка связи с целевой функцией
	<i>doOptimize</i>	Запуск генетического алгоритма для поиска оптимума
	<i>getQuality</i>	Выдача значения целевой функции по генотипу
	<i>makeIndividual</i>	Создание индивидуума с определенной структурой генотипа и функцией качества
	<i>codeGenotype</i>	Создание генотипа по набору параметров
	<i>decodeGenotype</i>	Получение значений параметров по генотипу
	<i>codeInt</i>	Кодирование целочисленного параметра в генотип
	<i>decodeInt</i>	Декодирование целочисленного параметра из генотипа
	<i>codeReal</i>	Кодирование вещественного параметра в генотип
	<i>decodeReal</i>	Декодирование вещественного параметра из генотипа
Атрибут	<i>m_vParameters</i>	Набор управляемых параметров – тип, диапазон изменения, точность
	<i>m_pQF</i>	Указатель на целевую функцию

Рисунок 6.5 – Жизненный цикл класса *CGAManager*

Например, переход (операция *doOptimize*) в основное состояние оптимизации возможен лишь при наличии этих данных ($m_vParameters.size() > 0$, $m_pQF \neq NULL$). В этом состоянии первым внутренним сообщением *codeGenotype* объект переводится в субсостояние кодировки, где по мере необходимости используются операции *codeInt* и *codeReal*.

Далее идет работа генетического алгоритма – состояние ожидания для объекта, откуда периодически идет обратный вызов сообщением *getQuality*, переводящий объект в состояние декодирования. По завершении работы генетического алгоритма объект переходит в состояние формирования оптимального решения в виде, понятном внешней системе. После получения лучшего индивидуума (*CPopulation::getBestIndividual*) декодируется генотип (*getQuality*) и, вернув результат, возвращается в исходное состояние. Особенность состояния получения значения целевой функции в том, что после декодирования объект переходит в состояние ожидания, вызвав модель имитации указателем *m_pQF* (см. таблицу 6.5).

Семантику абстракций, а также отношения между ними можно представить на диаграмме классов, ответственных за процесс оптимизации (рисунок 6.6) [221, с. 101–102].

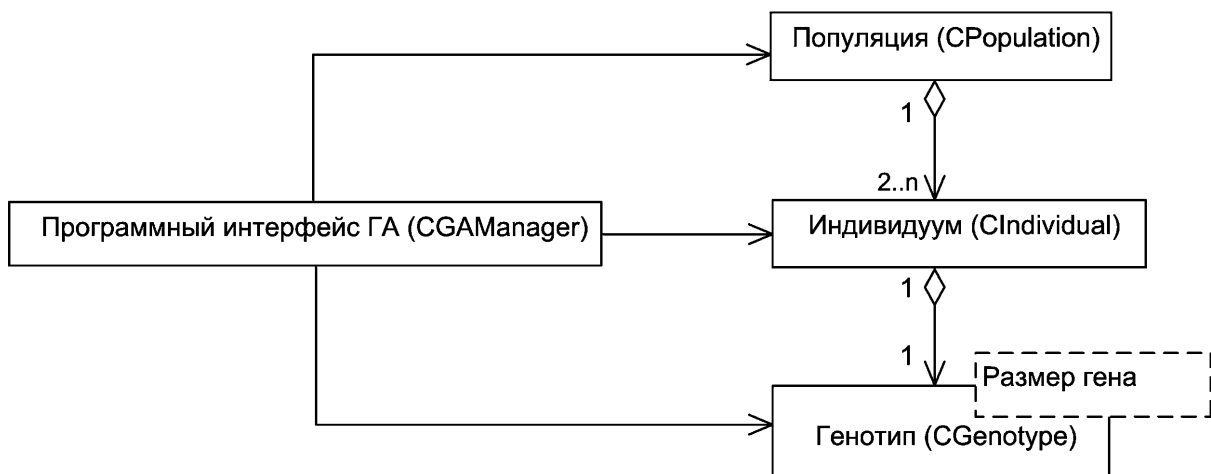


Рисунок 6.6 – Диаграмма классов, ответственных за процесс оптимизации

После определения основных абстракций строится сценарий работы системы. Работа системы заключается в получении оптимального решения, следовательно, основным сценарием является проведение оптимизации. Некий внешний объект (в общем случае экземпляр метакласса) в начале сценария путем посылки сообщений *addParameter*, *delParameter*, *setQualityFunction* объекту класса *CGAManager* создает прообраз интерфейса целевой функции, вызов которой осуществляет сам внешний объект.

Далее посылкой сообщения *doOptimize* тому же объекту инициируется процедура поиска оптимума. Объект класса *CGAManager* кодирует управляющие параметры в генотип (внутренняя операция *codeGenotype*), создает прототип объекта класса *CIndividual::make*, после чего запускает

процесс эволюции сообщением *nextGeneration* объекту класса *CPopulation*, на что данный объект, отобрав родителей (внутренняя операция *doParentSelection*), инициирует работу объекта класса *CIndividual* сообщениями *doCrossover*, *doMutation*.

CIndividual по мере необходимости обращается к объекту *CGenotype*, используя различные сообщения *swap*, *clear*, *assign* и др. В ходе всего этого процесса периодически объект *CPopulation* запрашивает приспособляемость объекта *CIndividual* сообщением *getQuality*.

В ответ идет одноименное сообщение интерфейсу взаимодействия *CGAManager*, который, декодируя генотип (внутренняя операция *decodeGenotype*), побуждает (функция обратного вызова (**m_pQF*)) внешний объект запустить имитационную модель и дать результат при данном наборе значений параметров. По окончании моделирования эволюции объект *CGAManager* запрашивает лучшего индивидуума (*getBestIndividual*) и, декодируя его значение, возвращает внешнему объекту оптимальное решение.

Внешний объект взаимодействует с генетическим алгоритмом только через специальный интерфейс, т. е. он не имеет никаких сведений о структуре алгоритма и наоборот (алгоритм о внешнем объекте). Имеет место контрактная модель взаимодействия клиент-сервер. Клиент, соблюдая правила интерфейса, выдает запрос на оптимизацию. Через интерфейс данные преобразуются в формат сервера (генетический алгоритм), и формируется запрос в понятной серверу форме. Сервер, приняв запрос, обрабатывает его, получает решение и, соблюдая правила интерфейса, возвращает его. Через интерфейс данные в требуемой форме выдаются клиенту [221, с. 102–103].

6.2 Имитационное моделирование контрактов в информационной системе предприятия

6.2.1 Основные типы контрактов на промышленном предприятии. Достаточно сложно обеспечить успешную работу предприятий в условиях рынка, представленных неопределенностью оценки его состояния и риском при реализации принятых решений. При этом одной из основных составляющих успеха предприятий является эффективная маркетинговая деятельность, в частности, правильное заключение контрактов.

Для исследования имитационных моделей контрактов выбраны следующие типы контрактов: контракт продажи; контракт купли; контракт на дилерское обслуживание; контракт по товарному обмену.

Приобретение материально-технических и энергетических ресурсов, реализация продукции осуществляются посредством заключения контрактов. Под контрактом будем понимать документ, являющийся юридической основой деятельности сторон – участников соглашения, регламентирую-

щей их права и обязанности, включая оплату в денежной форме в определенной согласованной сторонами валюте с применением оговоренных в контракте способа платежа и формы расчета (деньги, бартер, взаимозачет, вексель, налоговое освобождение, чек/аккредитив и др.), сроки, даты и периодичность отгрузок продукции и поступлений денежных средств.

В результате обобщения различных вариантов контрактов выделены основные виды (рисунки 6.7–6.12): на поставку сырья с предварительной оплатой; на поставку сырья с отсрочкой платежа; на реализацию продукции с предварительной оплатой; на реализацию продукции с отсрочкой платежа; с товарообменной операцией. Возможны также смешанные варианты, представляющие собой всевозможные комбинации указанных схем [221, с. 26–27].

С целью обобщения можно расширить понятие контракта, рассматривая его в более широком смысле, как определенный набор условий получения ресурсов и реализации продукции. Получение/выдачу кредитов и займов, использование трудовых ресурсов также можно рассматривать и учитывать через контракты. Так, выплату заработной платы можно представить как оплату за использованные трудовые ресурсы, которые по принципам использования аналогичны энергетическим, так как их нельзя накапливать. Кредит можно представить как особый вид контракта, по которому предприятию поставляются денежные средства, за что оно также расплачивается деньгами.



Рисунок 6.7 – Контракт на поставку сырья с предварительной оплатой



Рисунок 6.8 – Контракт на поставку сырья с отсрочкой платежа

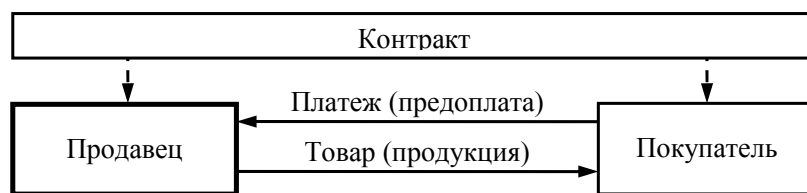


Рисунок 6.9 – Контракт на реализацию продукции с предоплатой

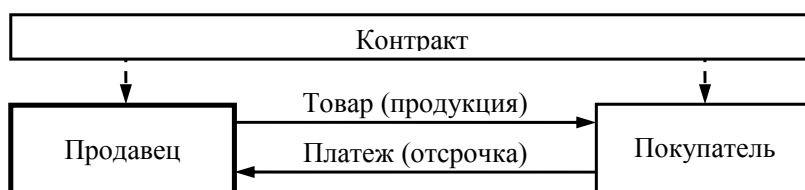


Рисунок 6.10 – Контракт на реализацию продукции с отсрочкой платежа

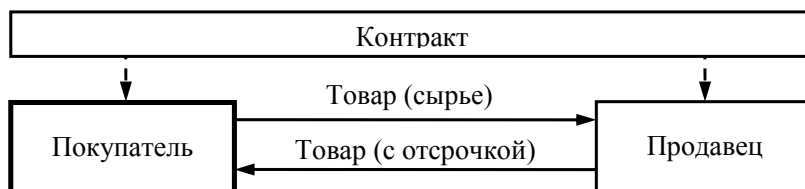


Рисунок 6.11 – Контракт по товарному обмену

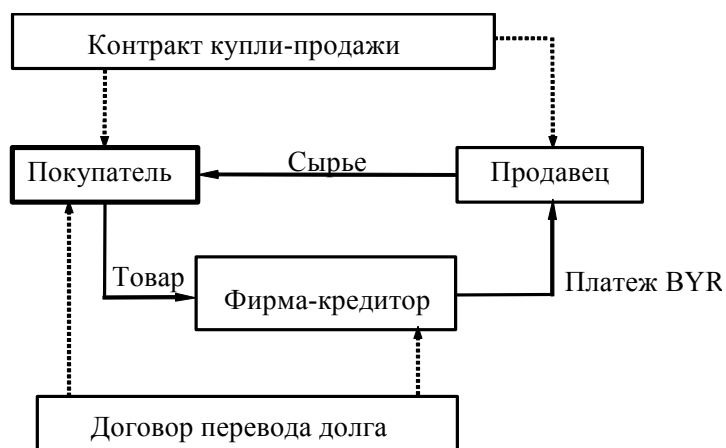


Рисунок 6.12 – Схема отношений по контракту с участием третьих лиц

6.2.2 Технология моделирования контрактов в ПТКИ BELSIM. Для моделирования контрактов в базовую модель предприятия [11, с. 11–16] добавлены процессы, алгоритмы которых моделируют выполнение соответствующих типов контрактов. Моделируются процессы, связанные непосредственно с контрактами, т. е. с производством и реализацией продукции: процессы снабжения, производства и сбыта продукции, финансовые процессы [238, с. 66].

Этапы технологии моделирования контрактов [221, с. 184–186].

Первый этап. Для проведения анализа эффективности контракта с применением ПТКИ Belsim на первом этапе определяется структура входной, выходной и управляющей информации. Для представления структуры этой информации используется методология *IDEF0* и *CASE*-пакет *BPwin*.

Исходными данными для моделирования параметров контракта являются, например, дата подписания контракта, сроки поставок ресурсов, сроки отгрузок продукции, объёмы поставок для каждого вида ресурсов, относительные объёмы отгрузок для каждого вида продукции (определя-

ются как отношение объема продукции в каждой отгрузке к общему объёму данного вида продукции в контракте), рентабельность каждого вида поставляемой продукции для каждой отгрузки.

Второй этап. Для моделирования контракта, например контракта по товарному обмену, выполняется следующая последовательность действий.

1 *Имитировать процесс поставки ресурсов.* Для каждой поставки сырья выполняется увеличение общей суммы контракта на сумму поставки ресурса, т. е. рассчитывается общая сумма контракта. Сумма поставки для каждого ресурса рассчитывается как произведение объема ресурса на цену ресурса.

2 *Имитировать факт заключения контракта.* Производится расчёт объемов каждого вида продукции для каждой отгрузки. Для этого общая сумма контракта умножается на относительный объем каждого вида продукции в каждой отгрузке и делится на цену данного вида продукции.

3 *Имитировать процесс отгрузки продукции по контракту.* Прежде чем моделировать отгрузку продукции, необходимо оценить наличие данного вида продукции в необходимом количестве. Если имеется достаточное количество продукции, производится её отгрузка. В противном случае моделируется отгрузка имеющейся продукции, для недопоставленной продукции моделируется задержка отгрузки на некоторое время (обычно на неделю), пока продукция не будет произведена в достаточном количестве и отгружена полностью [81, с. 33; 225, с. 184–188].

В соответствии с представленными процессами разрабатывается программа модели контракта, включается в ИМ и проводится их совместная отладка [221, с. 186–188].

Третий этап. После доработки и отладки ИМ вводятся исходные данные о текущем состоянии предприятия и состоянии внешней среды, строится план эксперимента. Пусть, например, требуется определить эффективность заключения контракта по критерию рентабельности продукции. Для этого ИМ исследуется на двух уровнях (0 – прогон модели без контракта, 1 – прогон модели с контрактом). Для ввода исходных данных модели и построения плана эксперимента используется специальная программа – *ExperimentDesigner* [221, с. 186].

Полученные в результате прогонов модели результаты сохраняются в виде XML-файлов, и их дальнейшая обработка выполняется при помощи математического пакета STATISTICA 6.0.

Рабочая область программы *Experiment Designer* отображает информацию в зависимости от этапа подготовки эксперимента. После выбора исполняемого файла модели (или файла данных модели) на экране появится окно для ввода исходных данных модели (рисунок 6.13) [221, с. 186]. Ввод исходных данных модели представлен на рисунке 6.14.



Рисунок 6.13 – Главное окно программы Experiment Designer для ввода данных модели

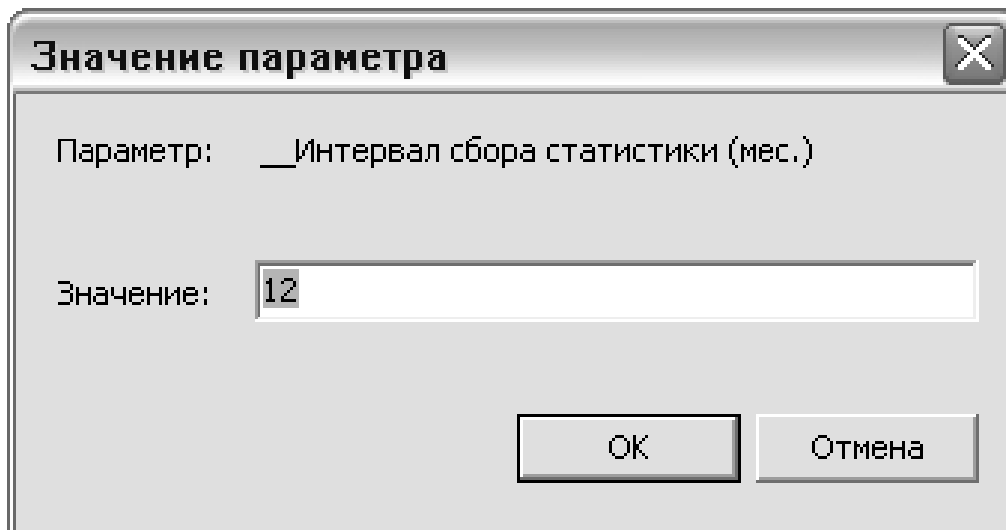


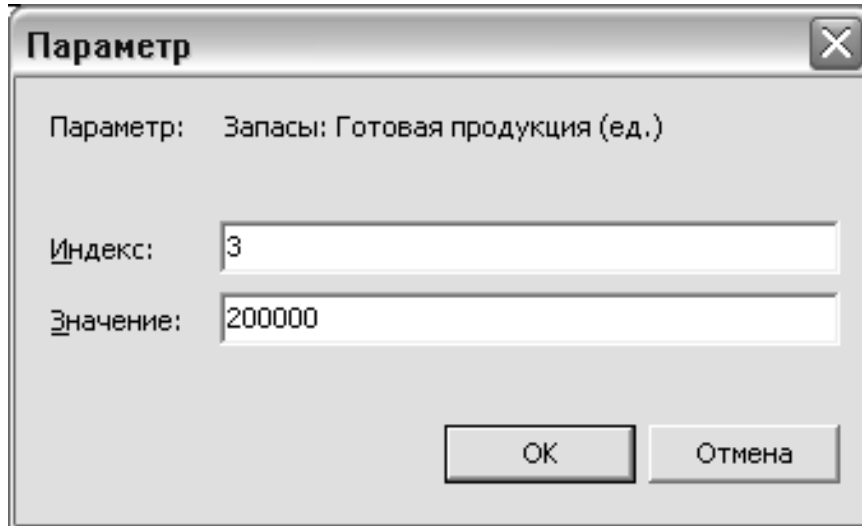
Рисунок 6.14 – Окно для ввода исходных данных модели

Параметры, после которых не следует знака равенства, представляют собой массивы данных для разных видов ресурсов, продукции и т. д.

Ввод массивов осуществляется после выделения курсором или левой кнопкой мыши параметра, который следует ввести, и нажатия клавиши Insert на клавиатуре. В появившемся окне, представленном на рисун-

ке 6.15, вводят значение индекса элемента массива и значение элемента массива.

После этого начинается подготовка эксперимента. Тип эксперимента задается в соответствующем окне (рисунок 6.16).



Параметр

Параметр: Запасы: Готовая продукция (ед.)

Индекс: 3

Значение: 200000

OK Отмена

Рисунок 6.15 – Ввод параметров-массивов



Experiment Designer v1.02

1. Модель 3. Эксперимент
2. Данные модели 4. Данные эксперимента

Experiment Designer
© 2003 Сергей А. Альховик

Тип эксперимента

Оценка погрешности моделирования
Анализ длины переходного процесса и устойчивости результатов моделирования
Анализ чувствительности откликов к изменению входных параметров
Полный факторный эксперимент
Дробный факторный эксперимент

<- Назад Сохранить Сохранить как... Справка Выход

Рисунок 6.16 – Окно выбора типа эксперимента

Данные для эксперимента выбираются через окно для выбора параметров и откликов модели. Для этого в области данных эксперимента следует выбрать параметры и отклики модели, щёлкнув левой кнопкой мыши по надписи «Переменные» в области «Данные эксперимента» и нажав на клавиатуре клавишу *Insert*. В результате на экране появится окно для выбора параметров и откликов модели (рисунок 6.17). Выбор параметров и откликов осуществляется путем установки соответствующих флажков. В качестве параметров выбираются независимые переменные, в качестве откликов – зависимые. После выбора параметров и откликов модели, ввода уровней изменения параметров и числа параллельных опытов (обычно 100) файл эксперимента сохраняют, используя ссылки «Сохранить» и «Сохранить как» из управляющей области программы.

После проведения эксперимента осуществляется обработка результатов средствами математического пакета STATISTICA. Для загрузки данных эксперимента в STATISTICA вызывается макрос ExperimentData (рисунок 6.18). При этом открывается диалоговое окно выбора файла с результатами моделирования, в котором следует выбрать файл данных эксперимента. В пакете STATISTICA можно, например, построить график зависимости какого-либо отклика от параметра, используя меню Graphs и т. д.

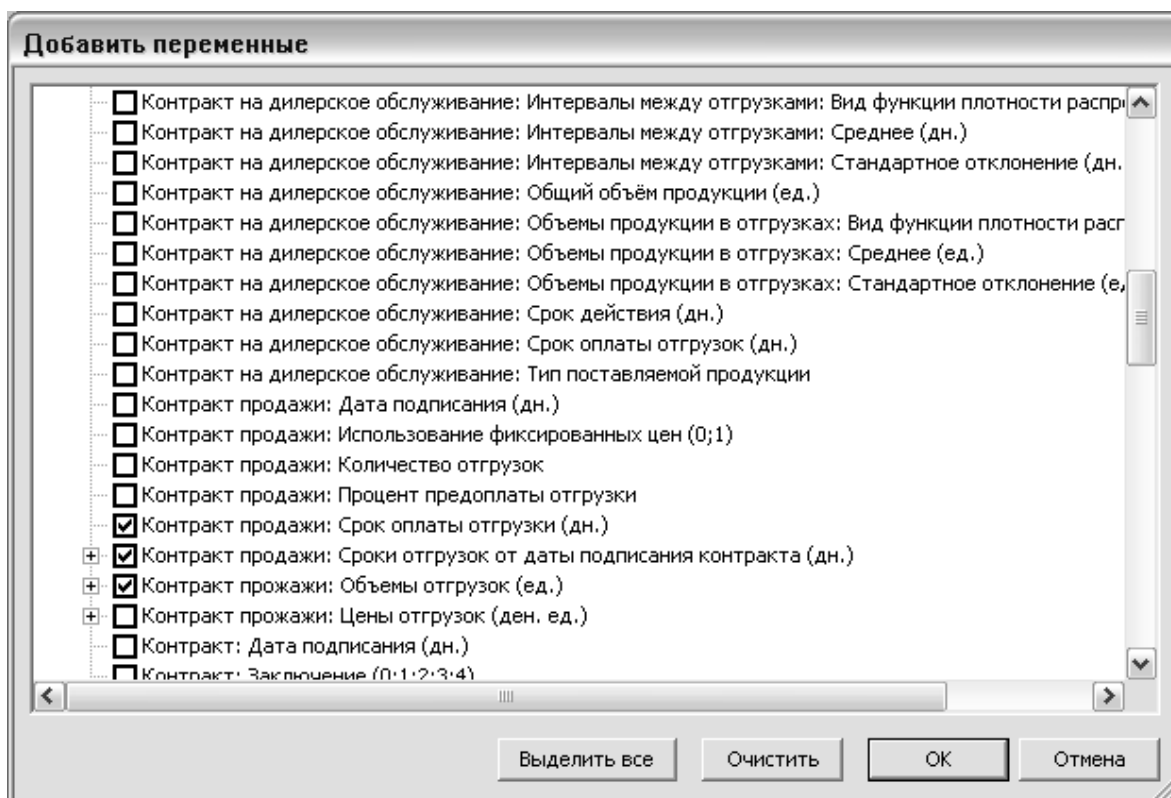
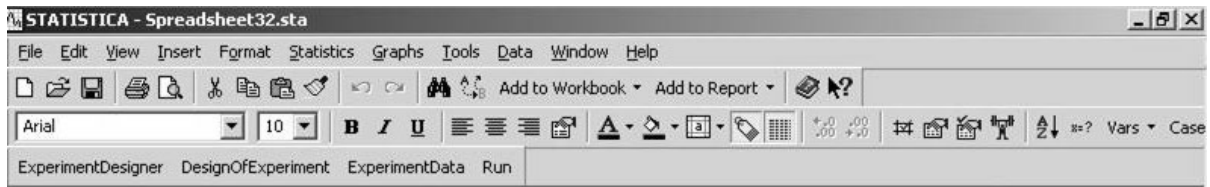


Рисунок 6.17 – Окно выбора параметров и откликов модели



Кнопка для загрузки
результатов моделирования

Рисунок 6.18 – Загрузка данных эксперимента в пакет STATISTICA

6.2.3 Вербальная модель контракта продажи. Контекстная диаграмма потоков данных контракта продажи представляет собой схему «Выполнить контракт продажи». Входные данные: размер предоплаты, объем отгрузки, цена вида продукции в каждой отгрузке, количество отгрузок – вводятся пользователем. Результатом функционирования являются: объем оставшейся продукции на складе, общая сумма отгрузок, формирование бухгалтерского баланса [78, с. 50]. Контекстная диаграмма договора продажи представлена на рисунке 6.19.

Декомпозиция процесса «Выполнить контракт продажи», или функциональная модель диаграммы первого уровня, проведенная в соответствии с принципом декомпозиции цели, представлена в виде диаграммы на рисунке 6.20.



Рисунок 6.19 – Контекстная диаграмма контракта продажи

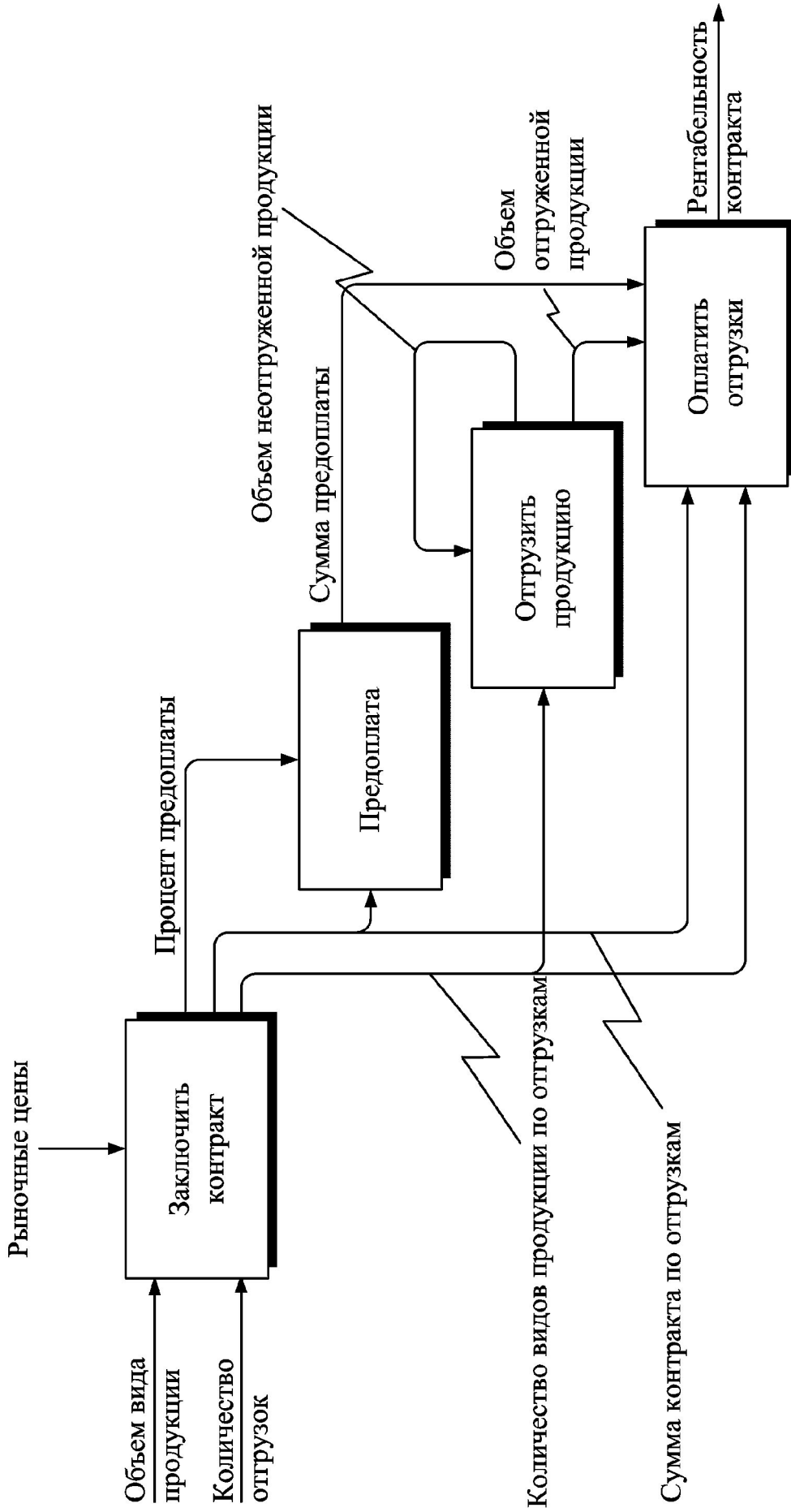


Рисунок 6.20 – Функциональная диаграмма первого уровня, или декомпозиция процесса «Выполнить контракт продажи»

6.2.4 Имитационная модель контракта продажи. В маркетинговой деятельности промышленного предприятия модели контрактов позволяют анализировать их эффективность на стадии заключения и принимать рациональные решения. Практика внедрения программно-технологического комплекса имитации сложных систем (ПТКИ) BelSim [262, с. 3–4] требует создания библиотеки имитационных моделей различных типов контрактов, в частности контракта продажи [226, с. 139–141].

Постановка задачи. Необходимо оценить эффективность контракта продажи K , заданного следующими подмножествами: $K = \langle Vп, Oк, Sк \rangle$, где $Vп$ – множество видов продукции, $Oк$ – множество отгрузок продукции по контракту; $Sк$ – объем контракта в денежном измерении.

Множество видов продукции по контракту $Vп = \{Vп_i\}$, $i = 1, \dots, n$, где n – количество видов продукции; $Vп_i$ определяется следующими подмножествами:

$$Vп_i = \langle Tо_j, \{OПo_{ij}, ЦПo_{ij}\} \rangle, i = 1, \dots, n, j = 1, \dots, m, \quad (6.1)$$

где m – количество отгрузок продукции;

$Tо_j$ – дата j -й отгрузки продукции;

$OПo_{ij}$ – объем i -го вида продукции в j -й отгрузке;

$ЦПo_{ij}$ – цена i -го вида продукции в j -й отгрузке.

Множество отгрузок продукции по контракту $Oк = \{Oк_j\}$, $j = 1, \dots, m$, (m – количество отгрузок продукции); $Oк_j = \langle Dо_j, \{Oо_{ij}, Цпо_{ij}\} \rangle$, $i = 1, \dots, n$, $j = 1, \dots, m$ (n – количество видов продукции); $Dо_j$ – дата j -й отгрузки продукции; $Oо_{ij}$ – относительный объем i -го вида продукции в j -й отгрузке; $Цпо_{ij}$ – цена i -го вида продукции в j -й отгрузке [226, с. 139].

Сумма контракта в денежном измерении определяется выражением

$$Sк = \sum_{i=1}^n \sum_{j=1}^m ЦПo_{ij} \cdot OПo_{ij}. \quad (6.2)$$

Промежуточным результатом является $OПк_j$ – оплата отгрузки продукции с учетом предоплаты:

$$OПк_j = Sк_j \cdot (1 - ПП), \quad (6.3)$$

где $ПП$ – процент предоплаты, отн. ед.;

$Sк_j$ – сумма j -й отгрузки.

При анализе контракта продажи используются и другие параметры модели: $Sр/с$ – сумма на расчетном счете; $ЗC_i$ – объем складского запаса i -го вида продукции; n – количество видов продукции; $ЦР_i$ – рыночная цена i -го вида продукции; $СП$ – сумма предоплаты:

$$СП = S_k \cdot ПП. \quad (6.4)$$

Имитационная модель контракта продажи с предварительной оплатой. Для оценки эффективности контракта продажи в ИМ предприятия добавлен процесс «Анализ контракта продажи». При заключении контракта учитываются виды продукции, их объемы и цены с учетом рентабельности. Если в контракте цены на реализуемую продукцию не определены, то используются рыночные цены на продукцию на момент заключения контракта. Предоплата определяется в процентном отношении от объема контракта.

Моделирование отгрузки продукции по контракту выполняется в несколько этапов. Для каждой партии продукции в качестве исходных данных задаются дата отгрузки и относительные объемы отгрузки по каждому виду продукции (определяются как отношение объема продукции в каждой отгрузке к общему объему данного вида продукции в контракте), а также рентабельность каждого вида продукции. После завершения отгрузки продукции перечисляются деньги на расчетный счет предприятия-продавца и формируются соответствующие показатели бухгалтерского баланса. Добавленный в ИМ процесс состоит из четырех активностей (рисунок 6.21) [226, с. 140].

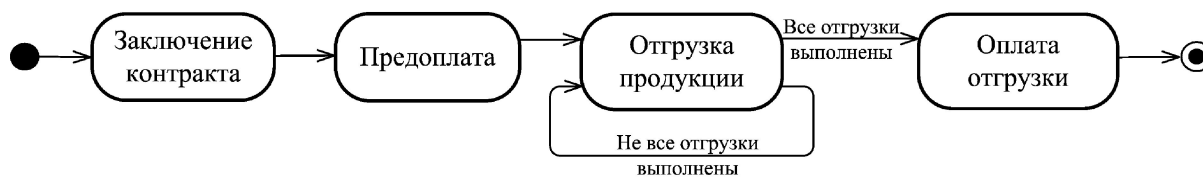


Рисунок 6.21 – Диаграмма активностей процесса анализа контракта продажи с предварительной оплатой

Активность «Заключение контракта». Для всех видов продукции по контракту продажи устанавливаются рыночные цены на момент его заключения, а также выполняется создание активностей отгрузок, моменты активизации которых соответствуют срокам поставок продукции покупателю (рисунок 6.22).

Активность «Предоплата». При выполнении этой активности рассчитывается сумма предоплаты, которая переводится на расчетный счет предприятия-продавца, осуществляется формирование показателей бухгалтерского баланса (рисунок 6.23) [226, с. 140]. Далее переходят к активности, моделирующей отгрузку продукции.

Активность «Отгрузка продукции». Осуществляется проверка наличия на складе в достаточном количестве всех видов продукции, которые нужно отгрузить. Если объем складских запасов каждого вида продукции превышает объем продукции, который необходимо отгрузить, то выполняется отгрузка продукции (уменьшается объем складских запасов, формируются показатели бухгалтерского баланса) и создается активность оплаты отгрузки.

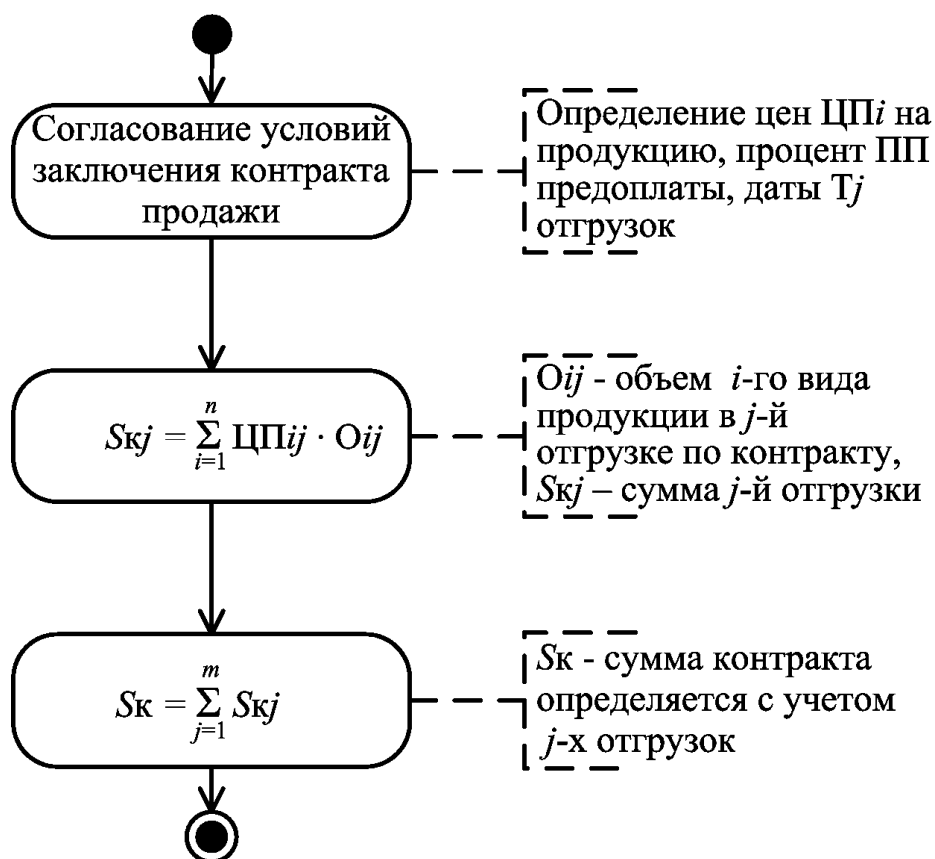


Рисунок 6.22 – Алгоритм активности «Заключение контракта»

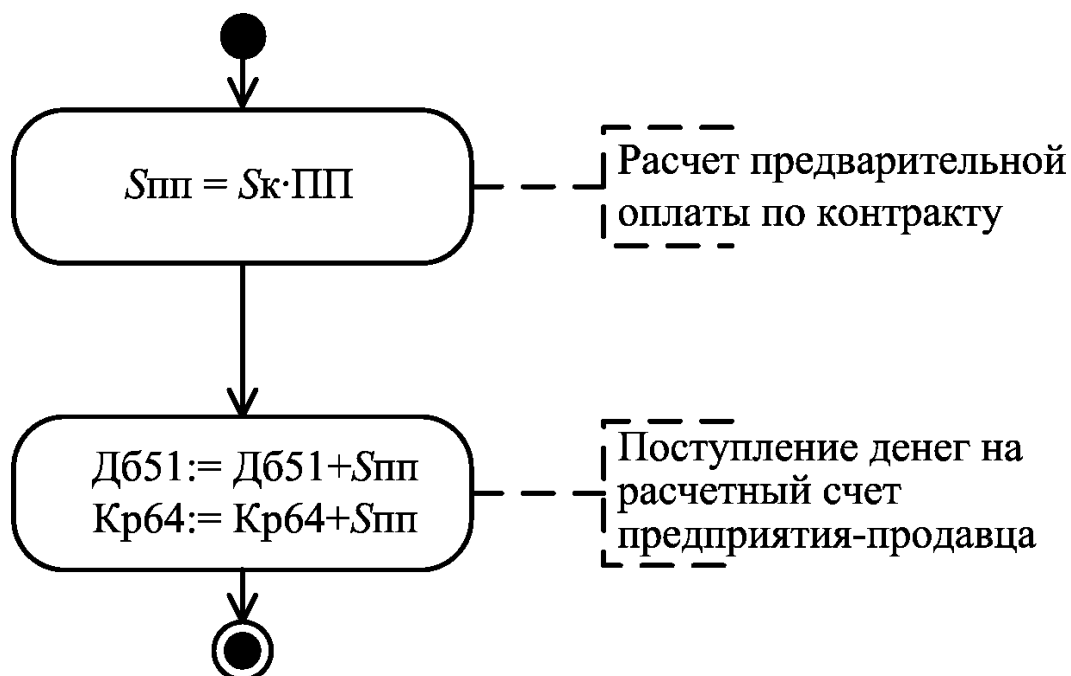


Рисунок 6.23 – Алгоритм активности «Предоплата»

Если на складе недостаточно какого-либо вида продукции, то эта активность переходит в состояние ожидания, пока продукция не будет произведена (рисунок 6.24). Следует отметить, что активность отгрузки продукции аналогична и для других видов контрактов, например для контракта по товарному обмену и контракту на дилерское обслуживание [78, с. 50–53].

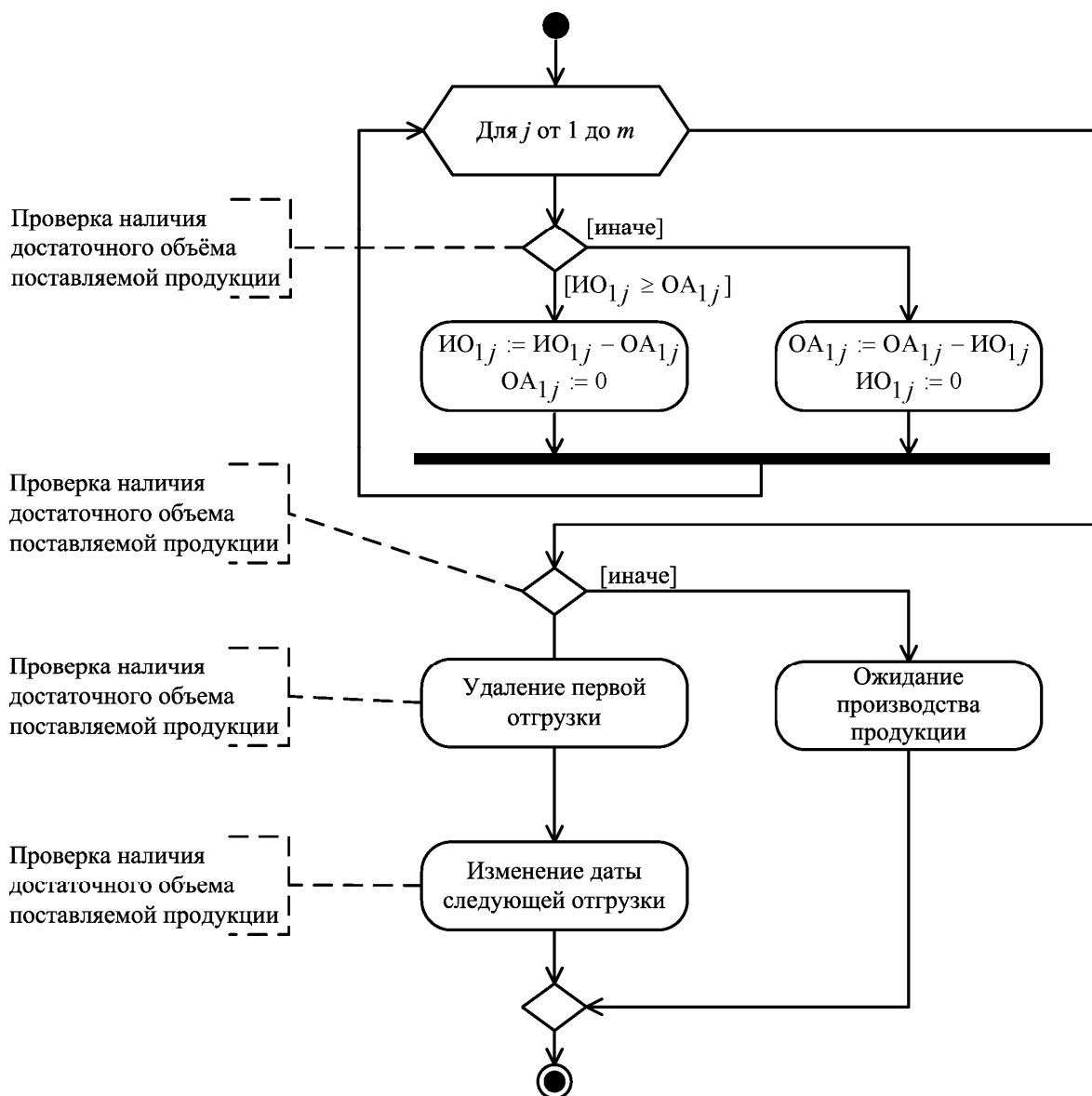


Рисунок 6.24 – Активность «Отгрузка продукции»

Активность «Оплата отгрузки». В ходе выполнения данной активности осуществляется расчет суммы отгрузки, перечисление денег на расчетный счет и формирование соответствующих показателей бухгалтерского баланса (рисунок 6.25) [226, с. 141]. Для оценки эффективности контракта ИМ предприятия прогоняется дважды: первый раз – без контракта, второй раз – с контрактом. Полученные финансово-экономические показатели работы предприятия сравниваются для этих двух случаев и делается вывод о целесообразности заключения контракта [221, с. 183].

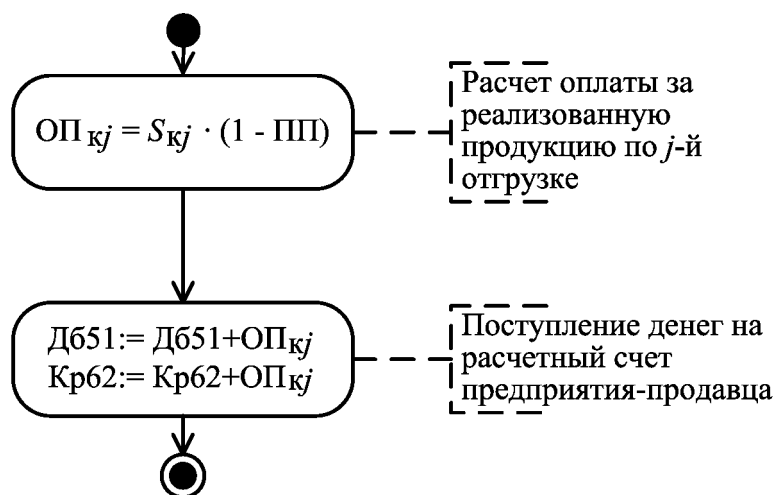


Рисунок 6.25 – Алгоритм активности «Оплата отгрузки»

6.3 Математическое моделирование в комплексной информационной системе

6.3.1 Модель оценки рентабельности контракта с предварительной оплатой. Обращение к имитационной модели для оперативной оценки рентабельности контрактов при их заключении не всегда представляется возможным. Представим математическую модель рентабельности контракта в виде набора упрощенных функциональных зависимостей от его параметров, состояния предприятия и его окружающей среды. Эта модель, используемая в качестве гипотезы, потребует проверки путем проведения соответствующих экспериментов на имитационной модели.

В маркетинговой деятельности предприятий при определении ценовой политики используют ориентированную на потребителя систему обоснования цен, применяют различные методы и стратегии их формирования, учитывают разнообразные факторы, определяющие уровень цен на отдельных товарных рынках [142]. Эффективность оборотных средств, полученных при реализации товарной продукции с предварительной оплатой, определяется их вложением в производство: приобретается сырье, производится готовая продукция и реализуется с плановой рентабельностью производства R_n .

Производителю предоставляется беспроцентная ссуда, за счет которой он получает дополнительную прибыль, отправляя все средства на развитие производства. В другом варианте он отказывается от краткосрочной ссуды банка и при этом также извлекает выгоду [180].

Учет прибыли за счет банковского процента. Пусть контракт заключен на сумму

$$X_n C = X_n C (1 + R), \quad (6.5)$$

где $X_nЦ$ – объем контракта в денежном выражении;
 X_n – объем товарной продукции в соответствии с контрактом;
 $Ц$ – цена единицы изделия по контракту;
 C – себестоимость единицы товарной продукции;
 R – рентабельность товарной продукции, реализуемой по контракту, отн. ед.

Сделана предварительная оплата, отгрузка произведена через T_{np} дней. Если полученные средства не направляются в производство, то прибыль может быть получена за счет банковского процента, начисленного за период T_{np} .

К моменту отгрузки товарной продукции выручка от ее реализации фактически определяется из выражения

$$S_6 = X_nЦ + I, \quad (6.6)$$

где S_6 – объем контракта с учетом дополнительной прибыли за счет банковского процента;

I – сумма процентной надбавки за предоставленную ссуду [45]:

$$I = X_nЦ \frac{T_{np}}{T_2} i_2 = X_n C (1 + R) \frac{T_{np}}{T_2} i_2, \quad (6.7)$$

где T_{np} – длительность от момента предварительной оплаты до отгрузки товарной продукции, дн.;

T_2 – длительность года, дн.;

i_2 – простая годовая ставка банковского ссудного процента, отн. ед.

На рисунке 6.26 представлена геометрическая модель контракта с предварительной оплатой [221, с. 29], по которой рентабельность контракта определяется из следующего выражения:

$$S_6 = X_n C (1 + R_6), \quad (6.8)$$

где R_6 – рентабельность контракта при предварительной оплате с учетом банковского процента, отн. ед.

С учетом равенств (6.6) и (6.8) можно составить тождество

$$X_n C (1 + R_6) = X_n C (1 + R) + C (1 + R) \frac{T_{np}}{T_2} i_2. \quad (6.9)$$

Разделив выражение (6.9) на $X_n C$ и проведя несложные преобразования, определяем искомую рентабельность контракта с предварительной оплатой:

$$1 + R_{\delta} = (1 + R)\left(1 + \frac{T_{np}}{T_2} i_2\right). \quad (6.10)$$

Окончательно искомая рентабельность R_{δ} определяется выражением

$$R_{\delta} = (1 + R)\left(1 + \frac{T_{np}}{T_2} i_2\right) - 1. \quad (6.11)$$

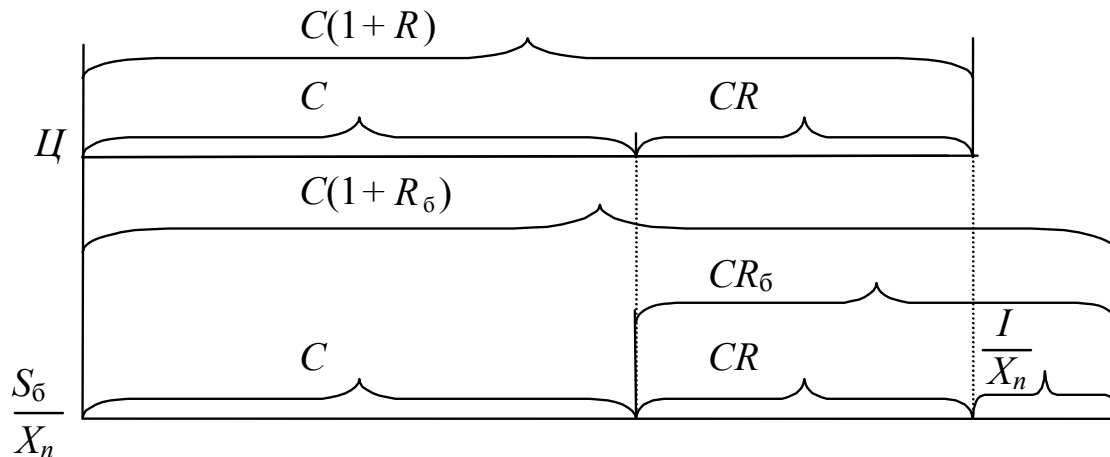


Рисунок 6.26 – Геометрическая модель оценки контракта при предварительной оплате с учетом банковского процента

Учет прибыли в производственном цикле. Для определения прибыли от привлечения оборотных средств в производственный цикл введем следующие обозначения: $C_P = ЦX_n$ – себестоимость выпуска изделий за счет оборотных средств, полученных по предварительной оплате; $\frac{C_P}{T_{ц}}$ – себестоимость выпуска изделий, приходящаяся на единицу производственного цикла; $C_P \frac{T}{T_{ц}}$ – себестоимость выпуска изделий за T дней производственного цикла; $C_P R_n$ – прибыль от изделий, изготовленных за счет средств по предварительной оплате; $\frac{C_P R_n}{T_{ц}}$ – прибыль от выпуска изделий, приходящаяся на единицу производственного цикла; $C_P R_n \frac{T}{T_{ц}}$ – прибыль, созданная за T дней производственного цикла.

С учетом приведенных обозначений прибыль, полученная в производственном цикле за время использования оборотных средств при предварительной оплате:

$$\Pi_P = C_P R_n \frac{T_{np}}{T_u}, \quad (6.12)$$

где R_n – рентабельность производства, отн. ед.;

T – длительность предварительной оплаты, $T = T_{np}$;

T_u – длительность производственного цикла, дн.

Для оценки рентабельности контракта добавляют к его сумме $X_n \Pi$ прибыль Π_P , полученную в производственном цикле за счет предварительной оплаты. Тогда расчетная сумма S_P контракта с учетом принятых обозначений и выражения (6.12):

$$S_P = X_n \Pi + \Pi_P = X_n C(1 + R) + X_n C(1 + R) R_n \frac{T_{np}}{T_u}. \quad (6.13)$$

Определим цену изделий по контракту с учетом прибыли Π_P , отнесенной на его счет:

$$\Pi_P = \frac{S_P}{X_n} = C(1 + R_p). \quad (6.14)$$

В итоге рентабельность контракта R_P с учетом прибыли в производстве определится следующим образом:

$$C(1 + R_p) = \Pi + \frac{\Pi_P}{X_n} = C(1 + R) + C(1 + R) R_n \frac{T_{np}}{T_u}. \quad (6.15)$$

Отсюда

$$1 + R_p = (1 + R) + (1 + R) R_n \frac{T_{np}}{T_u}, \quad (6.16)$$

или

$$R_p = (1 + R) \left(1 + R_n \frac{T_{np}}{T_u} \right) - 1. \quad (6.17)$$

Такие рассуждения справедливы, если объем контракта X_n значительно меньше объема производственной программы [234, с. 29–33]. В

противном случае необходимо учитывать изменение доли постоянных затрат в себестоимости единицы продукции при изменении объема выпуска [221, с. 31].

Учет инфляционного фактора. В условиях инфляции государственные или иные ограничения на рост цен снижают эффективность средств, вкладываемых в производство. Эти ограничения должны быть учтены при расчете рентабельности контракта с предварительной оплатой. В разработанной геометрической модели [221, с. 32] учитывается, что повышение цен с темпом инфляции α ограничивается темпом γ роста цен на выпускаемую продукцию, снижая тем самым плановую норму прибыли. Расчеты на практике удобно вести с индексами инфляции I_α и государственных ограничений I_γ , которые, как отмечается в [136, с. 25–30], наиболее просто определяются курсами валют.

Расчет рентабельности контракта с учетом инфляционного фактора выводится следующим образом по геометрической модели (рисунок 6.27) [221, с. 32; 269, с. 42]:

$$C(1 + \alpha)R_{np}^u = C(1 + \gamma)R_{np} - [C(1 + \alpha) - C(1 + \gamma)]. \quad (6.18)$$

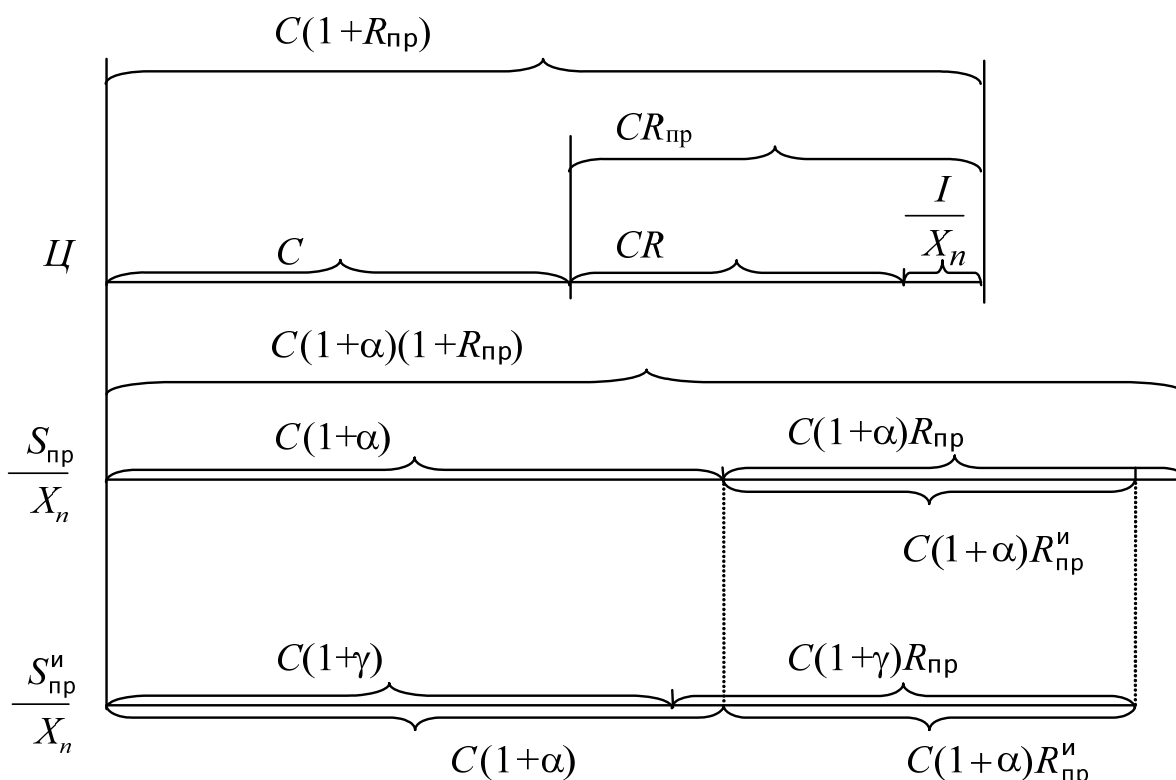


Рисунок 6.27 – Геометрическая модель предварительной оплаты с учетом инфляционного фактора

Из этого выражения определяется искомая рентабельность [269, с. 43]

$$R_{np}^u = \frac{R_{np}I_\alpha - (I_\alpha - I_\gamma)}{I_\alpha}, \quad (6.19)$$

здесь индекс инфляции I_α определяется в общем случае выражением

$$I_\alpha = (1 + \alpha)^{T_{np}/T_m}, \quad (6.20)$$

где α – темп инфляции, отн. ед.;

T_{np} – длительность предварительной оплаты, дн.;

T_m – длительность периода (например, один месяц) за который отмечают повышение цен, дн.

Индекс I_γ ограниченного роста цен на выпускаемую продукцию определяется аналогично выражению (6.20).

Полученное ранее выражение (6.19) может быть представлено в виде, удобном для анализа контрактов:

$$1 + R_{np}^u = \frac{I_\gamma}{I_\alpha} + R_{np}. \quad (6.21)$$

В соответствии с (6.21) эффективность контракта снижается, если рост цен на товарную продукцию отстает от темпа инфляции.

Представленные результаты можно перевести в область нечетких расчетов, представив эти оценки, например, нечеткими треугольными числами [196, с. 133–137]. Эти преобразования чрезвычайно просты. Минимальная оценка эффективности контракта считается нижней границей в нечетком треугольном числе, максимальная оценка – верхней границей.

Наиболее вероятное значение при отсутствии дополнительной информации может быть определено средним значением интервала. При таком подходе значение функции принадлежности $\mu(\alpha)$ в нижней и верхней границах считается равным нулю, а $\mu(\alpha)$ наиболее вероятного значения равным единице [196, с. 133–137].

В качестве примера построены интервальные оценки рентабельности контракта без учета и с учетом инфляционного фактора (рисунок 6.28).

Полученные выражения позволяют автоматизировать процесс оценки эффективности контракта и ценообразования в маркетинговой деятельности предприятий [209, с. 513; 212, с. 502–507].

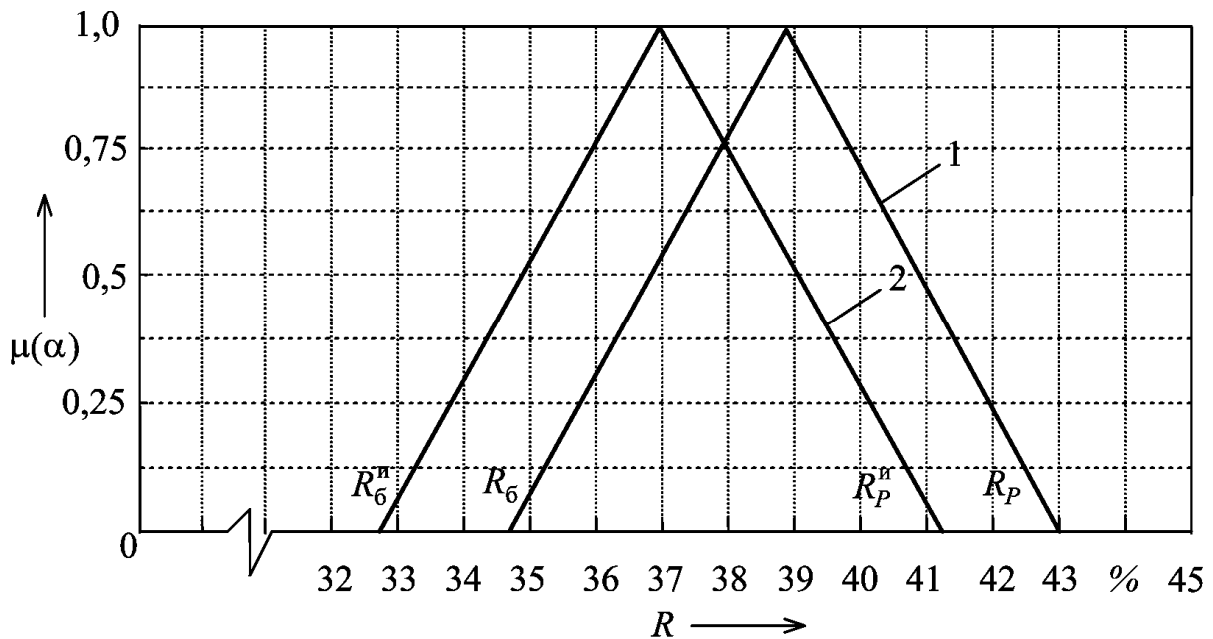


Рисунок 6.28 – Интервальная оценка эффективности контракта с предварительной оплатой без учета (1) и с учетом (2) инфляционного фактора

6.3.2 Модель оценки рентабельности контракта с отсрочкой платежа. При реализации товарной продукции различные условия контракта оказывают влияние на финансовое состояние предприятия. Контракт с отсрочкой платежа приводит к «замораживанию» части капитала, предприятие несет потери из-за невовлечения денежных средств в производственный оборот и вынуждено брать банковский кредит. Покупателю фактически выдана ссуда без процентов, которую он использовал для покупки товара и использует ее далее для получения прибыли [207].

При отсрочке платежа продавец идет на упущенную выгоду – тот же убыток, по крайней мере, в сумме банковского ссудного процента, который мог бы набежать на эту сумму за время отсрочки, если бы она была получена немедленно. Более того, если рентабельность предприятия превышает среднебанковскую ставку процента, то сумма платежа, немедленно пущенная в оборот, могла бы принести еще большее приращение [180].

Один из недостатков отсрочки платежа для продавца – сложность определения потерь от отсутствия оборотных средств и темпов инфляции и, соответственно, от связанного с этим риска. Для покупателя преимущество заключается в возможности оплаты товаров после их продажи на рынке, не отвлекая собственные оборотные средства. При этом, однако, необходимо уплатить продавцу проценты за кредит [14].

Возможны различные варианты формализации ценообразования при реализации товарной продукции с отсрочкой платежа. Рассмотрим вариант, при котором ситуация сводится к «замораживанию» капитала и продавец не получает выгоды в виде процентов от ссуды, предоставленной по-

купателю. При этом следует рассматривать потери из-за невовлечения оборотных средств в производство и потери от неполучения процентов за предоставленную покупателю ссуду.

Учет потерь от неполучения процентов за предоставленную ссуду. Отгружая товарную продукцию с отсрочкой платежа, продавец фактически выдает покупателю беспроцентную ссуду. Ведь в простейшем случае для закупки товара покупатель вынужден при отсутствии денежных средств взять кредит, ссуду и, естественно, при этом выплачивать ссудный процент.

Пусть реализовано товарной продукции в объеме X_n по цене $Ц$ за единицу продукции на сумму $X_nЦ$ с отсрочкой платежа на T_{om} дней. Анализ проведем с учетом простой ставки ссудных (декурсивных) процентов, применяемых обычно в краткосрочных финансовых операциях. При этом в зависимости от способа определения продолжительности финансовой операции рассчитывается либо точный, либо обыкновенный процент. Дата выдачи и дата погашения ссуды всегда считаются за один день. Возможны два варианта:

- 1) используется точное число дней ссуды;
- 2) применяется приближительное число дней ссуды, когда продолжительность полного месяца принимается равной 30 дням.

Вводятся следующие обозначения: i_2 – относительная величина простой годовой ставки ссудного процента; P – величина первоначальной денежной суммы; S – наращенная сумма; d – продолжительность периода начисления, дн.; T_2 – продолжительность года, дн. [45].

Основная формула для определения наращенной суммы:

$$S = P \left(1 + \frac{d}{T_2} i_2 \right). \quad (6.22)$$

Сумма процентных денег I за период начисления d находится по формуле

$$I = S - P = P \frac{d}{T_2} i_2. \quad (6.23)$$

Если первоначальная сумма P определяется суммой $X_nЦ$ товарной продукции по контракту, то с учетом периода начисления $d = T_{om}$ наращенная сумма S вычисляется по простой ставке ссудного процента:

$$S = P \left(1 + \frac{d}{T_2} i_2 \right) = X_nЦ \left(1 + \frac{T_{om}}{T_2} i_2 \right). \quad (6.24)$$

Выдавая покупателю кредит, продавец терпит убытки из-за невовлечения оборотных средств в производство и сам вынужден брать ссуду. В то же время покупатель извлекает дополнительную прибыль. При этом продавец недополучает ссудный процент при отсрочке платежа на величину

$$I = P \frac{d}{T_2} i_2 = X_n \Pi \frac{T_{om}}{T_2} i_2. \quad (6.25)$$

Пусть цена единицы товарной продукции определяется по формуле

$$\Pi = C (1 + R), \quad (6.26)$$

где C – себестоимость единицы товарной продукции;

R – рентабельность товарной продукции.

Отгружено товарной продукции в денежном выражении на сумму

$$X_n \Pi = X_n C (1 + R). \quad (6.27)$$

При условии выплаты покупателем ссудного процента наращенная сумма составит

$$S = P + I, \quad (6.28)$$

или

$$S = X_n C (1 + R) + X_n C (1 + R) \frac{T_{om}}{T_2} i_2. \quad (6.29)$$

Геометрическая интерпретация такой ситуации представлена на рисунке 6.29 [245, с. 442], в соответствии с которым определяется рентабельность контракта. Для наращенной суммы S рассчитывается цена Π_S товарной продукции:

$$\Pi_S = \frac{S}{X_n}. \quad (6.30)$$

Выражение (6.30) можно записать иначе:

$$\Pi_S = C (1 + R_S), \quad (6.31)$$

где R_S – рентабельность товарной продукции, реализованной по цене Π_S .

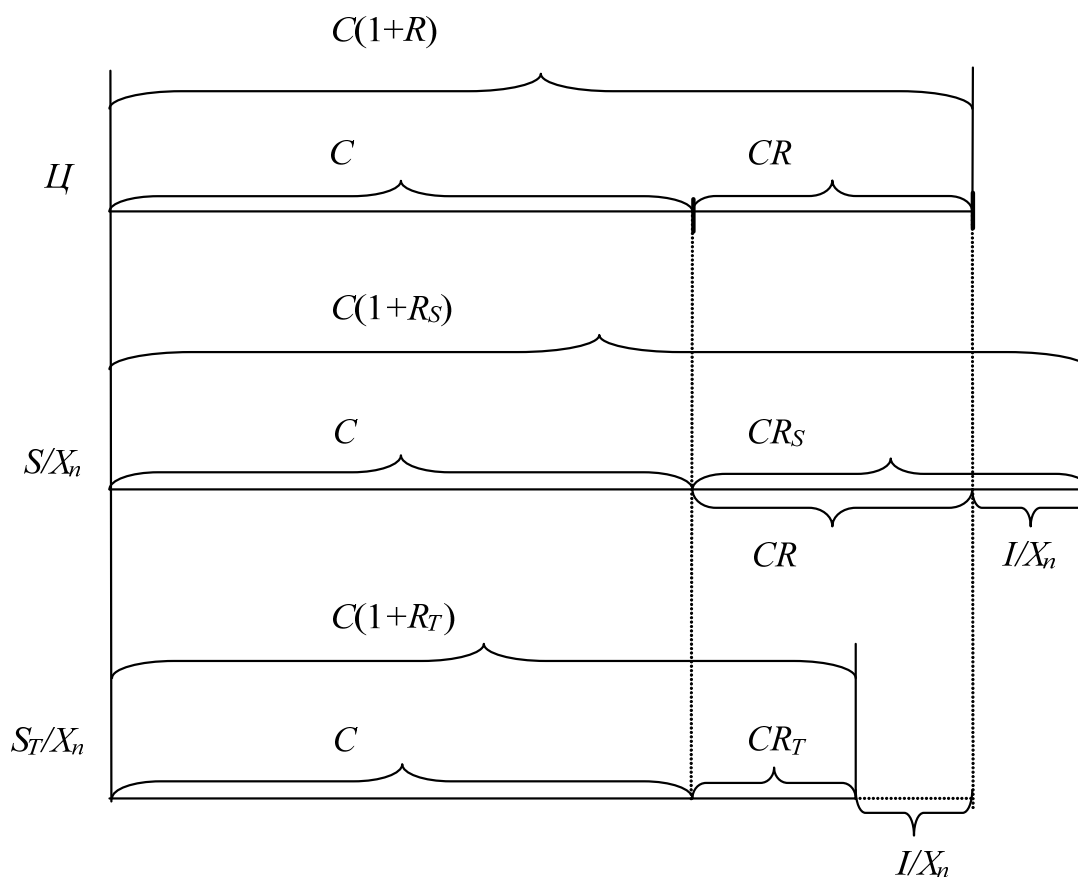


Рисунок 6.29 – Геометрическая интерпретация изменения цены C с учетом ссудного процента

В соответствии с геометрической моделью (см. рисунок 6.29)

$$C(1 + R_S) = C(1 + R) + C(1 + R) \frac{T_{om}}{T_2} i_2, \quad (6.32)$$

откуда искомая рентабельность контракта определяется рентабельностью товарной продукции:

$$R_S = R + (1 + R) \frac{T_{om}}{T_2} i_2. \quad (6.33)$$

Товарная продукция, реализованная с рентабельностью R_S , компенсирует потери продавца, вынужденного брать банковский кредит на время отсрочки платежа.

При невозврате покупателем процента за ссуду продавец сам несет потери. С их учетом сумма контракта

$$S_T = X_n C - I. \quad (6.34)$$

Продукция реализуется по цене

$$Ц_T = S_T / X_n, \quad (6.35)$$

или

$$Ц_T = Ц - I / X_n. \quad (6.36)$$

Рентабельность R_T товарной продукции или рентабельность контракта в этом случае определяется из следующего выражения (см. рисунок 6.29):

$$C(1 + R_T) = C(1 + R) - C(1 + R) \frac{T_{om}}{T_2} i_2. \quad (6.37)$$

Отсюда определяется искомая рентабельность

$$R_T = R - (1 + R) \frac{T_{om}}{T_2} i_2. \quad (6.38)$$

Учет потерь в производственном цикле. Невовлечение оборотных средств в производственный процесс приводит к недополучению прибыли Π_{S1} , определяемой рентабельностью производства R_n [234, с. 29–33]. Для определения потерь от невовлечения оборотных средств в производственный цикл вводятся следующие обозначения: CX_n – затраты и издержки на выпуск X_n изделий; $\frac{CX_n}{T_u}$ – затраты и издержки, приходящиеся на единицу производственного цикла; $CX_n \frac{T}{T_u}$ – затраты и издержки на выпуск X_n изделий за T дней производственного цикла; $CR_n X_n$ – прибыль от реализации X_n изделий; $\frac{CR_n X_n}{T_u}$ – прибыль от X_n изделий, приходящаяся на единицу производственного цикла; $CR_n X_n \frac{T}{T_u}$ – прибыль, созданная за T дней производственного цикла.

С учетом приведенных обозначений прибыль, недополученная в производственном цикле за время отсрочки платежа,

$$\Pi_{S1} = CR_n X_n \frac{T_{om}}{T_u}, \quad (6.39)$$

где R_n – рентабельность производства, отн. ед.;

T_{om} – длительность отсрочки платежа, $T_{om} = T$;

T_u – длительность производственного цикла, дн.

Наихудший вариант для продавца – покупателю предоставлена беспроцентная ссуда, оборотные средства не вовлечены в производственный цикл. Для оценки рентабельности контракта вычитают из его суммы $X_n C$ недополученную прибыль Π_{S1} в производственном цикле и сумму процентов I , не полученную от покупателя. Расчетная сумма S_T контракта

$$S_T = X_n C - (\Pi_{S1} + I). \quad (6.40)$$

Цена изделий по контракту с учетом потерь, отнесенных на его счет,

$$C_{S1} = \frac{S_T}{X_n} = C(1 + R_{S1}). \quad (6.41)$$

В итоге рентабельность контракта R_{S1} с учетом потерь в производстве и неполучении процентов за кредит определится следующим образом (рисунок 6.30):

$$C(1 + R_{S1}) = C - \frac{(\Pi_{S1} + I)}{X_n} = C(1 + R) - \left(CR_n \frac{T_{om}}{T_u} + C(1 + R) \frac{T_{om}}{T_u} i_z \right). \quad (6.42)$$

Отсюда

$$R_{S1} = R - \left(R_n \frac{T_{om}}{T_u} + (1 + R) \frac{T_{om}}{T_u} i_z \right). \quad (6.43)$$

С учетом инфляционного фактора рентабельность контракта может быть определена по формуле [240, с. 34–37] для отсрочки платежа:

$$R_k^u = \frac{1 + R_k - I_\alpha}{I_\alpha}, \quad (6.44)$$

где I_α – индекс инфляции, определяется по формуле

$$I_\alpha = (1 + \alpha)^{T_{om}/T_M}, \quad (6.45)$$

где T_m – период (например, один месяц), за который определен рост цен, дн.;

T_{om} – длительность отсрочки платежа, дн.

Пусть покупателю отгружено товарной продукции на сумму $X_n C = 100$ тыс. р. с рентабельностью $R = 30\%$ и отсрочкой платежа $T_{om} = 30$ дн. Длительность производственного цикла $T_y = 60$ дн., рентабельность производства $R_n = 20\%$, темп инфляции за месяц $\alpha = 0,05$, простая банковская ставка ссудного процента $i_2 = 0,44$ отн. ед. На рисунке 6.31 представлены интервальные оценки рентабельности контракта без учета и с учетом инфляционного фактора [221, с. 41].

При окончательном расчете необходимо при отсрочке платежа изменить цену, чтобы учесть потери прибыли из-за невовлечения оборотных средств в производство с учетом прогнозируемого индекса инфляции [245, с. 440–446].

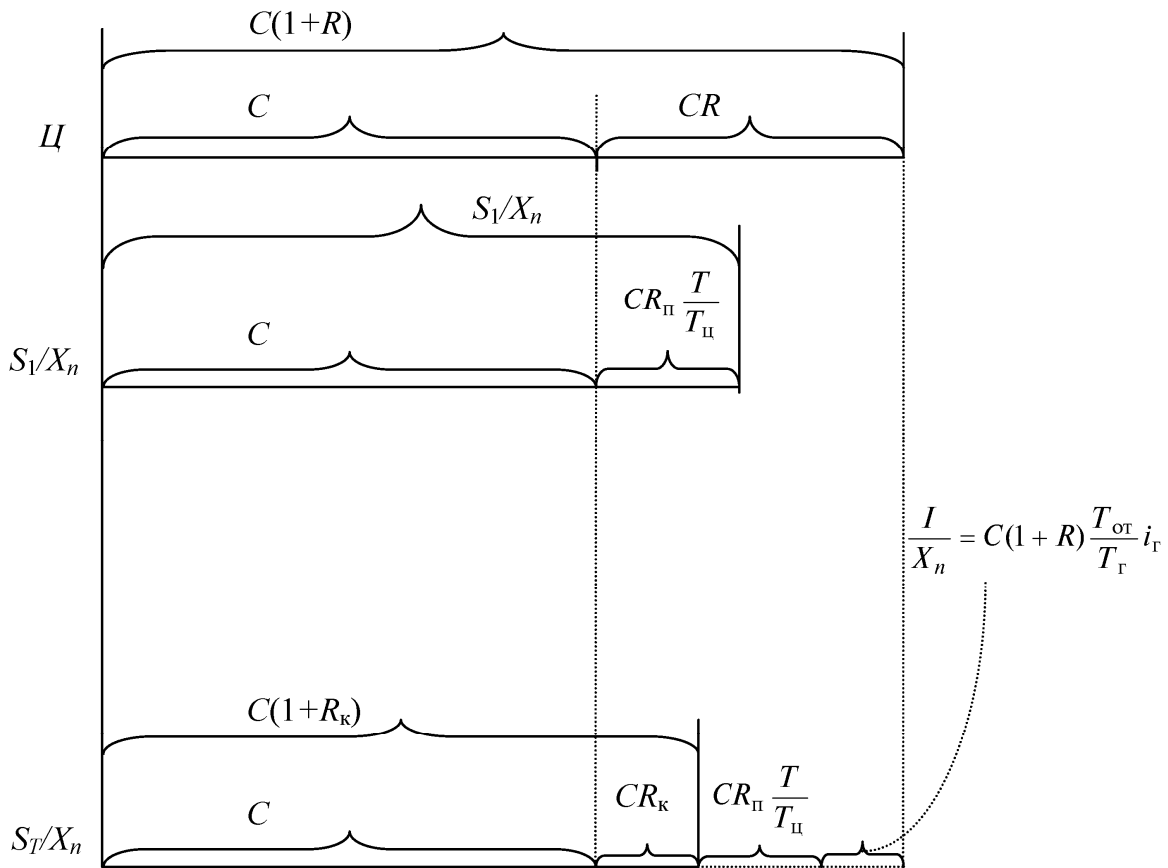


Рисунок 6.30 – Геометрическая модель рентабельности контракта

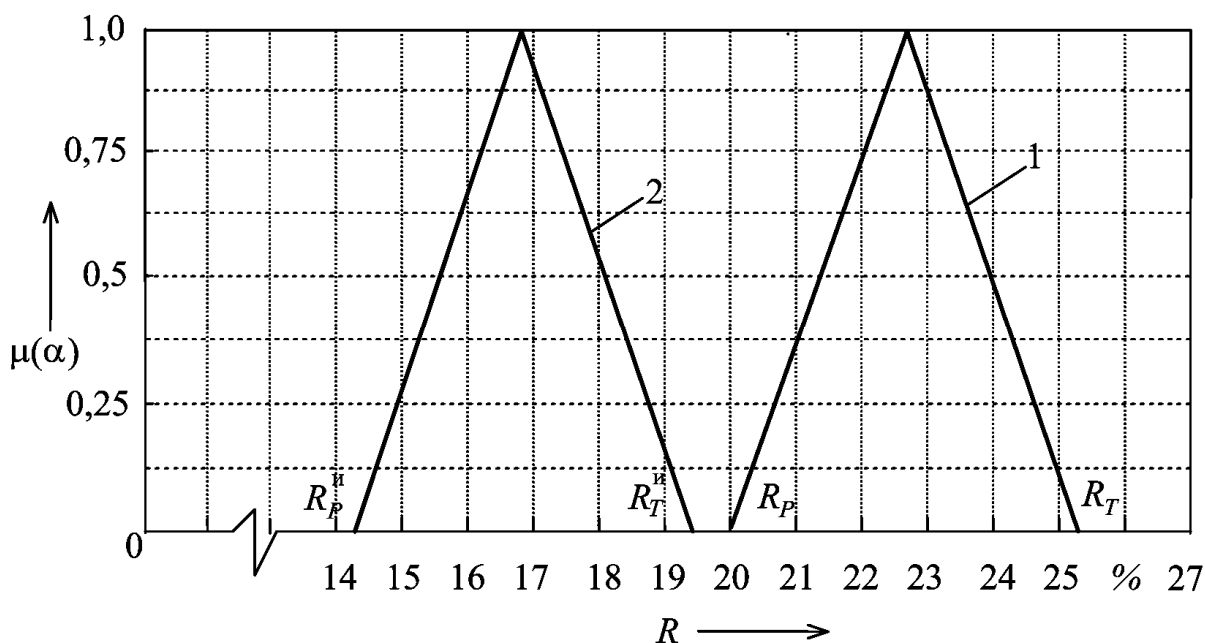


Рисунок 6.31 – Интервальная оценка эффективности контракта с отсрочкой платежа без учета (1) и с учетом (2) инфляционного фактора

6.4 Программный комплекс Contract Analyzer для оценки эффективности контрактов на промышленном предприятии

6.4.1 Основные характеристики программного комплекса Contract Analyzer. Решение задачи получения параметров (в частности, предполагаемой рентабельности) контракта вручную представляется очень трудоемким вследствие необходимости учета множества факторов (длительность технологического цикла, форма оплаты, сроки поставок, изменения курсов валют, ограничения на цены и т. д.), а также сложности математических зависимостей. В то же время анализ множества контрактов должен давать оценку финансового состояния предприятия. Прогнозирование финансового состояния предприятия на основе оценки эффективности контрактов во времени с учетом неопределенностей и возмущений в условиях современной экономики является актуальным для предприятий всех форм собственности и всех отраслей промышленности. Такая задача безусловно требует привлечения средств автоматизации [263, с. 146–149].

На основе разработанных математических моделей [221, с. 28–42; 240, с. 34–37; 245, с. 440–446] оперативного прогнозирования для автоматизации процедуры определения эффективности контракта на стадии его заключения разработан программный комплекс Contract Analyzer [213; с. 26–33; 255, с. 60–63; 258, с. 206–211] (таблица 6.6).

Таблица 6.6 – Технические характеристики программного комплекса Contract Analyzer

Операционная система для клиента	Windows 9x, Windows NT 4.0, Windows 2000
СУБД	MS Jet 4.0 (Access 2000)
Сервер	MSDE 1.0, MS SQL Server 7.0
Дата создания первой версии	2001 г.

Программное обеспечение Contract Analyzer предназначено для решения задач, относящихся к классу управленческих, предоставляет следующие функциональные возможности [255, с. 61]:

- расчет отпускной цены изделия для заданной рентабельности;
- расчет рентабельности изделия для договорной цены;
- расчет рентабельности контракта с учетом объемов реализации каждого артикула;
- пересчет отпускной цены с учетом курсов валют на определенную дату;
- оценка рентабельности контракта с учетом форм оплаты (предоплата, оплата по факту отгрузки, отсрочка платежа);
- прогнозирование рентабельности контракта с учетом индекса инфляции;
- прогнозирование рентабельности контракта с учетом ограничений на рост цен;
- формирование отчета с показателями эффективности контракта (рентабельность изделия плановая и договорная, рентабельность контракта плановая и договорная, последняя с учетом факторов времени, инфляции, ценовых ограничений);
- расчет показателей эффективности контракта с несколькими отгрузками за период действия контракта.

При этом обеспечивается ведение базы данных контрактов для контроля портфеля заказов [228, с. 34–35].

6.4.2 Технология оценки эффективности контрактов в Contract Analyzer. В программном комплексе предусмотрены различные варианты отношений между покупателем и продавцом. Например, оценивается эффективность контракта с предварительной оплатой и отсрочкой платежа в условиях инфляционных процессов и ограничений на рост цен (рисунок 6.32).

Среди схем отношений использованы наиболее часто встречающиеся в практике предприятий контракты продажи продукции, купли сырья и материалов и др.

The screenshot displays the 'Contract Analyzer 2' software window. The title bar reads 'Контракт #245 - Contract Analyzer 2'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Контракт', 'Расчеты', and 'Помощь'. The toolbar contains icons for printing, saving, opening, and other functions.

The main interface is divided into several sections:

- Contract Type:** A dropdown menu is set to 'Предоплата+Отсрочка'.
- Contract Diagram:** A flowchart titled 'Sale contract' shows a 'Customer' and a 'Seller'. An arrow labeled 'Goods' points from the Seller to the Customer, and an arrow labeled 'Payments' points from the Customer to the Seller.
- Payment Terms:**
 - Предварительная оплата (Advance payment):** 'на срок (в днях):' 10, 'в объеме (в %):' 50,0%.
 - Отсрочка платежа (Payment deferral):** 'на срок (в днях):' 20, 'в объеме (в %):' 50,0%.
- Отгрузки (Shipments):** A table with columns 'Дата' and 'Примечания'.

Дата	Примечания
19.07.2001	1-я отгрузка
26.07.2001	2-я отгрузка
- Отгружаемые артикулы (Shipped articles):** A table with columns 'Наименование', 'Количество', and 'Цена'.

Наименование	Количество	Цена
2С2КВн	1446,2	618
3С18КВн	11249	618
3С19КВн	11235,8	618
8С103КВн	11460,3	618
8С115КВн	4082,5	618
- Итого по отгрузке (Total by shipment):**

Артикулов:	Сумма:	Рентабельность:
19	60259330	4,70%
- Итого по контракту (Total by contract):**

Отгрузок: 2	Сумма: 134498166,1	Рентабельность: 18,60%
-------------	--------------------	------------------------

Рисунок 6.32 – Экранная форма одной из схем контракта

Программный комплекс *Contract Analyzer* позволяет быстро и наглядно оценить влияние того или иного фактора на эффективность конкретного контракта в заданных условиях. Последовательность действий может быть такой [212, с. 502–507]:

- вводятся данные по контракту (ассортимент и объемы поставки продукции, сроки поставки, формы оплаты и т. д.), если это не было сделано ранее;
- указываются границы изменения различных факторов, влияющих на эффективность контракта;
- анализируются полученные зависимости, представленные в виде интервальных оценок (рисунок 6.33) [221, с. 59];
- производится корректировка условий заключения контракта;
- выводится на печать отчет, содержащий всю информацию по контракту, включая данные об эффективности контракта и условия, при которых они получены.

Программа *Contract Analyzer* дает возможность проводить анализ эффективности заключения контрактов продажи с учетом предоплаты или без нее. На рисунке 6.32 представлена основная рабочая форма для опера-

тора. Здесь осуществляется выбор схемы контракта, и в соответствии с ней задаются параметры срока предоплаты либо срока и процента предоплаты в случае схемы «Предоплата + Отсрочка». Для внесения деталей по датам и артикулам отгружаемой продукции заключаемого контракта используются пользовательские формы «Отгрузка» и «Добавить артикулы». При анализе рентабельности контракта пользователю дается возможность внести данные о состоянии внутренней экономики на момент заключения контракта через форму «Исходные данные для расчетов» [255, с. 61–62].

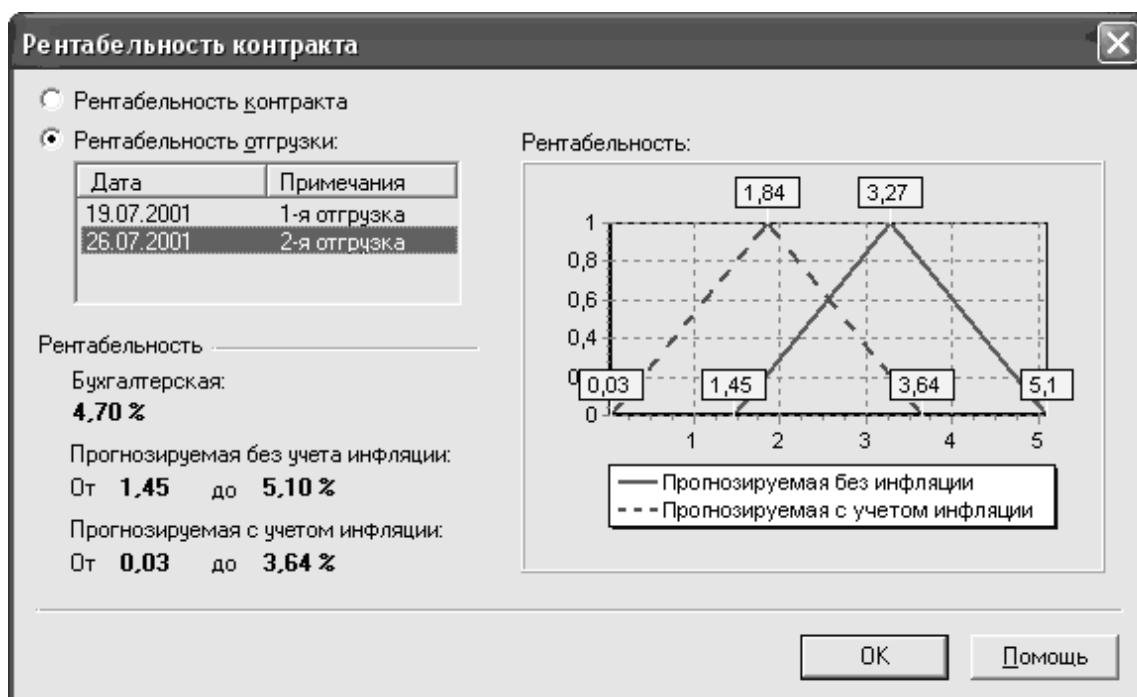
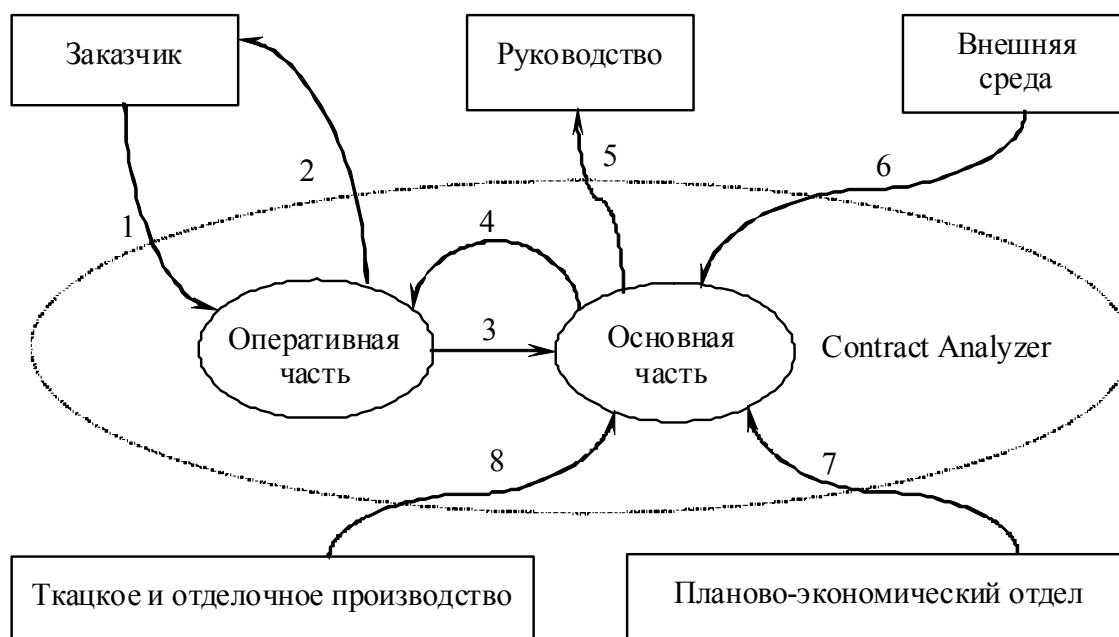


Рисунок 6.33 – Интервальные оценки эффективности контракта

6.4.3 Интеграция Contract Analyzer в информационную систему управления предприятием. Программное обеспечение *Contract Analyzer* интегрируется в информационную систему управления предприятием. Обобщенная структура информационных потоков для уровня прогнозирования эффективности контрактов представлена на примере предприятия текстильной промышленности (рисунок 6.34) [228, с. 34–35].

Запрос на продукцию поступает от заказчика в виде факса, письма, телекса и оформляется в виде заявки. Заявка содержит следующие данные: наименование заказчика, условия платежа, список требуемой продукции, срок исполнения, объем продукции, дата подачи заявки. Из внешних баз данных используются информация о клиентах, курсах валют, сведения по изделиям.



1 – имя заказчика, артикул, цена, объем, срок поставки, порядок расчетов, валюта; 2 – контракт; 3 – текущая информация о заключенных контрактах; 4 – цена артикула, себестоимость, плановая рентабельность, план выпуска артикула, индекс инфляции; 5 – основные показатели рентабельности контрактов; 6 – индекс инфляции, юридические ограничения; 7 – расчет себестоимости продукции, расчет цен, план производства и т. д.; 8 – объем выпуска продукции

Рисунок 6.34 – Обобщенная структура информационных потоков при определении эффективности контракта

Структура баз данных программного комплекса *Contract Analyzer* показана в [228, с. 34–35; 258, с. 209]. Таблицы с исходной информацией (Валюта, КурсВалюты, Артикул, Себестоимость, Заказчик) являются частью локального хранилища данных, формируемого из оперативных источников. Отсутствие централизованного подхода к хранению данных и, как следствие, разбросанность необходимой информации по локальным базам данных систем оперативного учета создают значительные трудности.

Выходными являются следующие данные:

- информация, формируемая в процессе анализа контракта;
- отпускная цена артикулов для заданной рентабельности;
- рентабельность артикула для договорной цены;
- прогнозируемая рентабельность контракта (договора) с учетом различных факторов;
- отчет по контракту (включая плановую и договорную рентабельность артикулов, плановую и договорную рентабельность контракта с учетом факторов времени, инфляции);
- база данных контрактов (содержит полную информацию по контрактам, включая показатели эффективности).

Программный комплекс предназначен для работы в локальной сети в многопользовательском режиме. Информация, с которой работает *Contract*

Analyzer, является критичной с точки зрения обеспечения нормальной работы предприятия. Ее искажение или потеря могут привести к значительным финансовым потерям, поэтому программное обеспечение комплекса удовлетворяет следующим требованиям:

- обладает средствами предотвращения несанкционированного доступа к информации;
- обеспечивает сохранность информации при сбоях аппаратного и системного программного обеспечения (исключая случаи физического уничтожения данных);
- предотвращает потерю информации в результате некорректных действий пользователя [255, с. 62].

7 Апробация метода поиска рациональных решений в ERP-системе управления промышленным предприятием

7.1 Состав задач, выбранных на апробацию

Задача 1. Проверка работоспособности, *исследование технических характеристик* программной системы «Экспериментатор» и исследование алгоритма случайного поиска на примере модели генетического алгоритма для поиска его лучших параметров. Поиск лучших параметров ГА проведен на примере исследования параметра алгоритма «*Размер популяции*». Функцией качества выбрана одна из известных функций Де Йонга (De Jong, 1975) [293]. При этом исследуются анализ длины переходного процесса и устойчивости результата, оценка погрешности моделирования работы ГА, анализ чувствительности откликов к изменению входных параметров, строится зависимость отклика «*Лучшее решение*» от изменения параметра «*Размер популяции*» [211, с. 3–9].

Задача 2. Одной из задач управления предприятием является задача *оптимизации распределения производственной нагрузки*. Особенно актуальной эта задача является для предприятий химической промышленности, что обусловлено следующими факторами. Себестоимость продукции химического производства определяется не только совершенством технологической схемы и степенью приближения режима процессов к оптимальному, но и в известной степени зависит от мощности производства. Эта зависимость носит в основном обратно пропорциональный характер. Расход средств по основным статьям, определяющим себестоимость продукта (расход сырья, расход энергетических ресурсов, амортизационные отчисления), с увеличением мощности производства при сохранении технологической схемы и режима проведения основных процессов имеет тенденцию к сокращению. Относительные потери сырья и готового продукта на разных стадиях технологической схемы обратно пропорциональны величине перерабатываемых потоков и уменьшаются с увеличением мощности производства почти линейно, приводя к снижению удельного расхода сырья на единицу готового продукта. На крупных предприятиях из-за больших объемов производства снижение себестоимости продукции даже на малую долю приводит к большой экономии в стоимостном выражении. Задача оптимизации распределения производственной нагрузки заключается в таком распределении ее между несколькими цехами, при котором себестоимость выпускаемой продукции для заданного ассортимента и объема выпуска будет минимальная.

Традиционно для решения подобных задач применяется корреляционно-регрессионный анализ. При этом требуется анализ большого количества данных, которые не всегда доступны. Кроме того, подобные модели не учитывают переходные процессы, неизбежно возникающие при изме-

нении производственной нагрузки на химическом предприятии, что значительно снижает их практическую ценность в условиях современной экономики, требующей быстрого реагирования на изменения на рынках продукции и сырья.

Возможным решением указанной проблемы может быть применение имитационных моделей, адекватно отражающих динамику технологических процессов. Однако в этом случае сложно, а порой и невозможно, представить целевую функцию в аналитическом виде, либо последняя имеет сложный вид (наличие множества локальных экстремумов, точек разрыва), делающий невозможным применение традиционных методов оптимизации [242, с. 47–51; 243, с. 44–46].

Задача 3. *Выявление свойств имитационных моделей контрактов* продажи, купли, на дилерское обслуживание и по товарному обмену для обоснования области применения аналитических моделей оценки эффективности указанных контрактов, поиск оптимальных решений при заключении контрактов в маркетинговой деятельности промышленного предприятия, оценка влияния параметров контрактов на финансово-экономическое состояние предприятия. В соответствии с нормативными документами для оценки финансово-экономического состояния приняты следующие показатели: K_1 – коэффициент текущей ликвидности; K_2 – доля собственных оборотных средств в общей их сумме; K_3 – коэффициент обеспеченности собственными средствами [84, с. 13; 85, гл. 2].

Задача 4. Промышленное объединение представляется многоуровневой системой, в которой решение нижестоящего уровня зависит от задания со стороны вышестоящего уровня, а решение вышестоящего уровня, в свою очередь, зависит от откликов нижестоящего уровня. Для такой системы важным является описание нижестоящего уровня как взаимодействие взаимосвязанных подсистем, каждая из которых преследует собственные цели. При этом основное внимание обращено на внутриуровневые и межуровневые взаимодействия. Такой подход обеспечивает при моделировании целостность рассмотрения сложной системы, удерживая в поле зрения всю систему в целом для решения задачи управления с позиции системной цели.

На самом высоком уровне рассматриваются вопросы распределения ресурсов и взаимовлияния между подсистемами с точки зрения всей системы в целом и вопрос об эффективности и допустимости ее режимов. Наиболее детально при моделировании рассматриваются только те подсистемы, в которых имеется существенное снижение степени эффективности и допустимости режимов. *Выбор типа координации процесса планирования производственной программы* позволяет выбрать оптимальную стратегию управления промышленным производством [216, с. 334–343; 217, с. 239–245].

Задача 5. Чтобы оценить возможность развития ПТКИ Belsim для решения проблем в смежных предметных областях ставится задача *разра-*

ботки библиотеки моделей для исследования информационной системы промышленного предприятия. Библиотека должна включать ИМ бизнес-процесса планирования продаж и производства, ИМ информационной системы производственного процесса, ИМ бизнес-процесса управления запасами ресурсов, ИМ функционирования информационной системы в локальной вычислительной сети [79, с. 20–27; 80, с. 37–41; 214, с. 74–79; 220, с. 136–138; 246, с. 299–301; 266, с. 415–421].

7.2 Исследование параметров генетического алгоритма для поиска лучшего решения

7.2.1 Применение программного комплекса «Экспериментатор» для исследования параметров генетического алгоритма.

Генетический алгоритм (ГА) – это простая модель эволюции в природе, реализованная в виде компьютерной программы. В нем используются аналоги механизма генетического наследования и естественного отбора. При этом для лучшего прослеживания аналогии сохраняется биологическая терминология в упрощенном виде. Основное использование ГА – решение оптимизационных проблем различного рода, причем качество получаемого решения сильно зависит от качества настройки алгоритма [174, с. 39–45; 308, с. 811–818; 249, с. 332–334].

Входные параметры ГА составляют множество $X = \{x_i\}$, $i = 1, \dots, 10$. Здесь x_1 – способ кодирования гена в генотипе; x_2 – длина генотипа; x_3 – размер популяции; x_4 – количество родителей; x_5 – механизм отбора родителей; x_6 – вероятность кроссовера; x_7 – тип кроссовера; x_8 – количество точек разрыва при фиксированном кроссовере; x_9 – вероятность мутации; x_{10} – механизм отбора в новое поколение. Выходные параметры ГА составляют множество откликов $Y = \{y_i\}$, $i = 1, \dots, 5$. Здесь y_1 – значение результата «*Лучшее решение*»; y_2 – время работы алгоритма на некоторой эталонной машине; y_3 – количество вызовов функции качества; y_4 – среднее значение приспособленности поколения; y_5 – дисперсия приспособленности поколения. Функции, на которых исследуются ГА, составляют множество условий $Z = \{z_i\}$, $i = 1, \dots, k$. Здесь k – количество функций, на которых будет производиться исследование для каждого набора фиксированных значений множества X [235, с. 156; 236, с. 204–208].

Проверка работоспособности программного комплекса проведена на примере исследования параметра ГА «*Размер популяции*». При этом выбрана одна из функций Де Йонга (De Jong, 1975) [293]:

$$F(x_1, x_2) = 100(x_1^2 - x_2) + (1 - x_1)^2, x_i \in [-2,048; 2,048], i = 1, 2. \quad (7.1)$$

Минимум $F_{\text{опт}}$ данной функции находится в точке (1,0; 1,0) и равен нулю:

$$F_{\text{опт}}(1,0; 1,0) = 0.$$

Значения параметров для этой функции предложены в таблице 7.1 [293].

Таблица 7.1 – Исследуемые значения параметров ГА

Параметры	Значение
Размер популяции	50–100
Число родителей	Равно размеру популяции
Тип кроссовера	Двухточечный
Вероятность кроссовера	0,6–0,9
Мутация	Простая
Вероятность мутации	0,01–0,001
Отбор родителей	Рулеточное колесо по приспособленности
Отбор в следующее поколение	Элитный

7.2.2 Этапы исследования ГА. С помощью программной системы «Экспериментатор» исследуются предложенные в [293] значения параметра «Размер популяции» на их достаточность.

Этапы решения задачи:

– определение *достаточного числа параллельных запусков модели* для статистической значимости результатов. Этап может проводиться по различным методикам (например, заранее задано значение 500 запусков [114]);

– *анализ длины переходного процесса и устойчивости* результатов моделирования для определения числа поколений, после которого заканчивается переходный процесс;

– *оценка погрешности* моделирования на найденном числе поколений для предложенных в [293] значений – получение контрольного значения вероятности того, что алгоритм найдет оптимальное решение;

– *анализ чувствительности* откликов к изменению параметра «Размер популяции»;

– *эксперимент* (таблица 7.2) на множестве значений параметра «Размер популяции» с целью получить зависимость вероятности оптимального решения от размера популяции и числа родителей;

– проверка достаточности предлагаемых в [293] значений параметра «Размер популяции».

Для проведения экспериментов выбраны значения параметров, сведенные в таблицу 7.2 [211, с. 6].

Таблица 7.2 – Основные параметры эксперимента

Название параметра плана эксперимента	Значение
Входные параметры X	См. таблицу 7.1
Исследуемые отклики Y	Лучшее решение
	Среднее значение приспособленности
	Дисперсия приспособленности поколения
Целевая функция Z	См. формулу (7.1)
Число параллельных запусков	500

7.2.3 Анализ длины переходного процесса и устойчивости результата. Для анализа выбираются наиболее характерные значения параметров модели [114; 254, с. 45] – максимальные значения из предложенных в таблице 7.1. Сбор данных проводится через каждые пять поколений.

С помощью приложения «*ExperimentDesigner*» составляется план эксперимента на основе значений, представленных в таблице 7.2.

Эксперимент проводится с использованием консольного приложения «*Experimenter*», и его результаты загружаются с помощью макроса «*ExperimentData*» в пакет STATISTICA. В этом приложении по полученным данным строятся зависимости изменения откликов от номера поколения. Для отклика «*Лучшее решение*» такая зависимость представлена на рисунке 7.1.

По этим зависимостям определяется момент окончания переходного процесса. В качестве границы переходного процесса выбирается 50 поколений. Переходный процесс у алгоритма заканчивается на 40–50-м поколении (см. рисунок 7.1) [211, с. 6–7].

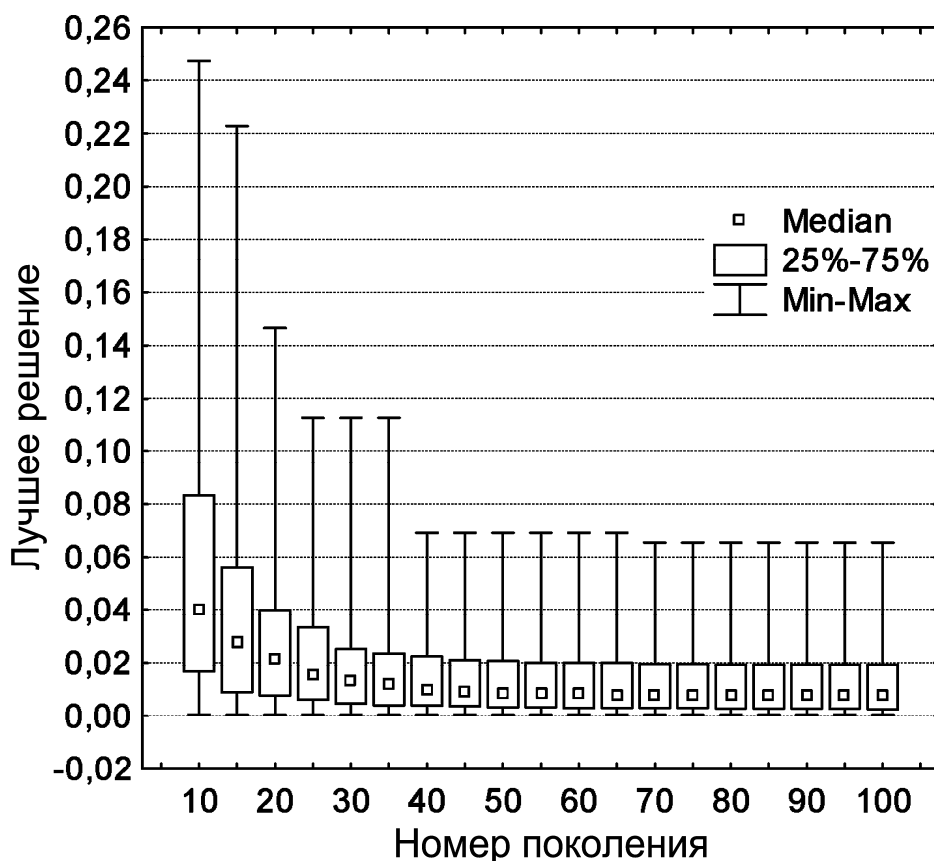


Рисунок 7.1 – Зависимость отклика «Лучшее решение» от номера поколения

7.2.4 Оценка погрешности моделирования работы ГА. Результатом моделирования работы ГА является лучший индивидуум (отклик «*Лучшее решение*»). В связи с этим определение погрешности моделиро-

вания сводится к оценке погрешности данного отклика по распределению значений отклика. Значения параметров берутся из предыдущего этапа. Для увеличения статистической значимости количество параллельных запусков удваивается [114].

Сбор данных проводится после 50 поколений, т. е. с учетом длины переходного процесса и устойчивости результата. Используя приложение «*ExperimentDesigner*», задают план эксперимента [64]. При помощи программы «*Experimenter*» проводится эксперимент, и его результаты загружаются макросом «*ExperimentData*» в пакет STATISTICA, где по результатам эксперимента строится распределение частот значений отклика «*Лучшее решение*», приведенное в таблице 7.3 (y_1 – значение отклика «*Лучшее решение*»).

Таблица 7.3 – Распределение значений отклика «Лучшее решение»

Интервал значений	Количество значений	Кумулятивное количество	Процент от общего числа значений	Кумулятивный процент
$0,0001 \leq y_1 < 0,0501$	965	965	96,5	96,5
$0,0501 \leq y_1 < 0,1001$	33	998	3,3	99,8
$0,1001 \leq y_1 < 0,1501$	2	1000	0,2	100,0
$0,1501 \leq y_1 < 0,2001$	0	1000	0	100,0
Пропущенных	0	1000	0	100,0

Из распределения значений отклика «*Лучшее решение*» видно, что в 95 % случаев отклонение результата от минимума функции не превышает 0,05, т. е. решение, полученное с помощью ГА, будет $F_{ГА} = F_{\text{опт}} \pm \Delta = 0 \pm 0,05$. Ошибка работы алгоритма случайного поиска не превышает 5 %.

7.2.5 Анализ чувствительности откликов к изменению входных параметров. Анализ чувствительности проводится для определения степени влияния изменения входного параметра на изменение отклика. Чувствительность используется для задания точности разбиения интервала изменения входного параметра при проведении факторных экспериментов с целью получения искомых зависимостей.

План эксперимента задается в приложении «*ExperimentDesigner*» на уровнях -1 ; 0 ; $+1$, соответствующих значениям параметра «*Размер популяции*» 10; 100; 500. Остальные параметры эксперимента не изменяются.

С использованием программы «*Experimenter*» проводится эксперимент, его результаты загружаются с помощью макроса «*ExperimentData*» в пакет STATISTICA. Результаты анализа, построенные макросом, приведены в таблице 7.4.

Изменение входных параметров и откликов рассчитывается по формулам (5.7) и (5.9) соответственно.

Таблица 7.4 – Анализ чувствительности отклика «Лучшее решение» от параметра «Размер популяции»

Статистический параметр	Значение
Доверительный интервал	0,05
Среднее отклика на уровне +1	0,005
Среднее отклика на уровне –1	0,307
Значение t -критерия*	–14,8 ($p = 0$)**
Значение F -критерия*	17200 ($p = 0$)**
Изменение параметра Δ_X , %	192,2
Изменение отклика Δ_Y , %	193,7
Отношение изменений Δ_Y/Δ_X	1,008

Примечание – * – проверка гипотезы о существовании зависимости между входным параметром «Размер популяции» и откликом «Лучшее решение» [37]; ** – уровень значимости (p -уровень) или вероятность ошибочно отклонить предложение об отсутствии данной зависимости

Как видно из таблицы 7.4, отношение изменения параметра «Размер популяции» к изменению отклика «Лучшее решение» приблизительно равно 1:1. При абсолютной погрешности $\Delta = 0,05$ значение точности изменения отклика выбирают вдвое меньшее: $0,05/2 = 0,025$. Учитывая приведенные выше значения, шаг h_X изменения параметра «Размер популяции» рассчитывается по формуле

$$\frac{|\bar{Y}_j^{(+1)}| - |\bar{Y}_j^{(-1)}|}{|X_i^{(+1)}| - |X_i^{(-1)}|} \cdot \frac{1}{1} = \frac{0,025}{h_X}, \quad (7.2)$$

где $\bar{Y}_j^{(+1)}$, $\bar{Y}_j^{(-1)}$ – средние значения j -го отклика на соответствующем уровне входного параметра X_i .

Значение шага h_X получено из формулы (7.2):

$$h_X = \frac{0,025 \cdot (500 - 10)}{0,307 - 0,00491} \rightarrow 50 \text{ особей.}$$

Таким образом, шаг изменения параметра «Размер популяции» выбирается равным $h_X = 50$ особей.

При помощи вычисленного значения шага h_X получают значения размера популяции на уровнях факторного эксперимента от 1 до 10; от 50 до 500 с шагом $h_X = 50$ особей. Значения остальных параметров не изменяются. С использованием полученных данных в приложении «*ExperimentDesigner*» задается план эксперимента.

Далее, используя программу «*Experimenter*», проводят эксперимент и его результаты загружают с помощью макроса «*ExperimentData*» в пакет STATISTICA. С помощью средств приложения STATISTICA по полученным данным строится зависимость изменения отклика от искомого

параметра, представленная на рисунке 7.2 [221, с. 95].

Как видно из полученного графика (границы 5–95 %), предложенная граница размера популяции, равная 50 особям, в 90 % гарантирует получение результата в пределах $F_{ГА} = 0 \pm 0,1$, а размер 100 особей – в пределах $F_{ГА} = 0 \pm 0,05$, т. е. при отклонении результата $\Delta = 0,05$ размер популяции должен быть не менее 100 (гарантированно 150) особей при числе поколений 50, являющемся границей переходного процесса [211, с. 9].

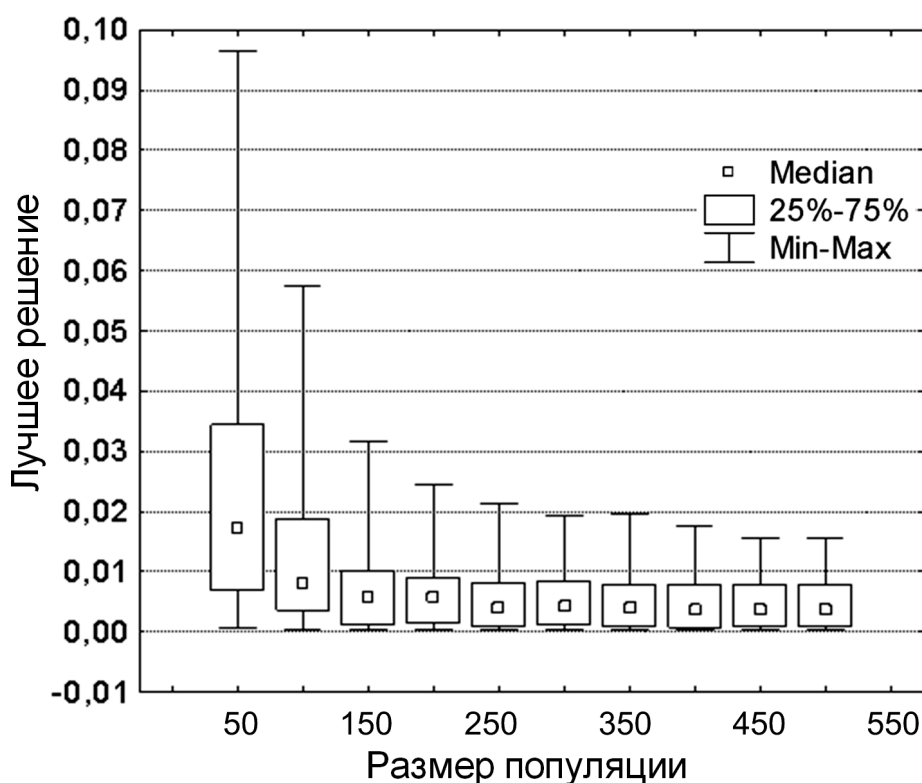


Рисунок 7.2 – Зависимость отклика «Лучшее решение» от изменения параметра «Размер популяции»

7.2.6 Использование средств оптимизации и распределенных вычислений в ПТКИ BelSim для решения задачи о грузоперевозках на предприятии. ПТКИ *BelSim* позволяет моделировать задачи не только для проблем промышленного предприятия, но и для смежных предметных областей. Типовой для оптимизационных задач является задача об оптимизации грузоперевозок на автотранспортном предприятии [10, с. 110–112], которая формулируется следующим образом.

Предприятие имеет N заказов на перевозку грузов. Каждый заказ $j = 1, \dots, N$ характеризуется объемом O_j , сроком выполнения Sr_j и координатой на карте местности, при этом $\|S_{fn}\|$ – матрица расстояний между базой и всеми пунктами назначений; $\|V_{fn}\|$ – матрица средних скоростей между всеми пунктами назначения и базой; $f, h = 0, \dots, N$.

Автотранспортное предприятие предоставляет услуги по грузопере-

возкам в территориальных пределах, задаваемых картой местности. Автопарк предприятия представлен некоторым количеством грузовых машин M . Каждая машина $i = 1, \dots, M_c$ имеет следующие характеристики: грузоподъемность T_i , расходы на топливо TO_i и оплату времени работы водителя Z_i . Для полного учета времени работы водителя существуют нормативы на скорость погрузки Vp и разгрузки Vr . Заказ всегда выполняется полностью за одну поездку. При превышении сроков выполнения некоторого заказа с предприятия взимается пеня Zpr . Требуется составить план грузоперевозок ($\|A_{ig}\|$ – матрица распределения заказов по машинам, $g = 1, \dots, 2 \cdot N + 1$; $i = 1, \dots, M$), при котором выполняются все заказы и при этом общие затраты автотранспортного предприятия минимальны:

$$\sum_{i=1}^M \left(\sum_{j=1}^{2 \cdot N} S_{A_{ij} A_{ij+1}} \cdot TO_i + \left(\sum_{j=1}^{2 \cdot N} S_{A_{ij} A_{ij+1}} / V_{A_{ij} A_{ij+1}} + \sum_{j=1}^{2 \cdot M_c + 1} O_{A_{ij}} \cdot (Vp + Vr) \right) \cdot Z_i \right) + Zpr \cdot Tpr \rightarrow \min. \quad (7.3)$$

Для решения задачи оптимизации грузоперевозок использована реализация ГА в подсистеме оптимизации ПТКИ BelSim.

Применение генетического алгоритма с параметрами по умолчанию с использованием матрицы $\|A_{ig}\|$ для представления решения дает отрицательный результат. Требуется более эффективный способ кодирования решения задачи и рациональный выбор параметров генетического алгоритма. Проведены исследования следующего набора параметров генетического алгоритма:

$$GA = \langle \varepsilon_f, L_g, NI, NG, \delta_r(NI), p_m, p_k, M_{bp}, D_{bp}, O_t \rangle, \quad (7.4)$$

где ε_f – точность представления параметра функции;

L_g – длина генотипа, $L_g = M + N$;

NI – размер популяции;

NG – количество поколений;

$\delta_r(NI)$ – доля родителей от размера популяции;

p_m – вероятность мутации;

p_k – вероятность кроссовера;

M_{bp} – математическое ожидание количества точек разрыва;

D_{bp} – дисперсия количества точек разрыва;

O_t – отбор в следующее поколение, $O_t = \langle \lambda_{om}, \{x_{om}\} \rangle$ (где λ_{om} и $\{x_{om}\}$ – тип и параметры оператора отбора родителей соответственно).

В целях сужения пространства поиска решение задачи представляет-

ся в виде N наборов вида $\langle M_{z_i}, p_i \rangle$, где i – номер заказа, $i = 1, \dots, N$; M_i – номер машины, которая должна выполнять i -й заказ, $M_i \in M_{z_i}$ (M_{z_i} – множество машин с подходящей для i -го заказа грузоподъемностью); p_i – приоритет, определяющий порядок выполнения заказов.

В результате пространство поиска решений значительно сокращается за счет исключения недопустимых вариантов, когда один и тот же заказ может доставляться более одного раза или не доставляться вовсе, а также из-за возможного превышения максимальной грузоподъемности машин. Для вычисления значения целевой функции выполняется преобразование указанного представления решения задачи в исходное.

Для выбора лучших значений указанных выше параметров ГА в зависимости от длины генотипа, которая определяется размерностью задачи (количеством машин и заказов), проведены исследования по указанной выше методике. В качестве функции качества использована целевая функция задачи, взятая со знаком минус [10, с. 110–112].

Задача о грузоперевозках принята для оценки эффективности распределенных вычислений. В качестве экспериментальной базы использована локальная вычислительная сеть из трех компьютеров с основными характеристиками, представленными в таблице 7.5.

Таблица 7.5 – Экспериментальная база для исследования распределенных вычислений в ПТКИ BelSim

Компьютер	Процессор	ОЗУ, МБ	Сетевая карта	Операционная система
Compaq Evo N610c	Intel Pentium 4 / 2,0 ГГц	512	Ethernet / 100 Мбит/с; TCP/IP	Windows XP Professional (версия 2002)
Compaq Evo N620c	Intel Pentium М / 1,4 ГГц			
HP Compaq nc 6000	Intel Pentium М / 1,6 ГГц			

Для оценки эффективности проведения эксперимента с распределенными вычислениями определена величина T_{ij} – длительность i -го эксперимента на j -м компьютере (i – порядковый номер эксперимента, $i = 1, \dots, 3$; j – идентификатор компьютера, $j = PC1, \dots, PC3$) при заданном количестве прогонов модели в i -м эксперименте N_i . Результаты проведенного исследования представлены в таблице 7.6.

Эффективность распределения вычислений оценена по следующему выражению

$$T_{i\max} = \max_{\forall PC_j} (T_i(PC_j)),$$

где максимальное значение $T_{i\max}$ продолжительности эксперимента определяется наибольшим значением $T_i(PC_j)$ для наименее производительного компьютера.

Таблица 7.6 – Результаты эксперимента с распределенными вычислениями в ПТКИ BelSim

Количество прогонов модели N_i	Продолжительность эксперимента T_{ij} , с					
	на одном PC	на двух PC		на трех PC		
	T1(PC1)	T2(PC1)	T2(PC2)	T3(PC1)	T3(PC2)	T3(PC3)
60	21,99	12,22	12,81	8,17	8,24	8,91
100	35,73	19,11	19,90	13,26	13,34	14,38
200	71,69	35,94	37,28	26,18	26,10	28,07
400	146,64	73,14	76,29	54,30	54,44	58,37
600	227,89	108,81	114,38	79,46	79,61	86,12

Например, при сравнении экспериментов с $N_i = 400$ на одном и двух компьютерах продолжительность прогонов модели сокращается в 1,9, т. е. почти в 2 раза. На трех компьютерах продолжительность эксперимента в сравнении с одним компьютером сокращается в 2,5 раза. В соответствии с сетевым законом Амдала [203, с. 12–13] при общей эффективности вычислений не наблюдается прямо пропорционального снижения длительности экспериментов с увеличением числа используемых компьютеров в сети [248, с. 137–142].

7.3 Решение задачи оптимизации производственного процесса

7.3.1 Структура исследуемого производственного объединения.

На крупных химических предприятиях, включающих в своей структуре несколько заводов [241, с. 56], расход средств по основным статьям, определяющим себестоимость продукта (расход сырья, расход энергетических ресурсов, амортизационные отчисления), с увеличением мощности производства при сохранении технологической схемы и режима проведения основных процессов имеет тенденцию к сокращению. В то время как расход сырья на стадии химического превращения практически не зависит от размера производственной аппаратуры, относительные потери сырья и готового продукта (например, диметилтерефтолата (ДМТ) на заводе органического синтеза (ЗОС), рисунок 7.3), обратно пропорциональные величине перерабатываемых потоков на других стадиях технологической схемы, уменьшаются с увеличением мощности производства почти прямолинейно, приводя к снижению удельного расхода сырья на единицу готового продукта. Если производство многопоточное, то для заданной производственной программы ставится задача определения оптимальной загрузки цехов предприятия [242, с. 47–51].

В соответствии со стандартом MRPII система управления предприятием должна включать такие функции, как планирование производственных мощностей, планирование производственных операций, управление на уровне производственных цехов. С ростом сложности производственных процессов возникает проблема определения оптимального управления

производством и планирования выпуска продукции. Следует подчеркнуть, что получившие широкое распространение и признание как в отечественной, так и в зарубежной литературе методики расчета производственной мощности предприятия, цеха, участка основаны либо на принципе прямого счета, либо на принципе оптимизации загрузки оборудования с помощью линейно-программной модели. Расчеты на их основе выполняются для целей текущего (годового, квартального) планирования выпуска продукции. Используемая для этого информация является не только существенно агрегированной, но и не учитывает непрерывность технологического процесса, его цикличность и партионный характер движения предметов труда от одной производственной операции к другой.

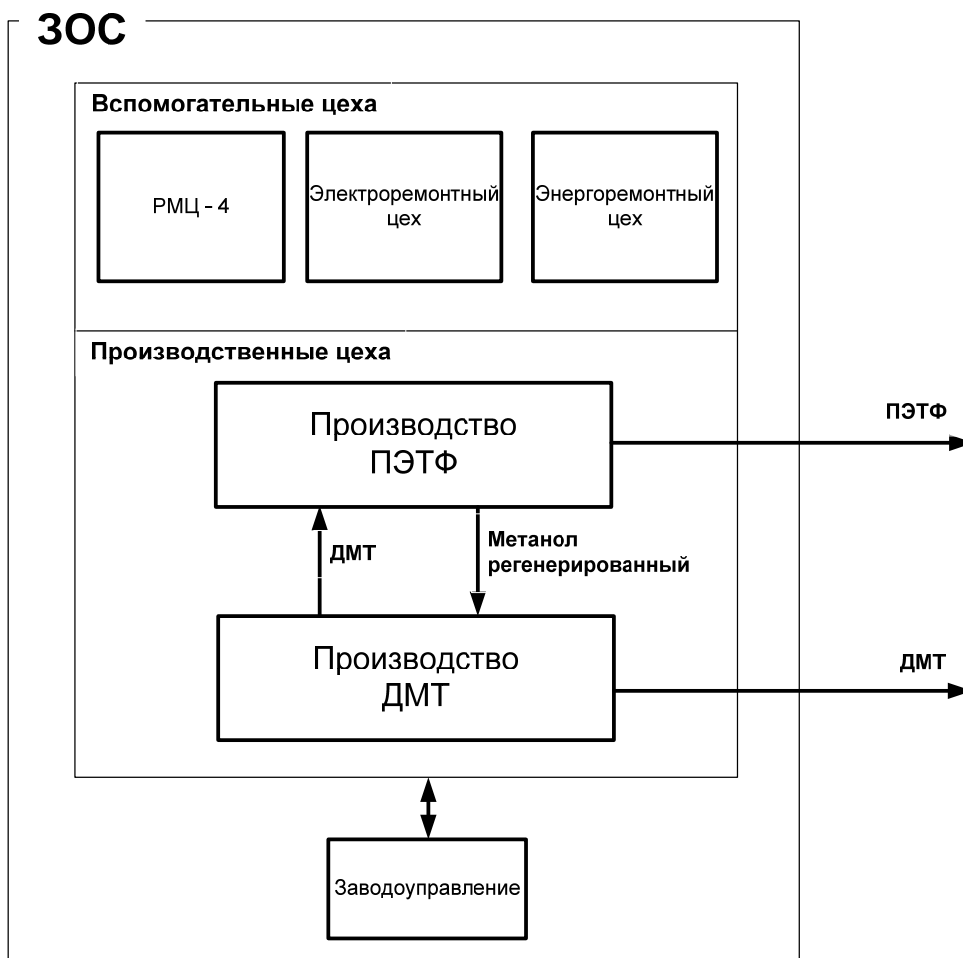


Рисунок 7.3 – Структура завода органического синтеза

Таким образом, рассматриваемая проблема требует учета существенных признаков непрерывного производственного процесса и показателей, адекватно характеризующих его динамику. В этой связи можно отметить, что одним из возможных подходов к ее решению является имитационное моделирование непрерывного производственного процесса. Эффективным средством построения имитационных моделей на основе процесс-

ного способа является программно-технологический комплекс имитации (ПТКИ) BelSim.

На рисунке 7.4 выделены основные операции производственного процесса. В результате проведенного анализа определено, что производственный процесс состоит из четырех производственных операций: Оп1, Оп2, Оп3, Оп4. Операции Оп2, Оп3, Оп4 преобразуют сырье С2, С3, С4 и дополнительное сырье ДС2, ДС3 и ДС4 в ресурсы Р2, Р3, Р4 соответственно, а операция Оп1 производит из ресурсов Р2, Р3, Р4, сырья С1 и дополнительного сырья ДС1 готовую продукцию П1. Каждая из производственных операций Оп1, Оп2, Оп3, Оп4 характеризуется своей нормой потребления сырья и ресурсов, а также и своей нормой выпуска продукции.

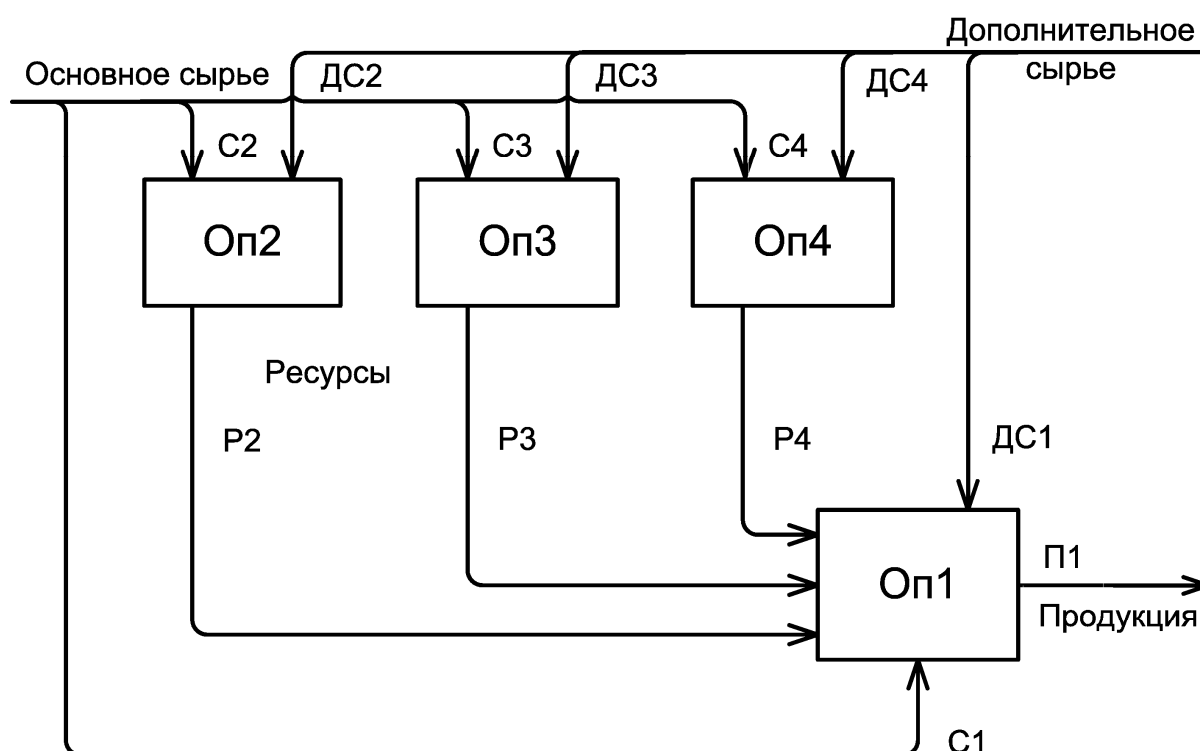


Рисунок 7.4 – Схема производственного процесса

Исходя из приведенной схемы (см. рисунок 7.4), можно сделать вывод, что себестоимость продукции П1 определяется загруженностью операций Оп2, Оп3, Оп4 и себестоимостью производимых ими ресурсов (Р2, Р3, Р4). Таким образом, меняя загруженность этих операций и определяя себестоимость продукции, при заданной норме выпуска можно определить оптимальные параметры производственного процесса.

7.3.2 Технология обработки исходных данных для моделирования. Удельные нормы расхода основного сырья для всех операций определяются технологией производства и являются постоянными. Иначе обстоя-

ит дело с дополнительным сырьем. Удельный расход дополнительного сырья, как известно из опытных исследований, изменяется по обратно пропорциональному закону в зависимости от объема выпуска (выработки) продукции. Для нахождения зависимостей удельных норм от выработки на предприятии собирается статистическая информация, которую необходимо проанализировать и обработать. Такой анализ и обработка проведены в пакете STATISTICA [223, с. 237]. На рисунке 7.5 представлены данные по удельному расходу дополнительного сырья $ДС2_i \in ДС2$ для операции Оп2 в зависимости от суточной выработки за последние 5 лет [216, с. 338].

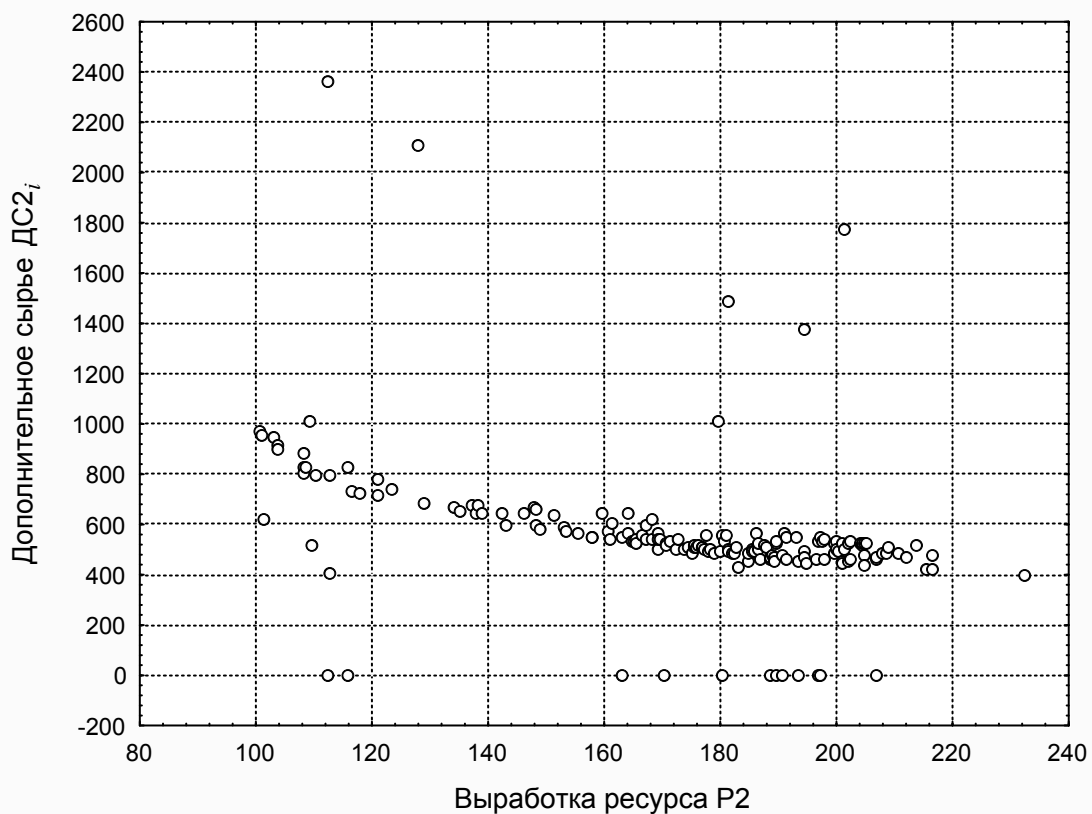


Рисунок 7.5 – Удельный расход дополнительного сырья $ДС2_i$ для операции Оп2 в зависимости от суточной выработки

На графике (см. рисунок 7.5) явно просматривается некоторая зависимость, а также присутствуют некоторые так называемые «выбросы», которые обусловлены различными причинами: ошибками при измерении, ошибками при снятии показаний с контрольно-измерительных приборов, ошибками при внесении информации в базу данных и т. д. Для того чтобы «очистить» данные от выбросов, необходимо задаться требуемым коэффициентом корреляции. Коэффициент корреляции Пирсона находят для линейных зависимостей, следовательно, необходимо перейти от обратно пропорциональной зависимости к линейной. Это можно сделать, построив зависимость не от выработки, а от величины, обратной выработке [216, с. 334–343].

На основании экспертных оценок сделан вывод о достоверности или ошибочности этих данных. В большинстве случаев разброс обусловлен ошибками измерений или ошибками внесения данных в базу данных. В пакете STATISTICA существует инструмент «щетка», при помощи которого можно удалить очевидные выбросы. На рисунке 7.6 представлена зависимость удельного расхода дополнительного сырья $ДС2_i$ от величины, обратной суточной выработке, но уже без выбросов [387, с. 205–206].

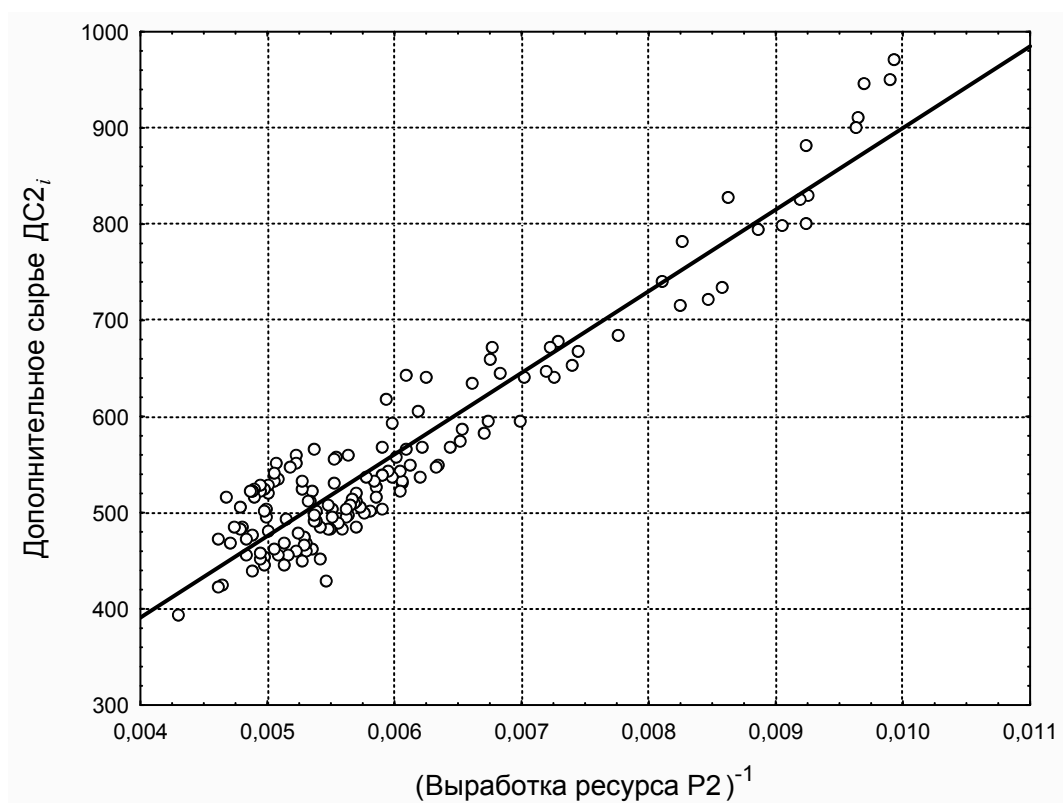


Рисунок 7.6 – Удельный расход дополнительного сырья $ДС2_i$ от величины, обратной суточной выработке ресурса P2 без выбросов

Для данных на рисунке 7.6 коэффициент корреляции составляет 0,95, что говорит о наличии достаточно сильной зависимости между двумя параметрами. Также STATISTICA автоматически выводит уравнение регрессии для данной зависимости. Можно построить график зависимости расхода дополнительного ресурса $ДС2_i$ от выработки уже без учета выбросов. Такой график представлен на рисунке 7.7 и аппроксимируется функцией вида

$$y = \frac{K}{x} + B, \quad (7.5)$$

где коэффициенты K и B принимаются из уравнения регрессии, построенного в пакете STATISTICA.

Сплошной линией на рисунке 7.7 показана линия регрессии. Таким образом, получена зависимость удельного расхода дополнительного сырья ДС_{2i} от суточной выработки ресурса Р2 для операции Оп2. Аналогичным образом находятся остальные зависимости удельного расхода каждого дополнительного сырья от суточной выработки для операций Оп2, Оп3, Оп4 (см. рисунок 7.4).

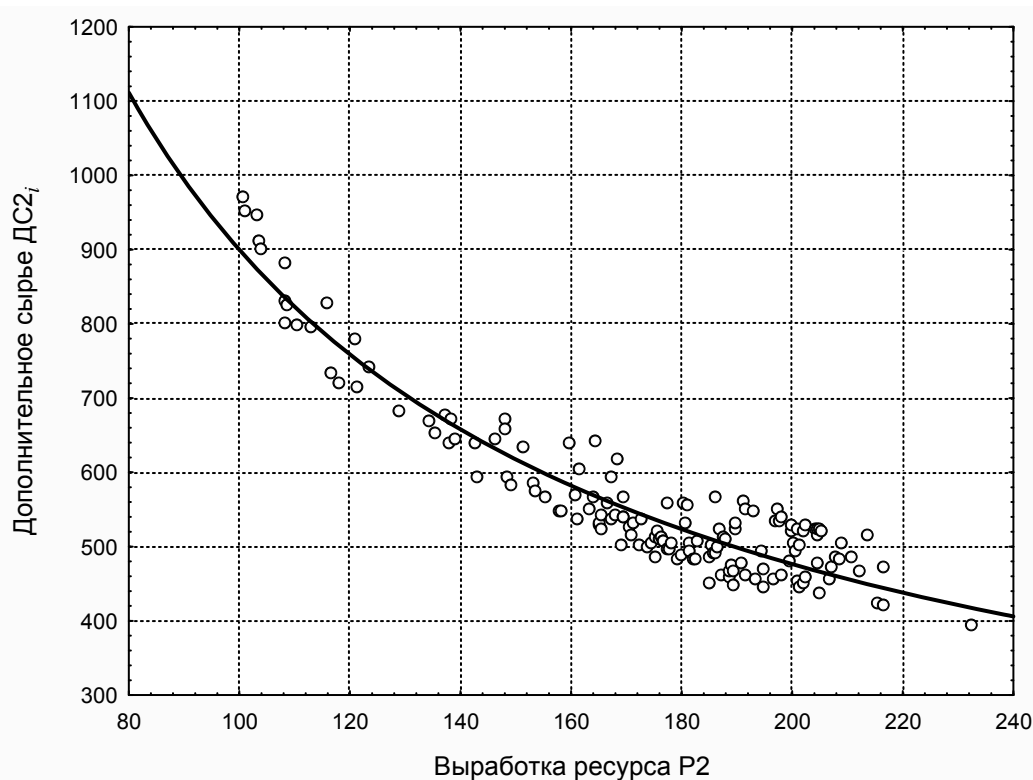


Рисунок 7.7 – Зависимость удельного расхода дополнительного сырья ДС_{2i} от суточной выработки ресурса Р2 без учета выбросов

На исследуемом предприятии основная информация о себестоимости продукции и нормах расхода сырья хранится в информационной системе 1С: Предприятие, а статистическая информация о нормах расхода дополнительного сырья – в программе MS Excel.

7.3.3 Имитационная модель непрерывного производственного процесса. Имитационная модель производственного процесса реализуется на основе процессного способа моделирования. Выделены следующие необходимые процессы:

- моделирующий работу производственной операции;
- определяющий состояние производства и собирающий статистические данные.

Алгоритм процесса, моделирующего производственную операцию, состоит из трех стадий: «Запуск», «Производство», «Окончание». Для модели-

рования непрерывного производства процесс разбивается на n отрезков. При выполнении такой процесс будет запускаться в n раз чаще, однако потреблять ресурсов и производить продукции он будет в n раз меньше. Порции произведенной продукции при запуске процесса будут сдвигаться на один отрезок вперед, а только что произведенная продукция будет помещаться в первый отрезок. Продукция $\Pi_{\text{гот}}$ из последнего отрезка Отр_n будет отгружаться на склад. Пример такого движения представлен на рисунке 7.8.

На стадии «Запуск» проверяется наличие ресурсов на складе. Если они есть, то происходит их выделение и добавление порции готовой продукции в первый отрезок; если сырья на складе недостаточно, то первый отрезок помечается как пустой.

На стадии «Производство» (рисунок 7.9) организуется движение продукции из одного отрезка в другой. Также на этой стадии моделируются сбои и отказы. Для этого в начале работы модели определяется время наступления отказа и сбоя для каждой операции, и при наступлении этого момента происходит задержка производства с возможной потерей продукции.

На стадии «Окончание» полученная продукция отгружается на склад.

Таким образом, выполняя рассмотренные стадии в порядке «Запуск», «Производство», «Окончание», получен алгоритм, полностью моделирующий непрерывную производственную операцию с учетом сбоев и отказов. Следует заметить, что при числе отрезков, равном единице, данный алгоритм будет моделировать дискретную производственную операцию [227, с. 135–137].

Алгоритм процесса, определяющего состояние производства и собирающего статистические данные, состоит из двух стадий: анализа текущих данных модели и вывода статистики.



Рисунок 7.8 – Движение ресурсов по отрезкам при выполнении производственной операции



Рисунок 7.9 – Алгоритм стадии «Производство»

7.3.4 Исследование имитационной модели непрерывного производственного процесса. При исследовании ИМ рассмотрен производственный процесс (ПрП), представленный на рисунке 7.4. Основным фактором, влияющим на длину переходного процесса, является полный запуск производства, т. е. когда все производственные операции (ПрО) входят в стационарный режим работы. Для определения длины переходного процесса выбран параметр «Себестоимость готовой продукции». Себестоимость продукции стабилизируется примерно через 135 единиц модельного времени, т. е. после полного начала работы всех ПрП. Статистика, которая собирается во время переходного процесса, исключается. Для этого процесс сбора статистики ИМ запускается через 135 единиц модельного времени.

Рациональный выбор параметров ПрП может быть проиллюстрирован следующим образом. Пусть для заданного непрерывного ПрП необходимо определить минимальную себестоимость выпускаемой продукции при заданной норме выпуска 600 единиц. Схема ПрП приведена на рисунке 7.4, а исходные данные реального ПрП – в таблице 7.7, где коэффициенты K и B (см. формулу (7.5)) определяются из уравнения регрессии, построенного по эмпирическим данным для определения объема j -го вида ресурса, потребляемого i -й операцией.

В таблице 7.8 указана себестоимость основного и дополнительного сырья. Ограничения на выработку ресурсов приведены в таблице 7.9. Операции Оп2, Оп3, Оп4 производят ресурсы Р2, Р3, Р4 соответственно

(см. рисунок 7.4). Определяется оптимальная загрузка операций (выработка ресурсов) для получения минимальной себестоимости продукции. Для операции Оп2 изменяется значение уровня загрузки от минимального до максимального возможного, а для операций Оп3 и Оп4 выбирается средний уровень загрузки. Строится зависимость себестоимости продукции от выработки ресурса Р2 (рисунок 7.10).

Исходя из полученных данных выбирается рациональное значение загрузки операции Оп2, соответствующее минимальной себестоимости выпускаемой продукции, и фиксируется. Для операции Оп3 также меняется загрузка, но уже при найденном значении ресурса Р2, и получается зависимость, представленная на рисунке 7.11.

Аналогичным образом строится зависимость себестоимости продукции от выработки ресурса Р4, но при фиксированных значениях ресурса Р2 и Р3 (рисунок 7.12). Так как себестоимость продукции минимальна и при максимальной выработке ресурса Р3, и при максимальной выработке ресурса Р4, то для выбора оптимальных значений строится зависимость себестоимости одновременно от двух ресурсов.

Таблица 7.7 – Исходные данные реального производственного процесса

Основное сырье	Количество		Основное сырье	Количество	
Операция-1			Операция-2		
Сырье-4	10,00		Сырье-1	0,62	
			Сырье-2	0,38	
			Сырье-19	2,00	
			Сырье-4	14,80	
			Сырье-7	0,10	
			Сырье-8	0,20	
Дополнительное сырье	<i>B</i>	<i>K</i>	Дополнительное сырье	<i>B</i>	<i>K</i>
Дополнительное сырье-11	15,53	6156,96	Дополнительное сырье-11	531,65	27864,13
Дополнительное сырье-12	3,77	8,96	Дополнительное сырье-12	0,59	543,04
			Дополнительное сырье-13	53,10	84669,53
Дополнительное сырье-14	1,31	21,44	Дополнительное сырье-14	0,71	8212,35
			Дополнительное сырье-15	73,84	10718,96
Операция-3			Операция-4		
Сырье-1	0,62		Сырье-1	0,62	
Сырье-2	0,38		Сырье-2	0,38	
Сырье-19	2,00		Сырье-19	2,00	
Сырье-4	14,80		Сырье-4	14,80	
Сырье-5	0,23		Сырье-5	0,23	
Дополнительное сырье	<i>B</i>	<i>K</i>	Дополнительное сырье	<i>B</i>	<i>K</i>
Дополнительное сырье-11	451,51	32576,49	Дополнительное сырье-11	346,19	51200,16
Дополнительное сырье-12	-1,54	593,65	Дополнительное сырье-12	-0,70	661,45
Дополнительное сырье-13	53,46	72708,80	Дополнительное сырье-13	105,08	103040,00
Дополнительное сырье-14	2,26	6852,12	Дополнительное сырье-14	0,4756	6007,26
Дополнительное сырье-15	75,38	6805,23	Дополнительное сырье-15	61,36	26490,78
Дополнительное сырье-16	0	12	Дополнительное сырье-16	0	19,6

Таблица 7.8 – Себестоимость основного и дополнительного сырья

Наименование ресурса	Цена, р.	Наименование ресурса	Цена, р.
Сырье-1	700000	Дополнительное сырье-11	50
Сырье-2	170000	Дополнительное сырье-12	10000
Сырье-4	1500	Дополнительное сырье-13	25
Сырье-5	200	Дополнительное сырье-14	50
Сырье-6	3500	Дополнительное сырье-15	60
Сырье-7	20	Дополнительное сырье-16	100000
Сырье-8	1500	Сырье-19	7042

Таблица 7.9 – Ограничения для операций по выработке ресурса

Значение выработки	Выработка ресурса по операциям		
	Операция-2 (Ресурс-2)	Операция-3 (Ресурс-3)	Операция-4 (Ресурс-4)
Минимальное	100	100	185
Номинальное	200	200	370
Максимальное	220	220	410

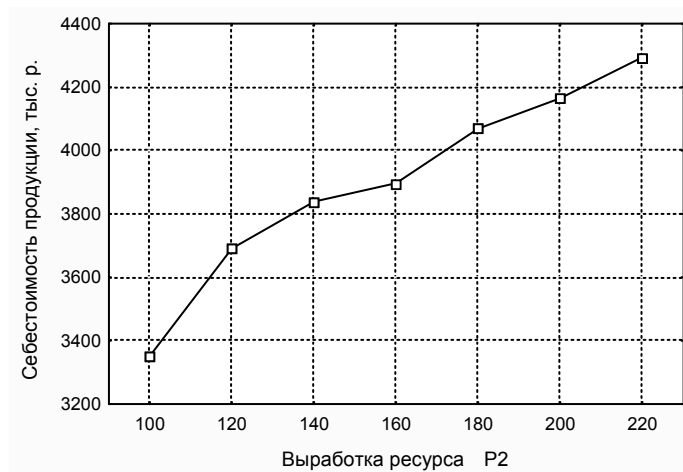


Рисунок 7.10 – Зависимость себестоимости продукции от выработки ресурса P2

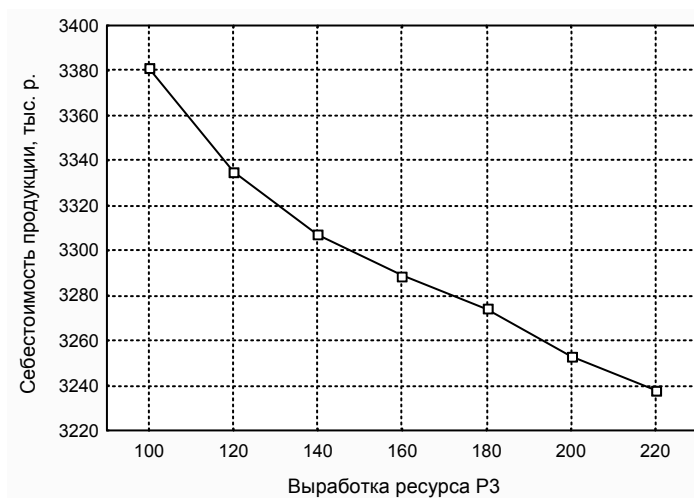


Рисунок 7.11 – Зависимость себестоимости продукции от выработки ресурса P3

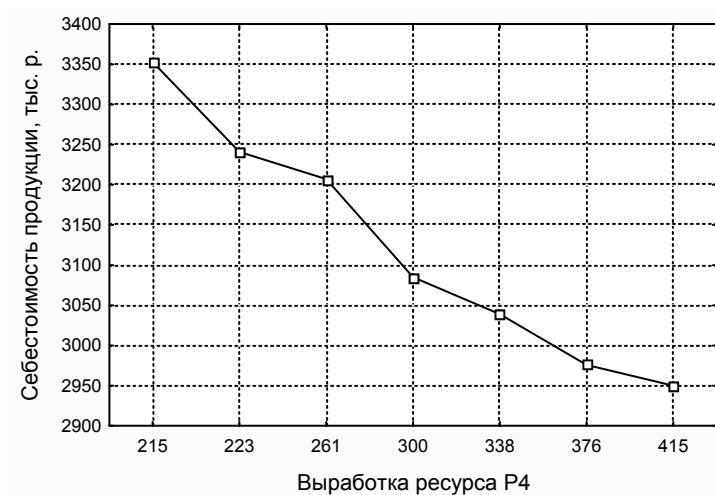


Рисунок 7.12 – Зависимость себестоимости продукции от выработки ресурса P4

Аналогично построена зависимость объема произведенной продукции от выработки ресурса P3 и ресурса P4. На построенном графике проводится плоскость, соответствующая заданному выпуску продукции, равному 600 единицам. При пересечении трехмерного графика с секущей плоскостью появляется кривая, определяющая соотношение выработок ресурса P3 и P4 для получения требуемого выпуска продукции. Полученная кривая переносится на график зависимости себестоимости продукции одновременно от выработки ресурса P3 и P4, и через нее строится плоскость, параллельная оси «Себестоимость продукции». Точки, лежащие на пересечении построенной поверхности и графика, соответствуют себестоимости выпускаемой продукции при заданной выработке.

Таким образом, минимальная себестоимость продукции получается при максимальной выработке ресурса P3, а выработка ресурса P4 определяется заданным количеством производимой продукции. Подбирают загрузку ресурса P4, обеспечивающую заданный выпуск продукции (рисунок 7.13). Для определенного значения выработки ресурса P4 находят себестоимость продукции. Найденные оптимальные параметры выработки ресурсов следующие: операция Оп2 – 100 ед.; операция Оп3 – 220 ед.; операция Оп4 – 280 ед.; себестоимость продукции – 3151 тыс. р./ед.

Свойства исследуемой модели в соответствии с формулой (1.12) представлены следующим образом:

$$Q = \langle A(S, D, E) \rangle = \langle A(4, 4, 5) \rangle,$$

т. к. моделируется производственная деятельность A для нескольких участков предприятия ($S = 4$). При этом модель динамически изменяется с изменением параметров и структуры технологического оборудования ($D = 4$), находится оптимальное распределение производства продукции между цехами ($E = 5$).

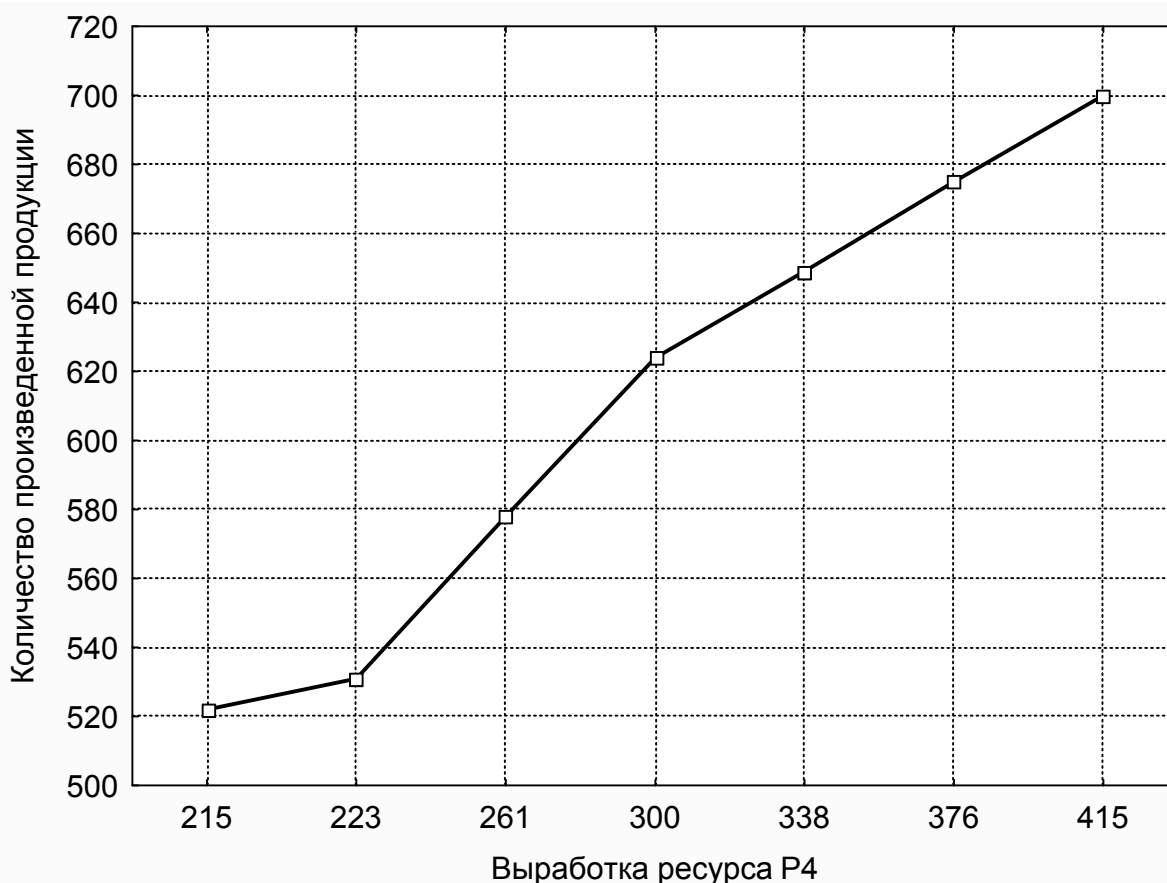


Рисунок 7.13 – Зависимость количества выпущенной продукции от выработки ресурса Р4 при максимальной выработке ресурса Р3 и минимальной выработке ресурса Р2

7.3.5 Реализация динамической производственно-экономической модели ЗОС на уровне производства ДМТ. Традиционная технология распределения производственной нагрузки между цехами состоит в следующем. Из-за отсутствия достаточного количества сырья и заказов на производство ДМТ предприятие не в состоянии использовать весь свой производственный потенциал и обеспечить работу всех трех цехов на проектных мощностях. Вследствие этого цех ДМТ-2 практически не работает, а производственные мощности распределяются между цехами ДМТ-3 и ДМТ-4 (см. рисунок 7.4). Причем руководствуются следующим соображением: так как проектная мощность цеха ДМТ-4 приблизительно в два раза больше, чем у ДМТ-3, то исходя из обратно пропорционального закона снижения себестоимости продукции с увеличением загрузки, принимается решение обеспечить номинальную работу цеха ДМТ-4, а остальную часть производства сосредоточить в цехе ДМТ-3. В случае производственной необходимости в строй вводится цех ДМТ-2. Как только план производства ДМТ снижается до определенного уровня, недостаточного для обеспечения работы трех цехов, цех ДМТ-2 останавливается.

При анализе эмпирических данных, хранящихся в комплексной ин-

формационной системе (КИС), было установлено, что одним из путей снижения себестоимости продукции является распределение производства продукции между цехами таким образом, чтобы удельная стоимость сырья и энергии была минимальной. В качестве исходных данных для решения поставленной задачи используются данные о выпуске продукции при различной загрузке цехов, в качестве ограничений – общее количество продукции, которое необходимо произвести за определённый период (месяц, квартал), а также максимальный и минимальный выпуск продукции для каждого цеха. Необходимо отметить, что цеха характеризуются разным расходом сырья и энергии на единицу выпускаемой продукции, что обусловлено характеристиками оборудования и особенностями технологических процессов.

Методика решения такой задачи состоит из двух этапов [247, с. 36].

1 *Обработка эмпирических данных.* Данные по расходу ресурсов при разной загрузке цехов импортируются из комплексной информационной системы в пакет STATISTICA. На основе полученных данных для каждого цеха строятся зависимости удельных значений расхода сырья и энергии от выпуска продукции:

$$PP_{ik} = f(KП_k),$$

где PP_{ik} – расход i -го вида ресурса (сырья или энергии) в k -м цехе на единицу продукции;

$KП$ – количество продукции.

При построении зависимостей из рассмотрения исключаются данные, которые не попадают в диапазон значений $(PP_{ik} - 2\sigma_{PPik}, PP_{ik} + 2\sigma_{PPik})$.

2 *Оптимизация загрузки цехов.* Полученные на предыдущем этапе зависимости переносятся в табличный процессор Excel, где осуществляется переход от количественных зависимостей к стоимостным путем умножения расходов ресурсов на их цены $ЦP_i$. При поиске оптимального решения используется целевая функция

$$\sum_i ЦP_i \cdot \sum_k PP_{ik} \rightarrow \min.$$

В качестве ограничений при минимизации себестоимости продукции используются минимальная и максимальная загрузка цехов и общий выпуск продукции $\sum_i \sum_k PP_{ik}$.

Представленная модель производственного процесса является сильно упрощенной и может применяться только при планировании производственных мощностей на некотором небольшом промежутке времени, поскольку не учитывает изменения цен на ресурсы. В реальных же условиях

цены на ресурсы подвержены влиянию инфляции, а также сезонным колебаниям (это относится в первую очередь к ценам на сырье из-за его дефицита в определённые месяцы).

При построении и эксплуатации модели решают задачи обеспечения подготовки и ввода исходных данных, разработки автоматизированной технологии расчета требуемых показателей. Для расчета, например, себестоимости выпускаемой продукции необходимо ввести в модель данные о ценах и нормах расхода основного сырья, вспомогательных материалов и энергоресурсов [216, с. 336].

Себестоимость готовой продукции рассчитывается по формуле

$$C = \frac{\sum_{i=1}^n c_{y\partial i}(x_i) \cdot v_i}{V}, \quad (7.6)$$

где C – себестоимость ДМТ по предприятию;

$c_{y\partial i}(x_i)$ – удельная себестоимость ДМТ в зависимости от нагрузки i -го цеха (x_i);

v_i – выпуск ДМТ i -м цехом;

V – суммарный выпуск ДМТ всеми цехами.

Функцию удельной себестоимости можно представить в виде

$$c_{y\partial i}(x_i) = oc_i(x_i) + vc_i(x_i) + er_i(x_i) + pz_i(x_i), \quad (7.7)$$

где $oc_i(x_i)$ – удельная себестоимость основного сырья в зависимости от нагрузки i -го цеха (x_i);

$vc_i(x_i)$ – удельная себестоимость вспомогательных материалов в зависимости от нагрузки i -го цеха (x_i);

$er_i(x_i)$ – удельная себестоимость энергоресурсов в зависимости от нагрузки i -го цеха (x_i);

$pz_i(x_i)$ – удельные постоянные затраты в зависимости от нагрузки i -го цеха (x_i).

Функции удельной себестоимости основных и вспомогательных материалов и энергоресурсов представляют собой сумму произведений удельной нормы j -го материала или энергоресурса на цену или себестоимость данного материала или энергоресурса и рассчитываются по формуле

$$f_r(x_i) = \sum_{j=1}^n N_j(x_i) \cdot Ц_j, \quad (7.8)$$

где $f_r(x_i)$ – функции удельной себестоимости основных и вспомогательных материалов и энергоресурсов;

N_j – удельная норма j -го ресурса;

C_j – цена или себестоимость j -го ресурса.

Удельные постоянные затраты

$$pz_i(x_i) = \frac{ZP + OPr + NL}{x_i}, \quad (7.9)$$

где ZP – заработная плата основных и вспомогательных рабочих с отчислением на социальное страхование;

OPr – общепроизводственные расходы;

NL – налоговые отчисления [216, с. 336–337].

Для построения модели можно воспользоваться различными средствами или языками программирования. Однако самой основной проблемой при построении модели является написание алгоритмов поиска оптимального решения исходя из нескольких критериев и условий ограничения. Создание и реализация данного алгоритма при помощи различных языков программирования на практике оказывается довольно сложной задачей. Проще воспользоваться табличным процессором MS Excel, в котором реализован метод «Поиск решения», позволяющий, задавая условия ограничения, находить оптимальное решение задачи. Кроме того, MS Excel представляет широкие возможности для хранения взаимосвязанных данных, используя связи между ячейками таблиц. Возможность написания макросов позволяет автоматизировать некоторые рутинные операции, например, по вводу и выводу данных.

Для расчета себестоимости необходимо ввести в модель данные о ценах и нормах расхода основного сырья, вспомогательных материалов и энергоресурсов. Нормы расхода сырья и вспомогательных материалов, а также цены на сырье, материалы и энергоресурсы хранятся в системе 1С: Предприятие. Поэтому в разрабатываемой модели необходимо осуществить автоматизированный импорт этих данных в модель. Данные о расходе энергоресурсов хранятся в комплексной информационной системе на предприятии в виде статистической информации, полученной снятием показаний приборов учета. Эти данные обрабатываются в пакете STATISTICA, полученные зависимости норм расхода энергоресурсов сохраняются и вводятся в модель. Далее вводят суточный план выпуска ДМТ и получают расчетную загрузку цехов по производству ДМТ и себестоимость ДМТ для каждого цеха и завода в целом. Зависимости норм расхода энергоресурсов, планы выпуска, расчетные значения загрузки и себестоимости могут быть сохранены в единой комплексной информационной системе на базе 1С: Предприятие 8.0, где также предусматривается чтение указанных данных при открытии модели [216, с. 340].

На рисунке 7.14 приведена схема автоматизированной технологии обработки информации [224, с. 163–165]. На рисунке 7.15 представлена

структурная схема программного комплекса для создания динамической производственно-экономической модели завода [218, с. 203–204].



Рисунок 7.14 – Схема автоматизированной технологии обработки информации

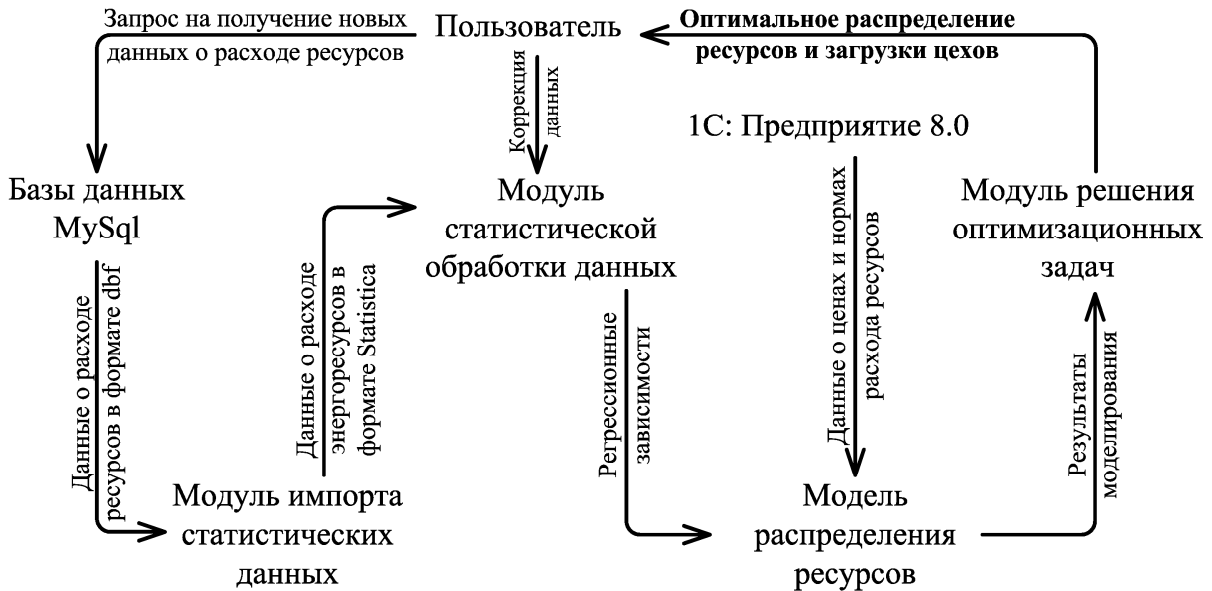


Рисунок 7.15 – Структурная схема программного комплекса

Программный комплекс состоит из следующих блоков [261, с. 116]:
 – блок импорта данных, состоящий из функций импорта данных по

ценам, нормам расхода материалов и энергоресурсов и сохранению данных в модели;

- блок статистической обработки данных, реализующий функции предварительной подготовки данных для анализа и непосредственного проведения регрессионного анализа;

- блок расчета загрузки цехов и себестоимости выпускаемой продукции, реализующий функции расчета загрузки цехов и себестоимости выпускаемой продукции;

- блок печати результата.

Пользователю доступны следующие настройки для каждого ресурса по каждому из цехов: фильтр на допустимые значения, анализируемый период, тип регрессионной модели, опция исключения ресурса из анализа, работа с выбросами, параметры вывода результата анализа [150; 242, с. 47–51].

7.3.6 Экспорт данных из системы 1С: Предприятие. В комплексной информационной системе 1С: Предприятие существует возможность создавать любые документы, справочники, регистры по своему усмотрению. Для разных предприятий существуют разнообразные информационные базы различной конфигурации. Основной задачей при моделировании производственного процесса является экспорт данных из справочников или документов в MS Excel.

Система 1С: Предприятие может использоваться внешними приложениями в качестве OLE-сервера. Для запуска системы 1С: Предприятие в качестве OLE-сервера из MS Excel выполняется следующая последовательность действий:

- создается OLE-объект с идентификатором v8.Application;

- выполняется инициализация системы 1С: Предприятие методом Connect;

- вызываются свойства и методы системы 1С: Предприятие как OLE-сервера.

Выбрать необходимые данные из справочника можно при помощи запроса. В системе 1С: Предприятие использование запросов является основным способом получения сводных данных. Для выполнения запроса к данным OLE-базы достаточно с помощью метода NewObject создать объект Запрос и далее работать с ним стандартными методами.

В справочниках содержатся цены на основные и вспомогательные материалы. Для получения данных при помощи запроса из справочников, содержащих информацию о ценах на материалы и энергоресурсы, используется программа на языке Visual Basic.

Полученные с помощью запроса из системы 1С: Предприятие данные сохраняются на отдельном листе MS Excel в строго определенных ячейках. Этого можно добиться, задавая различные условия при запросе и используя сортировку. Например, при запросе о ценах на материалы дан-

ные сортируются по коду и, несмотря на то, что запрос выбирает все поля, в ячейки можно вставлять значения любых атрибутов строки запроса, используя доступ к ним как к элементам объекта «Запрос». Далее на данные о ценах и нормах расхода материалов устанавливаются ссылки из модели. Таким образом обеспечивается автоматизированный ввод исходных данных в модель.

Целью создания технологии моделирования и оптимизации системы управления производственным процессом является решение следующих задач.

Задача 1. Оптимальное распределение производственной нагрузки по цехам ДМТ ЗОСа, обеспечивающее минимизацию себестоимости выпускаемой продукции.

Задача 2. Минимизация длительности переходного процесса и сокращение потерь при переходе на другой ассортимент полиэтилентерефталата (ПЭТФ).

Задача 3. Определение максимально допустимых закупочных цен на основное сырье.

Технология моделирования производства ДМТ включает следующие основные этапы (рисунок 7.16) [261, с. 115–116].

Этап 1. Импорт данных.

Шаг 1. В модель импортируются данные о ценах и нормах расхода основного сырья, вспомогательных материалов и о ценах на энергоресурсы из информационной системы 1С: Предприятие.

Шаг 2. Данные о расходе энергоресурсов в зависимости от выработки продукции сохраняются в специализированной информационной системе на базе MySQL в виде статистической информации, полученной снятием показаний приборов учета.

Этап 2. Обработка статистических данных.

Шаг 1. Данные о расходе энергоресурсов за период с 2000 года, хранящиеся в специализированной информационной системе на базе MySQL, импортируются в табличный процессор MS Excel, а затем в пакет статистической обработки и анализа данных STATISTICA. Автоматизация данного шага обеспечивает постоянное обновление информации в модели и, следовательно, возможность коррекции регрессионных зависимостей с учетом новых данных.

Шаг 2. Статистические данные обрабатываются средствами пакета STATISTICA и полученные зависимости норм расхода энергоресурсов от выработки импортируются в модель. Автоматизация этапа статистической обработки данных обеспечивает снижение трудоемкости данного шага и позволяет пользоваться системой лицам, не являющимся специалистами в области моделирования и статистической обработки данных.

Этап 3. Решение оптимизационных задач.

Для оптимизации используется программное обеспечение SOLVER табличного процессора MS Excel [216, с. 341–342].

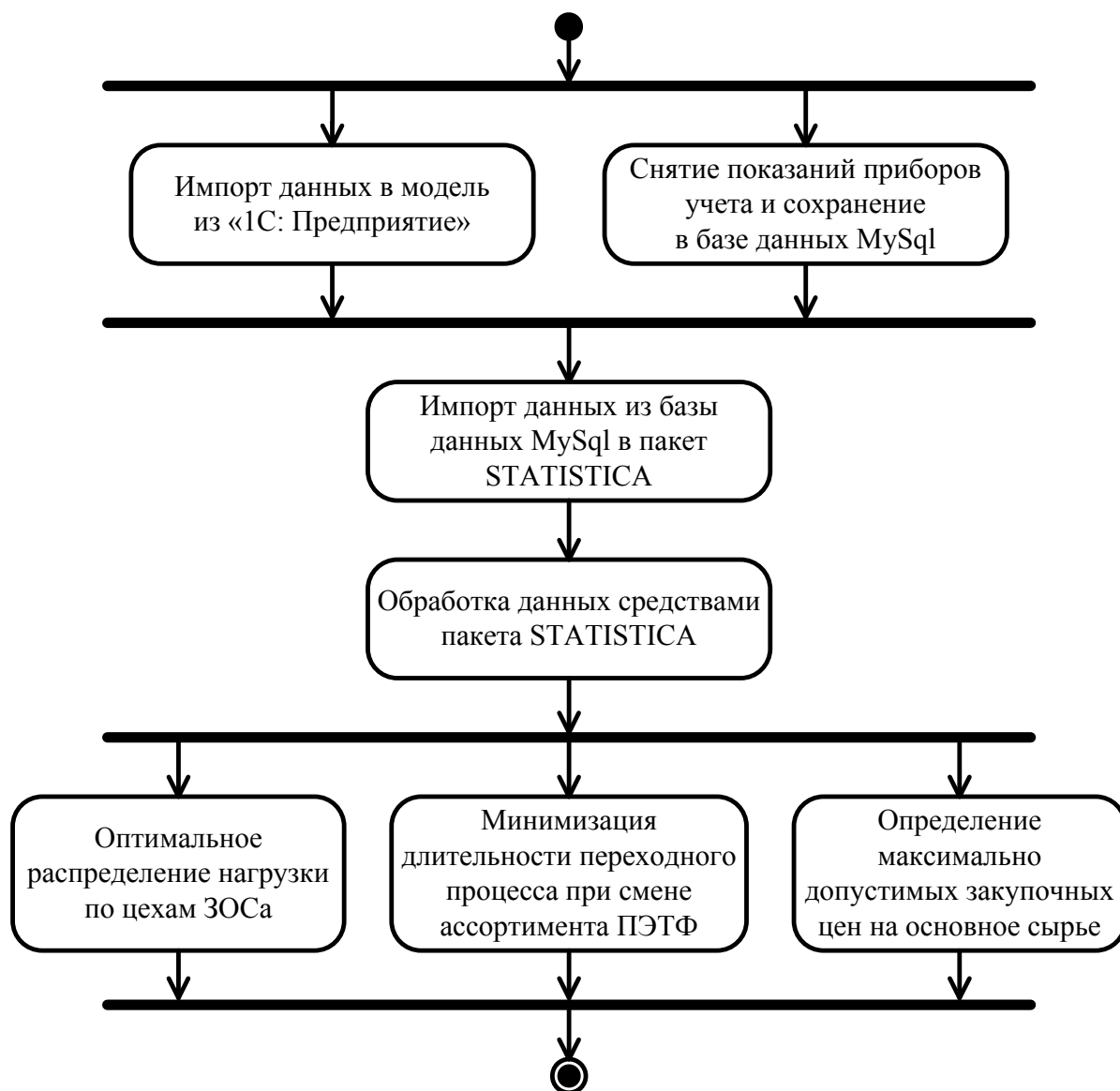


Рисунок 7.16 – Общая схема технологии моделирования и оптимизации системы управления производственным процессом на ЗОС

Решение задачи 1 (оптимальное распределение производственной нагрузки по цехам ДМТ ЗОСа, обеспечивающее минимизацию себестоимости выпускаемой продукции) реализуется следующей последовательностью шагов [223, с. 237; 261, с. 115–116].

Шаг 1. Вводятся зависимости расхода энергоресурсов от выработки, полученные для каждого цеха ДМТ ЗОСа на основе обработки статистических данных. Для основного сырья, вспомогательных материалов и постоянных затрат данные импортируются из системы 1С: Предприятие.

Шаг 2. Вводится суточный план выпуска ДМТ, который используется в качестве ограничения при расчете оптимальной загрузки цехов. В качестве ограничений при оптимизации используются также минимальная и максимальная суточная производительность каждого цеха.

Шаг 3. Вводится загрузка для каждого цеха, используемая в качестве начального приближения при оптимизации.

Шаг 4. Рассчитывается произведение расхода материалов и энергоресурсов на цену материала или энергоресурса. Вводится формула для расчета себестоимости ДМТ для каждого цеха как сумма затрат по статьям. В качестве целевой функции используется формула для расчета общей себестоимости по заводу.

Шаг 5. Минимизируется себестоимость продукции за счет изменения загрузки цехов при помощи программного обеспечения *SOLVER* табличного процессора MS Excel (рисунок 7.17).

Решение задачи 2 (минимизация длительности переходного процесса и потеря при переходе на другой ассортимент ПЭТФ) реализуется следующей последовательностью шагов.

Шаг 1. Вводятся зависимости расхода энергоресурсов от выработки на основе обработки статистических данных. Считается, что расход энергоресурсов не зависит от вида выпускаемого полимера. Для основного сырья, вспомогательных материалов и постоянных затрат данные импортируются из системы 1С: Предприятие.

Шаг 2. Вводится суточный план выпуска ПЭТФ, который используется в качестве ограничения при расчете оптимальной загрузки цехов.

Шаг 3. Задается интенсивность и продолжительность подачи дополнительного сырья.

Шаг 4. Рассчитывается себестоимость производства полимера при изменении ассортимента.

Шаг 5. Минимизируется время установления требуемой концентрации дополнительного сырья за счет изменения интенсивности его введения в технологический процесс [150, с. 64].

Решение задачи 3 (определение максимально допустимых закупочных цен на основное сырье) реализуется следующим образом.

При определении максимально допустимых закупочных цен себестоимость продукции для ДМТ и время перехода на другой ассортимент используются не в качестве целевых функций, а в качестве ограничений в процессе оптимизации. При этом возможны следующие варианты данной задачи.

Вариант 1. Необходимо установить максимальное значение закупочной цены для одного вида основного сырья. В этом случае задача сводится к максимизации значения закупочной цены для заданного вида основного сырья при однозначно заданном плане выпуска продукции (ДМТ и ПЭТФ), себестоимости продукции для ДМТ и времени перехода на другой ассортимент для ПЭТФ, не превышающем заданные значения. В качестве ограничения для каждого цеха ДМТ задается также минимальная и максимальная нагрузка.

```

' Процедура импорта данных из системы «1С: Предприятие»
Private Sub CommandButton1_Click()
    ' Подключение к базе данных средствами OLE Automation
    Set cc = CreateObject("v8.Application")
    cc.Connect ("File = C:\Progra~1\1cv8\Price; Usr = ")
    ' Формирование текста запроса
    Txt = "ВЫБРАТЬ * ИЗ Справочник.Материалы УПОРЯДОЧИТЬ ПО Код"
    ' Создание объекта Запрос
    Set Query = cc.NewObject("Запрос")
    Query.Text = Txt
    ' Выполнение запроса
    Set Result = Query.Execute()
    Set Chose = Result.Choose()
    i = 1
    ' Вывод содержимого запроса в ячейки Excel
    Do While Chose.Next()
        Worksheets("Цены").Cells(i + 2, 2) = Chose.Description
        Worksheets("Цены").Cells(i + 2, 3) = Chose.Ед
        Worksheets("Цены").Cells(i + 2, 4) = Chose.Цена
        i = i + 1
    Loop
    i = 1
    Txt2 = "ВЫБРАТЬ * ИЗ Справочник.Энергоресурсы УПОРЯДОЧИТЬ ПО Код"
    Query.Text = Txt2
    Set Result = Query.Execute()
    Set Chose = Result.Choose()
    i = 1
    Do While Chose.Next()
        Worksheets("Цены").Cells(i + 12, 2) = Chose.Description
        Worksheets("Цены").Cells(i + 12, 3) = Chose.Един
        Worksheets("Цены").Cells(i + 12, 4) = Chose.Цена
        i = i + 1
    Loop
    i = 1
End Sub

' Процедура расчета загрузки цехов
Private Sub CommandButton1_Click()
    ' Проверка входных данных на соответствие условиям
    ' ограничения выпуска
    If TextBox1.Value < 485 Then
        TextBox1.Value = 485
        MsgBox "Минимальное значение суточной выработки не должно быть ниже 485 тонн!"
    End If
    If TextBox1.Value > 850 Then
        TextBox1.Value = 850
        MsgBox "Максимальное значение суточной выработки не должно быть выше 850 тонн!"
    End If
    Cells(44, 2) = TextBox1.Value
    ' Реализация метода «Поиск решения»
    SolverOk SetCell:="$L$47", MaxMinVal:=2, ValueOf:="0", ByChange:= _ "$D$47,$G$47,$J$47"
    ' Установка условий ограничения
    SolverAdd CellRef:="$B$59", Relation:=2, FormulaText:="$B$44"
    SolverAdd CellRef:="$D$47", Relation:=1, FormulaText:="$B$57"
    SolverAdd CellRef:="$G$47", Relation:=1, FormulaText:="$C$57"
    SolverAdd CellRef:="$J$47", Relation:=1, FormulaText:="$D$57"
    SolverAdd CellRef:="$D$47", Relation:=3, FormulaText:="$B$55"
    SolverAdd CellRef:="$G$47", Relation:=3, FormulaText:="$C$55"
    SolverAdd CellRef:="$J$47", Relation:=3, FormulaText:="$D$55"
    SolverOk SetCell:="$L$47", MaxMinVal:=2, ValueOf:="0", ByChange:= _ "$D$47,$G$47,$J$47"
    SolverSolve UserFinish = False
    ' Вывод результатов в пользовательскую форму
    Label7.Caption = " " + FormatNumber(Cells(47, 4), 0)
    Label15.Caption = " " + FormatNumber(Cells(47, 7), 0)
    Label16.Caption = " " + FormatNumber(Cells(47, 10), 0)
    Label11.Caption = " " + FormatNumber(Cells(48, 4), 0)
    Label12.Caption = " " + FormatNumber(Cells(48, 7), 0)
    Label13.Caption = " " + FormatNumber(Cells(48, 10), 0)
    Label14.Caption = " " + FormatNumber(Cells(47, 12), 0)
    Label17.Caption = " " + FormatNumber(Cells(44, 2), 0)
End Sub

```

Рисунок 7.17 – Оптимальное распределение производственной нагрузки

Вариант 2. Необходимо установить максимальное значение закупочной цены C_i для N видов основного сырья ($i = 1, \dots, N$). В этом случае может быть использована процедура, аналогичная процедуре метода последовательных уступок: для выбранного одного вида сырья определяется максимальное значение закупочной цены $\max C_1$ при фиксированных регламентированных значениях цен на другие виды основного сырья. Для найденного максимального значения назначается некоторая уступка ΔC_1 .

Задавая в качестве ограничения $(\max C_1 - \Delta C_1) \leq C_1 \leq \max C_1$, определяют максимальное значение для второго вида основного сырья $\max C_2$. Далее назначается уступка для максимального значения цены второго вида основного сырья ΔC_2 и т. д. [150, с. 65].

7.4 Поиск оптимальных решений при заключении контрактов в маркетинговой деятельности промышленного предприятия

7.4.1 Исследование входных данных модели контракта продажи. Задачей исследования является выявление свойств имитационных моделей контрактов продажи, купли, на дилерское обслуживание и по товарному обмену для обоснования области применения аналитических моделей оценки эффективности указанных контрактов. Эксперименты проведены с применением ПТКИ BelSim на основе базовой имитационной модели промышленного предприятия. В ходе проведения экспериментов все варианты файлов данных модели контракта продажи имеют одинаковые значения параметров, кроме одного, изменение которого в ходе эксперимента осуществляется случайным образом по заданному закону распределения. Каждый вариант файла данных модели содержит только один из трех законов распределения случайных величин (равномерный, нормальный, экспоненциальный).

В качестве изменяемого в ходе эксперимента выбран параметр «Реализация: Объем отгрузки:» со средним значением, равным 500, и стандартным отклонением, равным 100, для каждого варианта модели. Вид функции плотности распределения задан для первого варианта модели – 0 (равномерный); для второго варианта модели – 1 (нормальный); для третьего варианта модели – 2 (экспоненциальный).

В качестве откликов выбрано несколько показателей, зависящих от изменяемого параметра:

- Деловая активность: Выручка от реализации (Y_1);
- Деловая активность: Чистая прибыль (Y_2);
- Рентабельность продукции (Y_3).

Тип эксперимента – полный факторный. При планировании эксперимента для каждого варианта задано по 100 прогонов модели.

Результаты проведенных исследований показывают, что закон изменения величин выходных откликов значительно зависит от закона изменения входных параметров. Следует отметить, что в общем виде задача установления типа выравнивающей кривой распределения и нахождения оценок параметров к настоящему времени до конца не решена [126, с. 32]. Для описания статистических распределений параметров производственной системы В. В. Нешиной [126, с. 41] предлагает использовать не отдельные распределения (нормальный закон, законы Вейбулла, Релея, Максвелла и др.), а обобщенные четырехпараметрические распределения, включающие, как частные случаи, большое количество известных распределений, образующих единую систему.

Три основные системы непрерывных распределений В. В. Нешиной представлены следующими плотностями распределения:

$$p(x) = Ne^{\gamma x} (1 - \alpha ue^{\beta x})^{\frac{1}{u}-1}; \quad (7.10)$$

$$p(t) = Nt^{\gamma-1} (1 - \alpha ut^{\beta})^{\frac{1}{u}-1}; \quad (7.11)$$

$$p(y) = \frac{N(\ln y)^{\gamma-1}}{y} \left[1 - \alpha u (\ln y)^{\beta} \right]^{\frac{1}{u}-1} \quad (7.12)$$

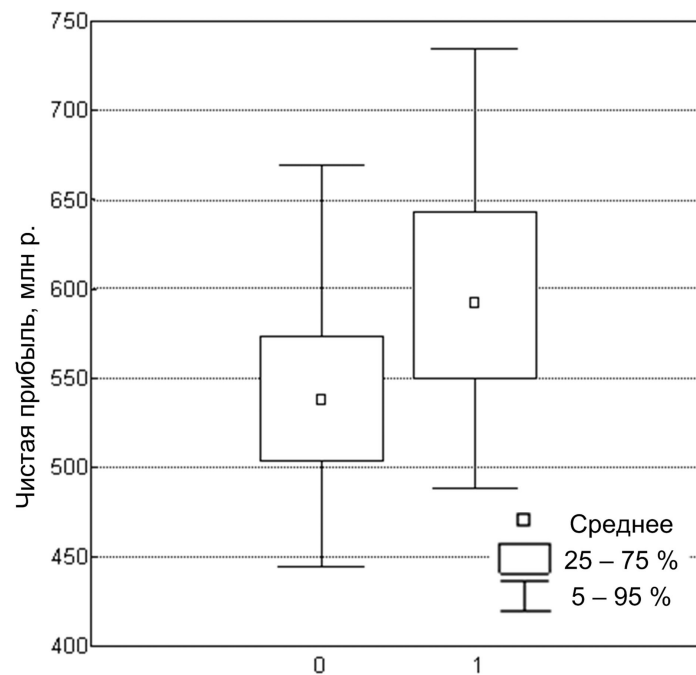
с четырьмя параметрами α , β , γ , u . Методики определения вида распределения, предложенные В. В. Нешиной, представляются перспективными для использования в имитационном моделировании производственно-экономических систем для задания исходных данных, получаемых из комплексной информационной системы предприятия. Этим достигается новый, значительно более высокий уровень точности решения прикладных задач, связанных с выравниванием статистических распределений.

7.4.2 Пример моделирования контракта продажи. Отгрузка продукции осуществляется после заключения контракта продажи в течение 76 дней. Основные данные по отгрузкам, видам изделий, ценам на момент заключения контракта и рентабельности продукции представлены в таблице 7.10.

Имитационная модель предприятия используется дважды. Сначала следует имитация без контракта, затем – с контрактом продажи. Результаты имитации с контрактом и без контракта представлены на рисунке 7.18.

Таблица 7.10 – Поставка продукции по контракту продажи

Наименование изделия, (цена, р./ед. изд.)	Отгрузка 1 (через 15 дн.): Объем, шт. (рентабельность, %)	Отгрузка 2 (через 34 дн.): Объем, шт. (рентабельность, %)	Отгрузка 3 (через 41 дн.): Объем, шт. (рентабельность, %)	Отгрузка 4 (через 68 дн.): Объем, шт. (рентабельность, %)	Отгрузка 5 (через 76 дн.): Объем, шт. (рентабельность, %)
1С119 (1420)		12866 (0)	5922 (0)	17548 (0)	
1С119К (2260)	750 (5,0)				
1С129К (2480)					3840 (16,0)
1С130К (760)					9360 (16,0)
2С2КВН (618)	40000 (5,0)	8753 (5,0)	13390 (5,0)	33670 (5,0)	98 (5,0)
4С31 (1240)			1249 (25,0)		
4С31КВ (1370)			2696 (18,0)		
4С3КВ (910)	815 (4,0)				
4С3КВГ (2080)				3890 (15,0)	
5С34КВ (1130)	35000 (22,0)		7043 (22,0)	87 (22,0)	
5С35КВ (870)			4075 (4,0)	3436 (4,0)	22511 (4,0)
5С36КВ (1218)			570 (5,0)	53231 (16,0)	
8С55 (2480)				1129 (23,0)	
8С58КВН (812)	40000 (8,3)		4082 (8,3)		
8С111 (1220)			1446 (7,0)		
8С111К (1150)					1950 (5,0)
8С41К (2300)		14720 (13,0)	9007 (13,0)		489 (13,0)
8С41КВ (1870)			2462 (10,0)	4586 (10,0)	
8С4КВГ (2390)				321 (22,0)	
9С88КВ (2050)		4940 (15,0)			
9С89КВ (1820)		4487 (6,0)			2192 (6,0)



0 – прогон модели без контракта; 1 – прогон модели с контрактом

Рисунок 7.18 – Изменение чистой прибыли при заключении контракта продажи

Моделирование проведено при следующих основных условиях: НДС – 0,2 отн. ед.; налог с выручки – 0,045 отн. ед.; налог с заработной платы – 0,05 отн. ед.; налоги с прибыли – 0,25 отн. ед.; налоги с прибыли, остающейся в распоряжении предприятия – 0,05 отн. ед.; экологический налог – 0,01 р./ед. изд.; коэффициент износа основных средств – 0,5 отн. ед.; коэффициент переоценки основных средств – 1,35 отн. ед./год; норма амортизации основных средств – 0,05 отн. ед./год; интервал планирования производства – 5 дн.; вид функции распределения интервалов между изменениями цен – экспоненциальный; средний интервал между изменениями цен – 30 дн.; стандартное отклонение интервалов между изменениями цен – 30 дн.; средний коэффициент изменения цен – 1,008 отн. ед./мес.; вид функции распределения коэффициента изменения цен – равномерный; стандартное отклонение коэффициента изменения цен – 0,01; сумма на расчетном счете – 100 млн р.

Таким образом, в соответствии с рисунком 7.18 исследователь может сделать вывод о целесообразности заключения исследуемого контракта при заданных исходных данных по критерию чистой прибыли предприятия. Аналогичные результаты получены и при использовании аналитической модели контракта продажи в программном комплексе *Contract Analyzer*, что подтверждает адекватность исследуемых моделей [223, с. 236–237].

7.4.3 Оценка влияния параметров контрактов на финансово-экономическое состояние предприятия. Оценка финансово-экономического состояния предприятия проводится по трем основным показателям: K_1 – коэффициент текущей ликвидности (Y74); K_2 – доля собственных оборотных средств в общей их сумме (Y71); K_3 – коэффициент обеспеченности собственными средствами (Y73). Дополнительно принят к анализу показатель рентабельности продукции (Y76).

Для исследований принята модель спроса, представленная на рисунке 7.19.

По этим же показателям оценивается и влияние сроков платежа на финансово-экономическое состояние предприятия. Эксперименты, результаты которых представлены на рисунках 7.20–7.25, проведены при следующих значениях сроков платежа за отгруженную продукцию: предоплата 30 дн.; предоплата 10 дн.; отсрочка платежа 10 дн. [13, с. 64–67; 224, с. 162–165; 238, с. 64–66].

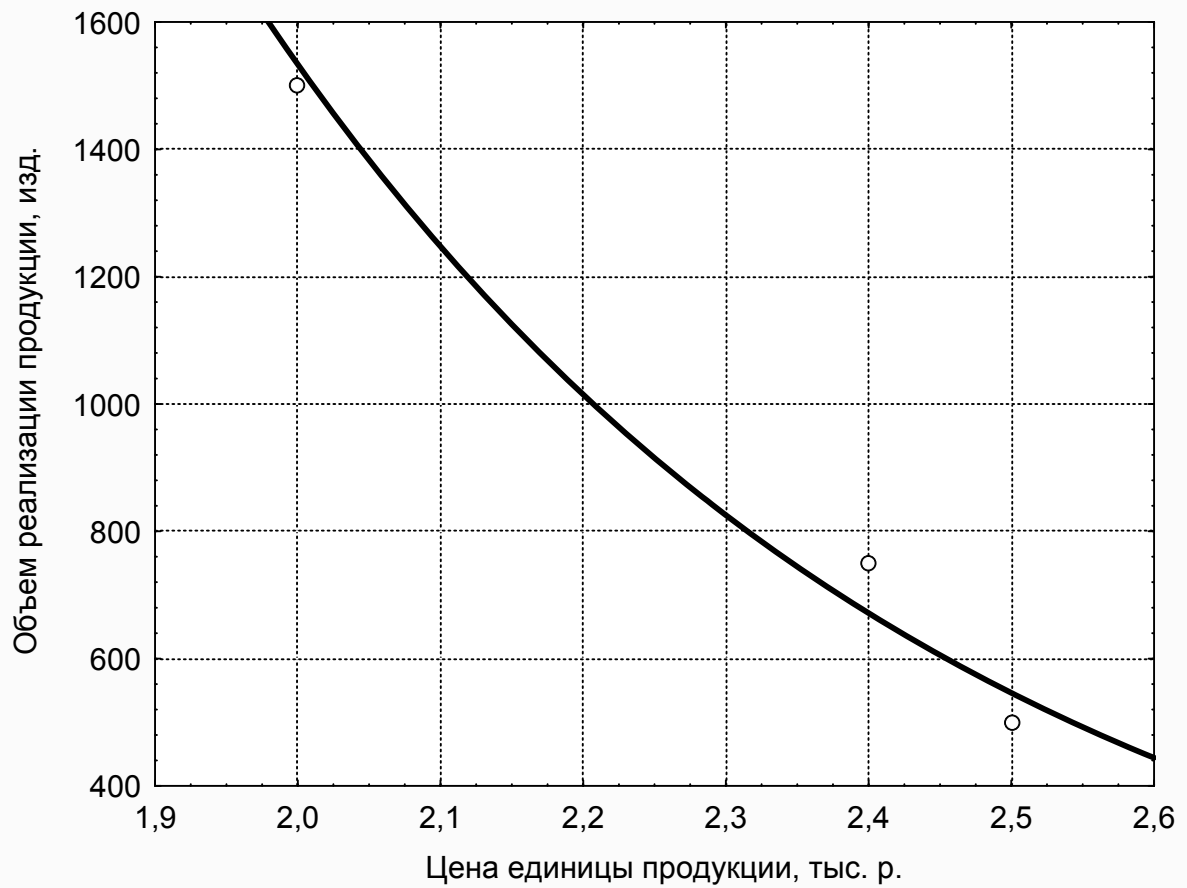
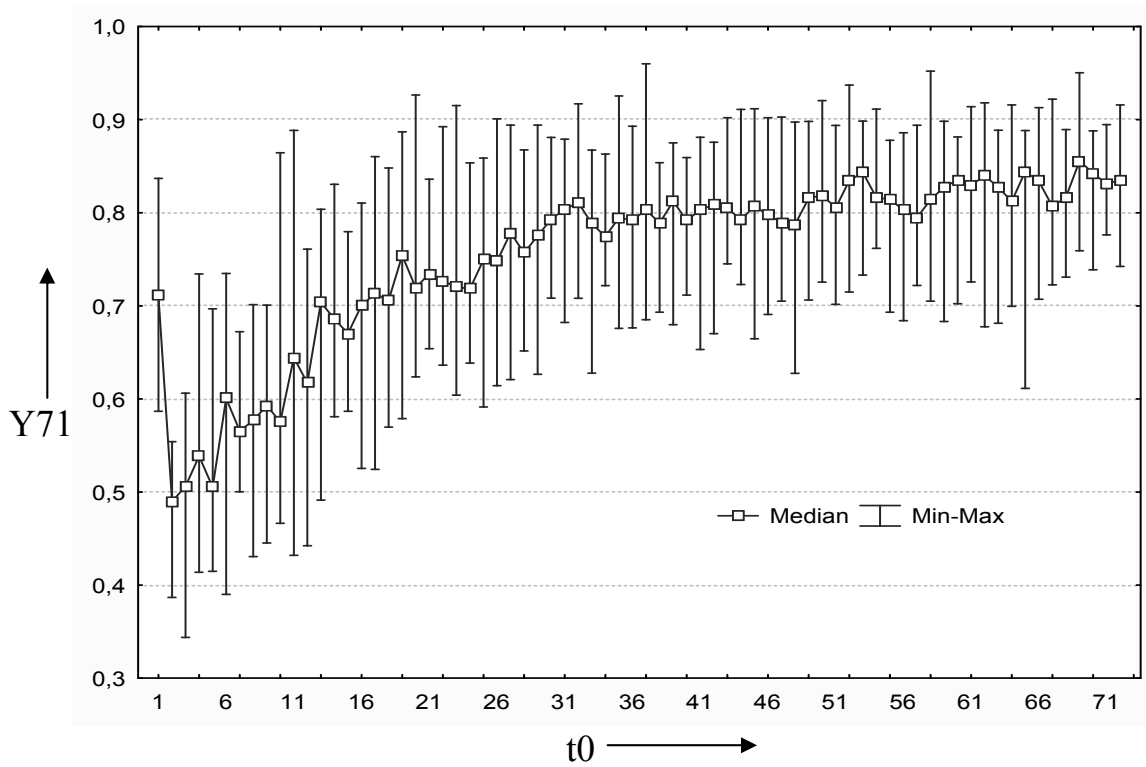


Рисунок 7.19 – Модель спроса на выпускаемую продукцию

Рисунок 7.20 – K_2 : срок платежа по контракту с предоплатой 30 дней

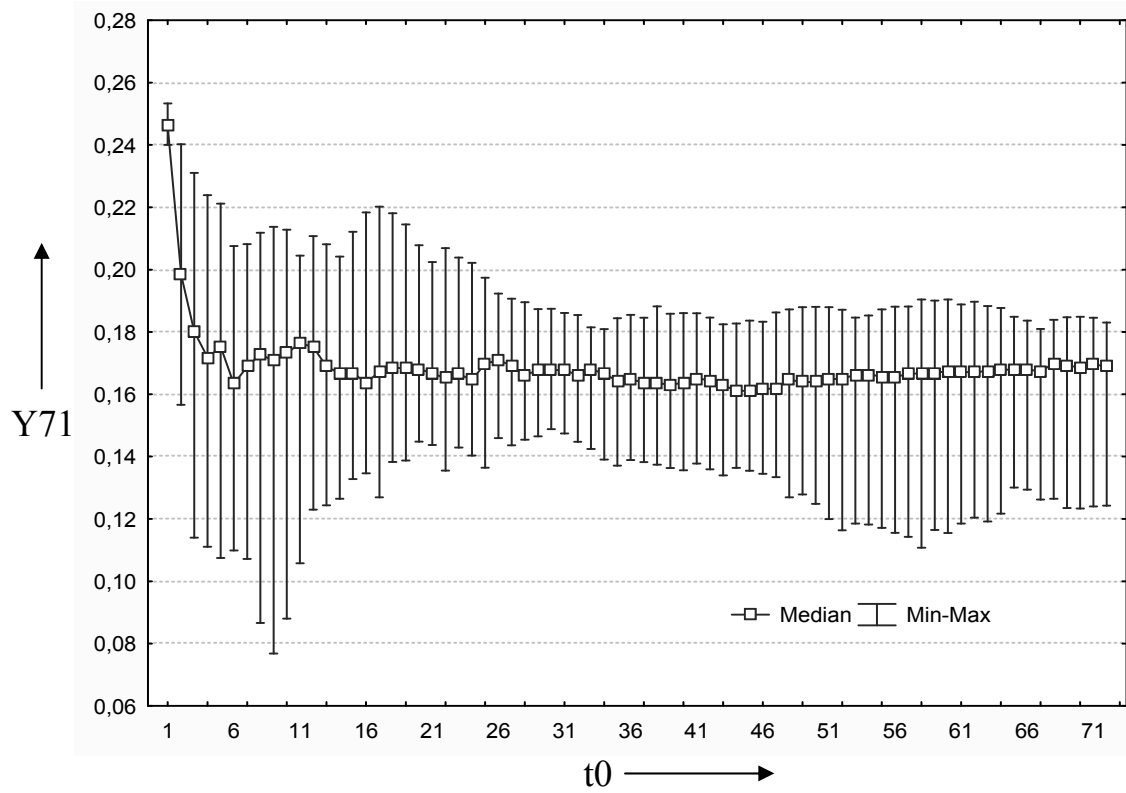


Рисунок 7.21 – Рентабельность продукции: срок платежа по контракту с предоплатой 30 дней

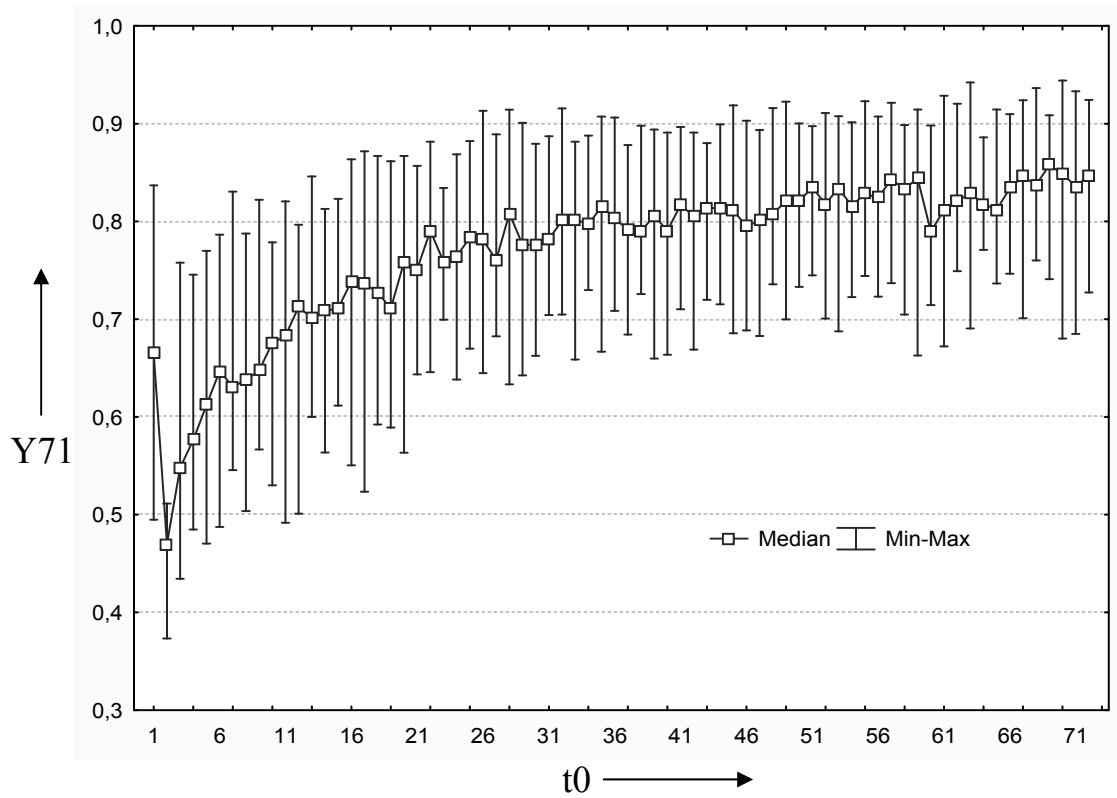


Рисунок 7.22 – K_2 : срок платежа по контракту с предоплатой 10 дней

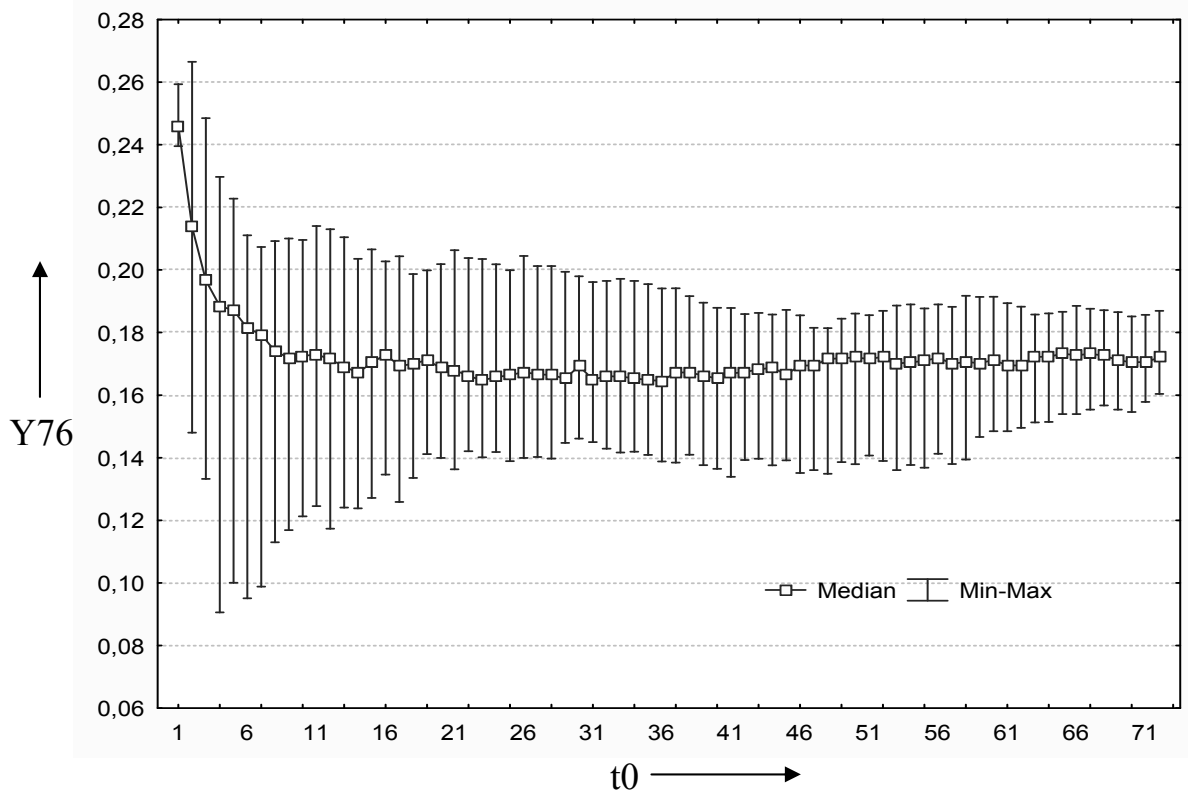


Рисунок 7.23 – Рентабельность продукции: срок платежа по контракту с предоплатой 10 дней

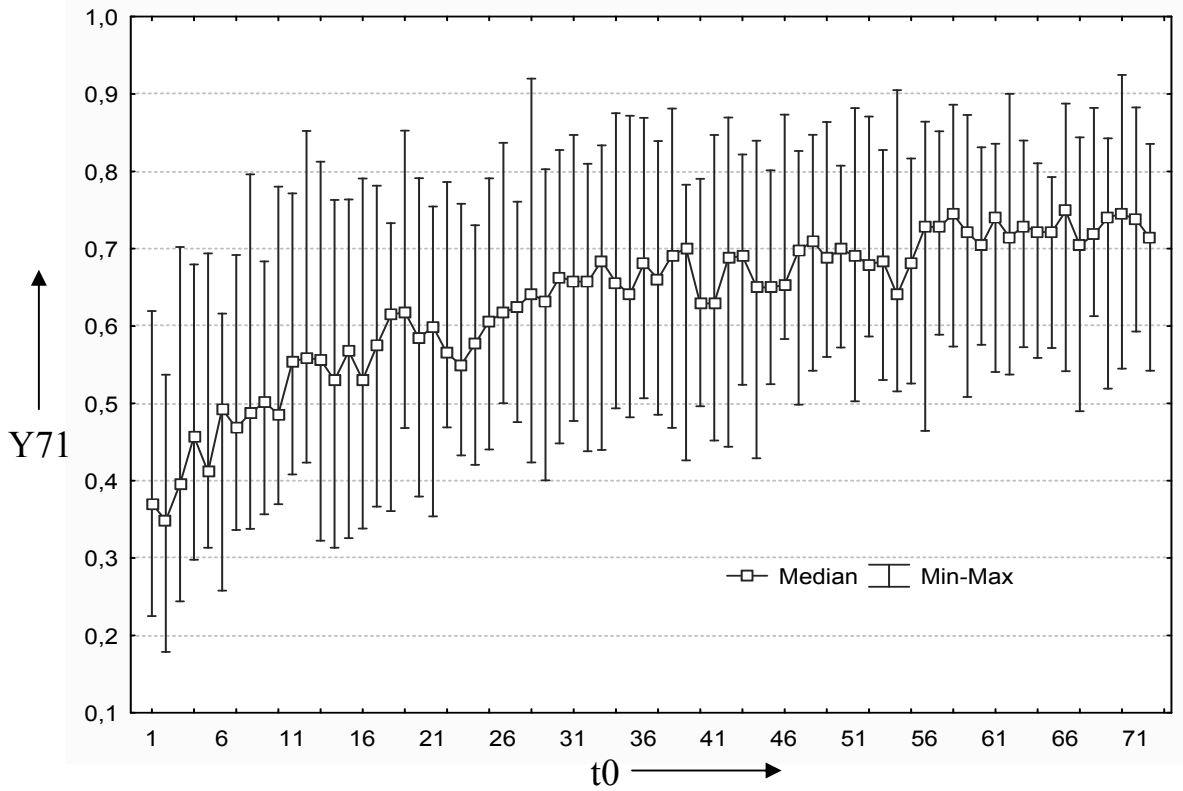


Рисунок 7.24 – K_2 : срок платежа по контракту с отсрочкой 10 дней

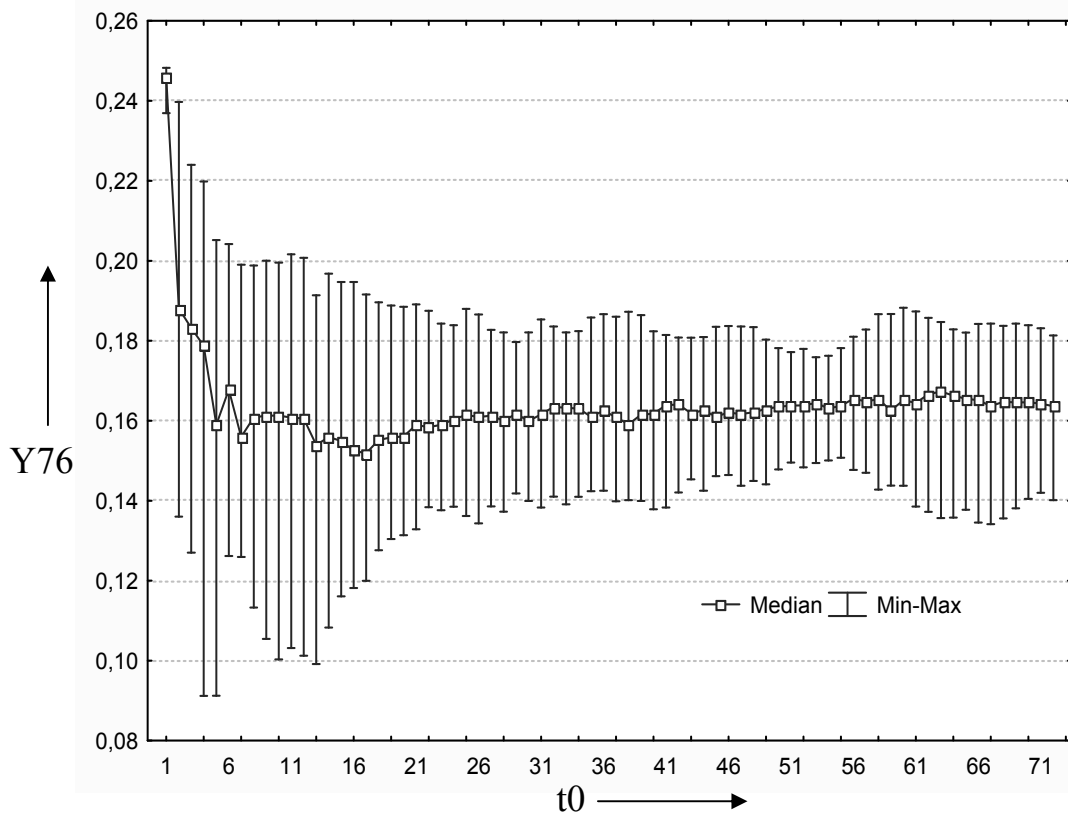


Рисунок 7.25 – Рентабельность продукции: срок платежа по контракту с отсрочкой 10 дней

7.5 Выбор типа координации процесса планирования производственной программы

7.5.1 Целевая функция планирования производственной программы. Производственная программа является результатом согласования следующих целей промышленного предприятия:

- получение максимальной прибыли;
- возможно полное удовлетворение потребностей рынка сбыта;
- учет реальных финансовых и иных ресурсных возможностей, в том числе динамики поставки сырья и энергоресурсов;
- максимальное снижение производственных издержек, в том числе и максимально полная загрузка технологического оборудования.

Для их достижения используется целевая функция, которую можно представить в следующем виде:

$$\Phi = F(P, M, C, X, S), \quad (7.13)$$

где P – получение максимальной прибыли, $P \rightarrow \max$;

M – максимальное использование производственных мощностей, $M \rightarrow \max$;

C – минимальная себестоимость выпускаемой продукции, $C \rightarrow \min$;

X – минимальные затраты на хранение выпускаемой продукции,
 $X \rightarrow \min$;

S – минимальные затраты на уплату штрафов за не вовремя отгруженную продукцию, $S \rightarrow \min$.

Процесс разработки информационной системы для поддержки принятия решений при планировании производственной нагрузки включает в себя следующие основные блоки:

- создание базы данных для анализа рыночной конъюнктуры сырья и производимой продукции, построения имитационных моделей рынка сырья и сбыта продукции;

- создание базы данных о финансовом состоянии предприятия, производственных мощностей и уровней запасов сырья на складе;

- разработка интерфейса интеграции имитационных моделей с базой данных комплексной информационной системы;

- разработка модуля для оценки вариантов и принятия решений при планировании производственной нагрузки на основе имитационного моделирования.

Укрупненная схема алгоритма решаемой задачи включает реализацию следующих основных функций:

- внесение данных;

- редактирование данных;

- выборка данных для моделирования;

- моделирование производственной нагрузки;

- анализ полученных результатов и выбор оптимального варианта планирования.

7.5.2 Модель планирования производственной нагрузки с жесткой координацией. Модель с жесткой координацией основана на том, что выпуск продукции (производственная нагрузка) меняется в зависимости от поступающих заказов на уровне объединения. Однако при этом производственная нагрузка не может превышать некоторое граничное значение $ВП\max_i$ и не может быть ниже минимального значения $ВП\min_i$, обусловленного особенностями технологического процесса производства. При этом продукция производится непрерывно, загружается в тару и перемещается на склад.

При необходимости изменения плана выпуска продукции в соответствии с требованиями рынка на предприятии используется традиционная технология распределения производственной нагрузки между цехами. Поскольку проектная мощность цехов различная, то обеспечивается номинальная работа наиболее мощного из них, а остальная часть выпуска продукции распределена в других цехах. В случае производст-

венной необходимости в строй вводится резервный цех, который периодически останавливается, если снижается план реализации продукции. При этом ввод резервного цеха сопряжен с дополнительными значительными затратами. В условиях, когда требуется использование производственных мощностей всех цехов, такая технология не определяет их оптимальную загрузку и не обеспечивает минимальную себестоимость продукции.

Производство продукции распределяется по дням, начиная со срока отгрузки. В случае, если даже при максимальном выпуске невозможно выполнить заказ в срок, можно попытаться выполнить заказ несколько позже, но при этом придется платить штраф за просрочку. При работе данной модели необходимо рассчитать максимальный объем выпуска продукции к сроку отказа заказчика от поставки продукции.

Для алгоритма с жесткой координацией определены следующие параметры: $ЗСн$ – незаказанный объем складского запаса продукции; $ОЗк_j$ – объем заказа по j -му контракту; $\sum_{i=2}^4 ПН_{i\max}$ – максимальная суточная про-

изводственная нагрузка ЗОС по выпуску ДМТ; $T \cdot \sum_{i=2}^4 ПН_i = ОЗк_j - ЗСн$ –

требуемый выпуск ДМТ за период T с учетом складского запаса $ЗСн$ для выполнения j -го заказа объемом $ОЗк_j$; $ОЗост$ – текущее значение объема ДМТ, оставшееся для выполнения заказа; $Tз$ – срок выполнения заказа по контракту; $ВП_j$ – суточная производственная нагрузка ЗОС по выпуску ДМТ. Алгоритм планирования производственной нагрузки с жесткой координацией с учетом введенных выше обозначений представлен на рисунке 7.26. Имитационная модель процесса планирования производства по указанному алгоритму реализована в среде ПТКИ BelSim.

7.5.3 Модель планирования производственной нагрузки с интегральной координацией. Модель с интегральной координацией основана на постоянной производственной нагрузке, когда за определенный длительный период времени требуется выпустить заданное количество продукции. Если спрос на продукцию отсутствует, она поступает на склад. С другой стороны, при отсутствии сырья требуется брать кредиты для его закупки. Финансовые средства могут отсутствовать, когда продукция не реализуется. При этом следует определить максимально допустимые цены на закупку сырья для завода органического синтеза и учесть дополнительные затраты на хранение товарной продукции на складе.

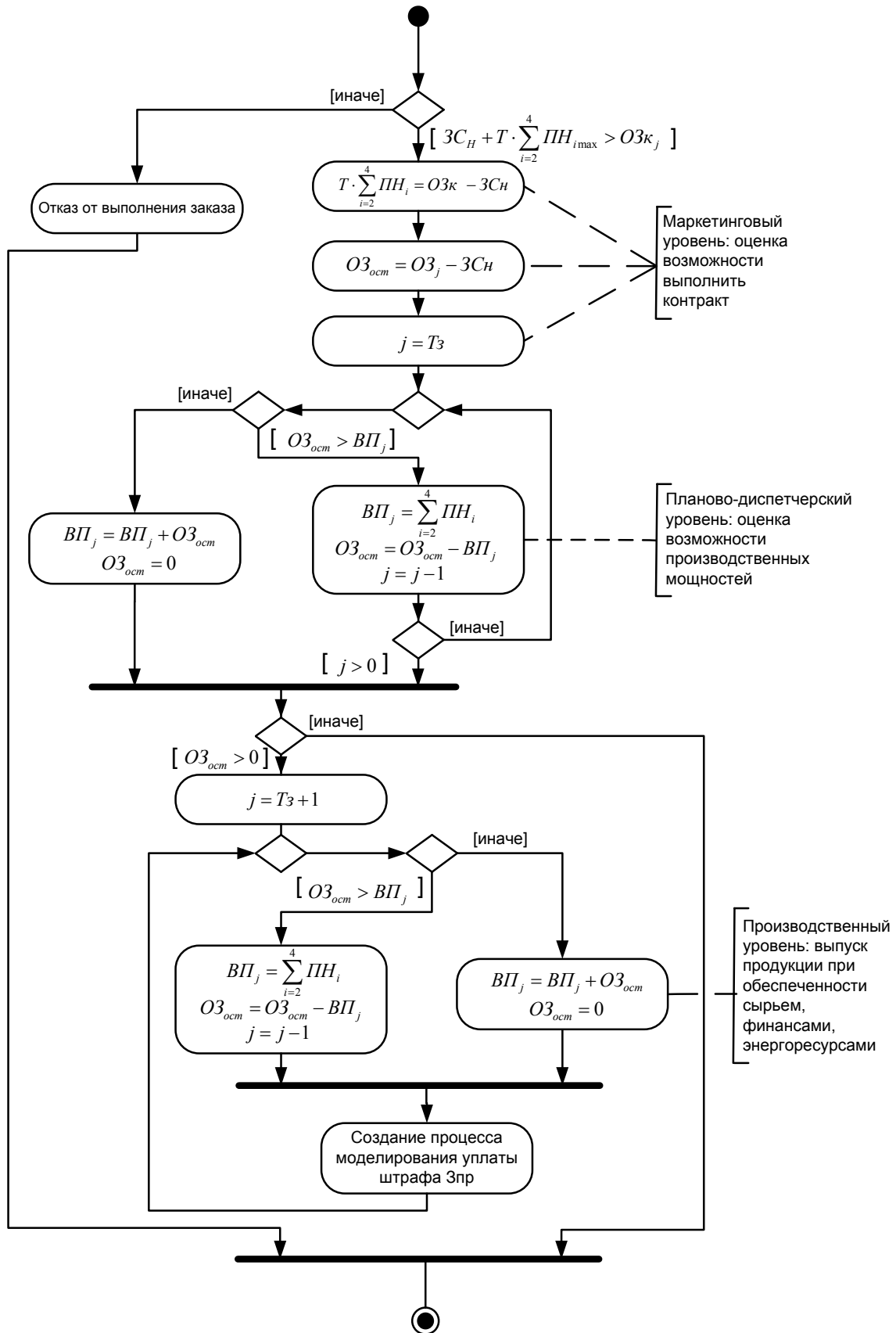


Рисунок 7.26 – Алгоритм процесса планирования с жесткой координацией HardCoord

В алгоритме интегральной координации на рисунке 7.27 выпуск продукции осуществляется с постоянной производственной нагрузкой. Такой вариант планирования обусловлен тем, что пуск цехов и останов производства сопряжены со значительными энергетическими затратами, которые увеличивают себестоимость выпускаемой продукции. Поэтому при отсутствии заказов предусмотрена поставка продукции на склад. Алгоритм построен при условии достаточного количества сырья для выпуска продукции.

Пример – Пусть m – количество критериев, $m = 2$, $Y = \{y^1, y^2, y^3\}$, причем $y^1 = \{1,4; 1,1\}$; $y^2 = \{1,3; 1,2\}$; $y^3 = \{1,1; 1,3\}$. Здесь все три возможных вектора являются Парето-оптимальными, т. е. принцип Парето не позволяет сузить область поиска выбираемых векторов.

Предположим, что имеется дополнительная информация о том, что первый критерий важнее второго с коэффициентом относительной важности $\theta_{ij} = 0,5$.

Согласно формуле (2.52) новый второй критерий принимает вид:

$$\hat{y}_2 = 0,5y_1 + 0,5y_2 \quad (7.14)$$

и, следовательно,

$$\hat{Y} = \{(1,4; 1,25), (1,3; 1,25), (1,1; 1,20)\}. \quad (7.15)$$

Парето-оптимальным в этом множестве является только один первый вектор. Значит, он, и только он один, может оказаться выбранным при условии, что выбираемые векторы существуют.

Пример – Пусть k – количество различных сообщений об относительной важности критериев [129], $k = 2$, $m = 3$, $Y = \{y^1, y^2, y^3, y^4\}$, где $y^1 = (1,1; 1,45; 1,2)$, $y^2 = (1,2; 1,3; 1,1)$, $y^3 = (1,3; 1,2; 1,1)$, $y^4 = (1,5; 1,15; 1,2)$, $u^1 = (1,0; 1,5; 1,1)$, $v^1 = (1,2; 1,2; 1,0)$, $u^2 = (1,5; 1,0; 1,2)$, $v^2 = (1,1; 1,1; 1,1)$.

Поскольку $u^1 - v^1 = (-0,2; 0,3; 0,1)$, $u^2 - v^2 = (0,4; -0,1; 0,1)$, то данные две пары векторов u^1, v^1, u^2, v^2 могут задавать (если они непротиворечивы) информацию об относительной важности критериев, состоящую из двух сообщений.

Для проверки совместности векторов u^1, v^1, u^2, v^2 в соответствии с алгебраическим критерием непротиворечивости составляется однородная система линейных уравнений:

$$\lambda_1 - 0,2\mu_1 + 0,4\mu_2 = 0; \quad (7.16)$$

$$\lambda_2 + 0,3\mu_1 - 0,1\mu_2 = 0; \quad (7.17)$$

$$\lambda_3 + 0,1\mu_1 + 0,1\mu_2 = 0. \quad (7.18)$$

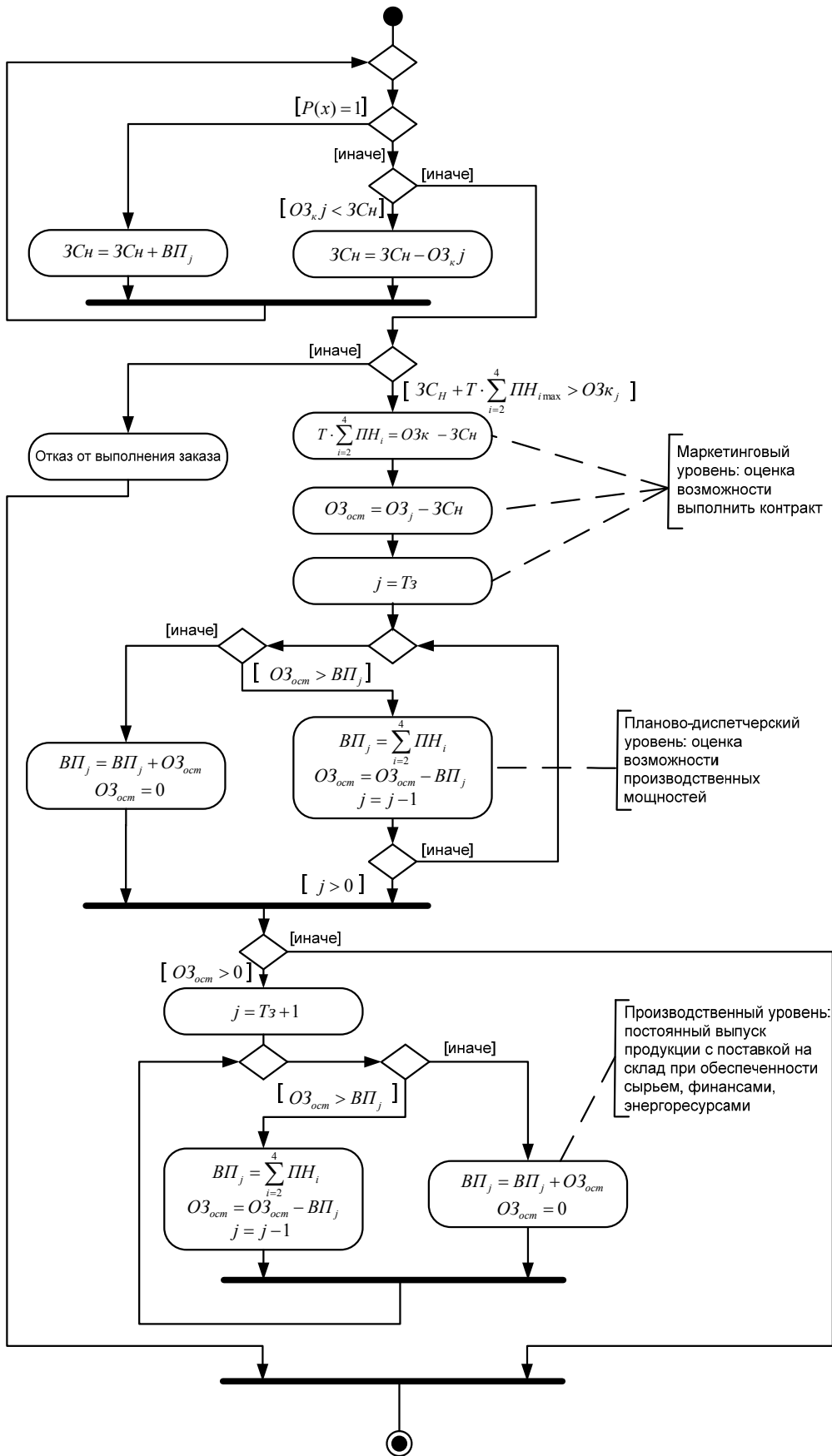


Рисунок 7.27 – Алгоритм процесса планирования с интегральной координацией IntegralCoord

Из уравнения (7.18) благодаря неотрицательности чисел λ_3, μ_1, μ_2 следует равенство нулю:

$$\lambda_3 = 0; \mu_1 = 0; \mu_2 = 0. \quad (7.19)$$

Тогда из уравнений (7.16) и (7.17) следует, что

$$\lambda_1 = \lambda_2 = 0. \quad (7.20)$$

Следовательно, рассматриваемая система линейных уравнений не имеет ненулевых неотрицательных решений, что означает совместность двух пар векторов u^1, v^1, u^2, v^2 .

Для построения множества недоминируемых векторов $\text{Ndom } Y$ составляется система линейных уравнений [129, с. 124] для векторов $y' = y^1$ и $y'' = y^2$:

$$\begin{aligned} \lambda_1 - 0,2\mu_1 + 0,4\mu_2 &= -0,1; \\ \lambda_2 + 0,3\mu_1 - 0,1\mu_2 &= 0,15; \\ \lambda_3 + 0,1\mu_1 + 0,1\mu_2 &= 0,1. \end{aligned} \quad (7.21)$$

Система уравнений (7.21) имеет ненулевое неотрицательное решение: $\lambda_1 = \lambda_2 = \mu_2 = 0$; $\lambda_3 = 0,05$; $\mu_1 = 0,5$. Следовательно, выполняется соотношение $y^1 \succ_M y^2$, а значит, вектор y^2 не может входить в множество недоминируемых векторов $\text{Ndom } Y$.

Для векторов $y' = y^4$ и $y'' = y^3$ система линейных уравнений (7.21) принимает вид:

$$\begin{aligned} \lambda_1 - 0,2\mu_1 + 0,4\mu_2 &= 0,2; \\ \lambda_2 + 0,3\mu_1 - 0,1\mu_2 &= -0,05; \\ \lambda_3 + 0,1\mu_1 + 0,1\mu_2 &= 0,1. \end{aligned} \quad (7.22)$$

У системы (7.22) имеются ненулевые неотрицательные решения $\lambda_1 = \lambda_2 = \mu_1 = 0$; $\lambda_3 = 0,05$; $\mu_2 = 0,5$. Поэтому вектор y^3 также не входит в множество недоминируемых векторов $\text{Ndom } Y$.

Пара систем линейных уравнений (7.21) для векторов $y' = y^1, y'' = y^4$ и $y' = y^4, y'' = y^1$:

$$\begin{cases} \lambda_1 - 0,2\mu_1 + 0,4\mu_2 = -0,4; \\ \lambda_2 + 0,3\mu_1 - 0,1\mu_2 = 0,3; \\ \lambda_3 + 0,1\mu_1 + 0,1\mu_2 = 0; \end{cases} \quad \begin{cases} \lambda_1 - 0,2\mu_1 + 0,4\mu_2 = 0,4; \\ \lambda_2 + 0,3\mu_1 - 0,1\mu_2 = -0,3; \\ \lambda_3 + 0,1\mu_1 + 0,1\mu_2 = 0. \end{cases} \quad (7.23)$$

Ни одна из этих двух систем не имеет ненулевых неотрицательных решений:

$$\begin{aligned} \lambda_1 = -0,26; \lambda_2 = 0,13; \lambda_3 = -0,05; & \quad \lambda_1 = 0,45; \lambda_2 = -0,4; \lambda_3 = 0,004; \\ \mu_1 = 0,55; \mu_2 = -0,08; & \quad \mu_1 = -0,36; \mu_2 = 0,05. \end{aligned}$$

Следовательно, ни одно из соотношений $y^1 \succ_M y^4$ и $y^4 \succ_M y^1$ не выполняется. В итоге получено следующее двухэлементное множество недоминируемых векторов:

$$\text{Ndom}_M Y = \{y^1, y^4\}.$$

Это множество представляет собой оценку сверху для множества выбираемых векторов $\text{Sel}Y$, т. е. $\text{Sel}Y \subset \{y^1, y^4\}$. Ни один из возможных векторов y^2, y^3 не вошел в это множество, а значит, ни один из них заведомо не должен быть выбранным.

Разработанные программные средства позволяют моделировать стратегии управления на основе базовой имитационной модели с формированием основных показателей деятельности предприятия и принимать решение на основе сформированной целевой функции.

7.6 Разработка библиотеки моделей для исследования информационной системы промышленного предприятия

7.6.1 Состав и структура библиотеки имитационных моделей информационной системы на основе ПТКИ BelSim.

Промышленные предприятия и их информационные системы (ИС) относят к сложным системам. Для них проблематично построить общую имитационную модель (ИМ), которая учитывала бы все аспекты их функционирования. Моделирование затруднено из-за большого количества входных характеристик X , управляющих параметров Z , откликов модели Y , происходящих в системе процессов W , сложной структуры Σ системы. Коды программ моделей таких систем получаются значительными, что усложняет их отладку. Появляются сложности в планировании экспериментов при большом числе X, Y, Z и их взаимном влиянии. Трудно решаются вопросы верификации и оценки адекватности моделей.

Случайным образом может изменяться ряд параметров ИС, например, количество и объем заказов клиентов, появление сбоев и отказов в работе оборудования, время пополнения запасов ресурсов. Это обуславливает необходимость создания имитационных моделей (ИМ) для оценки параметров и выбора состава и структуры ИС предприятия.

Моделирование ИС предприятия предполагает реализацию бизнес-процессов корпоративной информационной системой (КИС) в соответствии

с функциями MRP II на основе имитационного моделирования. Такой подход к моделированию ИС позволяет оценивать эффективность функционирования базового и проектного вариантов ИС предприятия и принимать решение по рациональному выбору состава, структуры и параметров ИС.

Для моделирования ИС предприятия разработана библиотека ИМ (рисунок 7.28), являющаяся предметным расширением ПТКИ BelSim при исследовании ИС. Библиотека ИМ ИС промышленного предприятия включает ИМ бизнес-процесса планирования продаж и производства, ИМ бизнес-процесса управления запасами ресурсов, ИМ ИС производственного процесса, а также ИМ функционирования ИС в корпоративной сети.

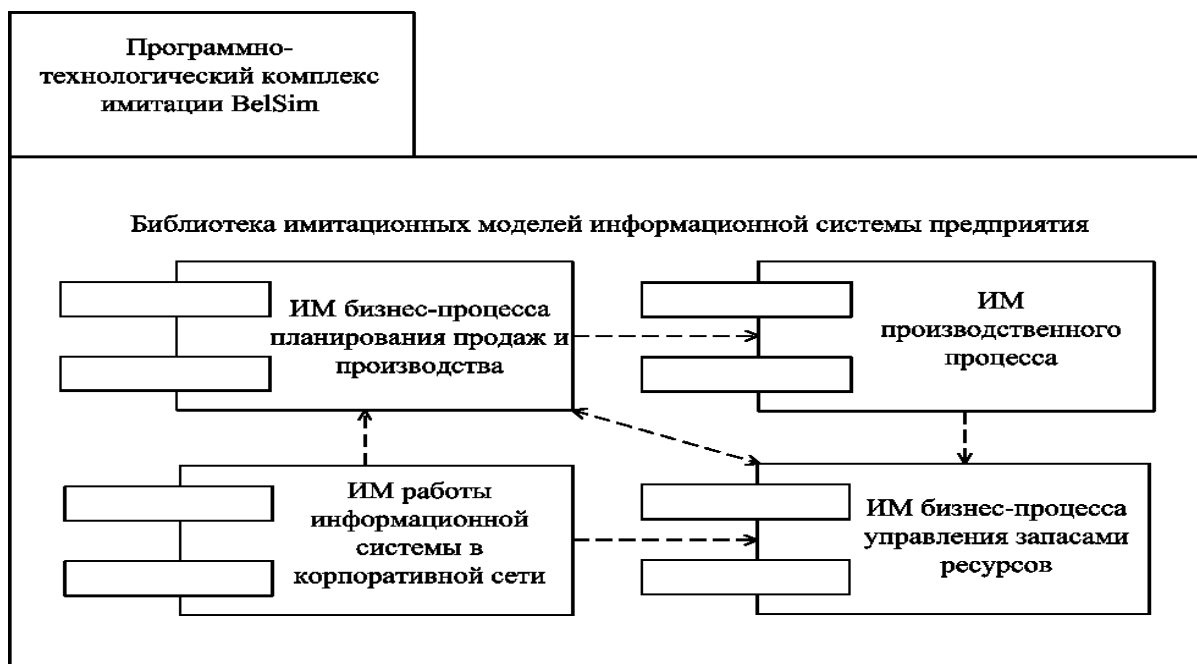


Рисунок 7.28 – Состав и структура библиотеки имитационных моделей информационной системы

Все ИМ, входящие в состав библиотеки ИМ ИС предприятия, реализованы на основе процессного способа имитации и предназначены для рационального выбора параметров подсистем ИС. Каждая ИМ, кроме процессов, реализованных в ИС, включает процессы финансово-хозяйственной деятельности промышленного предприятия, непосредственно связанных с процессами ИС.

7.6.2 Имитационная модель бизнес-процесса планирования продаж и производства. ИМ бизнес-процесса планирования продаж обеспечивает возможность анализа эффективности различных методов планирования продаж, реализованных в КИС, с точки зрения экономических показателей работы предприятия. Структуры данных ИМ предназначены для хранения параметров процессов модели, которые, в свою очередь, реали-

зуют бизнес-логику ее работы.

В структуре *TOrder* (рисунок 7.29) хранятся параметры заказов: даты поступления; виды продукции; объемы заказов; даты их выполнения; максимальное время ожидания выполнения заказов после истечения срока поставки; распределение объема заказов по видам продукции; распределение отгрузок продукции по датам; интервал времени между поступлениями заказов.



Рисунок 7.29 – Диаграмма компонентов ИМ бизнес-процесса планирования продаж и производства

Структура *TPlan* предназначена для хранения параметров планирования продаж и производства: числа заказов в очереди; максимального ежедневного выпуска условного вида продукции без сверхурочной работы; максимального ежедневного выпуска условного вида продукции при сверхурочной работе; дополнительных затрат на выпуск единицы условного вида продукции сверх установленных норм; длительности производственного цикла для каждого вида продукции; потерь производительности при переводе на выпуск другого вида продукции; изменений загрузки производственной мощности во времени; метода планирования продаж и производства.

Структура *TProduct* предназначена для хранения характеристик выпускаемой продукции: максимального ежедневного выпуска продукции

без сверхурочной работы; коэффициентов пересчета выпуска продукции на единицу условного вида продукции; норм хранения каждого вида продукции; штрафов за каждый день просрочки поставки; складского запаса каждого вида продукции.

Основным компонентом ИМ бизнес-процесса планирования продаж и производства является структура *TEnterprise*, содержащая большинство стандартных характеристик промышленного предприятия и внешней среды: виды продукции и цены каждого вида продукции, интервалы между изменениями цен и коэффициенты изменения цен; характеристики ресурсов (объёмы и цены), необходимых для производства каждого вида продукции.

Процесс *CShipmentGenerating* используется для генерации заказов через случайные промежутки времени с заданным законом распределения, средним значением и стандартным отклонением. При создании заказа генерируются его параметры: виды и объёмы продукции, дата выполнения и максимальное время ожидания выполнения заказа. Эти данные используются в процессах *Planning MPS1*, *Planning MPS2*, *Planning MPS3* (см. рисунок 7.29) планирования продаж и производства.

Процесс *Planning MPS1* реализует традиционный метод планирования продаж и производства, состоящий в том, что заказы на поставку продукции собираются в течение некоторого промежутка времени (обычно месяца), после чего планируется выпуск продукции с учетом имеющихся производственных мощностей. Заказы выполняются в порядке их поступления. При этом предполагается, что сроки исполнения заказов не ограничены. Данный процесс запускается через промежутки времени, равные периоду сбора заказов.

Процесс *Planning MPS2* реализует метод планирования продаж и производства, основанный на том, что выпуск продукции меняется каждый раз при поступлении заказа. Однако выпуск не может превышать некоторого предельного значения. Данный процесс создается процессом *CShipmentGenerating*. Продукция производится непосредственно перед отгрузкой, что позволяет снизить издержки, связанные с ее хранением. Производство продукции распределяется по дням, начиная со срока отгрузки. При этом максимальный выпуск продукции приходится на дни, непосредственно предшествующие отгрузке. В случае, если даже при максимальном выпуске невозможно выполнить заказ в срок, можно попытаться выполнить заказ несколько позже, но при этом придётся платить штраф за просрочку.

Процесс *Planning MPS3* реализует метод планирования продаж и производства, при котором все заказы выполняются только точно вовремя за счёт сверхурочной работы. Если заказ невозможно выполнить вовремя даже при максимальном выпуске продукции, планируется производство продукции за счёт сверхурочной работы в течение нескольких дней, непосредственно предшествующих сроку исполнения заказа. Если заказ невоз-

можно выполнить даже при сверхурочной работе с максимальным дополнительным выпуском от момента поступления заказа до срока его исполнения, предприятие отказывается от выполнения заказа.

Процессы *CShipmentPrepaying*, *CShipmentShipping* и *CShipmentPaying* моделируют предоплату заказа продукции, отгрузку продукции и оплату заказа в соответствии с условиями договора продажи. Процесс *CProductionStartup* предназначен для моделирования производства продукции. Процесс *CAccount* обеспечивает формирование показателей бухгалтерского баланса.

Откликами ИМ являются финансово-экономические показатели работы предприятия (чистая прибыль, рентабельность продукции и др.), а также основной план-график производства и план продаж, использующийся ИМ производственного процесса и управления запасами ресурсов.

7.6.3 Имитационная модель информационной системы производственного процесса. ИМ ИС производственного процесса (ПрП) предназначена для оценки основных параметров ПрП в условиях неопределенности и риска, связанных с возможностью возникновения сбоев и отказов в процессе выполнения производственных операций ПрО. Целью моделирования ИС ПрП является рациональный выбор параметров управления отдельными ПрО и ПрП в целом, обеспечивающий необходимый выпуск продукции при минимальных затратах.

Параметрами каждой ПрО являются: вид и объём потребляемого и производимого ресурсов; время между сбоями и отказами; время действия операции; время устранения сбоев и отказов в процессе действия операции. Учитывая, что сбои и отказы в ПрО имеют случайный характер, время между сбоями и отказами, а также время на их устранение задаются видом функции распределения, средним значением и стандартным отклонением. Параметрами ПрП являются количество видов ресурсов, количество ПрО и цены на ресурсы.

В качестве выходных данных (откликов) ИМ выступают следующие величины: длительность производственного цикла; количество запусков операций; количество сбоев и отказов операций; уровень запасов ресурсов; производственные мощности; средняя скорость изменения уровня запасов ресурсов; суммарное время ожидания операцией требуемого ресурса; суммарное время простоя операции; суммарное время устранения отказов операций; суммарное время устранения сбоев операций; себестоимость продукции. В ИМ ИС ПрП используются три типа ресурсов: складские, промежуточные, продукция.

При запуске процесса, моделирующего ПрО, проверяется наличие всех необходимых для запуска операции ресурсов. Если все ресурсы доступны, моделируется потребление ресурсов и процесс переводится на стадию производства.

Если ПрО находится на стадии производства, то через интервал времени, соответствующий длительности ПрО, процесс переходит на стадию окончания. Если в процессе работы ПрО происходит отказ, то процесс переходит на стадию запуска, определяется время следующего отказа, генерируется время устранения отказа. После устранения отказа процесс, моделирующий ПрО, запускается заново. Если в процессе работы ПрО происходит сбой, то рассчитывается время следующего сбоя, генерируется время устранения сбоя и процесс, моделирующий ПрО, переводится в состояние ожидания на время устранения сбоя. Когда процесс, моделирующий ПрО, находится на стадии окончания, увеличивается объем ресурса и процесс переводится на стадию запуска. Структуры данных, процессы ИМ ИС ПрП и взаимосвязь компонентов модели представлены на рисунке 7.30.

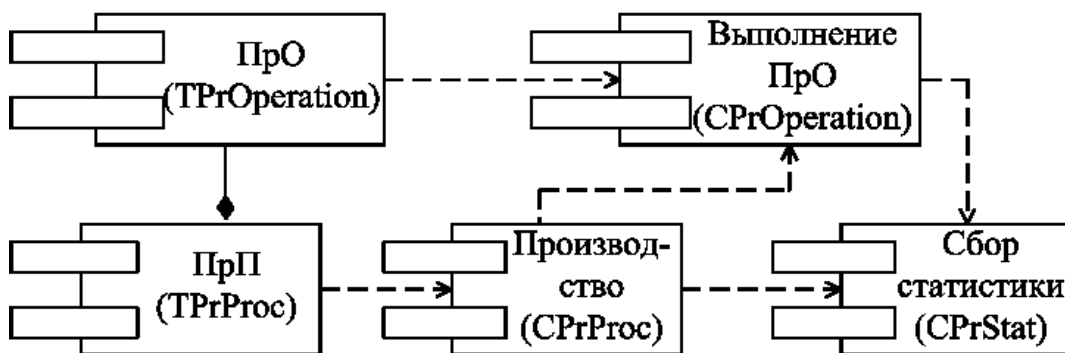


Рисунок 7.30 – Диаграмма компонентов ИМ ИС ПрП

В ИМ ИС ПрП структура данных *TPrOperation* (см. рисунок 7.30) предназначена для хранения параметров ПрО, структура данных *TPrProc* – для хранения параметров производственного процесса.

Назначение процесса *CPrProc* – задание параметров и запуск модели. Процесс *CPrOperation* моделирует процесс работы производственной операции в соответствии с алгоритмом, описанным выше. Для сбора статистики моделирования предназначен процесс *CPrStat*.

7.6.4 Имитационная модель бизнес-процесса управления запасами ресурсов. Управление запасами ресурсов (УЗР) является одной из основных функций ИС промышленного предприятия в соответствии со стандартом стандарта MRP II, поскольку эффективное УЗР обеспечивает не только сокращение складских запасов ресурсов (что, в свою очередь, позволяет сократить затраты на их хранение), но и бесперебойную работу производства и, следовательно, своевременное выполнение заказов клиентов.

В общей теории управления запасами известны методики управления запасами с фиксированным размером заказа, с постоянным уровнем запасов, с двумя уровнями запасов. При использовании методики УЗР с фиксированным размером заказа уровень запаса проверяется каждый раз при по-

треблении ресурса. Если уровень запаса ресурса становится ниже критического – подаётся заказ, размер которого является постоянной величиной.

При использовании методики УЗР с постоянным уровнем запаса заказ подаётся каждый раз при снижении уровня запаса ресурса. Величина заказа равна разности критического и имеющегося на момент заказа уровня запаса ресурсов. При использовании методики УЗР с двумя уровнями запасов заказ подаётся каждый раз при уменьшении запаса ресурса ниже критического уровня, но величина заказа равна разности между максимальным и имеющимся на момент заказа уровнями запаса ресурсов.

Особенности функционирования реального предприятия и производственного процесса накладывают ограничения на применение той или иной методики УЗР. В частности, предприятие не всегда может пополнить запасы ресурсов, когда в этом возникает необходимость, из-за отсутствия денежных средств, дефицита ресурсов на рынке и т. д.

Одним из методов выбора методики УЗР и ее параметров в условиях неопределенности является построение имитационной модели управления запасами (рисунок 7.31).

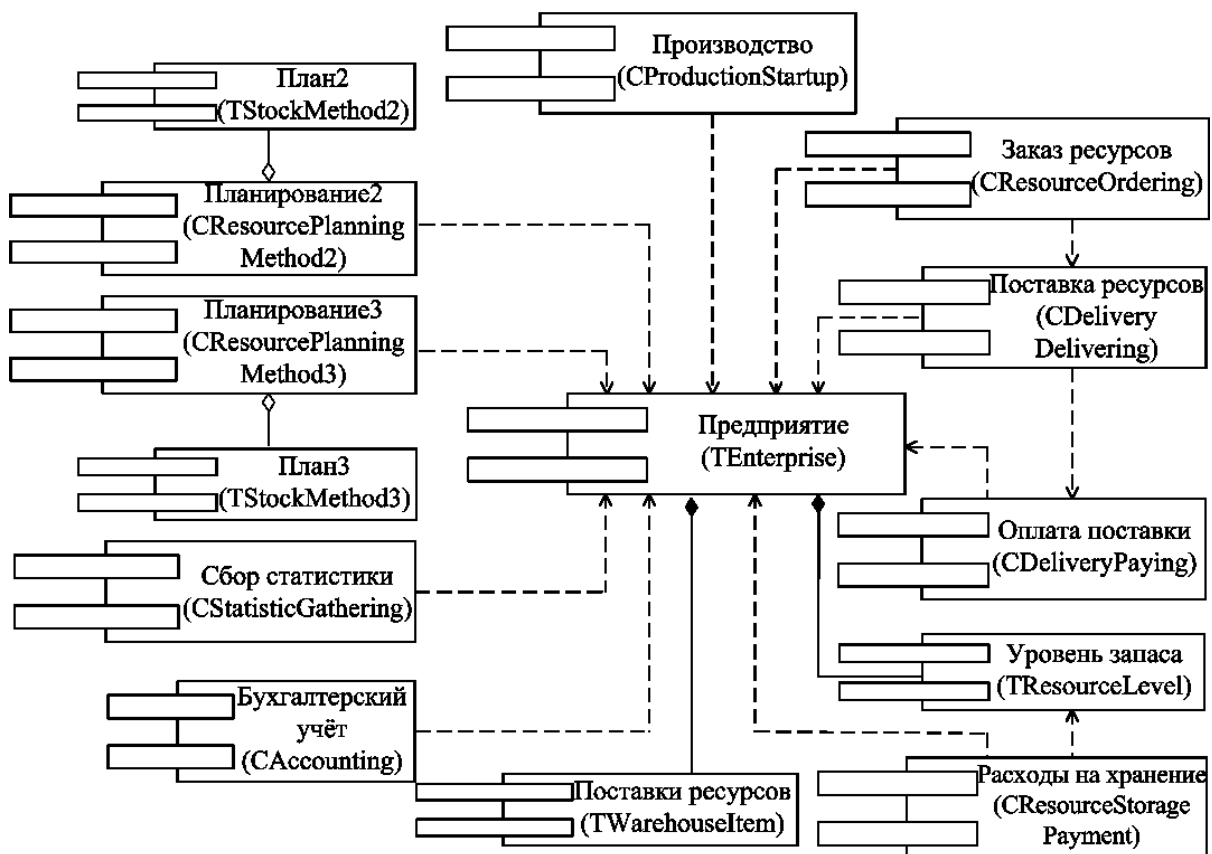


Рисунок 7.31 – Структуры данных, процессы ИМ бизнес-процесса УЗР и взаимосвязь компонентов модели

В ИМ структура *TStockMethod2* (см. рисунок 7.31) содержит параметры методики управления запасами с постоянным уровнем запаса ресур-

са: минимальный уровень запаса ресурсов, интервал проверки состояния запасов ресурсов, объем заказанных ресурсов.

Структура *TStockMethod3* содержит параметры методики управления запасами ресурсов с двумя уровнями: минимальный и максимальный уровни запаса ресурсов, интервал проверки состояния запасов, объем заказанных ресурсов. В структурах *TStockMethod2* и *TStockMethod3* объем заказанных ресурсов является промежуточным параметром, предназначенным для исключения повторного заказа ресурсов в том случае, если ресурсы заказаны, но запасы не пополнились из-за отсутствия денег на расчетном счете предприятия или из-за задержки выполнения заказа.

Структура *TResourceLevel* содержит параметры, характеризующие изменение уровня запасов ресурсов во времени: текущее время и уровень запаса ресурсов в этот момент.

Структура *TWarehouseItem* содержит информацию о поставках ресурсов на предприятие (в каком количестве и по какой цене осуществлена поставка ресурсов). Эта информация необходима для формирования показателей бухгалтерского баланса.

Структура *TEnterprise* является основным компонентом в имитационном моделировании деятельности предприятия и содержит большинство стандартных характеристик объекта моделирования (цены на ресурсы и продукцию, нормы хранения запасов ресурсов, параметры план-графика производства, интервалы между отгрузками продукции, нормы расхода ресурсов на производство продукции) и внешней среды (уровень инфляции). Структура *TEnterprise* содержит также параметры методики УЗР с фиксированным размером заказа: минимальный размер запаса (точка заказа) и объем заказанных ресурсов.

Процесс *CResourceOrdering* моделирует заказ ресурсов и бизнес-процесс УЗР, если выбрана методика с фиксированным размером заказа. В этом случае *CResourceOrdering* запускается при уровне запаса ниже точки заказа. В остальных случаях *CResourceOrdering* вызывается из процессов, моделирующих бизнес-процесс УЗР на основе других методик.

Процесс *CResourcePlanningMethod2* предназначен для моделирования бизнес-процесса планирования заказа ресурсов по методике УЗР с постоянным уровнем, процесс *CResourcePlanningMethod3* – для моделирования бизнес-процесса УЗР по методике УЗР с двумя уровнями.

Процессы *CResourcePlanningMethod2* и *CResourcePlanningMethod3* запускаются через промежутки времени, соответствующие интервалам проверки уровня запаса.

Процесс *CResourceStoragePayment* предназначен для расчета затрат, связанных с хранением ресурсов, запускается ежедневно в модельном времени.

В процессе *CResourcePriceChanging* рассчитываются цены на ресурсы в момент заказа. Данные заносятся в структуры *TWarehouseItem* и

TEnterprise.

Процессы *CDeliveryPrepaying*, *CDeliveryDelivering* и *CDeliveryPaying* предназначены для моделирования предоплаты поставки ресурса, поставки ресурса и оплаты поставки ресурса соответственно. Эти процессы запускаются из *CResourceOrdering*.

Процесс *CProductionStartup* предназначен для моделирования производства продукции. Формирование показателей бухгалтерского баланса обеспечивает процесс *CAccounting*. Сбор статистики моделирования через определённые интервалы времени выполняет процесс *CStatisticsGathering*.

7.6.5 Имитационная модель функционирования информационной системы на основе локальной вычислительной сети. ИМ работы ИС в локальной вычислительной сети (ЛВС) предназначена для оценки скорости выполнения запросов при заданной конфигурации компьютеров и известной структуре сети. При этом ИМ позволяет также выбрать рациональную конфигурацию компьютеров и структуру сети, обеспечивающую необходимую скорость выполнения запросов.

В качестве параметров ИМ работы ИС в ЛВС используются стандартные характеристики основных компонентов компьютера: процессора, оперативной памяти, накопителя на жестком диске, сетевого адаптера, устройств ввода-вывода – и показатели конфигурации сети: типы устройств и способ их соединения, скорость передачи данных. В качестве исходных данных для моделирования используются и параметры пользовательских запросов (интенсивность генерации запросов, длительность и объёмы использования ресурсов), которые задаются видом функции плотности распределения, средним значением и стандартным отклонением.

Откликами ИМ являются скорость выполнения запросов в каждом узле, пропускная способность узла и ЛВС в целом, степень загрузки устройств компьютера и сетевого оборудования. Структуры данных ИМ работы ИС в ЛВС хранят следующую информацию: *TQuery* – параметры запросов; *TResourceStatistic* – статистику моделирования; *TCore* – параметры процессора и ОЗУ; *THDD* – параметры накопителя на жестком диске; *TOutput* – параметры устройства вывода; *TNet* – параметры сетевого адаптера; *TPC* – состав узла ЛВС; *TLVS* – структуру сети ЛВС.

Перечень и назначение процессов, моделирующих работу ИС в ЛВС: *CgenerateQuery* – генерация запросов на узле сети; *CqueryProcessing* – обработка запросов на узле; *CcoreProcessing* – обработка запроса процессором и ОЗУ; *CHDDProcessing* – обработка запроса накопителем на жестком диске; *CoutputProcessing* – обработка запроса устройством вывода; *CnetProcessing* – обработка запроса сетевым адаптером; *CnetTransfer* – передача пакетов данных по сети; *Cstatistics* – сбор статистики моделирования.

Процессы ИМ реализуют типовые алгоритмы моделирования обработки пользовательских запросов в узлах ЛВС и процессы передачи ин-

формации по сети (рисунок 7.32).

Разработанная библиотека ИМ ИС позволяет выбирать состав, структуру и параметры ИС предприятия. Для этого на первом шаге выбираются проекты ИС и уровни изменения параметров каждого проекта. На втором шаге последовательно прогоняются ИМ, входящие в состав библиотеки ИМ ИС, и определяются рациональные значения параметров каждой ИМ при помощи интегрального показателя, рассчитанного как сумма произведений нормированных откликов модели на уровни их значимости для исследователя. На третьем шаге рассчитывается интегральный показатель качества функционирования ИС как сумма интегральных показателей ИМ, входящих в состав библиотеки ИМ ИС. Проект ИС с максимальным интегральным показателем считается наиболее предпочтительным.

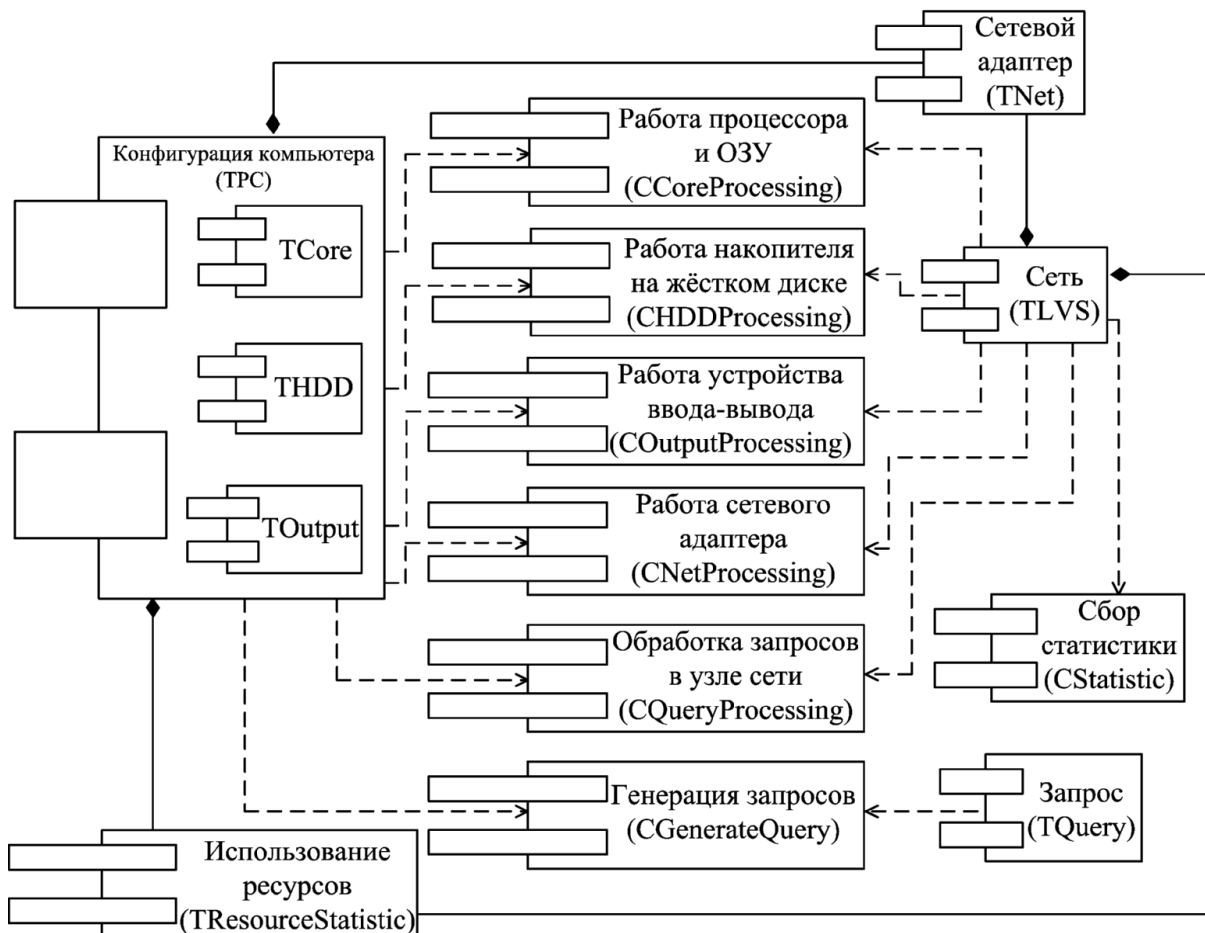


Рисунок 7.32 – Диаграмма компонентов ИМ работы ИС в ЛВС

Заключение

Основные научные результаты монографии

1 Разработана методология системного анализа, основанная на использовании имитационного моделирования многоуровневых иерархических структур в составе ERP-системы управления производственно-экономической деятельности промышленного предприятия, как развитие общего метода моделирования сложных систем на основе комплексного применения современных информационных технологий, *отличающаяся* использованием способа функционального моделирования IDEF0, эволюционных методов оптимизации, комплексным применением при программной реализации имитационных моделей технологии объектно-ориентированного анализа и проектирования, UML, XML, ADO, MPI информационных технологий, что *позволило* разработать теоретические основы и практические рекомендации для применения имитационного моделирования в структуре комплексной информационной ERP-системы управления ресурсами промышленного предприятия. *Формализация* многоуровневой системы с использованием координирующих элементов *позволяет создавать перспективные системы имитационного моделирования* для комплексных систем анализа деятельности предприятий.

2 Разработан метод построения имитационных моделей (ИМ), основанный на процессном способе имитации системы управления ресурсами промышленного предприятия путем использования комплекса взаимосвязанных имитационных моделей с высоким уровнем детализации, *отличающийся* тем, что включает *новые подходы* к составлению *вербальной модели* промышленного предприятия на основе IDEF0-концепции, учитывает иерархическую структуру управления ресурсами предприятия, использует координирующие элементы в технологии использования ИМ на каждом из уровней иерархии, итеративный характер использования ИМ других уровней производства, что обеспечивает построение модели технологии управления ресурсами предприятия, *позволяющей определить пути усовершенствования производственно-экономической системы*, о чем свидетельствуют исследования на ЗОС ОАО «Могилевхимволокно», ОАО «Моготекс», РУП «Могилевторгтехника» и др. *Выделение этапа эксплуатации* ИМ из общей технологии исследований позволяет использовать универсальные, заранее проверенные ИМ предприятия *самими специалистами предприятия*, эксплуатирующими ERP-систему управления ресурсами предприятия.

3 Разработан программно-технологический комплекс имитации сложных систем (ПТКИ) BelSim для комплексной автоматизации задач проектного моделирования технологий предприятий, использующих ERP-систему управления ресурсами предприятия, *отличающийся* тем, что

кроме стандартных возможностей система моделирования в среде ПТКИ BelSim обладает средствами проведения структурного анализа процессов, протекающих в сложных многоуровневых системах, и наличием *средств интеграции*, созданных ИМ в информационную систему предприятия для их использования в составе ERP-системы управления ресурсами предприятия, что *позволило сократить время* проведения имитационных экспериментов в среде ПТКИ BelSim не менее чем в 2,5 раза за счет автоматизации прогонов имитации согласно процедуре Монте-Карло, использования библиотеки процессов в среде BelSim и организации распределенной обработки информации в локальной вычислительной сети, *решить важную практическую задачу расчета оптимальной загрузки цехов* на ЗОС ОАО «Могилевхимволокно» для достижения минимальной себестоимости продукции с помощью средств программной системы «Экспериментатор» в среде ПТКИ BelSim.

4 Разработан метод поиска рациональных решений, основанный на *имитации* системы управления ресурсами промышленного предприятия с помощью базовой имитационной модели, в которой созданы типовые компоненты в соответствии с концепцией MRP II, являющиеся общей основой для комплексного анализа функционирования реальных предприятий и формирования показателей функционирования ERP-системы предприятия как многоуровневой иерархической системы, *отличающийся* использованием *постулата совместимости координирующих решений М. Месаровича* для определения наилучшего решения на каждом из уровней иерархии системы распределения ресурсов. При этом решение считают оптимальным, если оно удовлетворяет не только целевой функции исследуемой подсистемы, но и согласуется с глобальной целью системы.

5 Разработана технология верификации *базовой ИМ* функционирования промышленного предприятия *с помощью дополнений в текст программы* ИМ для вывода информации о ходе моделирования, дополнительно к которой *предложена* технология *верификации модели методом оценки трудоемкости алгоритмов компонентов модели* для обнаружения *непредвиденных исследователем взаимодействий* между компонентами, поскольку функционирование ИМ состоит из последовательности тех компонентов, чей порядок определен разработчиком модели в виде программной реализации ИМ и использования генераторов псевдослучайных чисел; оценка трудоемкости алгоритма принимается в качестве критерия достижения верификации имитационной модели.

6 Разработана технология определения переходного процесса базовой ИМ, *отличающаяся* тем, что используются значения тренда отклика $Y_{ict}(T_i)$ в моменты T_i , отклика $Y_{ict}(T_{i+1})$ в моменты T_{i+1} , для оценки применяются максимальное $Y_{imax}(T_i)$ и минимальное $Y_{imin}(T_{i+1})$ отклонения отклика относительно тренда. *Предложено* принимать в качестве оценки длительности переходного процесса $T_{пп}$ середину интервала $[T_i, T_{i+1}]$, если отклонения от-

клика Y_i не превышают допустимых значений на величину до 5 %.

7 Разработано программное средство реализации метода поиска рациональных решений *Оптимизатор*, реализованное в среде ПТКИ *BelSim*, отличающееся от аналогов следующей новизной: обладает специальной *процедурой адаптации модели* алгоритма случайного поиска оптимума целевой функции с помощью *генетического алгоритма сценария исследования функционирования* моделируемой системы. *Структурная и параметрическая декомпозиция* модели алгоритма случайного поиска позволяет *создать базу знаний алгоритмов* с лучшей структурой и лучшими значениями параметров при решении задач построения интеллектуальной системы поддержки принятия решений.

8 Впервые разработан алгоритм *количественной оценки координирующих действий в иерархической системе* при решении *многокритериальных задач* на основе базовой имитационной модели функционирования *предприятия*, отличающийся применением аксиоматического *принципа Парето* и *постулата совместимости М. Месаровича*, что позволило проводить комплексный анализ функционирования служб предприятия.

9 Разработаны математические модели для прогнозирования эффективности контрактов, составляющие библиотеку *ИМ контрактов*, наиболее часто используемых на промышленных предприятиях (контракты продажи, купли, дилерского обслуживания, по товарному обмену), отличающиеся наличием интервальных оценок рентабельности, которые реализованы в среде ПТКИ *BelSim* на основе *процессного способа имитации при высоком уровне детализации* процессов в ERP-системе предприятия, на основе которых разработан программный комплекс *Contract Analyzer*, реализующий методики прогнозирования эффективности и оптимизации параметров контрактов с интервальными оценками их рентабельности, обеспечивающий *автоматизацию маркетинговой деятельности* предприятий на уровне прогнозирования эффективности контрактов, что *позволяет* снизить потери от заключения малорентабельных контрактов и таким образом реализовать гибкую ценовую политику при сохранении заданного уровня рентабельности предприятия.

Рекомендации по практическому использованию результатов монографии

1 *Вербальные модели* подсистем промышленного предприятия на основе IDEF0-методологии *существенно ускоряют* построение концептуальной модели и разработку имитационной модели.

2 *Сокращение времени* проведения имитационных экспериментов (ИЭ) в среде ПТКИ *BelSim* обеспечивается за счет автоматизации прогнозов имитации согласно процедуре Монте-Карло, использования библиотеки *MPI* функций обмена данными между процессами в среде языка C++ с

помощью *MPICH*.

3 Высокопараметризованные детальные ИМ обеспечивают возможность решения задач анализа на ЭВМ состава и структуры ERP-систем управления ресурсами предприятия самими исследователями, *не являющимися специалистами* по программированию и имитации.

4 Модель технологии управления ресурсами предприятия *позволяет определить* пути усовершенствования производственно-экономической системы, о чем свидетельствуют исследования на ОАО «Могилевхимволокно», ОАО «Моготекс», РУП «Могилевторгтехника» и др. Модель позволяет сравнивать существующие технологии с проектируемой системой, поскольку существующая реальная система управления ресурсами позволяет измерить только часть информации о ее функционировании, а ИМ *существенно расширяет* состав статистик и откликов имитации.

5 *Базовая ИМ* промышленного предприятия, формируемая на основе показателей функционирования предприятия и использования статей бухгалтерского баланса (форма 1) и отчета о прибылях и убытках (форма 2), *позволяет* по результатам моделирования промышленного предприятия как многоуровневой иерархической системы *проводить его текущий анализ и принимать оперативные решения*.

6 Возможность *оперативной оценки рентабельности контрактов* предприятия обеспечивается еще на этапе заключения контрактов с помощью математических моделей, позволяющих оценить рентабельность на основе интервальных оценок эффективности контрактов. Методики исследования ИМ контрактов позволяют *корректировать аналитические модели контрактов* и таким образом *сократить время принятия решения о заключении контрактов*.

7 *Технология использования подсистемы Экспериментатор* и динамических производственно-экономических моделей завода органического синтеза позволяет экспортировать данные из комплексной информационной системы «1С: Предприятие» для постановки серии имитационных экспериментов, что *обеспечивает жизнеспособность предложенных моделей ERP-системы в реальных условиях* промышленного предприятия.

8 Показано, что *планирование производственной нагрузки для предприятия* с жестким и интегральным типом координации позволяет принимать рациональные решения в условиях быстроменяющихся рыночных отношений.

9 Библиотека ИМ бизнес-процессов планирования: продаж и производства, информационной системы производственного процесса, бизнес-процесса управления запасами ресурсов, функционирования информационной системы в локальной вычислительной сети – *существенно расширяет область применения ПТКИ BelSim*.

10 *Универсальность* ПТКИ BelSim состоит в том, что программная реализация ИМ подсистем предприятия может быть использована в каче-

стве средства автоматизации ИЭ любых сложных систем, которые можно исследовать процессным способом имитации. Результаты исследования представлены в стандартной форме и могут быть использованы при исследовании промышленных предприятий всех форм собственности.

11 *Универсальность системы Оптимизатор в среде ПТКИ BelSim и технологии его использования определяются тем, что они могут использоваться для анализа любых видов контрактов, связанных с реализацией продукции и поставкой ресурсов на предприятие. Сама библиотека ИМ контрактов также является универсальной, поскольку может применяться в качестве средства автоматизации операций оценки эффективности контрактов на промышленных предприятиях разных форм собственности.*

12 *Универсальность метода поиска рациональных решений и средств его реализации, а также технологий их использования состоит в том, что разработанная модификация ПТКИ BelSim и методики решения задач проектного моделирования можно использовать в смежных предметных областях.*

Список литературы

1 **Абрютина, М. С.** Анализ финансово-экономической деятельности предприятия : учеб.-практ. пособие / М. С. Абрютина, А. В. Грачев. – 2-е изд., испр. – М. : Дело и сервис, 2000. – 126 с.

2 **Айвазян, С. А.** Прикладная статистика. Исследование зависимостей / С. А. Айвазян, И. С. Енюков, Л. Д. Мешалкин ; под ред. С. А. Айвазяна. – М. : Финансы и статистика, 1985. – 487 с. : ил.

3 **Айвазян, С. А.** Прикладная статистика. Основы моделирования и первичная обработка данных / С. А. Айвазян, И. С. Енюков, Л. Д. Мешалкин ; под ред. С. А. Айвазяна. – М. : Финансы и статистика, 1983. – 481 с. : ил.

4 **Акофф, Р. Л.** Планирование в больших экономических системах : пер. с англ. / Р. Л. Акофф ; под ред. И. А. Ушакова. – М. : Сов. радио, 1972. – 233 с.

5 **Акулич, В.** Экспресс-анализ финансового положения организации / В. Акулич // Финансовый директор. – 2005. – № 7. – С. 44–56.

6 **Алтунин, А. Е.** Модели и алгоритмы принятия решений в нечетких условиях : монография / А. Е. Алтунин, М. В. Семухин. – Тюмень : Тюмен. гос. ун-т, 2000. – 352 с. : ил.

7 **Алиев, Р. А.** Методы и алгоритмы координации в промышленных системах управления / Р. А. Алиев, М. И. Либерзон. – М. : Радио и связь, 1987. – 208 с.

8 **Альсевич, В. В.** Оптимизация линейных экономических моделей: статические задачи : учеб. пособие / В. В. Альсевич, Р. Габасов, В. С. Глушенков. – Минск : БГУ, 2000. – 210 с. : ил.

9 **Альтшулер, И.** Мировые концепции управления производством / И. Альтшулер // PC Week. – 1997. – № 20(94). – С. 44–45.

10 **Альховик, С. А.** Генетический алгоритм в задаче оптимизации плана грузоперевозок / С. А. Альховик, А. В. Сазоненко, А. А. Ковалевич // Изв. Гомел. гос. ун-та им. Ф. Скорины. – 2006. – № 4(37). – С. 110–112.

11 **Альховик, С. А.** Имитационная модель промышленного предприятия для ERP-системы управления / С. А. Альховик, А. И. Якимов // Вестн. МГТУ. – 2004. – № 2(7). – С. 11–16.

12 **Альховик, С. А.** Моделирование маркетинговой деятельности промышленных предприятий для прогнозирования эффективности контрактов : дис. ... канд. техн. наук : 05.13.18 : защищена 13.02.04 / Альховик Сергей Александрович. – Гомель, 2004. – 325 с. – Библиогр. : с. 162–172.

13 **Альховик, С. А.** Прогнозирование финансово-экономического состояния предприятия на основе имитационного моделирования / С. А. Альховик, А. И. Якимов // Проблемы самоорганизующихся производственных и социально-экономических систем : материалы II-й междунар. науч.-практ. конф., 15–16 нояб. 2001 г. – Минск : Гипросвязь, 2001. – С. 64–67.

14 **Амуржуев, О. В.** Неплатежи: способы предотвращения и сокращения / О. В. Амуржуев, А. Е. Дороговцев. – М. : Аркаюр, 1994. – 56 с. : ил.

15 Анализ хозяйственной деятельности в промышленности : учебник / Л. А. Богдановская [и др.] ; под общ. ред. В. И. Стражева. – Минск : Выш. шк., 1995. – 363 с.

16 Аналитический обзор «Российский рынок B2B» [Электрон. ресурс]. – Вып. III, 2002. – Режим доступа : <http://www.rocit.ru/inform/index.php3?path=ibpartners>.

17 **Антонов, А. С.** Введение в параллельные вычисления: метод. пособие / А. С. Антонов. – М. : МГУ им. М. В. Ломоносова ; НИВЦ, 2002. – 69 с.

18 **Анташов В. А.** Экономический советник менеджера : учеб.-практ. пособие / В. А. Анташов, Г. В. Уварова. – Минск : Финансы, учет, аудит, 1996. – 320 с.

19 **Ахрамейко, А. А.** Агрегированная оценка финансового состояния предприятия / А. А. Ахрамейко, С. Н. Гнатюк, Д. В. Ксеневич // Проблемы экономики и организации производственных и социальных систем : материалы междунар. науч.-практ. конф., 7–8 дек. 2000 г. – Минск : Гипросвязь, 2000. – С. 51–54.

20 **Багатурова, О. С.** Математические модели маркетинга (аналитический обзор) / О. С. Багатурова, А. Т. Мамиконов // Автоматика и телемеханика. – 1991. – № 8. – С. 3–35.

21 **Бажин, И. И.** Информационные системы менеджмента / И. И. Бажин. – М. : ВШЭ, 2000. – 684 с.

22 **Балагин, В. В.** Теоретические основы автоматизированного управления : учеб. пособие для вузов / В. В. Балагин. – Минск : Выш. шк., 1991. – 252 с. : ил.

23 **Банди, Б.** Методы оптимизации. Вводный курс : пер. с англ. / Б. Банди. – М. : Радио и связь, 1988. – 128 с. : ил.

24 Методы и модели анализа данных : OLAP и Data Mining / А. А. Барсегян [и др.]. – СПб. : БХВ-Петербург, 2004. – 336 с. : ил.

25 **Батехин, С. Л.** Финансовая инженерия и оптимизация финансовых потоков / С. Л. Батехин // Финансы. – 2001. – № 1. – С. 68–69.

26 **Башаримов, В. В.** Выбор функции качества при исследовании генетического алгоритма / В. В. Башаримов, А. И. Якимов // Изв. Белорус. инженерной академии. – 2004. – № 1(17)/1. – С. 164–166.

27 **Башаримов, В. В.** Математические модели для автоматизированного проектирования рационального раскроя с применением численных методов / В. В. Башаримов, А. И. Якимов, С. А. Альховик // Современные методы проектирования машин : Респ. межведомств. сб. науч. тр. – Минск, 2004. – Т. 6, вып. 2. – С. 90–94.

28 **Башаримов, В. В.** Методика создания адаптивного метода оптимизации на основе генетического алгоритма / В. В. Башаримов, А. И. Якимов // Информационные технологии, энергетика и экономика : материалы

II-й Межрегион. науч.-техн. конф. : в 4 т. – Смоленск : ф-л ГОУ ВПО МЭИ(ТУ), 2005. – Т. 1. – С. 36–40.

29 **Башаримов, В. В.** Решение задач рационального выбора параметров систем в ПТКИ BelSim на основе алгоритма случайного поиска / В. В. Башаримов, А. И. Якимов // Интеллектуальные системы (AIS'06) и Интеллектуальные САПР (CAD'06) : тр. Междунар. науч.-техн. конф., Дивноморск, Россия, 3–10 сент. 2006 г. : в 3 т. – М. : Физматлит, 2006. – Т. 1. – С. 358–362.

30 **Бенькович, Е. С.** Практическое моделирование динамических систем / Е. С. Бенькович, Ю. Б. Колесов, Ю. Б. Сениченков. – СПб. : БХВ-Петербург, 2002. – 464 с.

31 **Бережная, Е. В.** Математические методы моделирования экономических систем : учеб. пособие / Е. В. Бережная, В. И. Бережной. – М. : Финансы и статистика, 2002. – 386 с. : ил.

32 **Бигель, Дж.** Управление производством : пер. с англ. / Дж. Бигель. – М. : Финансы и статистика, 1983. – 248 с.

33 Теория вероятностей и математическая статистика : учеб. пособие / Л. Г. Бирюкова [и др.]. – М. : ИНФРА-М, 2004. – 287 с. : ил.

34 **Богдановская, Л. А.** Порог рентабельности и запас финансовой устойчивости / Л. А. Богдановская // ФУА. – 1996. – № 4(28). – С. 25–27.

35 **Бодров, В. И.** Математические методы принятия решений : учеб. пособие / В. И. Бодров, Т. Я. Лазарева, Ю. Ф. Мартемьянов. – Тамбов : Тамбов. гос. техн. ун-т, 2004. – 124 с.

36 **Большаков, С. В.** Финансовое прогнозирование: теоретические аспекты / С. В. Большаков // Финансы. – 1999. – № 10. – С. 53–55.

37 **Большев, Л. Н.** Таблицы математической статистики / Л. Н. Большев, Н. В. Смирнов. – М. : Наука, 1983. – 416 с.

38 **Бодрунов, С. Д.** Маркетинг информационных услуг: модели управления / С. Д. Бодрунов, И. В. Максимей. – М. : Луч, 1992. – 127 с. : ил.

39 **Брейли, Р.** Принципы корпоративных финансов : пер. с англ. / Р. Брейли, С. Майерс. – М. : Олимп-Бизнес, 1997. – 1088 с. : ил.

40 **Бригхэм, Ю.** Энциклопедия финансового менеджмента : пер. с англ. / Ю. Бригхэм. – М. : Дело и сервис, 1998. – 816 с.

41 **Бринджолфсон, Э.** Исчез ли парадокс экономической эффективности информационных технологий? / Э. Бринджолфсон // Computer World. – 1994. – № 41(149). – С. 50.

42 **Букатов А. А.** Программирование многопроцессорных вычислительных систем / А. А. Букатов, В. Н. Дацюк, А. И. Жегуло. – Ростов н/Д : ЦВВР, 2003. – 208 с.

43 **Бусленко, Н. П.** Моделирование сложных систем / Н. П. Бусленко. – М. : Наука, 1968. – 356 с.

44 **Буч, Г.** Объектно-ориентированный анализ и проектирование с примерами приложений на C++ : пер. с англ. / Г. Буч. – 2-е изд. – М. : Би-

ном ; СПб. : Невский диалект, 1999. – 360 с. : ил.

45 **Ващенко, Т. В.** Математика финансового менеджмента / Т. В. Ващенко. – М. : Перспектива, 1996. – 82 с.

46 **Венцов, Н. Н.** Исследование и разработка генетических алгоритмов и автоматов адаптации для повышения эффективности доступа к данным САПР СБИС : автореф. дис. ... канд. техн. наук. – Таганрог : 2006. – 16 с.

47 **Винкоп, С.** Использование Microsoft SQL Server 7.0 : пер. с англ. / С. Винкоп. – Киев ; М. ; СПб. : Вильямс, 1999. – 816 с. : ил.

48 **Гаврилов, Д. А.** Управление производством на базе стандарта MRP II. Принципы и практика / Д. А. Гаврилов. – СПб. : Питер, 2002. – 352 с.

49 **Гейзлер, П. С.** Анализ производственной программы предприятия / П. С. Гейзлер, Н. Г. Синяк // Бухгалтерский учет. – 1998. – № 4. – С. 35–37.

50 **Герасенко, В. П.** Прогностические методы управления рыночной экономикой : учеб. пособие в 2 ч. / В. П. Герасенко. – Гомель : Альтаир, 1997. – Ч. 1. – 320 с.

51 **Гладков, Л. А.** Генетические алгоритмы : учеб. пособие / Л. А. Гладков, В. М. Курейчик, В. В. Курейчик ; под ред. В. М. Курейчика. – Ростов н/Д : Ростиздат, 2004. – 400 с.

52 **Гладкова, И.** Глобализация. Открытый мир. Коллаборативная модель управления предприятиями [Электрон. ресурс] / И. Гладкова // Оборудование. – 2002. – № 10(70). – Режим доступа : <http://home.expert.ru/obogud/02/10-02/data>.

53 **Глухов, В. В.** Экономико-математические методы и модели в планировании и управлении на металлургических предприятиях : пер. с нем. / В. В. Глухов, А. А. Спасов ; под ред. А. Г. Поршнева. – М. : Финансы и статистика, 1992. – 512 с. : ил.

54 **Голубков, Е. П.** Маркетинговые исследования: теория, методология и практика / Е. П. Голубков. – 2-е изд., перераб. и доп. – М. : Финпресс, 2000. – 464 с.

55 **Голубков, Е. П.** Прибыль как индикатор эффективности: необходимость комплексного подхода / Е. П. Голубков // Маркетинг в России и за рубежом. – 1998. – № 3. – С. 37–43.

56 **Голубков, Е. П.** Маркетинг: выбор лучшего решения / Е. П. Голубков, Е. Н. Голубкова, В. Д. Секерин ; под ред. Е. П. Голубкова. – М. : Экономика, 1993. – 222 с.

57 **Горчаков, А. А.** Компьютерные экономико-математические модели / А. А. Горчаков, И. В. Орлова. – М. : ЮНИТИ, 1995. – 134 с.

58 **Григорьева, А. А.** Математические модели задачи определения конкурентоспособности продукции / А. А. Григорьева, Ю. М. Осипов // Автоматизация и современные технологии. – 1999. – № 4. – С. 36–39.

59 **О’Лири, Д.** ERP-системы. Современное планирование и управление ресурсами предприятия. Выбор, внедрение, эксплуатация : пер. с

англ. / Д. О'Лири. – М. : Вершина, 2004. – 272 с.

60 **Гусаков, Б.** Критические цены и объемы производства / Б. Гусаков // Риск. – 1996. – № 10–12. – С. 92–95.

61 **Дейт, К.** Введение в системы баз данных : пер. с англ. / К. Дейт. – 6-е изд. – Киев ; М. ; СПб. : Вильямс, 1999. – 848 с. : ил.

62 **Денисов, А. А.** Теория больших систем управления : учеб. пособие для вузов / А. А. Денисов, Д. Н. Колесников. – Л. : Энергоиздат, 1982. – 288 с.

63 **Де Роза, К.** Планирование ресурсов, синхронизированное с покупателем (CSRP) [Электрон. ресурс] / К. Де Роза // Корпоративный менеджмент. – СПб. – 1999. – Режим доступа : <http://www.cfin.ru/vernikov/mrp/csrp.shtml>.

64 **Джонсон, Н.** Статистика и планирование эксперимента в технике и науке : пер. с англ. / Н. Джонсон, Ф. Лион. – М. : Мир, 1980. – 610 с. : ил.

65 **Дронов, Р. И.** Оценка финансового состояния предприятия / Р. И. Дронов, А. И. Резник, Е. М. Бунина // Финансы. – 2001. – № 4. – С.15–19.

66 **Дубров, А. М.** Моделирование рискованных ситуаций в экономике и бизнесе : учеб. пособие / А. М. Дубров ; под ред. Б. А. Лагоши. – М. : Финансы и статистика, 1999. – 225 с.

67 **Дудников, Е. Г.** Построение математических моделей химико-технологических объектов / Е. Г. Дудников, В. С. Балакирев. – М. : Химия, 1970. – 312 с.

68 **Дурович, А. П.** Маркетинг в предпринимательской деятельности / А. П. Дурович. – Минск : Финансы, учет и аудит, 1997. – 463 с.

69 **Елманова, Н.** Инструменты для разработчиков – обязательные и просто полезные / Н. Елманова // Компьютер-Пресс. – 2004. – № 3. – С. 64–67.

70 **Емельянов, А. А.** Имитационное моделирование экономических процессов : учеб. пособие / А. А. Емельянов, Е. А. Власова, Р. В. Дума ; под ред. А. А. Емельянова. – М. : Финансы и статистика, 2002. – 386 с. : ил.

71 **Емельянов, А. А.** Компьютерное моделирование / А. А. Емельянов, Е. А. Власова. – М. : МГУЭСИ, 2002. – 90 с. : ил.

72 **Ефимова, О. В.** Анализ безубыточности предприятия / О. В. Ефимова // Бухгалтерский учет. – 1993. – № 6. – С. 18–22.

73 **Ефимова, О. В.** Анализ финансовых результатов и эффективности использования имущества / О. В. Ефимова // Бухгалтерский учет. – 1994. – № 1. – С. 22–28.

74 **Жданов, С. А.** Экономические модели и методы в управлении / С. А. Жданов. – М. : Дело и сервис, 1998. – 176 с.

75 **Жиравок, А. Н.** Планирование эксперимента для построения математических моделей / А. Н. Жиравок // Соросовский образовательный журнал. – 2001. – Т. 7, № 9. – С. 121–127.

76 **Жучкевич, О. Н.** Планирование потребности в оборотном капитале / О. Н. Жучкевич // Проблемы экономики и организации производст-

венных и социальных систем : материалы междунар. науч.-практ. конф., 7–8 дек. 2000 г. – Минск : Гипросвязь, 2000. – С. 79–81.

77 **Замков, О. О.** Математические методы в экономике : учебник / О. О. Замков, А. В. Толстопятенко, Ю. Н. Черемных. – 2-е изд. – М. : МГУ им. М. В. Ломоносова ; Дело и Сервис, 1999. – 368 с.

78 **Захарченков, К. В.** Математические модели контрактов в маркетинговой деятельности промышленного предприятия / К. В. Захарченков, А. И. Якимов // Информационные технологии, энергетика и экономика : материалы II-й Межрегион. науч.-техн. конф. : в 4 т. – Смоленск : ф-л ГОУ ВПО МЭИ(ТУ), 2005. – Т. 1. – С. 50–53.

79 **Захарченков, К. В.** Метод моделирования и рационального выбора параметров информационной системы промышленного предприятия / К. В. Захарченков, А. И. Якимов // Инженерный вестник. – № 1(21)/1. – 2006. – С. 20–27.

80 **Захарченков, К. В.** Методика комплексного анализа информационной системы промышленного предприятия / К. В. Захарченков, А. И. Якимов // Информационные технологии, энергетика и экономика : материалы III-й Межрегион. науч.-техн. конф., 19–20 апр. 2006 г. : в 3 т. – Смоленск : ф-л ГОУ ВПО МЭИ(ТУ), 2006. – Т. 2. – С. 37–41.

81 **Захарченков, К. В.** Моделирование эффективности контракта с применением BelSim 2003 / К. В. Захарченков, Е. А. Якимов, А. И. Якимов // Информационные технологии, энергетика и экономика : материалы Межрегион. науч.-техн. конф. : в 4 т. – Смоленск : ф-л ГОУ ВПО МЭИ(ТУ), 2004. – Т. 1. – С. 31–34.

82 **Згуровський, М. З.** Основи системного аналізу / М. З. Згуровський, Н. Д. Панкратова. – Київ : ВНУ, 2007. – 544 с. : іл.

83 **Зедгинидзе, И. Г.** Планирование эксперимента для исследования многокомпонентных систем / И. Г. Зедгинидзе. – М. : Наука, 1976. – 390 с.

84 Инструкция о порядке формирования показателей бухгалтерской отчетности : утв. постановлением М-ва финансов Респ. Беларусь от 17 февр. 2004 г. № 16 (с учетом изменений, внесенных постановлением М-ва финансов Респ. Беларусь от 22 дек. 2004 г. № 178, от 31 марта 2005 г. № 40 и от 30 дек. 2005 г. № 169) : Нац. реестр правовых актов Респ. Беларусь 2004 г., № 55, 8/10745; 2005 г., № 8, 8/11946; № 57, 8/12393; 2006 г., № 21, 8/13792 // Вестн. М-ва по налогам и сборам Респ. Беларусь. – 2007. – № 8(105). – С. 7–22.

85 Инструкция по анализу и контролю за финансовым состоянием и платежеспособностью субъектов предпринимательской деятельности : утв. постановлением М-ва финансов Респ. Беларусь, М-ва экономики Респ. Беларусь и М-ва статистики и анализа Респ. Беларусь 14 мая 2004 г., № 81/128/65 : Нац. реестр правовых актов Респ. Беларусь 2004 г., № 90 // Консультант Плюс : Беларусь [Электрон. ресурс] / ЮрСпектр, Нац. Центр правовой информ. Респ. Беларусь. – Минск, 2004.

86 Как программировать на XML : пер. с англ. / Х. М. Дейтел [и др.]. – М. : Бином, 2001. – 994 с.

87 **Калянов, Г. Н.** CASE структурный системный анализ (автоматизация и применение) / Г. Н. Калянов. – М. : ЛОРИ, 1996. – 242 с. : ил.

88 **Канне, М. М.** Основы научных исследований в технологии машиностроения : учеб. пособие для вузов / М. М. Канне. – Минск : Выш. шк., 1987. – 231 с. : ил.

89 Вузовская наука: приборы, технологии, материалы, программное обеспечение : каталог III Международ. выставки / Сост. Т. А. Апанасевич [и др.]. – Минск : БГУ, 2000. – 44 с.

90 **Кельтон, В.** Имитационное моделирование. Классика CS / В. Кельтон, А. Лоу. – 3-е изд. – СПб. : Питер ; Киев : ВНУ, 2004. – 847 с. : ил.

91 **Кобелев, Н. Б.** Основы имитационного моделирования сложных экономических систем : учеб. пособие / Н. Б. Кобелев. – М. : Дело, 2003. – 336 с.

92 **Ковалев, А. И.** Анализ финансового состояния предприятия / А. И. Ковалев. – 4-е изд., испр. и доп. – М. : Центр экономики и маркетинга, 2000. – 208 с.

93 **Ковалев, В. В.** Финансовый учет и анализ: концептуальные основы / В. В. Ковалев. – М. : Финансы и статистика, 2004. – 720 с.

94 **Ковалев, М. Я.** Теория алгоритмов. Курс лекций : в 2 ч. / М. Я. Ковалев, В. М. Котов, В. В. Лепин. – Минск : БГУ, 2003. – Ч. 2. – 147 с.

95 **Кодацкий, В. П.** Методы расчета цены изделия / В. П. Кодацкий // Финансы. – 1997. – № 10. – С. 21–23.

96 **Колесник, В. Д.** Курс теории информации / В. Д. Колесник, Г. Ш. Полтырев. – М. : Наука, 1982. – 416 с.

97 **Кохно, Н.** Экспертная оценка технологии производства / Н. Кохно // Финансы, учет, аудит. – 1996. – № 10(34). – С. 29–30.

98 **Кузнецов, А. В.** Высшая математика. Математическое программирование : учеб. пособие / А. В. Кузнецов. – Минск : Выш. шк., 1994. – 286 с.

99 **Кузнецова Г. Ф.** Анализ современных методов ценообразования / Г. Ф. Кузнецова, Е. Б. Крупкова // Бухгалтерский учет. – 1998. – № 6. – С. 22–25.

100 **Курейчик, В. М.** Основные положения теории генетического поиска : конспект лекций / В. М. Курейчик, Л. А. Гладков. – Таганрог : ТРТУ, 2001. – 52 с.

101 Бухгалтерский учет / Н. И. Ладутько [и др.] ; под общ. ред. Н. И. Ладутько. – 3-е изд., перераб. и доп. – Минск : ФУАинформ, 2003. – 824 с.

102 **Ларман, К.** Применение UML и шаблонов проектирования. Введение в объектно-ориентированный анализ и проектирование / К. Ларман. – М. : Вильямс, 2001. – 496 с.

103 **Лафуенте А. М. Х.** Финансовый анализ в условиях неопределенности : пер. с исп. / А. М. Х. Лафуенте ; под ред. Е. И. Велеско,

В. В. Краснопрошина, Н. А. Лепешинского. – Минск : Тэхналогія, 1998. – 150 с.

104 **Лебег, Г.** Об измерении величин / Г. Лебег ; под ред. А. Н. Колмогорова. – М. : Учпедгиз, 1938. – 208 с.

105 **Левчук, В. Д.** Программно-технологические комплексы имитации сложных дискретных систем : монография / В. Д. Левчук, И. В. Максимей. – Гомель : ГГУ им. Ф. Скорины, 2006. – 263 с.

106 **Леоненков, А. В.** Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose / А. В. Леоненков. – М. : Интернет-Университет Информационных Технологий ; БИНОМ, 2006. – 320 с. : ил.

107 **Лешко, В.** Анализ показателей бухгалтерской отчетности / В. Лешко // Финансовый директор. – 2006. – № 7. – С. 56–62.

108 **Липсиц, И. В.** Коммерческое ценообразование / И. В. Липсиц. – М. : БЕК, 1997. – 353 с.

109 **Литвин, М. И.** О факторном методе планирования прибыли и рентабельности / М. И. Литвин // Финансы. – 1994. – № 2. – С. 29–34.

110 **Лычкина, Н. Н.** Имитационное моделирование экономических процессов : учеб. пособие / Н. Н. Лычкина. – М. : Академия, 2005. – 164 с.

111 **Мазурин А.** Корпоративные информационные системы в России: вчера, сегодня, завтра / А. Мазурин // Компьютер-Пресс. – 1999. – № 1. – С. 90–92.

112 **Макаров, С. С.** Разработка методов моделирования и информационного обеспечения при управлении медицинскими учреждениями : автореф. дис. ... д-ра техн. наук. – Таганрог : 2006. – 38 с.

113 Задачи и модели исследования операций. Технология имитации на ЭВМ и принятия решений : учеб. пособие / И. В. Максимей [и др.]. – Гомель : БелГУТ, 1999. – Ч. 3. – 150 с.

114 **Максимей, И. В.** Имитационное моделирование на ЭВМ / И. В. Максимей. – М. : Радио и связь, 1988. – 232 с. : ил.

115 **Малыхин, В. И.** Математическое моделирование экономики : учеб.-практ. пособие / В. И. Малыхин. – М. : УРАО, 1998. – 160 с.

116 **Мамаев, Е. В.** Microsoft SQL Server 2000 / Е. В. Мамаев. – СПб. : БХВ-Петербург, 2002. – 1280 с. : ил.

117 Математические модели, алгоритмы и программное обеспечение оптимизации контрактов на промышленных предприятиях : отчет о НИР (заключ.) / МГТУ ; рук. А. И. Якимов ; исполн. : С. А. Альховик [и др.]. – Могилев, 2003. – 34 с. – Библиогр. : с. 32. – № ГР 20023276. – Инв. № ГБ 030ф.

118 Математическое и программное обеспечение обработки трафика для регионального узла сети БелПАК : отчет о НИР (заключ.) / ММИ ; рук. А. И. Якимов ; исполн. : Ю. Д. Столяров [и др.]. – Могилев, 1999. – 48 с. – Библиогр. : с. 48. – № ГР 1998678. – Инв. № ХД 9779.

119 **Месарович, М.** Теория иерархических многоуровневых систем :

пер. с англ. / М. Месарович, Д. Мако, И. Такахара ; под ред И. Ф. Шахнова. – М. : Мир, 1973. – 344 с. : ил.

120 **Михайлов, А. В.** 1С: Предприятие 7.7/8.0: системное программирование / А. В. Михайлов. – СПб. : БХВ-Петербург, 2005. – 336 с. : ил.

121 **Михайлов, А. В.** Что такое MRP, MRP II, ERP, ERP II, CRM, SCM, CSRP, B2C, B2B? [Электрон. ресурс] / А. В. Михайлов. – 2003. – Режим доступа : <http://www.bcons.ru>.

122 Моделирование рискованных ситуаций в экономике и бизнесе : учеб. пособие / А. М. Дубров [и др.] ; под ред. Б. А. Лагоши. – 2-е изд., перераб. и доп. – М. : Финансы и статистика, 2001. – 224 с.

123 **Монтгомери, Д. К.** Планирование эксперимента и анализ данных : пер. с англ. / Д. К. Монтгомери. – Л. : Судостроение, 1980. – 384 с. : ил.

124 **Морозова, Т. Г.** Прогнозирование и планирование в условиях рынка : учеб. пособие для вузов / Т. Г. Морозова, А. В. Пикулькин, В. Ф. Тихонов. – М. : Юнити-Дана, 2000. – 318 с.

125 **Налимов, В. В.** Логические основания планирования эксперимента / В. В. Налимов, Т. И. Голикова. – 2-е изд., перераб. и доп. – М. : Металлургия, 1980. – 152 с.

126 **Нешиной, В. В.** Методы статистического анализа на базе обобщенных распределений : учеб.-метод. пособие / В. В. Нешиной. – Минск : Веды, 2001. – 168 с.

127 **Нешиной, В. В.** Статистический анализ и регулирование технологических процессов на базе обобщенных распределений с параметром сдвига : метод. рекомендации / В. В. Нешиной. – Минск : БелГИСС, 2001. – 40 с.

128 **Новиков, Ф. А.** Дискретная математика для программистов : учебник для вузов / Ф. А. Новиков. – 2-е изд. – СПб. : Питер, 2006. – 364 с. : ил.

129 **Ногин, В. Д.** Принятие решений в многокритериальной среде: количественный подход / В. Д. Ногин. – 2-е изд., испр. и доп. – М. : Физматлит, 2004. – 176 с.

130 **Норенков, И. П.** Эвристики и их комбинации в генетических методах дискретной оптимизации / И. П. Норенков // Информационные технологии. – 1999. – № 1. – С. 2–5.

131 **Ожегов, С. И.** Толковый словарь русского языка: 80000 слов и фразеологических выражений / С. И. Ожегов, Н. Ю. Шведова. – 2-е изд., испр. и доп. – М. : АЗЪ, 1994. – 928 с.

132 **Олзоева, С. И.** Особенности автоматизированного распределения вычислительного процесса для имитационного моделирования систем / С. И. Олзоева // Высокопроизводительные параллельные вычисления на кластерных системах : материалы 4-го междунар. науч.-практ. семинара. – Самара, СГАУ, 2004. – С. 210–214.

133 **Оливер, У. У.** Управление производством и материальными запасами в век ЭВМ : пер. с англ. / У. У. Оливер ; под ред. М. З. Штернгарца. – М. : Прогресс, 1978. – 304 с. : ил.

134 **Орешенков, А. А.** Механизм управления предприятием и его развитие в условиях рыночной экономики (на примере предприятий легкой промышленности) : автореф. дис. ... канд. экон. наук. – Минск : 1995. – 16 с.

135 **Ортнер, Г.** Применение ЭВМ на предприятиях. Критерии оптимальной организации : пер. с нем. / Г. Ортнер ; под ред. В. М. Савинкова. – М. : Прогресс, 1975. – 310 с. : ил.

136 **Панков, Д. А.** Аналитическая оценка влияния инфляционного фактора на результаты финансового анализа / Д. А. Панков // Бухгалтерский учет и анализ. – 1998. – № 4. – С. 25–30.

137 Перечень приоритетных направлений фундаментальных и прикладных научных исследований Республики Беларусь на 2006–2010 годы : утв. постановлением Совета Министров Респ. Беларусь 17 мая 2005 г., № 512 : Нац. реестр правовых актов Респ. Беларусь // Консультант Плюс : Беларусь [Электрон. ресурс] / ЮрСпектр, Нац. Центр правовой информ. Респ. Беларусь. – Минск, 2005.

138 **Петровский, В. И.** Системы автоматизации бухгалтерского учета и управления: история, современное состояние и перспективы развития / В. И. Петровский // Бухгалтер и компьютер. – 2000. – № 4. – С. 50–56.

139 **Пешкова, Е. П.** Маркетинговый анализ фирмы / Е. П. Пешкова. – М. : Ось-89, 1997. – 80 с.

140 **Поддиновский, В. В.** Парето-оптимальные решения многокритериальных задач / В. В. Поддиновский, В. Д. Ногин. – М. : Наука, 1982. – 256 с.

141 **Пол, А.** Объектно-ориентированное программирование на С++ : пер. с англ. / А. Пол. – 2-е изд. – СПб. ; М. : Невский Диалект ; БИНОМ, 1999. – 462 с. : ил.

142 **Полещук, И. И.** Ценообразование и маркетинг / И. И. Полещук. – Минск : Мисанта, 1997. – 132 с.

143 Портал ERP, CRM, MES, EAM-систем управления предприятиями и управленческого консалтинга [Электрон. ресурс]. – 2004. – Режим доступа : [http:// www.belerp.com](http://www.belerp.com). – Дата доступа 08.04.2005.

144 Правила составления и представления бухгалтерской отчетности : утв. постановлением М-ва финансов Респ. Беларусь 17 февр. 2004 г., № 16 (с учетом изменений, внесенных постановлением М-ва финансов Респ. Беларусь 22 дек. 2004 г., № 178) : Нац. реестр правовых актов Респ. Беларусь 2004 г., № 55, 8/10745; 2005 г., № 8, 8/11946 // Вестн. М-ва по налогам и сборам Респ. Беларусь. – 2007. – № 8(105). – С. 3–6.

145 **Картышов, С. В.** Marketing Analytic – программа для маркетингового анализа / С. В. Картышов // Компьютер в бухгалтерском учете и аудите. – 2000. – № 3. – С. 65–72.

146 Программное обеспечение и модели системного анализа / Под ред. Л. Н. Королева, П. С. Краснощекова. – М. : МГУ, 1991. – 190 с.

147 Программно-технологический комплекс имитации сложных систем BelSim 2. Перечень инновационных проектов третьего этапа конкурса

инноваций [Электрон. ресурс]. – Могилёв : Технологический парк, 2006. – Режим доступа : <http://www.technopark.by/konkurs/188.html>. – Дата доступа 17.06.2007.

148 **Радиевский, М. В.** Бизнес-план. Технико-экономическое планирование и обоснование финансовой стратегии предприятия / М. В. Радиевский. – Минск : Белпринт, 2000. – 264 с.

149 Разработка динамических имитационных экономико-математических моделей прогнозирования в условиях неопределенности : отчет о НИР (заключ.) / Белорус.-Рос. ун-т ; рук. А. И. Якимов ; исполн. : С. А. Альховик [и др.]. – Могилев, 2005. – 48 с. – Библиогр. : с. 41–45. – № ГР 20021541. – Инв. № ГБ 021ф.

150 Разработка динамической производственно-экономической модели завода органического синтеза ОАО «Могилевхимволокно» : отчет о НИР (заключ.) / Белорус.-Рос. ун-т ; рук. А. И. Якимов ; исполн. : С. А. Альховик [и др.]. – Могилев, 2007. – 166 с. – Библиогр. : с. 148–150. – № ГР 20066024. – Инв. № ХД 0663.

151 Разработка и исследование динамических экономико-математических моделей для прогнозирования финансового состояния предприятия на основе линейных алгебраических моделей оценки эффективности контрактов: отчет о НИР (заключ.) / МГТУ ; рук. А. И. Якимов ; исполн. : С. А. Альховик [и др.]. – Могилев, 2002. – 35 с. – Библиогр. : с. 30–31. – № ГР 2001594. – Инв. № ГБ 0110ф.

152 Разработка и реализация функциональной подсистемы АСУ «Маркетинг». Программное обеспечение контроля портфеля заказов и прогнозирования рентабельности контрактов: отчет о НИР (заключ.) / ММИ ; рук. А. И. Якимов ; исполн. : В. В. Бодунов [и др.]. – Могилев, 1999. – 34 с. – Библиогр. : с. 34. – № ГР 1998677. – Инв. № ХД 9780.

153 **Растринг, Л. А.** Адаптация сложных систем / Л. А. Растринг. – Рига : Зинатне, 1981. – 375 с. : ил.

154 **Резников, В. Б.** Разработка и исследование метода построения программного комплекса моделирования для распределенных систем с многоуровневым представлением сложных объектов : автореф. дис. ... канд. техн. наук. – Таганрог : 2006. – 16 с.

155 **Рихтер, Дж.** Программирование на платформе Microsoft .NET Framework : пер. с англ. / Дж. Рихтер. – 2-е изд., испр. – М. : Русская редакция, 2003. – 512 с. : ил.

156 **Робсон, М.** Практическое руководство по реинжинирингу бизнес-процессов : пер. с англ. / М. Робсон, Ф. Уллах ; под ред. Н. Д. Эрншвили. – М. : ЮНИТИ, 1997. – 224 с.

157 **Роджерсон, Д.** Основы СОМ : пер. с англ. / Д. Роджерсон. – М. : Русская редакция ; Channel Trading Ltd., 1997. – 376 с. : ил.

158 **Романов, А. Н.** Советующие информационные системы в экономике : учеб. пособие для вузов / А. Н. Романов, Б. Е. Одинцов. – М. :

ЮНИТИ-ДАНА, 2000. – 487 с.

159 **Рыжиков, Ю. И.** Имитационное моделирование: теория и технологии / Ю. И. Рыжиков. – М. : Альянс, 2004. – 384 с.

160 **Савицкая, Г.** Анализ деловой активности предприятия / Г. Савицкая // Финансовый директор. – 2005. – № 7. – С. 57–65.

161 **Савицкая, Г.** Анализ интенсивности и эффективности использования основных средств / Г. Савицкая // Финансовый директор. – 2006. – № 10. – С. 38–44.

162 **Савицкая, Г. В.** Анализ хозяйственной деятельности предприятия : учебник / Г. В. Савицкая. – 3-е изд., перераб. и доп. – М. : ИНФРА-М, 2006. – 425 с.

163 **Савицкая, Г. В.** Анализ эффективности деятельности предприятия: методологические аспекты / Г. В. Савицкая. – 2-е изд., испр. – М. : Новое знание, 2004. – 160 с.

164 **Савицкая, Г.** Методика расчета показателей оборачиваемости средств / Г. Савицкая // Финансовый директор. – 2006. – № 7. – С. 63–69.

165 **Савицкая, Г.** Методика факторного анализа прибыли с использованием международных стандартов / Г. Савицкая, С. Матальцкая // Финансы, учет, аудит. – 1997. – № 6(42). – С. 41–47.

166 **Савич, Л. К.** Теория вероятностей и математическая статистика : учеб. пособие / Л. К. Савич, Н. А. Смольская ; под науч. ред. О. И. Лаврова. – Минск : Адукацыя і выхаванне, 2006. – 208 с. : ил.

167 **Саймон, Г.** Науки об искусственном : пер. с англ. / Г. Саймон. – М. : Мир, 1972. – 148 с. : ил.

168 **Сакович, В. А.** Исследование операций (детерминированные методы и модели) : справ. пособие / В. А. Сакович. – Минск : Выш. шк., 1984. – 256 с.

169 **Саркисян, С. А.** Прогнозирование развития больших систем / С. А. Саркисян, Л. В. Голованов. – М. : Статистика, 1975. – 192 с. : ил.

170 **Сафронова, В. В.** Финансовый менеджмент на предприятии в условиях неплатежей / В. В. Сафронова // Финансы. – 1999. – № 7. – С. 21–22.

171 **Светуньков, С. Г.** Модели спроса и предложения в пространстве цена–объем–доход / С. Г. Светуньков. – Ульяновск : УлГУ, 1999. – 173 с.

172 **Сенченко, С. А.** Совершенствование процесса управления предприятием на основе внедрения и адаптации комплексных информационно-ориентированных систем : автореф. дис. ... канд. экон. наук. – Таганрог : 2006. – 28 с.

173 **Сильнова, С. В.** Моделирование реализации продукции в условиях рынка / С. В. Сильнова, Г. Р. Валеева // Автоматизация и современные технологии. – 2004. – № 7. – С. 40–46.

174 **Симанков, В. С.** Генетические алгоритмы и поиск оптимальных решений / В. С. Симанков, В. А. Частикова // Автоматизация и современные технологии. – 2003. – № 6. – С. 39–45.

175 **Советов, Б. Я.** Моделирование систем : учебник для вузов / Б. Я. Советов, С. А. Яковлев. – 3-е изд., перераб. и доп. – М. : Высш. шк., 2001. – 343 с. : ил.

176 **Сорокин, А. П.** Антикризисное управление персоналом предприятия / А. П. Сорокин // Проблемы самоорганизующихся производственных и социально-экономических систем : материалы II-й междунар. науч.-практ. конф., 15–16 нояб. 2001 г. – Минск : Гипросвязь, 2001. – С. 81–84.

177 **Спиридонов, А. А.** Планирование эксперимента при исследовании технологических процессов / А. А. Спиридонов. – М. : Машиностроение, 1981. – 184 с. : ил.

178 **Кузнецов, Ю. А.** Применение пакетов имитационного моделирования для анализа математических моделей экономических систем / Ю. А. Кузнецов, В. И. Перова. – Н. Новгород : НГУ им. Н. И. Лобачевского, 2007. – 98 с.

179 Статистическое моделирование и прогнозирование : учеб. пособие / Г. М. Гамбаров [и др.] ; под ред. А. Г. Гранберга. – М. : Финансы и статистика, 1990. – 383 с. : ил.

180 **Стоянова, Е. С.** Финансы маркетинга / Е. С. Стоянова. – М. : Перспектива, 1994. – 88 с. : ил.

181 **Страуструп, Б.** Язык программирования C++ : пер. с англ. / Б. Страуструп. – 3-е изд. – СПб. ; М. : Невский Диалект, 1999. – 580 с.

182 Структура приоритетных направлений научно-технической деятельности в Республике Беларусь на 2006–2010 годы : утв. Указом Президента Респ. Беларусь 06 июля 2005 г., № 315 : Нац. реестр правовых актов Респ. Беларусь // Консультант Плюс : Беларусь [Электрон. ресурс] / Юр-Спектр, Нац. Центр правовой информ. Респ. Беларусь. – Минск, 2005.

183 **Тельнов, Ю. Ф.** Реинжиниринг предприятия на основе имитационной модели / Ю. Ф. Тельнов // Корпоративный менеджмент. – 2001. – № 4. – С. 37–50.

184 Теория и практика проектирования интегрированных автоматизированных систем управления как многоуровневых иерархических структур : отчет о НИР (заключ.) / МГТУ ; рук. А. И. Якимов ; исполн. : Н. И. Подскробко [и др.]. – Могилев, 2001. – 57 с. – Библиогр. : с. 45–49. – № ГР 19962036. – Инв. № ГБ 9618.

185 Теория прогнозирования и принятия решений : учеб. пособие / Под ред. С. А. Саркисяна. – М. : Высш. шк., 1977. – 351 с. : ил.

186 ТИБО-2006. Отчет. [Электрон. ресурс]. – Минск, 2006. – Режим доступа : <http://www.technopark.by/konkurs/188.html>.

187 **Тихомиров, Ю. В.** Microsoft SQL Server 7.0: разработка приложений / Ю. В. Тихомиров. – СПб. : БХВ, 1999. – 352 с. : ил.

188 **Томашевський, В. М.** Моделювання систем / В. М. Томашевський. – Киев : ВНУ, 2005. – 352 с. : ил.

189 **Томас, Р.** Количественные методы анализа хозяйственной дея-

тельности : пер. с англ. / Р. Томас. – М. : Дело и сервис, 1999. – 432 с.

190 Имитационное моделирование. Теория и практика. ИММОД-2007 [Электрон. ресурс] : материалы III-й Всерос. науч.-практ. конф. по имитационному моделированию и его применению в науке и промышленности. – 2007. – Режим доступа : <http://www.simulation.org.ua/news.php>.

191 **Ульянов, М. В.** Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ / М. В. Ульянов. – М. : Физматлит, 2008. – 304 с.

192 **Уолш, К.** Ключевые показатели менеджмента. Как анализировать, сравнивать и контролировать данные, определяющие стоимость компании : пер. с англ. / К. Уолш. – 2-е изд. – М. : Дело, 2001. – 360 с. : ил.

193 **Файнгольд, М. Л.** Принципы расчета производственной мощности и загрузки оборудования / М. Л. Файнгольд, Д. В. Кузнецов ; под науч. ред. М. Л. Файнгольда. – Владимир : ВГПУ, 2002. – 85 с.

194 **Фомин, Г. П.** Математические методы и модели в коммерческой деятельности: учебник / Г. П. Фомин. – М. : Финансы и статистика, 2001. – 544 с. : ил.

195 **Форрестер, Дж.** Основы кибернетики предприятия (индустриальная динамика) : пер. с англ. / Дж. Форрестер ; под общ. ред. Д. М. Гвишиани. – М. : Прогресс, 1971. – 340 с. : ил.

196 **Хил Лафуенте, А. М.** Финансовый анализ в условиях неопределенности : пер. с исп. / А. М. Хил Лафуенте ; под ред. Е. И. Велесько, В. В. Краснопрошина, Н. А. Лепешинского. – Минск : Тэхналогія, 1998. – 150 с.

197 **Холзнер, С.** XML: энцикл. / С. Холзнер. – 2-е изд. – СПб. : Питер, 2004. – 1101 с. : ил.

198 **Цимбал, А.** Технология CORBA для профессионалов / А. Цимбал. – СПб. : Питер, 2001. – 624 с.

199 **Черемных, О. С.** Стратегический корпоративный реинжиниринг: процессно-стоимостной подход к управлению бизнесом : учеб. пособие / О. С. Черемных, С. В. Черемных. – М. : Финансы и статистика, 2005. – 238 с. : ил.

200 **Шеремет, А. Д.** Методика финансового анализа / А. Д. Шеремет, Р. С. Сайфулин, Е. В. Негашева. – М. : ИНФРА-М, 2000. – 208 с. : ил.

201 **Шенк, Х.** Теория инженерного эксперимента : пер. с англ. / Х. Шенк ; под ред. чл.-корр. АН СССР Н. П. Бусленко. – М. : Мир, 1972. – 380 с.

202 **Шилейко, А. В.** Введение в информационную теорию систем / А. В. Шилейко, В. Ф. Кочнев, Ф. Ф. Химушин : под ред. А. В. Шилейко. – М. : Радио и связь, 1985. – 280 с. : ил.

203 **Шпаковский, Г. И.** Программирование многопроцессорных систем в стандарте MPI / Г. И. Шпаковский, Н. В. Серикова. – Минск : БГУ, 2002. – 323 с.

204 **Щекалев, И.** Автоматизация предприятия: модель, архитектура, инструменты / И. Щекалев // PC Week. – 1997. – № 20(94). – С. 44–45.

205 **Эддопс, М.** Методы принятия решений : пер. с англ. / М. Эддопс, Р. Стэнсфилд ; под ред. И. И. Елисеевой. – М. : Аудит ; ЮНИТИ, 1997. – 590 с.

206 **Элджер, Дж.** С++: библиотека программиста : пер. с англ. / Дж. Элджер. – СПб. : Питер, 1999. – 320 с. : ил.

207 **Энтони, Р.** Учет: ситуации и примеры : пер. с англ. / Р. Энтони, Дж. Рис ; под ред. А. М. Петрачкова. – М. : Финансы и статистика, 1993. – 560 с. : ил.

208 **Юхневич, И. Н.** Оптимизация производственных запасов в условиях перехода к рынку Республики Беларусь : автореф. дис. ... канд. экон. наук. – Минск : 1995. – 20 с.

209 **Якимов, А. И.** Автоматизация маркетинговой деятельности предприятия в условиях современного рынка / А. И. Якимов, С. А. Альховик // Новые конкурентоспособные и прогрессивные технологии, машины и механизмы в условиях современного рынка : материалы междунар. науч.-техн. конф., 18–19 мая 2000 г. – Могилев : ММИ, 2000. – С. 513.

210 **Якимов, А. И.** Автоматизация производственных процессов в многоуровневых иерархических системах / А. И. Якимов, С. А. Альховик // Современные направления развития производственных технологий и робототехника : материалы междунар. науч.-техн. конф., 22–23 апр. 1999 г. – Могилев : ММИ, 1999. – С. 76.

211 **Якимов, А. И.** Автоматизация эксперимента на примере исследования генетического алгоритма / А. И. Якимов, В. В. Башаримов, С. А. Альховик // Автоматизация и современные технологии. – 2006. – № 5. – С. 3–9.

212 **Якимов, А. И.** Автоматизированная технология прогнозирования эффективности контрактов / А. И. Якимов, С. А. Альховик // Перспективные технологии, материалы и системы : сб. науч. тр. – Могилев, 2001. – С. 502–507.

213 **Якимов, А. И.** Анализ эффективности контрактов с использованием программного комплекса Contract Analyzer / А. И. Якимов, С. А. Альховик // Маркетинг в России и за рубежом. – 2001. – № 2. – С. 26–33.

214 **Якимов, А. И.** Библиотека имитационных моделей информационной системы промышленного предприятия / А. И. Якимов, К. В. Захарченков // Вестн. Брест. гос. техн. ун-та. Физика, математика, информатика. – 2005. – № 5(35). – С. 74–79.

215 **Якимов, А. И.** Верификация имитационной модели методом оценки трудоемкости алгоритма / А. И. Якимов // Изв. Гомел. гос. ун-та им. Ф. Скорины. – 2008. – Ч. 1, № 5(50). – С. 173–177.

216 **Якимов, А. И.** Выбор оптимальной стратегии управления производством на основе статистической обработки данных и моделирования / А. И. Якимов // Интеллектуальный анализ информации ИАИ-2006 : сб. тр. VI-й междунар. конф., 16–19 мая 2006 г. – Киев : Просвіта, 2006. – С. 334–343.

217 **Якимов, А. И.** Выбор стратегии межуровневой координации в

сложной производственно-экономической системе на основе имитационного моделирования / А. И. Якимов, С. А. Альховик // Интеллектуальные системы (AIS'06) и Интеллектуальные САПР (CAD'06) : тр. междунар. науч.-техн. конф., Дивноморск, Россия, 3–10 сент. 2006 г. : в 3 т. – М. : Физматлит, 2006. – Т. 2. – С. 239–245.

218 **Якимов, А. И.** Динамическая производственно-экономическая модель завода органического синтеза / А. И. Якимов // *Материалы, оборудование и ресурсосберегающие технологии : материалы междунар. науч.-техн. конф.*, 20–21 апр. 2006 г. : в 3 ч. – Могилев : Белорус.-Рос. ун-т, 2006. – Ч. 2. – С. 203–204.

219 **Якимов, А. И.** Имитационная модель финансово-хозяйственной деятельности предприятия и банка / А. И. Якимов, А. П. Тренихин, К. В. Захарченков // *Прогрессивные технологии, технологические процессы и оборудование : материалы междунар. науч.-техн. конф.*, 15–16 мая 2003 г. – Могилев : МГТУ, 2003. – С. 113–115.

220 **Якимов, А. И.** Имитационная модель функционирования информационной системы промышленного предприятия на уровне корпоративной сети / А. И. Якимов, К. В. Захарченков, А. Б. Падалка // *Изв. Гомел. гос. ун-та им. Ф. Скорины.* – 2006. – № 4(37). – С. 136–138.

221 **Якимов, А. И.** Имитационное моделирование в ERP-системах управления / А. И. Якимов, С. А. Альховик. – Минск : Белорус. наука, 2005. – 197 с. : ил.

222 **Якимов, А. И.** Имитационное моделирование в комплексных информационных системах / А. И. Якимов // *Изв. Гомел. гос. ун-та им. Ф. Скорины.* – 2003. – № 3(18). – С. 106–110.

223 **Якимов, А. И.** Имитационное моделирование в контуре управления информационной системы промышленного предприятия / А. И. Якимов // *Актуальные проблемы математики, механики, информатики : материалы междунар. науч.-метод. конф., посвящ. 90-летию высш. матем. образования на Урале.* – Пермь : Перм. гос. ун-т, 2006. – С. 236–237.

224 **Якимов, А. И.** Имитационное моделирование в системе принятия решений на промышленном предприятии / А. И. Якимов // *Проблемы управления и приложения (техника, производство, экономика) : тр. междунар. конф.* : в 3 т. – Минск : Ин-т математики НАН Беларуси, 2005. – Т. 3. – С. 161–165.

225 **Якимов, А. И.** Имитационное моделирование процессным способом эффективности контракта по взаимозачету на основе BelSim 2003 / А. И. Якимов, К. В. Захарченков, С. А. Альховик // *Изв. Гомел. гос. ун-та им. Ф. Скорины.* – 2004. – № 6(27). – С. 184–188.

226 **Якимов, А. И.** Имитационное моделирование процессным способом эффективности контракта продажи на основе ПТКИ BELSIM / А. И. Якимов // *Изв. Гомел. гос. ун-та им. Ф. Скорины.* – 2006. – № 4(37). – С. 139–141.

227 **Якимов, А. И.** Имитационное моделирование сложного дис-

кретного производственного процесса на основе ПТКИ BELSIM / А. И. Якимов, К. В. Захарченко, Р. В. Петров // Изв. Гомел. гос. ун-та им. Ф. Скорины. – 2005. – № 5(32). – С. 135–137.

228 **Якимов, А. И.** Информационное обеспечение прогнозирования эффективности контрактов на предприятиях текстильной промышленности / А. И. Якимов, С. А. Альховик, А. И. Степанов // Текстильная промышленность. – 2002. – № 9. – С. 34–35.

229 **Якимов, А. И.** Исследование методов кластерного анализа для решения задач оптимизации в имитационном моделировании / А. И. Якимов, В. В. Башаримов, Е. М. Борчик // Интеллектуальный анализ информации ИАИ-2007 : материалы VII-й междунар. конф., Киев, 15–18 мая 2007 г. – Киев : Просвіта, 2007. – С. 373.

230 **Якимов, А. И.** К вопросу об интеграции программных информационных систем / А. И. Якимов, А. И. Степанов, О. Г. Добромислин // Изв. Белорус. инженерной академии. – 2004. – № 1(17)/2. – С. 60–62.

231 **Якимов, А. И.** К вопросу о поиске рациональных решений в ERP-системе управления промышленным предприятием / А. И. Якимов // Динамические системы: устойчивость, управление, оптимизация : тез. докл. междунар. конф. к 90-летию со дня рождения академика Е. А. Барбашина, Минск, 29 сент.–4 окт. 2008 г. – Минск : Ин-т математики НАН Беларуси, 2008. – С. 168–170.

232 **Якимов, А. И.** Метод имитационного моделирования многоуровневых иерархических систем / А. И. Якимов // Электронное моделирование. – 2008. – № 5. – С. 69–80.

233 **Якимов, А. И.** Метод моделирования координирующих решений в многоуровневой иерархической системе / А. И. Якимов // Математическое и имитационное моделирование. МОДС 2007 : тез. докл. II-й науч.-практ. конф. с междунар. участием, 25–29 июня 2007 г. – Киев : Ин-т проблем математ. машин и систем НАН Украины, 2007. – С. 111–114.

234 **Якимов, А. И.** Методика анализа влияния платежных операций на рентабельность товарной продукции / А. И. Якимов, С. С. Михалюк, С. А. Альховик // Бухгалтерский учет и анализ. – 1999. – № 5(29). – С. 29–33.

235 **Якимов, А. И.** Методика выбора значений параметров алгоритма оптимизации для функций определенного класса / А. И. Якимов, В. В. Башаримов, Е. М. Борчик // Системный анализ и информационные технологии : материалы X междунар. науч.-техн. конф., Киев, 20–24 мая 2008 г. – Киев : КПИ, 2008. – С. 156.

236 **Якимов, А. И.** Методика кластеризации целевых функций в ПТКИ BelSim / А. И. Якимов, В. В. Башаримов, Е. М. Борчик // Информационные технологии в профессиональной деятельности и научной работе : материалы Всерос. науч.-практ. конф. с междунар. участием, 18–19 апр. 2008 г. : в 2 ч. – Йошкар-Ола : Марий. гос. техн. ун-т, 2008. – Ч. 2. – С. 203–207.

237 **Якимов, А. И.** Методика оценки эффективности индексации амортизационных отчислений в учетной политике предприятий / А. И.

Якимов, С. А. Альховик // Бухгалтерский учет и анализ. – 2001. – № 9(57). – С. 38–42.

238 Якимов, А. И. Методы и средства имитационного моделирования производственно-экономических систем / А. И. Якимов // Математичні машини і системи. – 2007. – № 1. – С. 62–66.

239 Якимов, А. И. Методы и средства имитационного моделирования производственно-экономических систем / А. И. Якимов // Математичне та імітаційне моделювання. МОДС 2006 : тез. перша наук.-практ. конф. з міжнар. участю. – Киев : Ін-т проблем математ. машин и систем НАН України, 2006. – С. 75–76.

240 Якимов, А. И. Моделирование влияния платежных операций на рентабельность товарной продукции с учетом инфляционного фактора / А. И. Якимов // Бухгалтерский учет и анализ. – 2000. – № 7(43). – С. 34–37.

241 Якимов, А. И. Моделирование в системе управления химическим предприятием. Задача распределения производственной нагрузки на уровне объединения / А. И. Якимов // Компьютеризация в химической и смежных отраслях промышленности : материалы междунар. науч.-техн. семинара. – Новополоцк : ПГУ, 2006. – С. 52–57.

242 Якимов, А. И. Моделирование в системе управления химическим предприятием. Задача распределения производственной нагрузки на уровне завода / А. И. Якимов, К. В. Захарченков // Компьютеризация в химической и смежных отраслях промышленности : материалы междунар. науч.-техн. семинара. – Новополоцк : ПГУ, 2006. – С. 47–51.

243 Якимов, А. И. Моделирование в системе управления химическим предприятием. Задача оптимизации распределения производственной нагрузки / А. И. Якимов, С. А. Альховик // Компьютеризация в химической и смежных отраслях промышленности : материалы междунар. науч.-техн. семинара. – Новополоцк : ПГУ, 2006. – С. 44–46.

244 Якимов, А. И. Моделирование инновационных технологий обучения в управлении персоналом / А. И. Якимов // Опыт и проблемы организации научно-исследовательской работы студентов : сб. науч. тр. – Минск, 2003. – С. 136–138.

245 Якимов, А. И. Моделирование интервальной оценки контракта при реализации товарной продукции с отсрочкой платежа / А. И. Якимов // Перспективные технологии, материалы и системы : сб. науч. тр. – Могилёв, 2003. – С. 440–446.

246 Якимов, А. И. Моделирование информационной системы предприятия для повышения ее эффективности / А. И. Якимов, С. А. Альховик, К. В. Захарченков // Изв. Белорус. инженерной академии. – 2003. – № 1(15)/1. – С. 299–301.

247 Якимов, А. И. Моделирование производственного процесса для поиска путей снижения себестоимости продукции / А. И. Якимов, К. В. Захарченков // Изв. Белорус. инженерной академии. – 2005. – № 1(19)/2. –

С. 36–39.

248 **Якимов, А. И.** Модернизация программно-технологического комплекса имитации сложных систем BelSim для организации распределенных вычислений / А. И. Якимов // Информатика. – 2008. – № 2(18). – С. 137–142.

249 **Якимов, А. И.** Настройка параметров генетического алгоритма для решения практических оптимизационных задач / А. И. Якимов, С. А. Альховик, В. В. Башаримов // Изв. Белорус. инженерной академии. – 2003. – № 1(15)/2. – С. 332–334.

250 **Якимов, А. И.** Об исследовании длительности переходного процесса в имитационной модели производственной деятельности промышленного предприятия / А. И. Якимов // Труды БГТУ. Сер. VI, Физ.-мат. науки и информ. – 2008. – Вып. XVI. – С. 116–120.

251 **Якимов, А. И.** Об одной методике принятия решений в иерархической структуре предприятия / А. И. Якимов // Изв. Гомел. гос. ун-та им. Ф. Скорины. – 2007. – № 5(44). – С. 132–137.

252 **Якимов, А. И.** Особенности интеграции имитационной модели объекта с информационной системой предприятия / А. И. Якимов // Изв. Гомел. гос. ун-та им. Ф. Скорины. – 2005. – № 5(32). – С. 90–92.

253 **Якимов, А. И.** Оценка влияния платежных операций на эффективность контракта / А. И. Якимов, С. А. Альховик // Проблемы экономики и организации производственных и социальных систем : материалы междунар. науч.-техн. конф. – Минск : Гипросвязь, 2000. – С. 57–59.

254 **Якимов, А. И.** Оценка длительности переходного процесса имитационной модели с трендом отклика в стационарном режиме / А. И. Якимов // Инновационные технологии управления в экономике–2007 : материалы респ. науч.-практ. конф., Брест, 24–25 апр. 2007 г. – Брест : БрГУ, 2007. – С. 45.

255 **Якимов, А. И.** Оценка эффективности контрактов с использованием программного обеспечения Contract Analyzer / А. И. Якимов, С. А. Альховик // Компьютер в бухгалтерском учете и аудите. – 2000. – № 3. – С. 60–63.

256 **Якимов, А. И.** Применение имитационного моделирования в современных концепциях управления предприятием / А. И. Якимов // Изв. Гомел. гос. ун-та им. Ф. Скорины. – 2004. – № 4(25). – С. 77–80.

257 **Якимов, А. И.** Применение математической теории систем / А. И. Якимов // Вес. АН Беларусі. Сер. фіз-мат. навук. – 1999. – № 3. – С. 132.

258 **Якимов, А. И.** Применение программного комплекса Contract Analyzer на промышленных предприятиях / А. И. Якимов, С. А. Альховик // Вузовская наука, приборы, технологии, материалы, программное обеспечение : материалы 3-й междунар. науч.-практ. конф., 25–27 окт. 2000 г. – Минск : БГУ. – С. 206–211.

259 **Якимов, А. И.** Проблема выбора корпоративной информационной системы промышленного предприятия / А. И. Якимов, К. В. Захарченков // *Материалы, оборудование и ресурсосберегающие технологии : материалы междунар. науч.-техн. конф.*, 22–23 апр. 2004 г. : в 3 ч. – Могилев : Беларус.-Рос. ун-т, 2004. – Ч. 3. – С. 107–108.

260 **Якимов, А. И.** Прогнозирование развития предприятия в условиях финансово-экономической напряженности / А. И. Якимов // *Создание и применение высокоэффективных наукоемких ресурсосберегающих технологий, машин и комплексов : материалы междунар. науч.-техн. конф.*, 25–26 окт. 2001 г. – Могилёв : МГТУ, 2001. – С. 555–556.

261 **Якимов, А. И.** Программное обеспечение интеграции имитационной модели с комплексной информационной системой / А. И. Якимов, К. В. Захарченков // *Докл. БГУИР.* – 2008. – № 2(32). – С. 111–117.

262 **Якимов, А. И.** Программно-технологический комплекс имитации сложных систем BelSim 2003 / А. И. Якимов // *Новые материалы, оборудование и технологии в промышленности : материалы респ. науч.-техн. конф.*, 29 янв. 2004 г. – Могилев : Беларус.-Рос. ун-т, 2004. – С. 3–4.

263 **Якимов, А. И.** Программный комплекс для прогнозирования финансово-экономического состояния и оптимизации управления предприятием / А. И. Якимов, С. А. Альховик // *Наука и инновации в регионах Беларуси : материалы респ. науч.-практ. конф.*, 19–20 дек. 2001 г. – Могилев : ИТМ НАН Беларуси, 2002. – С. 146–149.

264 **Якимов, А. И.** ПТКИ BelSim 2003 для поиска эффективных решений на основе эволюционного моделирования / А. И. Якимов, С. А. Альховик, В. В. Башаримов // *Белорус.-пол. науч.-практ. семинар : тез. докл. (г. Ольштын, Польша, 4–7 окт. 2004 г.)*. – Минск : БелИСА, 2004. – С. 88–89.

265 **Якимов, А. И.** ПТКИ BelSim 2003 для принятия решений в маркетинговой деятельности промышленных предприятий / А. И. Якимов, С. А. Альховик, К. В. Захарченков // *Белорус.-пол. науч.-практ. семинар : тез. докл. (г. Ольштын, Польша, 4–7 окт. 2004 г.)*. – Минск : БелИСА, 2004. – С. 90–91.

266 **Якимов, А. И.** Развитие ПТКИ BelSim для реализации функции моделирования в ERP-системах управления / А. И. Якимов, С. А. Альховик, К. В. Захарченков // *Перспективные технологии, материалы и системы : сб. науч. тр.* – Могилёв, 2005. – С. 415–421.

267 **Якимов, А. И.** Разработка технологии решения задач оптимизации моделей различных объектов машиностроения / А. И. Якимов, В. В. Башаримов, С. А. Альховик // *Вестн. МГТУ.* – 2004. – № 1(6). – С. 196–199.

268 **Якимов, А. И.** Технология имитационного моделирования сложных многоуровневых систем / А. И. Якимов // *Моделирование-2006 : материалы междунар. конф.* – Киев : Ин-т проблем моделир. в энергетике им. Г. Е. Пухова НАН Украины, 2006. – С. 455–460.

269 **Якимов, А. И.** Учет и анализ влияния инфляционного фактора на рентабельность товарной продукции / А. И. Якимов, С. А. Альховик //

Бухгалтерский учет и анализ. – 1999. – № 2(26).– С. 41–43.

270 21st European Conference on Modelling and Simulation ECMS 2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.scs-europe.net/conf/ecms2007>.

271 6th EUROSIM Congress on Modelling and Simulation [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.eurosim2007.org/Scope.html>.

272 19th European Modeling and Simulation Symposium (Simulation in Industry) EMSS 2007 [Электрон. ресурс]. – 2007. – Режим доступа : http://www.ingegneria.unical.it/impiantiindustriali/index_file/EMSS2007.htm.

273 8th International Conference Computer Data Analysis and Modeling: Complex Stochastic Data and Systems CDAM 2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.cdam.bsu.by/>

274 8th International Conference Computer Data Analysis and Modeling: Complex Stochastic Data and Systems CDAM 2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.cdam.bsu.by/>

275 **Al-Aomar R.** A Robust Simulation-Based Multicriteria Optimization Methodology / R. Al-Aomar // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 1931–1939.

276 **Al-Khamis, T. M.** Simulation-based Optimization for Repairable Systems Using Particle Swarm Algorithm / T. M. Al-Khamis, M. A. Ahmed // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 857–861.

277 **Al-Shihabi, S.** Subcontracting in a Make-to-stock Production System, IPA Gradients for a SFM / S. Al-Shihabi // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 819–824.

278 Applied Simulation and Modelling ASM 2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.iasted.org/conferences/home-581.html>.

279 Enhancing Business Process Management with Simulation Optimization / J. April [etc.] // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 642–649.

280 Practical Introduction to Simulation Optimization / J. April [etc.] // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 71–78.

281 Expanding our Horizons in Verification, Validation, and Accreditation Research and Practice / O. Balci [etc.] // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 653–663.

282 The Future of the Simulation Industry / J. Banks [etc.] // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 2033–2043.

283 **Barton, R. R.** Designing Simulation Experiments [Электрон. ресурс] / R. R. Barton // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 73–79. – Режим доступа : <http://www.informs-sim.org/wsc04papers/010.pdf>. – Дата доступа : 28.12.2007.

284 Panel: Simulation – Past, Present and Future / R. R. Barton [etc.] // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 2044–2050.

285 Future of simulation optimization / J. Boesel [etc.] // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 1466–1469.

286 **Brady, T. F.** Simulation Data Mining: A New Form of Computer Simulation Output / T. F. Brady, E. Yellig // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 285–289.

287 **Buchholz, P.** Enhancing Evolutionary Algorithms with Statistical Selection Procedures for Simulation Optimization / P. Buchholz, A. Thümmler // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 842–852.

288 **Buss, A.** Component based Simulation Modeling with Simkit / A. Buss // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 243–249.

289 **Carson, J. S.** Introduction to Modeling and Simulation / J. S. Carson // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 7–13.

290 **Carson, J. S.** Model Verification and Validation / J. S. Carson // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 52–58.

291 **Chung, C. A.** Simulation modeling handbook : a practical approach / C. A. Chung. – Boca Raton : CRC Press LLC, 2004. – 574 p.

292 **Chwif L.** A Prescriptive Technique for V&V of Simulation Models When No Real-life Data Are Available [Электрон. ресурс] / L. Chwif, L. M. Shimada // Proceedings of 2006 Winter Simulation Conference. – 2006. – P. 911–918. – Режим доступа : <http://www.informs-sim.org/wsc06papers/114.pdf>. – Дата доступа : 29.12.2007.

293 **De Jong, K. A.** An analysis of behavior of a class of genetic adaptive systems / K. A. De Jong // Doctoral dissertation, University of Michigan, University Microfilms, 1975. – № 76–9381.

294 The Current and Future Status of Simulation Software (Panel) / R. Diamond [etc.] // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 1633–1640.

295 **Ding, H.** A Simulation-Optimization Approach Using Genetic Search for Supplier Selection / H. Ding, L. Benyoucef, X. Xie // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 1260–1267.

296 **Drew, S.** Quasi-Monte Carlo Strategies for Stochastic Optimization / S. Drew, T. Homem-de-Mello // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 774–782.

297 **Drewry, D. T.** An Optimization-Based Multi-Resolution Simulation

Methodology / D. T. Drewry, P. F. Reynolds Jr., W. R. Emanuel // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 467–475.

298 **Eskandari, H.** Multiobjective Simulation Optimization Using an Enhanced Genetic Algorithm / H. Eskandari, L. Rabelo, M. Mollaghasemi // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 833–841.

299 **Fishwick, P. A.** Using XML for Simulation Modeling / P. A. Fishwick // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 616–622.

300 **Fowler, M.** UML distilled: a brief guede to the standard object modeling language / M. Fowler, K. Scott. – 2nd ed. – Addison Wesley Longman, 2000. – 195 p.

301 **Fu, M. C.** Simulation Optimization: A Review, New Developments, and Applications / M. C. Fu, F. Glover, J. April // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 83–95.

302 **Gehlsen, B.** A Framework for Distributed Simulation Optimization / B. Gehlsen, B. Page // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 508–514.

303 **Goldman, D.** Statistical Selection of the Best System / D. Goldman, S.-H. Kim, B. L. Nelson // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 178–187.

304 **Gupta, A.** Input Modeling Using Quantile Statistical Methods [Электрон. ресурс] / A. Gupta, E. Parzen // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 728–736. – Режим доступа : <http://www.informs-sim.org/wsc04papers/089.pdf>.

305 **Hanisch, A.** Initialization of Online Simulation Models / A. Hanisch, J. Tolujew, T. Schulze // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 1795–1803.

306 **Heavey, C.** Process Modelling Support for the Conceptual Modelling Phase of a Simulation Project / C. Heavey, J. Ryan // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 801–808.

307 **Hedlund, H. E.** A Genetic Algorithm and an Indifference-Zone Ranking and Selection Framework for Simulation Optimization / H. E. Hedlund, M. Mollaghasemi // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 417–421.

308 **Hu J.** Stochastic Optimization Using Model Reference Adaptive Search / J. Hu, M. C. Fu, S. I. Marcus // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 811–818.

309 **SASSY: A Design for a Scalable Agent-Based Simulation System Using a Distributed Discrete Event Infrastructure / M. Hybinette [etc.] // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 926–933.**

310 **IDEF0**. FIPS Integration Definition for Function Modeling (IDEF0), Federal Information Processing Standards Publication 183, Computer Systems Laboratory, National Institute of Standards and Technology. 1993.

311 **INFORMS Simulation Society Research Workshop**. Theme: Simulation for better decisions in an uncertain world / [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.insead.edu/issrw/>

312 **Jakimow, A. I.** Kompleks programowo-technologiczny BelSim ds. realizacji funkcji modelowania w systemach ERP [Электрон. ресурс] / A. I. Jakimow // Białorusko-Rosyjski Państwowy Uniwersytet, Mogilew. – 2004. – Режим доступа : <http://www.belembassy.org/poland/pol/index.php?id=575>.

313 **Kelton, W. D.** Experimental Design for Simulation / W. D. Kelton, R. R. Barton // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 59–65.

314 **Kelton, W. D.** Simulation with Arena / W. D. Kelton, R. P. Sadowski, D. A. Sadowski. – 2nd Ed. – New York : McGrawHill, 2001. – 632 p.

315 **Kilgore, R. A.** Multi-Language, Open-Source Modeling Using the Microsoft .Net Architecture / R. A. Kilgore // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 629–633.

316 **Kim, S.-H.** Selecting the Best System: Theory and Methods / S.-H. Kim, B. L. Nelson // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 101–112.

317 **Kirkpatrick, S.** Optimization by Simulated Annealing / S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi // Science. – 1983. – № 220. – P. 671–680.

318 **Krahl, D.** Debugging Simulation Models / D. Krahl // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 62–68.

319 Introduction to Modeling and Generating Probabilistic Input Processes for Simulation / E. K. Lada [etc.] // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 41–55.

320 A Methodological Framework for Business-Oriented Modeling of IT Infrastructure [Электрон. ресурс] / A. Landau [etc.] // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 474–482. – Режим доступа : <http://www.informs-sim.org/wsc04papers/057.pdf>. – Дата доступа : 18.11.2006.

321 **Law, A. M.** How to Build Valid and Credible Simulation Models / A. M. Law // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 24–32.

322 **Law, A. M.** Simulation-Based Optimization / A. M. Law, M. G. McComas // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 41–44.

323 **Law, A. M.** Statistical Analysis of Simulation Output Data: The Practical State of the Art [Электрон. ресурс] / A. M. Law // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 67–72. – Режим доступа : <http://www.informs-sim.org/wsc04papers/009.pdf>. – Дата доступа : 14.09.2005.

324 **L'Ecuyer, P.** Quasi-Monte Carlo Methods for Simulation /

P. L'Ecuyer // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 81–89.

325 **Leemis, L.** Input Modeling / L. Leemis // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 14–24.

326 **Lobo, F. G.** The parameter-less genetic algorithm: Rational and automated parameter selection for simplified genetic algorithm operation [Электрон. ресурс] / F. G. Lobo . – IlliGAL Report No 2000030, July 2000. – 2001. – Режим доступа: <http://www.illigal.ge.uiuc.edu>.

327 **Malak, Jr. R. J.** Foundations of Validating Reusable Behavioral Models in Engineering Design Problems [Электрон. ресурс] / Jr. R. J. Malak, C. J. J. Paredis // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 420–428. – Режим доступа : <http://www.informs-sim.org/wsc04papers/050.pdf>. – Дата доступа : 27.08.2005.

328 **Martens, J.** A Neural Network Approach to the Validation of Simulation Models / J. Martens, K. Pauwels, F. Put // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 905–910.

329 **Maryak, J. L.** Global Random Optimization by Simultaneous Perturbation Stochastic Approximation / J. L. Maryak, D. C. Chin // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 307–312.

330 **McDonnel, L.** Business driving in hierarchical systems / L. McDonnel, K. Miatliuk, S. Gancharova // Proceedings of 8th IFAC / IFORS / IMACS / IFIP Symposium on Large Scale Systems: Theory and Applications, LSS'98. – Greece, Rio Patras : University of Patras, 1998. – P. 766–771.

331 Simulation Standards: Current Status, Needs, and Future Directions / C. McLean [etc.] // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 2019–2026.

332 **Modarres, M.** Predicting and Improving Complex Business Processes: Values and Limitations of Modeling and Simulation Technologies / M. Modarres // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 598–603.

333 **Müller, W.** Visualization Methods for Time-Dependent Data – An Overview / W. Müller, H. Schumann // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 737–745.

334 Engineering statistics handbook NIST [Электрон. ресурс]. – 2003. – Режим доступа : <http://www.itl.nist.gov/div898/handbook>. – Дата доступа : 11.09.2006.

335 **Nakayama, M. K.** Analysis of Simulation Output / M. K. Nakayama // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 49–58.

336 **Nance, R. E.** Software Requirements Engineering: Exploring the Role in Simulation Model Development [Электрон. ресурс] / R. E. Nance, J. D. Arthur // Proceedings of the 2006 OR Society Simulation Workshop. –

2006. – Режим доступа : http://www.theorsociety.com/conf/simulation2006/proceedings/SW06_15NanceArthur.pdf.

337 **Nordgren, W. B.** Flexsim Simulation Environment / W. B. Nordgren // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 250–252.

338 Hierarchical mathematics: theory of sway / S. Novikava [etc.] // Proceedings of 8th IFAC / IFORS / IMACS / IFIP Symposium on Large Scale Systems: Theory and Applications, LSS'98., Rio Patras (Greece), July 15–17, 1998. – Rio Patras : University of Patras, 1998. – P. 480–487.

339 **Ólafsson, S.** Towards a Framework for Black-Box Simulation Optimization / S. Ólafsson, J. Kim // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 300–306.

340 **Onggo, S.** A Dynamic Business Model for Component-Based Simulation Software / S. Onggo, D. Soopramanien, M. Pidd // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 954–959.

341 **O'Reilly, J.** Introduction to AweSim [Электрон. ресурс] / J. O'Reilly // Proceedings of 2002 Winter Simulation Conference. – 2002. – P. 221–224. – Режим доступа : <http://www.informs-sim.org/wsc02papers/028.pdf>. – Дата доступа : 28.07.2004.

342 **Overstreet, C. M.** Model Testing: Is it Only a Special Case of Software Testing? / C. M. Overstreet // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 641–647.

343 Addressing Complexity Using Distributed Simulation: A Case Study in Spaceport Modeling / J. Park [etc.] // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 1804–1812.

344 Is Problem Solving, or Simulation Model Solving, Mission Critical? / R. J. Paul [etc.] // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 547–554.

345 **Paul, R. J.** Simulation for Business Processes and Information Systems Design / R. J. Paul, A. Serrano // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 1787–1796.

346 **Petredis, V.** Varying quality function in genetic algorithms and the cutting problem / V. Petredis, S. Kasarlis // Proceedings of 8th IFAC / IFORS / IMACS / IFIP Symposium on Large Scale Systems: Theory and Applications, LSS'98. – Greece, Rio Patras : University of Patras, 1998. – P. 766–771.

347 **Pichitlamken, J.** Selection-of-the-Best Procedures for Optimization Via Simulation / J. Pichitlamken, B. L. Nelson // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 401–407.

348 **Pidd, M.** Simulation Worldviews – So What? [Электрон. ресурс] / M. Pidd // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 288–292. – Режим доступа : <http://www.informs-sim.org/wsc04papers/034.pdf>. – Дата доступа: 30.04.2005.

349 **Pilana, S.** UML based Modeling of Performance Oriented Parallel

and Distributed Applications / S. Pillana, T. Fahringer // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 497–505.

350 **Priami, C.** Design Environments for Complex Systems / C. Priami // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 897–905.

351 **Reeves, C. R.** Genetic algorithms for the Operations Researcher / C. R. Reeves // Journal on Computing. – 1997. – № 9. – P. 231–250.

352 **Reeves, C. R.** Genetic algorithms. Modern heuristic techniques for combinatorial problems / C. R. Reeves // Blackwell Scientific Publications. – Oxford, 1993. – P. 151–196.

353 **Robinson, S.** A Statistical Process Control Approach for Estimating the Warm-up Period / S. Robinson // Proceedings of 2002 Winter Simulation Conference. – San Diego : IEEE Press, 2002. – P. 439–446.

354 **Robinson, S.** Automated Analysis of Simulation Output Data / S. Robinson // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 763–770.

355 **Robinson, S.** Conceptual modeling for simulation: issues and researches requirements / S. Robinson // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 792–800.

356 Modelling and Improving Human Decision Making with Simulation / S. Robinson [etc.] // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 913–920.

357 **Roehl, M.** Flexible Integration of XML into Modeling and Simulation Systems / M. Roehl, A. M. Uhrmacher // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 1813–1820.

358 **Rueß, H.** From Simulation to Verification (and Back) [Электрон. ресурс] / H. Rueß, L. de Moura // Proceedings of 2003 Winter Simulation Conference. – 2003. – P. 888–896. – Режим доступа : <http://www.informs-sim.org/wsc03papers/108.pdf>.

359 **Saad, S. M.** Simulation of Distributed Manufacturing Enterprises: A New Approach / S. M. Saad, T. Perera, R. Wickramarachchi // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 1168–1173.

360 **Sadowski, D. A.** Tips for Successful Practice of Simulation [Электрон. ресурс] / D. A. Sadowski, M. R. Grabau // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 61–66. – Режим доступа : <http://www.informs-sim.org/wsc04papers/008.pdf>. – Дата доступа : 30.04.2005.

361 **Sanchez, S. M.** A Two-Phase Screening Procedure for Simulation Experiments / S. M. Sanchez, T. W. Lucas, H. Wan // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 223–230.

362 **Sanchez, S. M.** Work Smarter, Not Harder: Guidelines for Designing Simulation Experiments / S. M. Sanchez // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 69–82.

363 The Simulation Project Life-Cycle: Models and Realities / R. G. Sargent [etc.] // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 863–871.

364 **Sargent, R. G.** Validation and Verification of Simulation Models / R. G. Sargent // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 130–143.

365 **Sargent, R. G.** Verification and Validation of Simulation Models / R. G. Sargent // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 37–48.

366 **Schmeiser, B. W.** Some Myths and Common Errors in Simulation Experiments / B. W. Schmeiser // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 39–46.

367 **Schriber, T. J.** Inside Discrete-Event Simulation Software: How it Works and Why it Matters / T. J. Schriber, D. T. Brunner // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 167–177.

368 **Shapiro, A.** Monte Carlo Simulation Approach to Stochastic Programming / A. Shapiro // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 428–431.

369 Problems of Visualization of Technological Processes / P. Slavik [etc.] // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 746–754.

370 Software Programs Managers Network SPMN [Электрон. ресурс]. – 2001. – Режим доступа : <http://www.spmn.com>. – Дата доступа : 30.04.2005.

371 Spring Simulation Multiconference SpringSim 2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.scs.org>. – Дата доступа : 29.01.2008.

372 **Steiger, N. M.** Performance Evaluation of ASAP3 for Steady-state Output Analysis / N. M. Steiger, E. K. Lada, J. R. Wilson // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 781–789.

373 **Sterman, J. D. A.** Skeptic's Guide to Computer Models / J. D. A. Sterman // Managing a Nation: The Microcomputer Software Catalog. – Boulder, CO : Westview Press, 1991. – P. 209–229.

374 **Sturm, J.** Developing XML Solutions [Электрон. ресурс] / J. Sturm. – Microsoft Press, 2000. – 1 CD ROM.

375 **Szechtman, R.** Constrained Monte Carlo and the Method of Control Variates / R. Szechtman, P. W. Glynn // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 394–400.

376 Panel on Future Challenges in Modeling Methodology [Электрон. ресурс] / S. J. E. Taylor [etc.] // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 327–335. – Режим доступа : <http://www.informs-sim.org/wsc04papers/039.pdf>. – Дата доступа : 14.05.2005.

377 The Huntsville Simulation Conference HSC2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.scs.org/confernc/hsc/hsc07/>

cfp /hsc07.htm.

378 The IASTED Asian Conference on Modelling and Simulation AsiaMS 2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.iasted.org/conferences/home-571.html>.

379 The Third IASTED International Conference on Environmental Modelling and Simulation EMS 2007 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.iasted.org/conferences/home-579.html>.

380 **Truong, T. H.** Simulation based Optimization for Supply Chain Configuration Design / T. H. Truong, F. Azadivar // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 1268–1275.

381 **Wainer, G. A.** Modeling and Simulation of Complex Systems with Cell-DEVS [Электрон. ресурс] / G. A. Wainer // Proceedings of 2004 Winter Simulation Conference. – 2004. – P. 49–60. – Режим доступа : <http://www.informs-sim.org/wsc04papers/007.pdf>. – Дата доступа : 19.04.2005.

382 **Wenzel, S.** A Taxonomy of Visualization Techniques for Simulation in Production and Logistics / S. Wenzel, J. Bernhard, U. Jessen // Proceedings of 2003 Winter Simulation Conference. – New Orleans : IEEE Press, 2003. – P. 729–736.

383 Winter Simulation Conference WSC'07 [Электрон. ресурс]. – 2007. – Режим доступа : <http://www.wintersim.org/program.htm>.

384 **Whitman, L.** A Taxonomy of a Living Model of the Enterprise / L. Whitman, K. Ramachandran, V. Ketkar // Proceedings of 2001 Winter Simulation Conference. – New Jersey : IEEE Press, 2001. – P. 848–855.

385 **Wolpert, D. H.** No Free Lunch Theorems for Optimization / D. H. Wolpert, W. G. Macready // Evolutionary Computation, IEEE Transactions on. – 1997. – V. 1, № 1. – P. 67–82.

386 **Xu, Y.** A Distributed Computing Architecture for Simulation and Optimization / Y. Xu, S. Sen // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 365–373.

387 **Yakimov, A. I.** Computer Data Analysis in Modeling and Optimization of Manufacturing Process Control System / A. I. Yakimov, S. A. Alkhovik // Computer Data Analysis and Modeling : Complex Stochastic Data and Systems : Proc. of the Eighth Intern. Conf., Minsk, Sept. 11–15, 2007 : in 2 vol. – Minsk : Publ. center BSU, 2007. – Vol 2. – P. 204–207.

388 **Yamane, S.** Meta-Level Control Architecture for Massively Multi-agent Simulations / S. Yamane // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 889–896.

389 **Yang, F.** Estimation of Percentiles of Cycle Time in Manufacturing Simulation / F. Yang, B. E. Ankenman, B. L. Nelson // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press, 2005. – P. 475–484.

390 **Zhao, W.** A Framework for Configurable Hierarchical Simulation in a Multiple-User Decision Support Environment / W. Zhao, A. Verbraeck // Proceedings of 2005 Winter Simulation Conference. – Orlando : IEEE Press,

2005. – P. 327–335.

391 **Zhao, Y.C.H.Q.C.** The No Free Lunch Theorems: Complexity and Security / Y.C.H.Q.C. Zhao, D. L. Pepyne // Automatic Control, IEEE Transactions on. – 2003. – V. 48, № 5.– P. 783–793.

392 **Zhou, M.** What Can Be Done to Automate Conceptual Simulation Modeling / M. Zhou, Q. Zhang, Z. Chen // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 809–814.

393 **Zülch, G.** Modelling and Simulation of Human Decision-Making in Manufacturing Systems / G. Zülch // Proceedings of 2006 Winter Simulation Conference. – Monterey : IEEE Press, 2006. – P. 947–953.

Оглавление

Перечень условных обозначений, терминов и сокращений	3
Введение	5
1 Обзор методов и средств моделирования и принятия решений в управлении промышленным предприятием	9
1.1 Обзор современных концепций управления промышленным предприятием	9
1.2 Математическая постановка системной задачи управления сложным иерархическим объектом	12
1.3 Аналитический обзор моделей и способов моделирования промышленного предприятия	15
1.4 Обзор методов и средств имитационного моделирования сложных систем	21
1.5 Обзор технологий имитационного моделирования	25
1.6 Обзор применения методов оптимизации в имитационном моделировании	30
2 Метод поиска рациональных решений в ERP-системе управления ресурсами промышленного предприятия	36
2.1 Формальное описание многоуровневой иерархической системы	36
2.2 Идея метода поиска рациональных решений	39
2.3 Формальное описание оптимизации объекта исследования на основе имитационной модели	43
2.4 Методика решения задачи рационального выбора алгоритма оптимизации систем	46
2.5 Целевая функция оценки эффективности взаимодействия компонентов предприятия	50
3 Метод имитационного моделирования ERP-системы управления ресурсами промышленного предприятия	66
3.1 Идея метода имитационного моделирования	66
3.2 Вербальная модель структурных компонентов промышленного предприятия	68
3.3 Концептуальные модели аспектов функционирования промышленного предприятия	75
3.4 Имитационная модель промышленного предприятия	84
4 Программно-технологический комплекс имитации сложных систем BELSIM	105
4.1 Состав и структура комплекса	105
4.2 Система имитационного моделирования PSTL	107
4.3 Технология построения имитационной модели	112
4.4 Технология программирования имитационной модели	114
4.5 Программная реализация базовой имитационной модели промышленного предприятия	117

4.6 Программные средства автоматизации планирования, проведения и обработки результатов имитационных экспериментов	122
4.7 Программная поддержка интеграции ПТКИ BelSim с комплексной информационной системой.....	135
4.8 Технология модернизации ПТКИ BelSim для организации распределенных вычислений	140
5 Апробация метода, средств и технологии имитационного моделирования ERP-системы управления.....	144
5.1 Состав задач, решаемых при апробации, и задание исходной информации для моделирования	144
5.2 Верификация имитационной модели	148
5.3 Исследование погрешности имитации	155
5.4 Исследование длительности переходного процесса в модели...	162
5.5 Исследование чувствительности имитационной модели	168
5.6 Проверка адекватности базовой имитационной модели по тесту Тьюринга	170
6 Программные средства для реализации метода поиска рациональных решений	172
6.1 Система проведения оптимизации «Оптимизатор»	172
6.2 Имитационное моделирование контрактов в информационной системе предприятия.....	180
6.3 Математическое моделирование в комплексной информационной системе.....	193
6.4 Программный комплекс Contract Analyzer для оценки эффективности контрактов на промышленном предприятии	206
7 Апробация метода поиска рациональных решений в ERP-системе управления промышленным предприятием	212
7.1 Состав задач, выбранных на апробацию	212
7.2 Исследование параметров генетического алгоритма для поиска лучшего решения.....	214
7.3 Решение задачи оптимизации производственного процесса	222
7.4 Поиск оптимальных решений при заключении контрактов в маркетинговой деятельности промышленного предприятия	243
7.5 Выбор типа координации процесса планирования производственной программы	250
7.6 Разработка библиотеки моделей для исследования информационной системы промышленного предприятия	257
Заключение	267
Список литературы	272

Научное издание

Якимов Анатолий Иванович

**ТЕХНОЛОГИЯ ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ
ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ**

Монография

Редактор Т. А. Рыжикова

Художественное оформление обложки И. А. Алексеюс

Технический редактор А. А. Подошевка

Компьютерный дизайн И. А. Алексеюс

Подписано в печать 17.02.2010. Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл.-печ. л. 17,67. Уч.-изд. л. 17,0. Тираж 100 экз. Заказ № 135.

Издатель и полиграфическое исполнение
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет»
ЛИ № 02330/375 от 29.06.2004 г.
212000, г. Могилев, пр. Мира, 43