

Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный университет
имени Ф. Скорины»

И. В. Максимей, В. С. Смородин, О. М. Демиденко

**РАЗРАБОТКА
ИМИТАЦИОННЫХ МОДЕЛЕЙ
СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ**

Гомель
ГГУ им. Ф. Скорины
2014

Авторы:

доктор технических наук, профессор **И. В. Максимей** ,
доктор технических наук, профессор В. С. Смородин,
доктор технических наук, профессор О. М. Демиденко

Максимей, И. В.

Разработка имитационных моделей сложных технических систем / И. В. Максимей, В. С. Смородин, О. М. Демиденко ; М-во образования РБ, Гом. гос. ун-т им. Ф. Скорины. – Гомель : ГГУ им. Ф. Скорины, 2014. – 298 с. – ISBN 978-985-439-951-5

В монографии содержится описание технологических этапов создания программ имитационных моделей сложных систем на ЭВМ. Для каждого этапа технологии создания имитационной модели сложной системы излагается методика преобразования результатов предыдущего этапа в исходную информацию, используемую при выполнении следующего этапа. Методика демонстрируется на шести примерах сложных систем из трех предметных областей: вычислительный процесс в ЛВС; поточное производство изготовления и ремонта изделий сложной структуры на предприятиях дискретного типа; вероятностные технологические процессы производства. Общая разработка программ демонстрируется путем использования двух средств автоматизации имитационного моделирования: программно-технологического комплекса имитации, реализующего транзактно-процессный способ имитации, и системы автоматизации имитационного моделирования, реализующей агрегатно-процессный способ имитации.

Табл. 25. Ил. 59. Библиогр.: 34 назв.

Рекомендовано к изданию научно-техническим советом
учреждения образования
«Гомельский государственный университет имени Франциска Скорины»

Рецензенты:

доктор физико-математических наук, профессор Ю. С. Харин;
доктор технических наук, профессор К. А. Бочков

ISBN 978-985-439-951-5

© Максимей И. В., Смородин В. С.,
Демиденко О. М., 2014
© УО «Гомельский государственный
университет им. Ф. Скорины», 2014

Список условных обозначений и терминов

СС	– сложные системы
АНМ	– аналитические модели
ИМ	– имитационные модели
ЛВС	– локальные вычислительные сети
ВС	– вычислительные системы
ВП	– вычислительный процесс
РН	– рабочая нагрузка
ВТПП	– вероятностные технологические процессы производства
СРО	– сборочно-разборочные операции
ИЭ	– имитационный эксперимент
НЭ	– натурный эксперимент
МТХО	– микротехнологическая операция
ВСГР	– вероятностный сетевой график
ТП	– технологический процесс
ПО	– программное обеспечение
ОС	– операционная система
УД	– уровни детализации
СРУ	– центральный процессор
НДД	– внешняя память на диске
DR и DIS	– диалоговый и дистанционный доступ к ресурсам
РАК	– пакет задач отложенного счета
ОСТ	– обрабатывающий станок
РМ	– рабочее место
RES	– ресурс предприятия
РОП	– ремонтная операция
СОП	– содержательное описание
АК	– активность
ФД	– функциональные действия
УПМ	– управляющая программа моделирования
ТК	– технологическая карта
КОМ	– концептуальная модель
TR	– транзакты
UTR	– управляющий транзакт
ДИСП	– диспетчер
ПММ	– полумарковская модель
ВД	– временная диаграмма
УЗО	– узел обработки
СМОУ	– системы массового обслуживания с управлением
TRIF	– транзакты информационные
SOB	– события
GKRP	– критический путь на ВСГР
ASOB и АТОР	– агрегаты события и технологическая операция
PR.MONTEK	– процедура Монте-Карло
ПТК	– программно-технологический комплекс имитации
САИМ	– система автоматизации имитационного моделирования

ВВЕДЕНИЕ

В настоящее время накоплен большой опыт построения математических моделей объектов исследования из различных отраслей науки, техники, промышленности и экономики. Однако проявляющаяся последнее время тенденция к использованию общесистемных принципов и методов исследований в различных областях знаний наталкивается на определенные трудности. Это связано, как известно, с целым комплексом различных причин: наличием больших объемов разнородной информации в различных областях знаний, использованием различных понятийных аппаратов, профессиональной разобщенностью исследователей и других.

Многочисленные попытки унификации системного подхода, при решении конкретных задач науки и практики, привели к понятию сложной технической системы как многопараметрического объекта, представимого конечным множеством математических моделей, каждая из которых отражает конкретную группу свойств исходной системы. При этом появляется возможность выделить классы сложных систем со специфическими свойствами, на основании которых разрабатываются методологические принципы построения математических моделей, которые характеризуются единой математической терминологией и могут быть доступны специалистам различных предметных областей.

Такой подход дает основания рассчитывать в дальнейшем на создание необходимой базы, позволяющей исследователю работать с системами любой степени сложности, вне зависимости от ее физической сущности или ограниченности рамками определенной формализации.

В тексте данной монографии к классам сложных технических систем относятся производственные и экономические системы, вычислительные сети, системы управления, системы энергообеспечения, а также другие технические системы, призванные обеспечивать безопасность жизнедеятельности и производства.

Системный анализ сложных систем положен в основу исследований производственных систем с вероятностными характеристиками их функционирования (вероятностных производственных систем) как класса технических сложных систем. При этом под вероятностными характеристиками функционирования могут пониматься характеристики надежности функционирования оборудования, задействованного в ходе реализации технологического цикла производства, характеристики выполнения технологических операций, качественные характеристики используемых в процессе производства материалов и комплектующих изделий и другие.

Как показал анализ современного состояния разработок в области исследования и проектирования технологических объектов

с вероятностными характеристиками их функционирования, проблема моделирования вероятностных производственных систем состоит в недостаточной результативности методов их исследования при увеличении количества учитываемых параметров, в особенности для тех случаев, когда структура технологического цикла изменяется в процессе функционирования объекта исследования.

Это связано, в первую очередь, с многообразием сложных технологических систем, в ходе реализации которых могут изменяться параметры их функционирования и структура технологического цикла; сложностью практических задач при оценке уровня надежности и безопасности потенциально опасных промышленных объектов; необходимостью учета человеческого фактора при выполнении работ на потенциально опасных объектах.

Данная монография посвящена решению актуальных задач обеспечения надежности и безопасности функционирования производственных систем, а также повышению эффективности их использования. Следует при этом отметить, что единственной технологией, обладающей возможностью структурной реконфигурации технологического процесса, остается технология имитационного моделирования, применяемая ко всему спектру задач синтеза оптимальной структуры сложных технических систем в рамках предложенного авторами подхода.

В настоящей монографии «Разработка имитационных моделей сложных технических систем» развиваются теоретические основы классического аппарата имитационного моделирования на основе нового подхода к унификации методов системных исследований в области проектного моделирования сложных технических систем с вероятностными параметрами их функционирования с использованием конечного множества математических моделей.

В качестве инструмента реализации предлагаемого подхода используется динамическое имитационное моделирование, базирующееся на разработке динамических имитационных моделей вероятностных производственных систем, что связано с погружением моделей технических систем, имеющих графовую структуру, в пространство возможных состояний объекта исследования (при наличии функциональных зависимостей между операциями) для реализации сквозной технологии объектно-ориентированного программирования контура управления технологическим циклом производства.

Представленные в настоящей монографии результаты могут использоваться в качестве составных компонентов систем анализа функционирования при автоматизации технологических процессов и производств, а также при разработке и автоматизации систем проектирования новых технологических объектов.

1 ЭТАПЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

1.1 Использование имитационных моделей при проектировании СС

В производстве понятие «технология» включает в себя совокупность производственных процессов и операций, с помощью которых создаются определенные виды изделий, а также описание способов производства. Поскольку программы для ЭВМ это также изделия, создаваемые коллективами специалистов разной квалификации при значительных трудовых затратах, то актуальна задача создания современной технологии производства таких программ. Имитационные модели являются специфичным и достаточно сложным программным изделием, и их разработка должна вестись с применением прогрессивной технологии. Технология проектирования ИМ включает в себя методы и средства, обеспечивающие их создание и развитие в течение всего периода их жизни. Этот период включает этапы проектирования, изготовления и эксплуатации ИМ. Он начинается с формирования назначения и принципов построения ИМ и завершается после полного прекращения эксплуатации модели. Как правило, ИМ является средством проектирования сложных систем. Технология их изготовления и использования увязана с технологией проектирования и эксплуатации объекта моделирования. Поэтому вначале рассмотрим вопросы применения ИМ на различных этапах проектирования сложных систем, затем уделим внимание собственно технологии создания и эксплуатации ИМ.

Проектирование сложной системы представляет собой весьма трудоемкий процесс, в котором участвуют разные специалисты. Большие размеры системы, сложность поведения ее компонент, высокая стоимость разработки требуют методов математического моделирования на всех этапах проектирования такой системы. Поэтому моделирование сопровождает и процесс проектирования, и процесс испытания, и процесс эксплуатации сложной системы. ГОСТами на разработку программных изделий, например ЕСПД, предусматриваются следующие этапы проектирования: формулировка технического задания, разработка технических предложений по созданию изделия, разработка эскизного проекта, разработка технического проекта, рабочее проектирование, испытание изделия. На всех этапах проектирования специалистам приходится рассматривать две стороны объекта проектирования:

требования к системе со стороны внешней среды (внешнее проектирование) и организацию функционирования сложной системы (внутреннее проектирование).

Формулировка технического задания. Начало работы над проектом сложной системы имеет целью сформулировать задачи проектирования и организовать рабочие группы по решению поставленных задач проектирования. Вначале образуется небольшая инициативная группа специалистов, которая на основе всесторонних обсуждений вырабатывает документ, содержащий в себе следующую информацию:

- формулировку проблем (результаты обсуждений);
- предлагаемые варианты решений (с указанием достоинств и недостатков каждого из них);
- предложения по необходимому составу специалистов, комплектованию и срокам подключения рабочих групп;
- грубую оценку времени и денежных средств, необходимых для разработки проекта.

В результате создается группа специалистов-проектировщиков, имеющая четко поставленную задачу проектирования сложной системы.

Разработка технических предложений. На этом этапе предполагается: выбор наилучшего решения, укомплектование группы проектировщиков системы до полного состава, составление плана всех работ над проектом. Для выбора наилучшего решения часть сил и средств направляется на поиск альтернативных решений. Для этого используются аналитические и имитационные модели. Затем наступает момент выбора решения. Многократно пересматривается формулировка задачи проектирования сложной системы. Основными средствами исследования являются математические модели проектируемого объекта. При этом выполняются следующие виды моделирования компонент: проектирование единичной нити, проектирование большой нагрузки, состязательное проектирование.

В процессе проектирования единичной нити прорабатываются вопросы выполнения функций системы. При проектировании большой нагрузки просматриваются действия системы, которые зачастую случайно распределены во времени. Цель состязательного проектирования – найти на модели ответные действия системы на тот случай, когда внешняя среда пытается затруднить работу алгоритма системы. Эти три вида работ должны осуществляться одновременно в течение проектирования. Этот этап характеризуется появлением большого количества новых идей решения проблемы, новых проработок по частным вопросам отдельных направлений работы.

Новые идеи должны документироваться в виде рабочих записок или отчетов. С ними необходимо знакомить всех разработчиков.

К этому времени должна быть укомплектована группа проектировщиков системы. Обязательно постоянство основного состава группы в процессе проектирования сложной системы. В результате их работы формулируется проблема, изыскиваются пути ее реализации, составляется план выполнения проекта системы, который должен:

- содержать график распределения времени по этапам работ с указанием необходимых средств и состава исполнителей;
- предусматривать консультации крупных специалистов по сложным вопросам;
- содержать заявки на проведение натурного и модельного экспериментов.

Эскизное проектирование. Этап начинается с разработки первого варианта того, что называется системой. Для этих целей требуется большое число экспериментов, в результате которых получают данные, позволяющие создать лучшую математическую модель сложной системы. На этапе эскизного проектирования решаются вопросы, допускающие многозначные ответы. Отчетная документация этапа должна содержать:

- подробное описание работы всей системы в целом;
- четкое описание подсистемы (форма, количество и времена ее создания, алгоритм ее функционирования);
- перечень допустимых значений характеристик системы;
- хотя бы один метод физической реализации предложенного способа функционирования сложной системы;
- информацию об исследованиях других методов физической реализации предложенного способа.

Техническое проектирование. Разработка технического проекта выполняется, когда система уже «заморожена» и идет отладочная работа. Уточняются и детализируются алгоритмы функционирования компонент системы. На имитационных моделях проверяется, соответствуют ли компоненты системы своему назначению. На этом этапе наряду с множеством подмоделей компонент системы обязательны создание и испытание обобщенной имитационной модели самой системы, с помощью которой решаются все вопросы внешнего и внутреннего проектирования. Отчетность аналогична отчетности предыдущего этапа, но содержит более детализированные описания системы и ее компонент.

Рабочее проектирование. Этот этап характеризуется созданием опытного образца, предназначенного для испытания свойств системы. Опытный образец системы должен иметь в своем составе контрольную аппаратуру. Зачастую при конструировании контрольной аппаратуры используются ЭВМ, позволяющие заменить ряд имитаторов-устройств

соответствующими программами-имитаторами и таким образом автоматизировать процесс исследования свойств системы с помощью образца. На этом этапе должна быть разработана техническая документация, включающая в себя: технические характеристики по эксплуатации опытного образца системы; полное описание всех технических решений, использованных при создании опытного образца.

Испытание изделия. На этом этапе осуществляются испытание и оценка принятых проектных решений по созданию сложной системы. Основная цель испытания состоит в получении подтверждения, что система работает так, как предусматривалось при проектировании. В ходе испытания система отлаживается, исключаются неизбежные дефекты ее функционирования. Испытание проводится по плану испытаний, предусматривающему усложнение условий работы опытного образца в процессе испытания по специально разработанным методикам. В результате испытаний составляются отчеты. В них даются рекомендации по улучшению использования системы и производственных условий и приводятся результаты анализа различий между реальным образцом системы и ее моделью.

1.2 Этапы разработки программы имитации функций СС

Независимо от способа проектирования сложной системы и назначения моделирования можно выделить следующие восемь этапов создания и использования математических моделей:

- определение объекта имитации, установление границ и ограничений моделирования, выбор показателей для сравнения эффективности вариантов системы (составление *содержательного описания* объекта моделирования);
- формулировка замысла модели, переход от реальной системы к логической схеме ее функционирования (составление *концептуальной модели*);
- реализация описания объекта в терминах математических понятий и алгоритмизация функционирования ее компонент (составление *формального описания* объекта);
- преобразование формального описания объекта в описание имитационной модели (составление *описания имитационной модели*);
- программирование и отладка модели (*программирование модели*);
- проверка модели, оценка ее свойств и затрат ресурсов на имитацию (*испытание и исследование модели*);

– организация модельного эксперимента на ЭВМ (*эксплуатация модели*);

– интерпретация результатов моделирования и их использование в ходе проектирования сложной системы (*анализ результатов*).

Рассмотрим порядок действий разработчиков на каждом из перечисленных этапов создания и использования имитационных моделей на ЭВМ. На рисунке 1.1 представлена схема взаимосвязи технологических этапов моделирования.

Составление содержательного описания объекта моделирования представляет собой выполнение следующих действий.

Вначале определяются объект имитации и состав исходной технической информации, достаточной для изучения тех сторон его функционирования, которые представляют интерес для исследователя.

Устанавливаются границы изучения функционирования объекта. Составляется возможный список ограничений модели, которые допустимы при организации имитации или при наличии которых еще имеет смысл имитация функционирования объекта моделирования.

Перед разработчиками ИМ ставятся вполне конкретные цели моделирования и формулируются основные критерии эффективности, по которым предполагается проводить сравнение на модели различных проектных решений или вариантов организации сложной системы.

Результатом работ на данном этапе является содержательное описание объекта моделирования с указанием целей имитации и аспектов функционирования объекта моделирования, которые необходимо изучить на ИМ. Обычно оно представляет собой техническое описание объекта моделирования, описание внешней среды, с которой он взаимодействует, и временную диаграмму этого взаимодействия.

Составление концептуальной модели производится в следующей последовательности. На основе анализа поставленной задачи определяется общий замысел модели [15]. Выдвигаются гипотезы и фиксируются все допущения, необходимые для построения ИМ.

На основании содержательного описания уточняется задача моделирования, определяются процедура и график ее решения. Уточняется методика всего имитационного эксперимента в зависимости от наличных ресурсов, выделенных для имитации. Общая задача моделирования разбивается на частные подзадачи. Устанавливаются приоритеты решения подзадач моделирования. Обосновываются требования в рабочей силе и выделения ресурсов ЭВМ. Затем проводится тщательный анализ задач, стоящих перед имитацией.

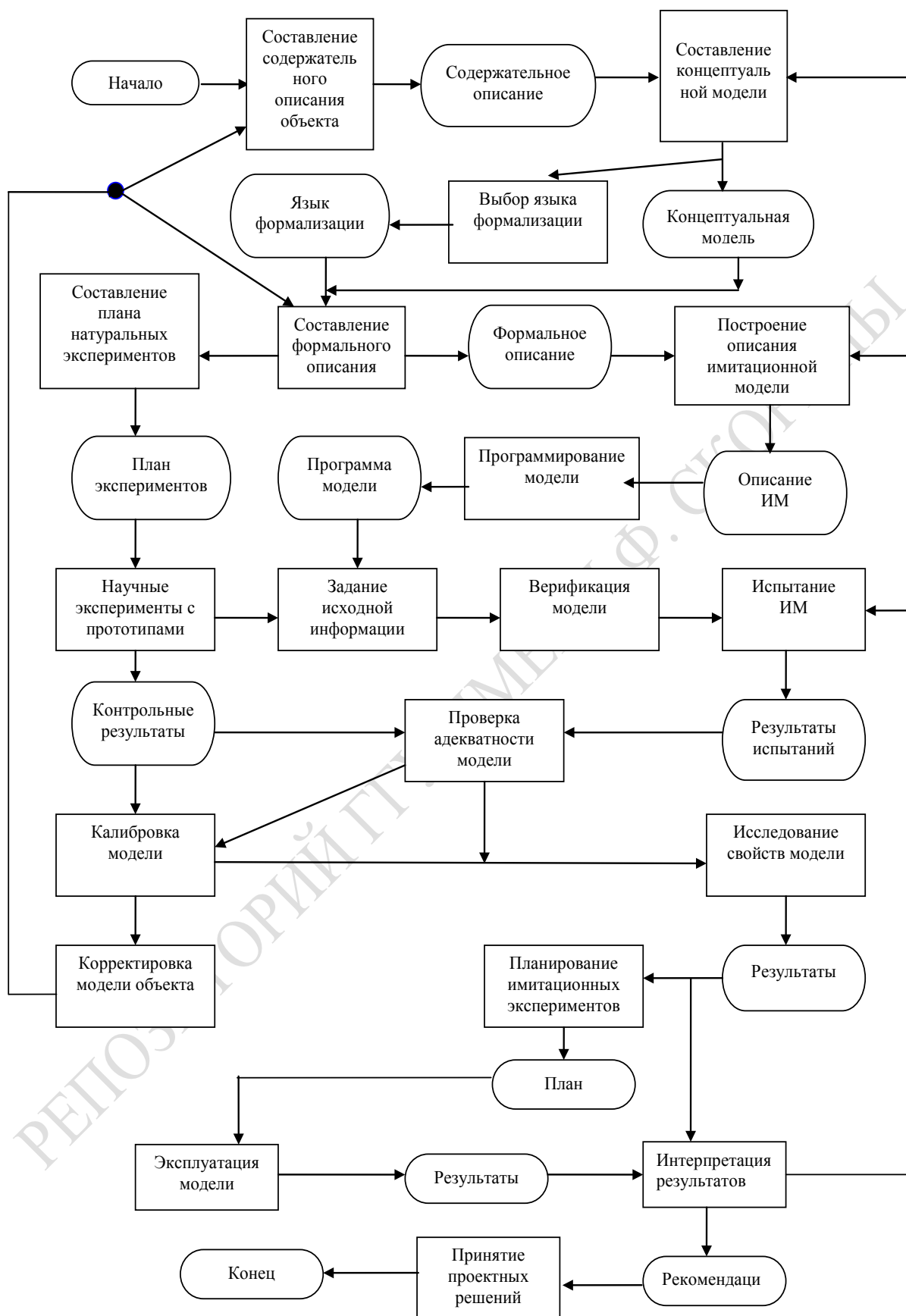


Рисунок 1.1 – Схема взаимосвязи технологических этапов имитационной модели СС

Выполняются такие работы, как выбор параметров и переменных системы, представляющих интерес для моделирования; уточнение критериев эффективности функционирования различных вариантов проектируемой системы; выбор типов аппроксимации отдельных компонент модели. Проводятся также: предварительный анализ требований к модели сложной системы; определение необходимых математических уравнений, описывающих реальные процессы; поиск возможных методов проверки правильности функционирования модели.

Определяются методы проверки программной реализации модели системы, формулируются технические требования на программирование моделей. Изучаются возможности применения известных методов обработки и анализа результатов, выбираются способы представления ожидаемых результатов моделирования. Одновременно с этим исследователь должен выбрать язык будущей формализации процессов в объекте моделирования.

Результатом выполнения работ являются: концептуальная модель, выбранный язык формализации и способ организации имитации, реализуемой языком формализации. В состав концептуальной модели входят: уточненное содержательное описание объекта моделирования, свободное от всего того, что не представляет интереса для имитации поведения системы; список параметров и переменных моделирования; критерии эффективности функционирования вариантов системы; список используемых методов обработки результатов имитации и перечисление способов представления результатов моделирования.

Отметим, что при создании небольших моделей данный этап работ совмещается с этапом составления содержательного описания моделируемой системы. Только с усложнением объекта моделирования и задач имитации появляется необходимость в данном этапе, главная цель которого состоит в определении того способа формализации, который наиболее подходит для решения конкретной задачи проектирования.

Составление формального описания объекта моделирования представляет собой ответственный этап создания имитационной модели сложной системы. Цель – получение исследователем формального представления алгоритмов поведения компонент сложной системы и отражение вопросов взаимодействия между собой этих компонент. При составлении формального описания модели исследователь использует тот или иной язык формализации. В зависимости от сложности объекта моделирования и внешней среды могут использоваться три вида формализации: аппроксимация явлений функциональными зависимостями, алгоритмическое описание процессов в системе,

смешанное представление в виде последовательности формул и алгоритмических записей.

В зависимости от выбранного способа организации квазипараллелизма используются свой язык формализации и своя методика составления формального описания объекта имитации. После составления формального описания объекта моделирования приступают к его проверке. Это первая главная проверка достоверности будущей модели сложной системы в процессе проектирования. Для обеспечения контроля правильности функционирования модели вводят классические модели, достоверность которых доказана. Они фигурируют в модели в виде составных частей. На вход таких моделей поступают данные, вычисляемые в других частях модели, достоверность которых проверяется. Если результат работы классической модели окажется недостоверным, то считают, что предшествующая часть формального описания системы также недостоверна.

В процессе проверки достоверности модели необходимо ответить на следующие вопросы: позволяет ли модель решить поставленные задачи моделирования, насколько полна предложенная схема модели и отражает ли она фактическую последовательность развития процессов в реальной системе. Необходимо провести анализ каждой функции модели и убедиться, что она нашла свое отражение в формальном описании системы. В том случае, когда уравнения получены на основании анализа опытных данных, необходимо провести выборочную проверку согласия уравнений с исходной информацией, по которой они получены. Для уравнений, полученных теоретическим путем, следует провести вычисления в нескольких контрольных точках с целью определения приемлемости результатов. Для дополнительной проверки уравнений желательно провести анализ размерностей и масштабов переменных системы.

Важно правильно выбрать вычислительные средства, которые обеспечили бы исследователю легкость программирования, минимальные затраты на моделирование, доступность выбранной ЭВМ, быстрое получение результатов. На практике начинающие исследователи либо из-за спешки, либо из-за организационных трудностей доступа к соответствующей ЭВМ зачастую недооценивают важность правильного выбора базовой ЭВМ, на которой им придется проводить эксперименты. Результатом являются огромные потери материальных ресурсов и труда исследователя из-за низкой технологической оснастки средств моделирования на выбранной ЭВМ. Конкретные рекомендации по выбору ЭВМ и математического обеспечения моделирования предоставить чрезвычайно трудно. В основу выбора ЭВМ необходимо положить

наличие средств оперативного построения и испытания модели. Результатом этапа является проверенное формальное описание исследуемой системы на выбранном языке формализации.

Составление имитационной модели. Как только средства реализации и тип ЭВМ выбраны, исследователь приступает к этапу преобразования формального описания в описание имитационной модели. Многие специалисты по имитации не делают различия между этими этапами. И это оправдано, когда имитационную модель можно представить с помощью таких универсальных средств описания, как агрегативные схемы или системы массового обслуживания. При переходе к более сложным системам это различие проявляется. Прежде всего исследователя не удовлетворяет состав стандартной статистики моделирования, реализуемой соответствующими системами моделирования. Кроме того, появляется множество дополнительных трудностей, связанных с выбранным способом организации квазипараллелизма. Как правило, в этих случаях исследователю приходится решать множество дополнительных вопросов, не связанных с описанием поведения моделируемой системы. Сюда входят следующие вопросы реализации модели: декомпозиция объекта на составляющие и формирование элементов модели; отработка вопросов синхронизации частей компонент модели друг с другом в модельном времени; организация сбора статистики; задание начальных условий моделирования; планирование процесса имитации отдельных вариантов системы; проверка окончания моделирования; обработка результатов имитации. Все эти действия являются чрезвычайно трудоемкими и ответственными, их успешное выполнение зависит прежде всего от опыта и интуиции исследователя. Результатом этапа является описание имитационной модели сложной системы.

Программирование. На этом этапе выполняются следующие действия. Составляется план создания и использования программной модели. Как правило, программа модели создается с помощью средств автоматизации моделирования на ЭВМ. Поэтому в плане указываются: тип ЭВМ; средство автоматизации моделирования; примерные затраты памяти ЭВМ на создание программы модели и ее рабочих массивов; затраты машинного времени на один цикл работы модели; оценки затрат на программирование и отладку программы модели.

Затем исследователь приступает к программированию модели. В качестве технического задания на программирование используется описание имитационной модели. Специфика работ по программированию модели зависит от средств автоматизации моделирования, которые доступны исследователю. Не существует значительных отличий

создания программы модели от обычной автономной отладки программных модулей большой программы или пакета программ. В соответствии с текстом производится деление модели на блоки и подблоки. В отличие от обычной автономной отладки программных модулей, при автономной отладке блоков и подблоков программной модели объем работ существенно увеличивается, поскольку для каждого модуля необходимо создать и отладить еще имитатор внешнего окружения. Весьма существенно выверить реализацию функций модуля в модельном времени и оценить затраты машинного времени на один цикл работы модели как функцию от значений параметров модели. Завершаются работы при автономной отладке компонент модели подготовкой форм представления входных и выходных данных моделирования.

1.3 Этапы испытания имитационных моделей СС

Вначале переходят ко второй *проверке достоверности программы модели* системы. В процессе этой проверки устанавливается соответствие операций в программе и описании модели. Для этого производится обратный перевод программы в схему модели (ручная «прокрутка» позволяет найти грубые ошибки **статики** модели).

После исключения грубых ошибок ряд блоков объединяется и начинается комплексная отладка модели с использованием тестов. Отладка по тестам начинается с нескольких блоков, затем в этот процесс вовлекается все большее число блоков модели. Отметим, что комплексная отладка программы модели намного сложнее отладки пакетов прикладных программ, поскольку ошибки **динамики** моделирования в этом случае найти значительно труднее вследствие квазипараллельной работы различных компонент модели. По завершении комплексной отладки программы модели необходимо вновь оценить затраты машинного времени на один цикл расчетов на модели. При этом полезно получить аппроксимацию времени моделирования на один цикл имитации.

Следующим действием является составление технической документации на модель сложной системы. Результатом этапа к моменту окончания комплексной отладки программы модели должны быть следующие документы:

- описание имитационной модели;
- описание программы модели с указанием системы программирования и принятых обозначений;
- полная схема программы модели;
- полная запись программы модели на языке моделирования;

- доказательство достоверности программы модели (результаты комплексной отладки программы модели);
- описание входных и выходных величин с необходимыми пояснениями (размерностей, масштабов, диапазонов изменения величин, обозначений);
- оценка затрат машинного времени на один цикл моделирования;
- инструкция по работе с программой модели.

Для проверки адекватности модели объекту исследования после составления формального описания системы исследователь составляет план проведения натурных экспериментов с прототипом системы. Если прототип системы отсутствует, то можно использовать систему вложенных ИМ, отличающихся друг от друга степенью детализации имитации одних и тех же явлений [15]. Тогда более детальная модель служит в качестве прототипа для обобщенной ИМ. Если же построить такую последовательность невозможно либо из-за отсутствия ресурсов на выполнение этой работы, либо из-за недостаточности информации, то обходятся без проверки адекватности ИМ. Согласно этому плану параллельно с отладкой ИМ осуществляется серия натурных экспериментов на реальной системе, в ходе которых накапливаются контрольные результаты. Имея в своем распоряжении контрольные результаты и результаты испытаний ИМ, исследователь проверяет *адекватность* модели объекту.

Как видно из схемы на рисунке 1.1, при обнаружении ошибок на этапе отладки, устранимых только на предыдущих этапах, может иметь место возврат на предыдущий этап. Кроме технической документации к результатам этапа прилагается машинная реализация модели (программа, оттранслированная в машинном коде ЭВМ, на которой будет происходить имитация).

Испытание модели. Это важный этап создания модели. При этом необходимо выполнить следующее. Во-первых, убедиться в правильности **динамики** развития алгоритма моделирования объекта исследования в ходе имитации его функционирования (провести *верификацию* модели). Во-вторых, определить степень адекватности модели и объекта исследования. Под *адекватностью* программной имитационной модели реальному объекту понимают совпадение с заданной точностью векторов характеристик поведения объекта и модели. При отсутствии адекватности проводят *калибровку* имитационной модели («подправляют характеристики алгоритмов компонент модели»).

Наличие ошибок во взаимодействии компонент модели возвращает исследователя к этапу создания имитационной модели. Возможно, что в ходе формализации исследователь слишком упростил физические

явления, исключил из рассмотрения ряд важных сторон функционирования системы, что привело к неадекватности модели объекту. В этом случае исследователь должен вернуться к этапу формализации системы (см. рисунок 1.1). В тех случаях, когда выбор способа формализации оказался неудачным, исследователю необходимо повторить этап составления концептуальной модели с учетом новой информации и появившегося опыта. Наконец, когда у исследователя оказалось недостаточно информации об объекте, он должен вернуться к этапу составления содержательного описания системы и уточнить его с учетом результатов испытания предыдущей модели системы.

Исследование свойств имитационной модели. При этом оцениваются *точность* имитации явлений, *устойчивость* результатов моделирования, *чувствительность* критериев качества к изменению параметров модели. Получить эти оценки в ряде случаев бывает весьма сложно. Однако без успешных результатов этой работы доверия к модели не будет ни у разработчика, ни у заказчика ИМ. У разных исследователей в зависимости от вида ИМ сложились различные интерпретации понятий точности, устойчивости, стационарности, чувствительности ИМ. Пока не существует общепринятой теории имитации явлений на ЭВМ. Каждому исследователю приходится полагаться на свой опыт организации имитации и на свое понимание особенностей объекта моделирования. Ниже авторы излагают личный опыт выполнения такой работы и не претендует на безусловность определений.

Точность имитации явлений представляет собой оценку влияния стохастических элементов на функционирование модели сложной системы.

Устойчивость результатов моделирования характеризуется сходимостью контролируемого параметра моделирования к определенной величине при увеличении времени моделирования варианта сложной системы (когда проверяются маловероятные ситуации в системе).

Стационарность режима моделирования характеризует собой некоторое установившееся равновесие процессов в модели системы, когда дальнейшая имитация бессмысленна, поскольку новой информации из модели исследователь не получит и продолжение имитации практически приводит только к увеличению затрат машинного времени. Такую возможность необходимо предусмотреть и разработать способ определения момента достижения стационарного режима моделирования.

Чувствительность ИМ определяется величиной минимального приращения выбранного критерия качества, вычисляемого по статистикам моделирования, при последовательном варьировании параметров моделирования на всем диапазоне их изменений, которая может влиять на результаты ИМ.

1.4 Этапы эксплуатации программ ИМ СС

Эти этапы начинаются с *составления плана эксперимента*, позволяющего исследователю получить максимум информации при минимальных усилиях на вычисления. Обязательно статистическое обоснование плана эксперимента. Планирование эксперимента представляет собой процедуру выбора числа и условий проведения опытов, необходимых и достаточных для решения поставленной задачи с требуемой точностью. При этом существенно следующее: стремление к минимизации общего числа опытов, обеспечение возможности одновременного варьирования всеми переменными; использование математического аппарата, формализующего многие действия экспериментаторов; выбор четкой стратегии, позволяющей принимать обоснованные решения после каждой серии экспериментов на модели.

Затем исследователь приступает к этапу *проведения рабочих расчетов* на модели. Это весьма трудоемкий процесс, требующий больших затрат ресурса ЭВМ. Отметим, что уже на ранних этапах создания ИМ необходимо тщательно продумывать состав и объемы информации моделирования, чтобы существенно облегчить дальнейший анализ результатов имитации. Итогом работы являются результаты моделирования.

Анализ результатов моделирования. Данный этап завершает технологическую цепочку этапов создания и использования имитационных моделей. Получив результаты моделирования, исследователь приступает к интерпретации результатов. Здесь возможны следующие циклы имитации. В первом цикле имитационного эксперимента в ИМ заранее предусмотрен выбор вариантов исследуемой системы путем задания начальных условий имитации для машинной программы модели. Во втором цикле имитационного эксперимента модель модифицируется на языке моделирования, и поэтому зачастую требуются повторная трансляция и редактирование программы.

Возможно, что в ходе интерпретации результатов исследователь установил наличие ошибок либо при создании модели, либо при формализации объекта моделирования. В этих случаях осуществляется возврат на этапы построения описания имитационной модели или на этап составления концептуальной модели системы соответственно.

Трудно дать общие рекомендации по реализации этапов эксплуатации программ ИМ СС, поскольку каждая ИМ создается под конкретный объект исследования или проектирования. Однако, в дальнейшем мы постараемся на типовых примерах описать технологию эксплуатации программ ИМ СС. В качестве примеров объекта имитации выступают:

вычислительный процесс в локальных вычислительных сетях ЭВМ (ВП ЛВС); вероятностные технологические процессы производства (ВТПП); технологические процессы сборочно-разборочных операций ремонта изделий (ТП СРО). С нашей точки зрения, примеров реализации технологии исследования и проектного моделирования шести типов объектов имитации достаточно для обучения технологии имитации сложных систем.

1.5 Формулировка рекомендаций по результатам имитации СС

Результатом этапа интерпретации результатов моделирования являются рекомендации по проектированию системы или ее модификации. Имея в своем распоряжении рекомендации, исследователи приступают к принятию проектных решений. На интерпретацию результатов моделирования оказывают существенное влияние возможности используемой ЭВМ и реализованной на ней системы моделирования. Рекомендации по интерпретации результатов моделирования на имитационных моделях в общем случае давать довольно трудно. Исследователю можно порекомендовать использовать графики как наиболее удобное изобразительное средство. В ряде случаев широко применяются дисплеи и средства для создания кинофильмов о моделируемом явлении. С их помощью можно помочь исследователю наблюдать и изучать явление в замедленном или в ускоренном темпе по сравнению с реальной скоростью протекания исследуемых процессов.

Например, при *проектном моделировании ЛВС* в ходе серии ИЭ исследователь может: получить оценки характеристик качества обслуживания запросов пользователей и загрузки компонентов оборудования; осуществить поиск «узких мест» в заданной структуре ВП ЛВС; оценить эффективность стратегий «свертки» векторов откликов моделирования и критериев принятия решений из выбранного состава вариантов комбинации параметров ЛВС и рабочей нагрузки (РН) на ЛВС; установить степень снижения функциональных характеристик компонентов ВП и РН из-за наличия отказов и восстановления работоспособности основного оборудования ЛВС по сравнению с абсолютно надежной их работой; определить параметры регрессионных зависимостей откликов отказа появления работоспособности компонент оборудования узлов ЛВС. В качестве дополнения к выработке рекомендаций по проектному моделированию ВП ЛВС рекомендуем работы [7; 9; 13].

При проектном моделировании вероятностных технологических процессах производства на ИМ ВТПП исследователь может: определить статические оценки критического времени реализации множества взаимосвязанных микротехнологических операций $\{МТХО_{ij}\}$, образующих вероятностный технологический процесс (ВТПП), получить граф критических путей, образуемых при каждой реализации ИМ ВТПП согласно процедуре Монте-Карло, с помощью которых оценить наиболее вероятный состав $\{МТХО_{ij}\}$, входящих в критический путь реализации вероятностного сетевого графика (ВСГР); определить наиболее опасные участки развития ВТПП; оценить общую пропускную способность вариантов организации ВТПП при имеющемся составе ресурсов и оборудования для их реализации. Методика решения части из перечисленных задач исследования ВТПП с помощью ИМ приведена в третьей части пособия. Дополнением по выработке рекомендаций при проектном моделировании ВТПП могут послужить методики, изложенные в работах [20; 26, 27].

При проектном моделировании или при эксплуатации технологических процессов сборочно-разборочного производства ремонта изделий сложной структуры (ТП СРП) с помощью имитационной модели исследователь может: оценить пропускную способность ТП СРП ремонта изделий сложной структуры; найти «узкие места» в организации ТП СРП и исследовать динамику изменения «узких мест» при изменении параметров ТП СРП; оценить эффективность стратегий ремонта изделий на предприятии; определить допустимые диапазоны изменения откликов ИМ ТП СРП и общего показателя эффективности при росте интенсивности поступления на предприятие изделий, требующих ремонта согласно заданной технологии. Методика решения некоторых из перечисленных задач с помощью ИМ ТП СРП приведена в третьей части издания. Для более подробного ознакомления рекомендуем работу [14].

В конечном итоге после выполнения всех перечисленных выше итерационных этапов исследователь либо окажется удовлетворенным результатами моделирования и будет их учитывать при проектировании сложной системы, либо забракует проектируемую систему и сформирует техническое задание на разработку новой архитектуры системы.

1.6 Итоги

Подведем итоги главы 1. Вначале в 1.1 анализируются роль и место использования ИМ при проектировании СС, рассматриваются виды

моделирования. В 1.2 формулируется состав действий исследователя при создании ИМ СС на восьми этапах исследования. На этих испытаниях ИМ СС в 1.3 исследователю предлагается следующая последовательность операций: проверка достоверности программы ИМ; создание ТД на ИМ; верификация ИМ; калибровка ИМ; исследование свойств ИМ; оценка точности ИМ; анализ устойчивости имитации; определение области стационарного режима имитации; анализ чувствительности ИМ. Далее в 1.4 обсуждаются вопросы: составления плана ИЭ; организации рабочих расчетов с помощью ИМ; анализа результатов моделирования. Завершается глава обсуждением способов формулировки решения при проектном моделировании и при эксплуатации ИМ СС.

2 СОСТАВЛЕНИЕ СОДЕРЖАТЕЛЬНОГО ОПИСАНИЯ ОБЪЕКТА ИМИТАЦИИ

2.1 Методика составления содержательного описания СС

Создание имитационной модели сложной системы начинается с постановки задачи. Но зачастую заказчик ставит задачу недостаточно четко. Поэтому работа обычно начинается с поискового изучения системы. Она порождает новую информацию, касающуюся ограничений, задач и возможных альтернативных вариантов. Отправной точкой при построении модели сложной системы будем считать исходную документацию на существующую систему или техническое задание на проектируемую систему. Совокупность сведений об объекте моделирования представляется в ней в виде схем, текстов, таблиц экспериментальных данных, характеризующих предполагаемую структуру и функционирование системы. В исходной документации одновременно может быть как избыток информации об одних аспектах поведения системы, так и недостаток информации о других аспектах функционирования системы. Имеется также информация о внешних воздействиях окружающей среды. Как правило, информации, представленной в исходной документации, бывает недостаточно для описания поведения сложной системы и изучения различных сторон ее поведения. Зачастую этого и не требуется. Задача может оказаться значительно уже. Поэтому необходимо сначала четко определить цель будущего исследования на модели, а затем в соответствии с этой целью, переработать весь объем исходной информации и восполнить недостающую информацию. Такой процесс назовем составлением *содержательного описания* сложной системы.

Рекомендуется следующая последовательность действий при составлении содержательного описания сложной системы: выбор показателей качества, отображающих цели моделирования; определение управляющих переменных, выбор состава контролируемых характеристик объекта моделирования; детализация описания режимов функционирования системы; дополнение описания информацией о внешней среде. На этом этапе исследователь широко применяет опрос инженерно-технического персонала или специалистов, имеющих опыт работы с прототипом проектируемой или реально существующей системы, поведение которой необходимо исследовать.

Выбор показателей качества моделируемой системы. Этот достаточно важный шаг этапа определяется теми задачами, для решения которых и строится модель. Часто наблюдается плохая тенденция имитировать все, что касается поведения объекта исследования. Следует строить модель, ориентированную на решение лишь тех вопросов, на которые требуется найти ответы, а не имитировать реальную систему во всех подробностях. Можно сослаться на закон Парето, который гласит, что «в каждой группе или совокупности существуют жизненно важное меньшинство и тривиальное большинство. Ничего действительно важного не происходит, пока не затронуто жизненно важное меньшинство» [30]. На первом этапе важно отделить главное от второстепенного. Например, если исследователя интересуют вопросы взаимодействия компонент в системе, то ему не следует изучать проблемы энергоснабжения и экономические аспекты функционирования системы. Выбор цели моделирования определяет характеристики, которые отражают поведение сложной системы. В дальнейшем вся работа сводится к выявлению и детализации тех аспектов функционирования системы, которые имеют отношение к выбранным показателям.

Определение управляющих переменных системы. Изучается техническая документация, по которой прослеживается информация, относящаяся к управлению системой, и анализируется на предмет общности и различия. В соответствии с описанием компонент реальной системы прототипа или согласно техническому заданию на проектируемую систему устанавливается состав управляемых и контролируемых характеристик объекта моделирования. Прежде всего выделяются те характеристики управления системой и контроля за ее работой, которые имеют отношение к цели моделирования. Все составляющие функциональной зависимости, определяющие значение показателя качества системы, включаются в состав управляющих переменных и контролируемых характеристик моделирования.

Детализация описания режимов функционирования системы. Перерабатывается и дополняется имеющаяся информация для возможного выделения алгоритмов функционирования в каждом из режимов работы системы. Любые неясности устраняются. Составляются временные диаграммы функционирования системы. Определяются наиболее неясные или сложные моменты функционирования компонент системы, устанавливается последовательность действий этих компонент, выделяются вероятные места возникновения конфликтных ситуаций и приводится принятый порядок их разрешения в системе.

Составление описания внешней среды. На этом завершающем шаге этапа информация о поведении внешней среды берется из

технического задания. В случае моделирования отдельных аспектов функционирования существующей системы проводится исследовательская работа, цель которой состоит в определении алгоритмов взаимодействия системы с внешней средой. Иногда возможны модификация или пополнение состава управляющих переменных системы из-за детализации алгоритмов взаимодействия между системой и внешней средой.

Таким образом, на каждом шаге данного этапа перерабатывается и дополняется имеющаяся информация о поведении системы в соответствии с поставленными целями моделирования. Результатом является содержательное описание сложной системы, выполненное на языке, принятом при составлении технических отчетов и понятном заказчику. Все лишнее, не относящееся к задаче моделирования, отбрасывается. Цель моделирования поставлена – и можно приступать к математической постановке задачи моделирования. Общего рецепта составления содержательного описания не существует. Успех зависит от интуиции и профессиональной подготовки разработчиков имитационной модели.

В последующих пунктах мы рассмотрим ряд примеров составления содержательного описания шести типов объектов имитации сложных систем разной степени сложности. Во всех этих случаях объекты имитации имеют вероятностную природу и для них (как правило) не выполняются предпосылки для использования аналитических моделей. Поэтому приходится использовать имитацию.

2.2 Описание вычислительного процесса в локальных вычислительных сетях

Вычислительная система (ВС) представляет типовой пример сложной системы. Исследования динамики организации вычислительного процесса (ВП) в ВС проводились многими исследователями [9]. Для выполнения технологического моделирования динамики ВП в ВС достаточно рассмотреть только процессы расхода ресурсов ВС запросами пользователей. Мы рассмотрим два примера организации ВП в ВС: **Пример 1** – организация обработки информации в локальных вычислительных сетях (ЛВС); **Пример 2** – организация решения фоновых режимов в узле ЛВС.

В примере 1 объектом исследования является организация ВП при обслуживании запросов пользователей локальных вычислительных сетей (ЛВС). Множество запросов пользователей ЛВС представим как рабочую нагрузку (РН) на ЛВС. **В примере 2** исследуется организация одного из режимов ВП в узле ЛВС, являющаяся сервером ЛВС.

Исследование ВП в ЛВС возможно различными способами. Общеизвестна многоуровневая классификация режимов обработки в этих системах [7, 13]. Исследование ВП в таких сложных системах можно проводить либо исходя из интересов разработчиков персональных компьютеров (ПК) и их программного обеспечения (ПО), либо исходя из интересов проектировщиков систем. Если первых интересует влияние структурных или функциональных изменений в системе на эффективность их использования и производительность обработки информации с их точки зрения, то вторые рассматривают систему в контексте взаимодействия одних компонентов системы с другими компонентами. Важной задачей исследования ЛВС является необходимость отображения на высоком уровне детализации динамики организации передачи информации на канальном и транспортном уровнях. Фиксированный характер передачи информации в таких системах и зависимость расхода ресурсов ЛВС на единицу передаваемой информации позволили исследователям провести моделирование процессов на двух верхних уровнях детализации этой модели обработки информации [5, 24], в результате чего были разработаны стандартные проколы передачи информации. В связи с этим в работе [21] ВП в ЛВС рассматривается на пяти уровнях детализации:

- последовательности взаимодействия узлов ЛВС при выполнении запросов пользователей («топология ЛВС», на двух УД);
- динамики использования ресурсов ЛВС при выполнении задач пользователей («задачи ЛВС»);
- отображения технологии выполнения процессов пользователей в узлах («Процессы ЛВС»);
- реализации операционной системой интерфейсов между процессами при использовании ресурсов ЛВС («Функции ЛВС»).

Каждый из этих аспектов представления ВП, несмотря на высокий уровень детализации технологии обслуживания запросов пользователей, представляет лишь отдельную сторону объекта исследования. Любая ЛВС на каждом из уровней детализации (УД) согласно [8] состоит из трех компонент: ресурсов, процессов, операционной системы сети (ОС). Как уже упоминалось ранее, рабочая нагрузка на ЛВС представляется совокупностью использований ресурсов ЛВС. На рисунке 2.1 приведен многоуровневый граф отображения РН на ЛВС при пяти уровнях детализации ВП в ЛВС.

На первом УД ВП в ЛВС («Топология») воздействия РН представляются в виде множества запросов на ресурсы системы $\{ZP_i\}$; i – номера запросов, каждый из которых в общем виде имеет графовую структуру (на рисунке 2.1 ZP_i представляется графом топологий ЛВС (G_{r1})).

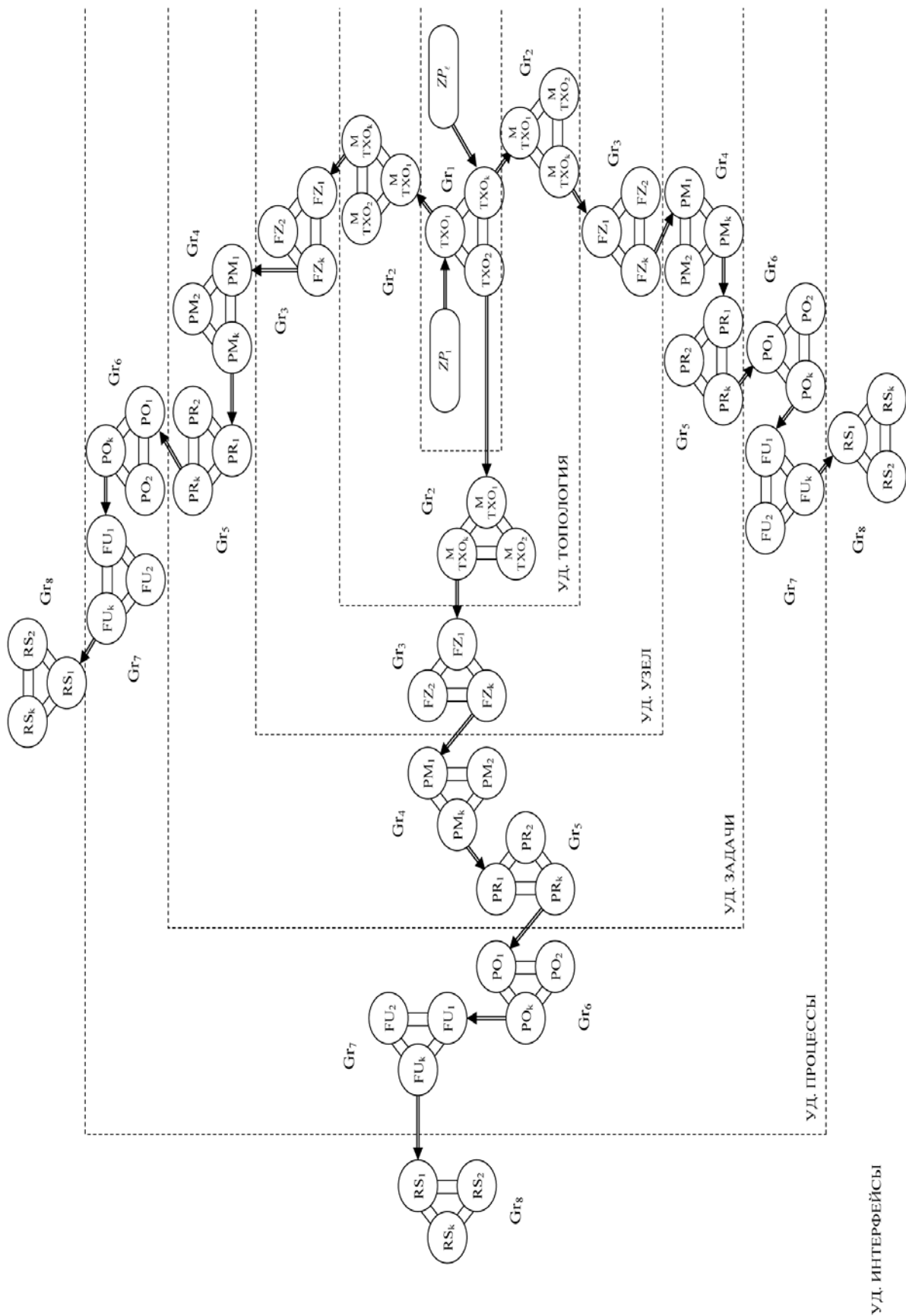


Рисунок 2.1 – Граф отображения ВП и РН на ЛВС

В свою очередь, каждый элемент этого графа есть k -я технологическая операция ($ТХО_j$), означающая определяющую обработку информации с использованием ресурсов ЛВС. Динамика взаимодействия РН с ВП ЛВС представляется графом связи технологических операций (G_{r2}), которые необходимо выполнить для реализаций $\{ТХО_k\}$. На рисунке 2.1 каждая из $ТХО_k$ представляется графом G_{r3} связей микротехнологических операций ($МТХО_k$). На этом УД ВП в ЛВС только $МТХО_k$ использует ресурсы узла ЛВС. Таким образом, на *первом* УД ЛВС («Топология») ВП представляется множеством использований ресурсов ЛВС с помощью вложенных друг в друга двух типов графов (G_{r2} в G_{r1}). Это означает, что для первого УД ЛВС представить ВП можно двухярусной имитационной моделью (ИМ). На *втором* УД ВП ЛВС выполнение каждой $МТХО_k$ в свою очередь представляется графом G_{r3} , узлами которого будут функциональные модули обработки информации (FZ_k). На втором УД ВП ЛВС представляется имитационной моделью узла ЛВС. В общем случае динамика имитации ВП ЛВС на втором УД уже представляется взаимодействием трех вложенных друг в друга графов (G_{r1} , G_{r2} , G_{r3}). На *третьем* УД ВП ЛВС («Задачи ЛВС») каждый функциональный модуль FZ_k реализуется взаимодействием вложенных друг в друга графов: отображения выполнения программных модулей (PM_k) в виде графа G_{r4} и графа реализации PM_k с помощью программных процессов (PR_k), совокупность которых представима в виде графа G_{r5} . Как видим, на третьем УД ВП в ЛВС («Задачи ЛВС») представляет собой совокупность вложенных друг в друга графов G_{r4} и G_{r5} . На *четвертом* УД ВП ЛВС («Процессы ЛВС») каждый программный процесс (PR_k) также представляет собой взаимодействие двух вложенных друг в друга графов G_{r6} и G_{r7} . Здесь могут использоваться частные имитационные модели использования ресурсов ЛВС ИМ («Процессов»). Граф G_{r6} отображает связи программных потоков (PO_k), реализуемых ОС для выполнения программных процессов PR_k . В свою очередь, каждый из программных потоков PO_k может быть представлен графом использования ресурсов ЛВС при выполнении системных функций (G_{r7}). Элементами G_{r7} являются системные функции (FU_k), которые реализуются ОС ЛВС на процессоре узлов ЛВС. Как видим, на четвертом УД ВП ЛВС можно проводить исследования с помощью частной ИМ, представляющей собой комбинацию вложенных друг в друга графов G_{r6} и G_{r7} . Наконец, на *пятом* УД ВП ЛВС отображается использование ресурсов узла ЛВС при выполнении функций (FU_k), которое представляется графом G_{r8} . Каждый узел графа G_{r8} представляет собой использование ресурсов (RS_k) узлов ЛВС.

На этом уровне детализации для имитации технологии использования ресурсов узлов ЛВС ресурсами, относящимися к аппаратной базе ЛВС, являются: центральный процессор (CPU), внешняя память (НДД), оперативная память (Mem), видеоресурсы (Wideo), сетевой ресурс обеспечения удаленных соединений (NEt). Каждая из этих аппаратных компонент обладает собственным алгоритмом поведения, зависящим от запросов пользователей на этот ресурс. Но для технологической имитации динамики взаимодействия этих компонент друг с другом и рабочей нагрузкой (запросами пользователя) достаточно отображать это взаимодействие либо с помощью времени их использования, либо путем задания объемов обрабатываемой информации. Программное обеспечение ЛВС, функционирующее на аппаратной базе ЛВС, представляется взаимодействием процессов использования ресурсов ЛВС. Таких процессов существует два типа: пользовательские и системные пользовательские процессы генерируются непосредственно по запросам РН на ЛВС, а системные процессы генерируются ОС ЛВС для выполнения системных функций по организации ВП ЛВС. При технологической имитации эмуляции алгоритмов ОС не требуются, и поэтому достаточно представить выполнение ОС в виде времени расхода ресурса CPU.

Как видим, динамика использования ресурсов узла ЛВС при выполнении РН представляется на самом высоком УД ВП ЛВС («Интерфейсы ОС») также с помощью имитационной модели ВП ЛВС. Отметим, что транспортный и канальный уровни семиуровневой модели ЛВС обработки информации [7] в ЛВС с помощью модели многократно исследованы и обычно даже представляются с помощью различных протоколов информационной связи в узлах ЛВС. Поэтому можно отказаться от дальнейшего повышения УД ВП в ЛВС.

Таким образом, ВП ЛВС как объект исследования представляет собой иерархическую сложную систему. Очевидно, что такого рода сложную систему реально исследовать только на основе декомпозиции при последующем представлении ВП в ЛВС частных ИМ. Первый УД ВП ЛВС исследован в работе [24]. Исследованиям на втором УД ВП ЛВС посвящена работа [8]. В работе [5] приведено исследование ВП ЛВС на третьем УД. Моделированию ВП ЛВС на четвертом и пятом уровнях детализации посвящена работа [21]. Завершает цикл исследований работа [7], в которой приведены результаты четырех предыдущих исследований.

Вышеизложенное представляет собой содержательное описание взаимодействия компонент такой сложной системы, как ВП ЛВС, на пяти уровнях детализации. Для каждого УД ВП ЛВС задаются списки:

динамики взаимодействия показателей качества компонентов ВП; описания характеристик объекта имитации (откликов и управляющих переменных параметров, моделирования); формулировка целей и задач технологического моделирования ВП. Проще всего это можно сделать с помощью соответствующих ИМ.

Содержательное описание ВП ЛВС первого уровня детализации «Топология» (Пример 1).

Динамика взаимодействия компонентов ВП уровня детализации «Топология» приведена выше. Блок-схема взаимодействия компонентов ВП приведена на рисунке 2.2. Как видим, ВП в ИМ ВП ЛВС на первом уровне его детализации представлен взаимодействием двух вложенных друг в друга графов (G_{r1} и G_{r2}).

Поскольку детализации механизма использования ресурсов ЛВС в данной частной ИМ ВП ЛВС не предусмотрено, то будем считать, что при выполнении $MTXO_{lk}$ используются основные ресурсы узлов ЛВС: центральный процессор (CPU_{lk}), внешняя память (HDD_{lk}), транзитные устройства передачи информации (TP_{lk}). На входы узлов ЛВС поступают запросы i -го типа (ZP_i), и поэтому важной характеристикой качества обслуживания запросов является время жизни запроса в системе ($T_{жi}$). Оборудование устройств TR_{lk} и программное обеспечение, осуществляющее передачу информации в ЛВС, представим в виде канала передачи данных. Номера rs (здесь r и s идентифицируются номерами узлов ЛВС). На рисунке 2.2 CPU_{lk} , HDD_{lk} , TP_{lk} и терминал пользователя (R_{lk}) представляют собой ресурсы ЛВС. Пунктирными линиями отображаются границы графов G_{r1} , и факт, что динамика выполнения TXO_k отображается соответствующим графом G_{r2k} .

Откликами имитационного моделирования (Y_1) являются: коэффициенты занятости ресурса CPU l -го узла обслуживания запросов i -го типа (η_{CPUil}); коэффициенты занятости l -го узла транзитными запросами (η_{TP1}); среднее время жизни в ЛВС запросов i -го типа (T_{ki}); количество промежуточных узлов ЛВС на трафике запросов i -го типа (KN_i).

Параметрами (X_{12}) являются: общее количество узлов и порядковые номера узлов в графе топологий G_r (K_l); пропускная способность канала номера rs и стоимость использования в единицу времени (W_{rs} , C_{rs}); средняя длина сообщений, формулируемых каналом номера rs (d_{rs}); среднее время обработки i -го запроса в ЛВС ($\tau_{обр}$); скорость выполнения запросов на CPU в k -м узле ЛВС (v_{CPUk}); общий объем внешней памяти в k -м узле ЛВС ($V_{ВНПk}$); средний объем транзитного сообщения через узел k запроса i -го типа или средний объем транзитного ответа через ЛВС для i -го запроса ($V_{ТРОki}$).

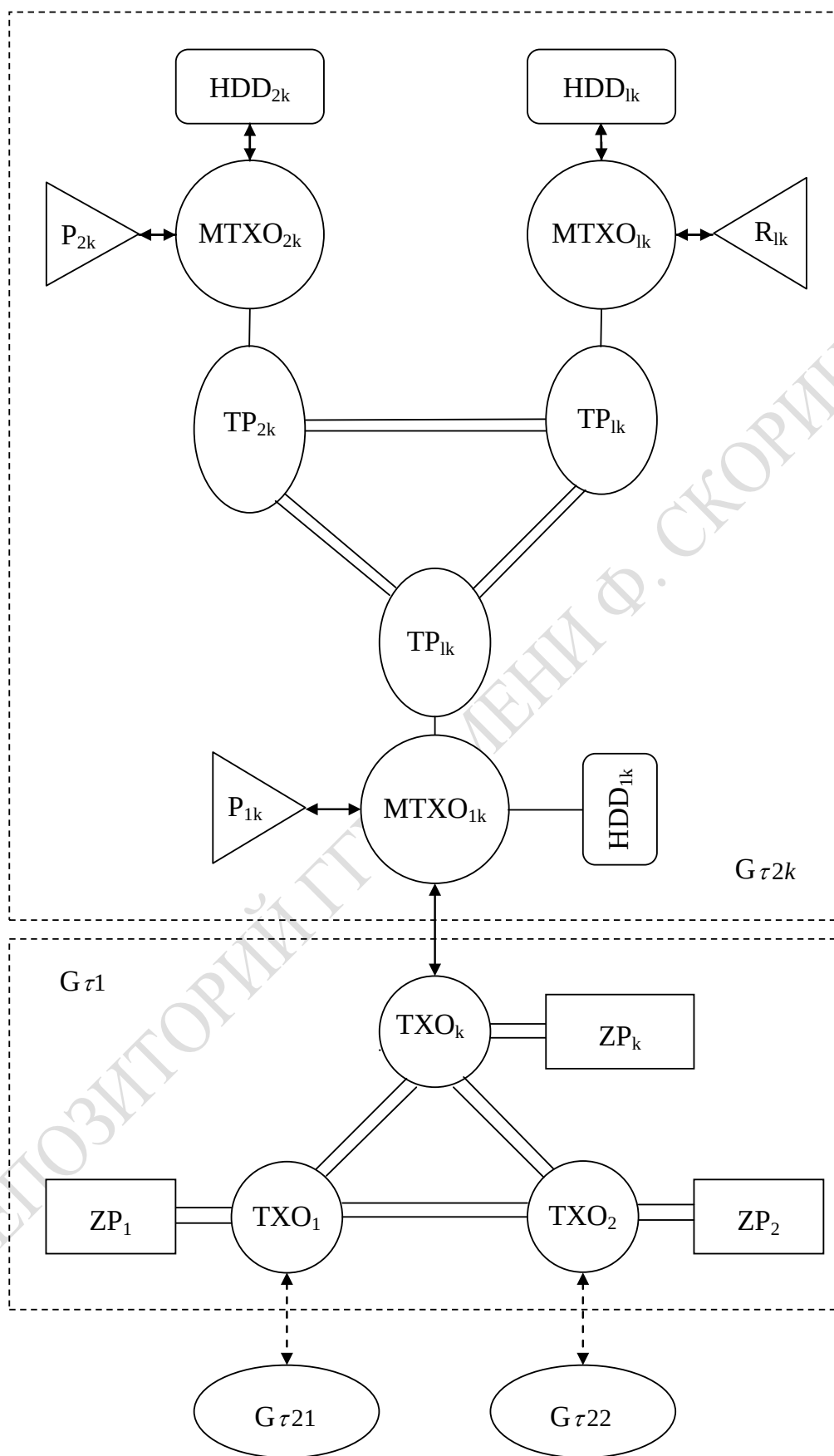


Рисунок 2.2 – Граф взаимодействия РН и ВП в ЛВС
уровня детализации «Топология»

В качестве вектора *ограничений* (OG_1) используются: стоимость услуг узла ЛВС i -му запросу (C_{1i}); пропускная способность k -го канала связи в ЛВС (Q_k); задержка запроса в системах передачи k -го канала ЛВС (T_k).

Сопутствующими *статистиками* имитации на первом УД ВП ЛВС являются: общее время использования всех CPU $\left(\sum_k \tau_{CPU_k}\right)$; средняя длина очереди запросов к l -му узлу ЛВС (l_{orl}); среднее время ожидания запросов в узлах l -го типа (l_{orl}) компонентами вектора. Состояниями первого УД ВП в ЛВС (S_1) на момент времени t_0 являются: число узлов ЛВС, занятых обслуживанием ($v_{обт0}$); число свободных линий связи ($v_{св0}$).

Компонентами вектора *параметров РН* являются: вероятность запросов i -го типа в данном варианте топологий ЛВС (p_{1i}), интенсивность поступления в ЛВС запросов пользователя i -го типа (λ_{1i}), средние времена обдумывания пользователем i -го типа ответа ЛВС ($\tau_{обд1i}$).

Структура графа FR_1 определяется: вероятностью выбора в качестве начального узла ЛВС номера k в запрос ZP_i (P_{01k}); матрицей следования друг за другом узлов ЛВС в трафике движения запросов (MP_{1k}); распределением числа смен узлов ЛВС в трафике обслуживания ZP_i ($F_{ik}(h)$).

Структура графа G_{r2} определяется: вероятностью (P_{02k}) выбора начального элемента k в последовательности $\{TХO_k\}$, реализующей ZP_i ; матрицей вероятностей выполнения друг за другом $TХO_k$ при выполнении ZP_i (MP_{2k}); распределением вероятностей числа смен $TХO_k$ при выполнении ZP_i ; матрицей вероятностей условных распределений длительностей использования узла ЛВС при реализации $TХO_k$ ($MFV_2(t)$), матрицей коэффициентов пересчета времен выполнения $TХO_k$ при переходе узла ЛВС на другую элементную базу (M_{y2k}).

С содержательным описанием ВП и РН на ЛВС для других уровней детализации можно познакомиться в работе [24], где приведено описание УД ВП ЛВС «Узел»; в [4 и 5] дано описание УД ВП ЛВС «Задачи»; в [21] содержится описание ВП ЛВС на УД и «Процессы» и «Функции ОС».

Целью имитационного моделирования является проектное моделирование рациональной структуры и состава ресурсов ЛВС, обеспечивающих максимально возможную загрузку ресурсов ЛВС при удовлетворительном качестве обслуживания запросов РН на ЛВС. Для этого необходимо решить следующие задачи:

1) за счет предложенного автором [24] алгоритма маршрутизации на ИМ исследовать характеристики наиболее вероятных маршрутов запросов пользователей по ЛВС фиксированного состава;

2) осуществить поиск «узких мест» в структуре ВП и РН, заданной графами G_{r1} и G_{r2} ;

3) провести оценку изменения величины пропускной способности ЛВС различных топологий и выбор состава и структуры ВП в ЛВС, обеспечивающего максимум суммарного коэффициента использования ресурсов ЛВС для i -го типа запросов пользователей.

При этом предполагается, что оборудование ЛВС является абсолютно надежным.

Содержательное описание ВП ЛВС третьего уровня детализации «УЗЕЛ» (Пример 2).

Объектом исследований является реализация ВП на узле ЛВС под воздействием РН. Сам ВП представляет собой способ организации использования ресурсов узлов ЛВС при выполнении заданий пользователей. Пользователи в ЛВС обычно взаимодействуют с ЛВС в следующих режимах: диалоговом (DR), дистанционном доступе к ресурсам узлов ЛВС (DIS), фоновом режиме путем отложенного счета (РАК). На рисунке 2.3 приведена блок-схема использования РН ресурсов узла ЛВС.

В режиме DR пользователи ЛВС обслуживаются приоритетным образом, и расход ресурсов ЛВС ограничивается тем узлом ЛВС, дисплеи которого они используют. В режиме DIS пользователи осуществляют обработку информации на других узлах ЛВС. Поэтому по сети движутся транзитные запросы, которые могут быть двух видов: либо это запросы на обработку информации в другом узле ЛВС, либо это результаты обработки возвращаются на тот узел, где находится пользователь, пославший заказ на обработку на соседнем узле ЛВС. В режиме РАК вся обработка информации осуществляется на одном узле, и чаще всего она выполняется на сервере ЛВС.

В режиме РАК задачи пользователей многократно используют основные ресурсы узла ЛВС (CPU_k и HDD_k) в течение длительных интервалов времени. Поэтому они выполняются на фоне приоритетного чередования вычислений по запросам режимов DR и DIS.

Рабочая нагрузка на узел ЛВС представляет собой множество запросов пользователей на ресурсы узла ЛВС (CPU_k и HDD_k). Запросы пользователей в режимах DR и DIS имеются у исследователя в виде временных диаграмм использования CPU_k и HDD_k узла ЛВС, поступающих в ВП узла ЛВС соответственно с интенсивностями λ_{Dk} и λ_{Tk} . Запросы пользователей режима РАК представляют собой задачи ZD_{ik} (i – номер задачи в пакете) и могут выполняться в моменты, когда

приоритетные запросы уже обнаружены. Поэтому можно считать, что в режиме РАК множество задач $\{ZD_{ik}\}$ объединяется в «пакет задач», который затем в виде отложенного счета обслуживается ОС узла ЛВС в то время, которое удобно системе при использовании ресурсов CPU_k и HDD_k k -го узла ЛВС.

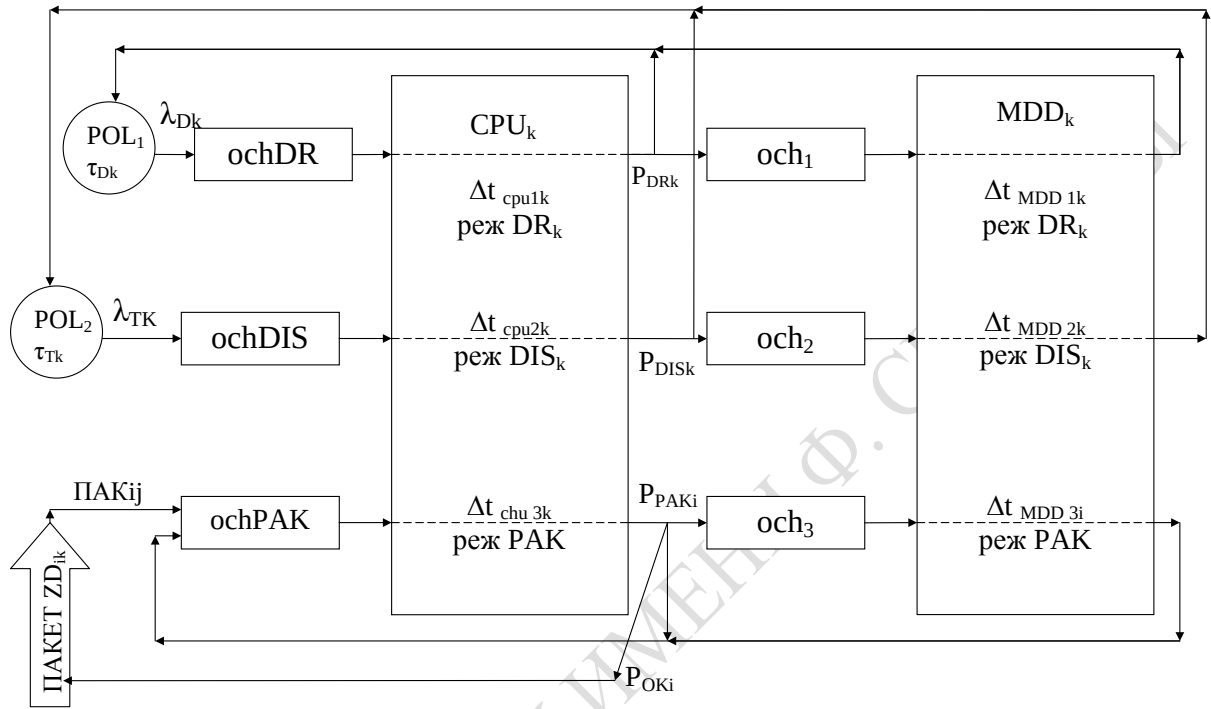


Рисунок 2.3 – Блок-схема использования ресурсов k -ого узла ЛВС.
Запросами пользователей в трёх режимах (DR, DIS, PAK)

Как видно из рисунка 2.3, запросы пользователей поступают в узел ЛВС в трех режимах (DR, DIS, PAK). Запросы пользователей режима DR поступают в самую приоритетную очередь (ochDR) на захват ресурса CPU_{1k} на время Δt_{CPU1k} . По завершении обслуживания этого запроса на CPU_{1k} с вероятностью $(1 - P_{DRk})$ пользователю возвращается ответ процесса, выполнившего на CPU_k заказ пользователя в режиме DR. С вероятностью P_{DRk} запрос поступает в очередь och1 на захват ресурса HDD_k , который после выполнения обработки информации возвращает результаты пользователю в режиме DR. Через время τ_{Dkj} в систему поступает новый запрос пользователя в режиме DR. Запросы пользователей в режиме DIS поступают в менее приоритетную очередь (ochDIS) на захват ресурса CPU_{2k} на время Δt_{CPU2k} . По завершении обслуживания k -го запроса на CPU_{2k} также с вероятностью $(1 - P_{DISk})$ транзитный запрос покидает k -й узел ЛВС. При наличии обмена с HDD_k с вероятностью P_{DISk} транзитный запрос поступает в очередь och2 на захват

ресурса HDD_k , который использует этот ресурс в течение интервала Δt_{HDD2k} , затем покидает k -й узел ЛВС и возвращается к пользователю. Далее циклы обслуживания запросов пользователя в режимах DR и DIS повторяются.

Источником запросов РН в режиме PAR является задача ZD_{jk} из пакета задач (PAK_k), которая в момент окончания выполнения предыдущей задачи поступает в самую неприоритетную очередь ($ochPAK$) на захват ресурса CPU_{3k} . Далее задача i -го типа ZD_{ik} постоянно использует ресурсы k -го узла, являющегося сервером ЛВС; число таких использований сервера ЛВС может быть равно (Ω_i) . Каждый j -й цикл использования задачей ZD_{ik} ресурсов k -го узла ЛВС выполняется с вероятностью $(1 - P_{oki})$ (в общем случае $j = \overline{1, \Omega_i}$). В j -м цикле расхода ресурсов k -го узла ЛВС после ресурса CPU_{3k} , ZD_{ikj} использует ресурс HND_{3k} с вероятностью P_{PAKki} (в этом случае ZD_{ikj} поступает в очередь $och3$ на захват ресурса HND_k). Поэтому после окончания обработки информации на CPU_{3k} запрос ZD_{ikj} с вероятностью $(1 - P_{PAKi} - P_{OKi})$ возвращается в начало очереди $ochPAK$ для повторного использования ресурсов сервера в ЛВС. После окончания использования HDD_{3k} длительностью Δt_{HDD3k} задача ZD_{ikj} также возвращается в начало очереди $ochPAK$, начиная таким образом очередной квант использования задачей ZD_{ikj} основных ресурсов сервера (CPU_{3k} и HDD_{3k}). Как видим, при выполнении j -го цикла ZD_{ikj} использует ресурсы CPU_{3k} и HDD_{3k} многократно в соответствии с вероятностями P_{PAKi} и P_{OKi} . Множество интервалов использования ресурсов сервера ЛВС ($\{\Delta t_{CPU3ij}\}$ и $\{\Delta t_{HDD3ij}\}$), вероятности (P_{PAKi} и P_{OKi}) и число циклов использования ресурсов (Ω_i) являются инвариантами расхода ресурсов i -й задачи и не зависят от операционной обстановки в ВП и порядка их выполнения. В пакете задач может находиться любое число задач ZD_{ik} , и поэтому интервал выполнения всех задач пакета (T_{PAK}) является важной статистикой ВП, характеризующей качество организации режима отложенного счета.

На основе вышеизложенного предметом имитационного моделирования ВП в узлах ЛВС, являющемся сервером ЛВС, является исследование зависимости T_{PAK} от порядка выполнения $\{ZD_{ik}\}$ пакета задач.

Цель имитации – определение такой очередности выполнения множества $\{ZD_{ik}\}$, которая обеспечит $\min T_{PAK}$. Итак, речь идет об адаптации РН режима РАК к имеющемуся составу ресурсов сервера ЛВС при наличии режимов более приоритетного использования ресурсов запросами пользователей, взаимодействующих с узлом ЛВС в режимах DR и DIS.

Параметрами РН для режима DR являются: время обдумывания пользователем результатов предыдущего выполнения j -го запроса на узле k (τ_{Dkj}); средняя длительность использования CPU_k за одно взаимодействие пользователя со своей программой ($\Delta \bar{t}_{\text{CPU}1k}$), средняя длительность использования HDD_k за одно взаимодействие пользователя с ЛВС ($\Delta \bar{t}_{\text{HDD}1k}$). В ходе мониторинга РН формируются соответствующие функции распределения этих параметров РН режима DR: $F_{D1k}(\Delta t_{\text{CPU}})$ и $F_{D2k}(\Delta t_{\text{HDD}})$. Кроме того, определяется:

- среднее время обдумывания пользователем результатов счета в режиме DR (τ_{Dk}), которое позволяет установить величину средней интенсивности поступления на сервер диалоговых запросов ($\lambda_{Dk} = 1/\tau_{Dk}$);

- вероятность использования диалоговыми запросами внешней памяти k -го узла ЛВС (P_{DRk}).

Для режима DIS аналогичным образом определяются подобные характеристики использования ресурсов сервера ЛВС: $F_{T1k}(\Delta t_{\text{CPU}})$, $F_{T2k}(\Delta t_{\text{HDD}})$, $\lambda_{Tk} = 1/\tau_k$, P_{DISk} .

Для режима РАК в ходе мониторинга процесса выполнения пакета задач отложенного счета формируются инварианты использования каждой задачей ZD_{ik} ресурсов CPU сервера ЛВС. Инвариантами ZD_{ik} являются две перемежающиеся последовательности использования ресурсов k -го узла ЛВС: $\{\tau_{\text{CPU}i1l}\}$ – интервалы использования CPU_{3k} задачами ZD_{ik} в режиме РАК, $l = \overline{1, \Omega_l}$ – номер кванта использования CPU_{3k} ; $\{\tau_{\text{HDD}3kl}\}$ – интервалы использования HDD_{3k} задачами ZD_{ik} . Каждый j -й интервал использования CPU_{3k} и HDD_{3k} запросами i -го типа в режиме РАК в общем случае представляет собой непрерывный интервал использования ресурсов сервера ЛВС соответственно ($\Delta t_{\text{CPU}3kj}$ и $\Delta t_{\text{HDD}3kj}$). При этом интервалы использования ресурсов ЛВС (инварианты) представляют собой суммы этих квантов $\tau_{\text{CPU}3ikl} = \sum_j \Delta t_{\text{CPU}3kj}$ и $\tau_{\text{HDD}3ikl} = \sum_j \Delta t_{\text{HDD}3kj}$. Весь пакет ZD_{ik} , решаемых в режиме РАК, характеризуется набором следующих параметров РН:

- $\{F_{pi1l}(\tau)\}$ – множество функций распределения длительностей интервалов использования CPU_{3k} в ZD_{ik} , $l = \overline{1, \Omega_{ik}}$;

- $\{F_{Dikl}(\tau)\}$ – множество функций распределения длительностей интервалов использования HDD_k в ZD_{ik} ;

- Ω_{ik} – количество интервалов в ZD_{ik} . В ходе мониторинга узла ЛВС определяется только общее число ZD_{ik} в пакете обычно решаемых на узле ЛВС равно n_{3k} . Поэтому номер задачи в пакете меняется в диапазоне $ik = \overline{1, n_{3k}}$.

Параметрами ВП узла ЛВС являются:

- скорость обработки кванта информации на основных ресурсах узла ЛВС (v_{CPU_k} и v_{HDD_k});
- квант времени непрерывного выполнения ОС узла ЛВС (Δt_{OC}) на обслуживание одного запроса пользователей (единый для режимов DR, DIS и PAK).

Откликами ВП k -го узла ЛВС являются:

- суммарный коэффициент использования ресурса CPU_k (η_{CPU_k});
- суммарный коэффициент использования ресурса HDD_k (η_{HDD_k});
- общее время выполнения пакета задач ZD_{ik} (T_{PAK}).

Статистиками ВП k -го узла ЛВС являются:

- средние значения длин очередей (och_m) и времени нахождения в них запроса m -го типа (L_{ochm} и t_{ochm}) на захват ресурса CPU_k ;
- средние значения длин очередей och_s и времени нахождения в них запросов m -го типа (L_{ochs} и t_{ochs}) на захват ресурса HDD_k .

При имитации ВП в узле ЛВС необходимо решить следующие задачи:

- оценка пропускной способности (W) пакетного режима решения задач отложенного счета при имеющемся на сервере составе ресурсов ЛВС, существующем РН в режимах DR и DIS и обычно используемом составе пакета задач $\{ZD_{ik}\}$;
- оценка эффективности применения алгоритма адаптации РН к составу ресурсов сервера ЛВС, который был представлен в работе [4];
- выбор такой структуры задач режима PAK (STR PAK), который обеспечивает $\max W$.

2.3 Содержательное описание поточного производства изготовления и ремонта изделий

Рассмотрим два примера поточного производства, в которых сложность связей возрастает: участок обработки деталей несколькими станками; процесс ремонта изделий сложной структуры, сопровождающийся операциями сборки и разборки компонентов изделия. Для каждого примера ТПП последовательно рассмотрим реализацию методики составления содержательного описания объекта имитации, изложенную ранее в 2.1.

Описание участка обработки деталей несколькими станками.

Рассмотрим некоторый участок цеха обработки деталей двух типов четырьмя станками [15]. Схема обработки деталей в цехе приведена на рисунке 2.4. Деталь l -го типа обрабатывается строго последовательно

станками $ОСТ_1$, $ОСТ_2$, $ОСТ_3$. Выбор типа детали при ее обработке на станке $ОСТ_1$ случайный. Запрещена одновременная обработка деталей разных типов станками $ОСТ_1$ и $ОСТ_2$. Обработка деталей 1-го и 2-го типов соответственно станками $ОСТ_3$ и $ОСТ_4$ осуществляется только в том случае, когда эти станки свободны. На каждом из станков $ОСТ_j$ ($j = \overline{1,4}$) обработка детали производится в порядке поступления деталей. Приведенный текст схемы обработки является простейшей документацией, из которой для простоты изложения исключены технологические аспекты, относящиеся к процессу обработки деталей каждым станком.

Выбор показателя качества моделируемого участка обработки деталей определяется назначением моделирования. *Целью моделирования* является изучение влияния интенсивности поступления деталей в цех на загрузку станков, пропускную способность цеха, среднее время полной обработки одной детали. При этом с помощью модели исследователь надеется выявить «узкие места» в технологической епочке обработки деталей. Итак, в качестве показателей качества моделируемой системы будут выступать перечисленные выше характеристики обработки деталей.

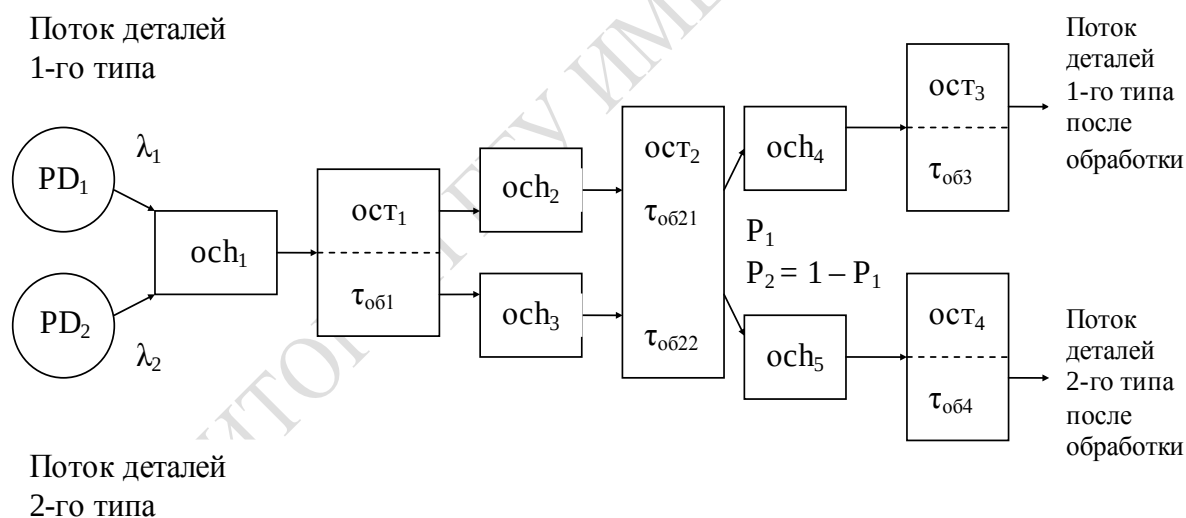


Рисунок 2.4 – Схема обработки деталей в цехе

Управляемыми переменными моделирования (*параметрами*) будут интенсивности поступления деталей в цех на обработку λ_1 и λ_2 (рисунок 2.4). Параметрами цеха обработки являются интервалы обработки деталей $\tau_{обij}$. *Контролируемыми характеристиками* объекта моделирования для данного случая являются *статистики* моделирования, позволяющие определить: загрузку станков (суммарное время обработки деталей каждым станком), пропускную способность цеха

(количество деталей каждого типа, обработанных цехом), среднее время полной обработки одной детали (интервал от момента поступления детали в цех на обработку до момента завершения ее обработки в цехе).

Детализация описания режимов функционирования сводится к следующим действиям. С помощью хронометража или на основании нормативных данных определяются выборки значений времен обработки деталей станками $\{\tau_{ij}\}$. Здесь индексы i и j означают принадлежность выборки значений τ соответственно к i -му типу деталей, обрабатываемых j -м станком. Затем известными статистическими методами по выборкам $\{\tau_{ij}\}$ формируются соответствующие функции распределения времени обработки i -й детали j -м станком $F_{ij}(\tau)$. Заметим, что на данном шаге для нашего примера не рассматриваются контроль качества деталей, обеспечение станков обслуживающим персоналом, порядок и технология обработки деталей каждым станком и т. д. Эти вопросы в соответствии с рассматриваемой целью моделирования являются лишними.

Описание внешней среды сводится к указанию механизма выбора типа деталей i . Детали 1-го типа поступают на обработку с вероятностью p_1 , а детали 2-го типа – с вероятностью p_2 . Поступление деталей в цех не зависит ни от ситуации в цехе, ни от времени, ни от продолжительности работы цеха.

Приведенный выше текст является содержательным описанием объекта моделирования – в нашем примере участка цеха обработки деталей двух типов четырьмя станками. Содержательное описание объекта моделирования для нашего примера включает следующую документацию:

- схему, приведенную на рисунке 2.4;
- текст, описывающий порядок обработки деталей станками;
- функции распределения $F_{ij}(\tau)$, представленные либо аналитически, либо в табличном виде.

Как видим, в силу простоты примера объекта имитации само содержательное описание достаточно короткое. Но этот пример показывает, насколько сильно упрощается описательное функционирование цеха обработки деталей из-за простоты цели имитации состава компонентов СС.

Описание технологии сборочно-разборочных операций **ремонта деталей сложной структуры**. Здесь объект имитации гораздо сложнее, и поэтому нужно более подробное содержательное описание объекта моделирования. Рассматривается технология поточного производства ремонта изделий сложной структуры на предприятии,

осуществляющем несколько типов ремонта и сборки изделий. Ремонт изделий сопровождается операциями их разборки в различных цехах. Цеха специализируются по типам ремонтных операций, (РОП) для выполнения которых на k -х рабочих местах (PM_k) используются различные ресурсы предприятия номера r (RES_r) [23].

Целью имитации является оценка пропускной способности предприятия (W) и поиск оптимального состава рабочих мест $\{PM_k\}$ при заданных интенсивностях поступления i -го типа изделий на ремонт (λ_i) и имеющемся множестве ресурсов предприятия $\{RES_r\}$ для выполнения РОП.

Особенностями объекта исследования являются:

- разделение технологических потоков ремонтируемых изделий на несколько уровней иерархии с разной технологией ремонта изделий;

- наличие трех типов операций (разборка, сборка, ремонт деталей изделия);

- цеховая структура объединения $\{PM_k\}$, каждый из цехов специализируется по определенному составу микротехнологических операций $\{MTXO_{ij}\}$, где i – порядковый номер РОП, ij – номер их следования $MTXO_{ij}$ в технологической цепочке ремонта изделия;

- привязанность PM_k к ресурсам предприятия, ограниченный состав ресурсов RES_r ($r \leq R$) на предприятии;

- ограниченный состав ремонтных бригад $\{PR_m\}$ и исполнителей $\{ISP_n\}$, которые совместно выполняют множество РОП на соответствующих PM_k ;

- при выполнении РОП на PM_k возможно использование ресурсов индивидуального и общего пользования (используемых частично несколькими РОП) на время выполнения РОП, и затем – снова возврат их для использования уже другой РОП;

- для некоторых РОП на ряде PM_k возможно использование специализированного оборудования, индивидуально закрепляемого PM_k ;

- имеется возможность заказа дополнительного оборудования, материалов и комплектующих изделий при выполнении РОП.

Отметим особенности ремонтируемых изделий (РИ) на предприятии:

- в изделии имеет место иерархия вложенности компонентов друг в друга;

- каждая компонента верхнего уровня вложенности содержит в себе несколько составляющих более низкого уровня вложенности (причем характер этой вложенности определяется местом $MTXO_{ij}$ в составе ТП СРП);

– для РОП исполнителем на PM_k с помощью соответствующего оборудования используется необходимое количество материалов (mt) и комплектующих деталей (КО);

– только часть компонентов изделий может требовать выполнения определенных РОП, однако для других компонентов РОП ремонтные операции могут не потребоваться; но при этом все же может потребоваться выполнение сборочно-разборочных операций над компонентами изделий.

Бл 1		Бл2					Бл3					Бл1	
Эл1		Эл6		Эл4	Эл3		Эл7	Эл5		Эл2	Эл1		
D ₂₁		D _{6.1}		D _{4.1}	D _{3.0}		D _{7.1}	D _{5.1}	D _{2.16}		D ₂₄	D ₇	
D ₂₀					D _{3.1}				D _{2.15}				
D ₁₉		D _{6.2}		D _{4.2}	D _{3.2}		D _{5.2}	D _{2.14}		D ₂₃			
D ₁₇					D _{3.3}			D _{2.13}					
D ₁₅		D _{6.10}	D _{6.3}		D _{3.4}		D _{5.3}	D _{2.12}		D ₂₂			
D ₁₃					D _{6.11}	D _{6.4}		D _{3.5}			D _{2.11}		D ₁₈
D ₁₁		D _{6.12}	D _{6.5}				D _{4.3}	D _{3.6}		D _{2.10}		D ₁₆	
D ₉				D _{6.13}	D _{6.6}	D _{4.4}		D _{3.7}		D _{2.9}			D ₁₄
D ₆ MTXO ₁₋₆		D _{6.14}	D _{6.7}				D _{4.5}	D _{3.8}		D _{2.8}		D ₁₂	
D ₅ MTXO ₁₋₅				D _{6.15}	D _{6.8}	D _{4.6}		D _{3.9}		D _{2.7}			D ₁₀
D ₄ MTXO ₁₋₄		D _{6.16}	D _{6.9}				D _{4.6}	D _{3.10}		D _{2.6}		D ₁₀	
				D _{6.17}		D _{4.6}		D _{3.11}		D _{2.5}			D ₁₀
		D _{6.17}					D _{4.6}	D _{3.12}		D _{2.4}		D ₁₀	
				D _{6.17}		D _{4.6}		D _{3.13}		D _{2.3}			D ₁₀
		D _{6.17}					D _{4.6}	D _{3.14}		D _{2.2}		D ₁₀	
				D _{6.17}		D _{4.6}				D _{2.1}			D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}		D _{4.6}							D ₁₀
		D _{6.17}					D _{4.6}					D ₁₀	
				D _{6.17}									

Рисунок 2.5 – Структурная схема сложного изделия Д

Динамика взаимодействия сборочно-разборочных операций (СРО) определяется структурой сложного изделия. На рисунке 2.5 приведена структура сложного изделия. Как видно из рисунка 2.5, имеет место вложенность компонентов изделия Д друг в друга. В изделие Д входит 108 деталей, которые сложным образом объединены в три блока (блоки состоят из элементов, а элементы komponуются из множества деталей). Основой изделия Д является блок № 1. На нем закрепляются блоки (№ 1; № 2; № 3), а также ряд деталей. В блок № 2 входят 3 элемента (эл. № 3; эл. № 4; эл. № 6). Блок № 3 также состоит из 3 элементов (эл. № 7; эл. № 5; эл. № 8).

Перед ремонтом блоков № 2 и № 3 необходимо произвести их отсоединение от основы изделия (блок № 1). Далее, перед ремонтом

блоков № 2 и № 3, необходимо разобрать их на составные части (элементы, входящие в состав блоков). Наконец, для осмотра и ремонта деталей ненужно отделить их от основы элемента. Кроме того, как видно из рисунка 2.5, перед разборкой блоков 2 и 3 следует произвести отделение, осмотр и ремонт 6 деталей изделия Д с помощью соответствующих микротехнологических операций ($MTXO_{lm}$); где l – номер блока, m – порядковый номер детали (D_m). После отсоединения деталей $D_1 - D_6$ с помощью операции разборки ($RAZBOR_1$) от основы отделения блок № 2, который направляется для ремонта в цех № 4. В этом цехе над блоком № 2 с помощью операции $RAZBOR_8$ осуществляется операция разборки на три элемента (эл. № 6; эл. № 4; эл. № 3). Далее эл. № 6 направляется в цех № 6, где выполняются осмотр и ремонт деталей элемента № 6 с помощью операций ($MTXO_{6.1} : MTXO_{6.1.7}$). Над элементом эл. № 4 выполняются две операции – ремонт деталей D_{47} и D_{48} . С помощью операции $RAZBOR_2$ от блока № 2 отделяется эл. № 3, который направляется на ремонт в цех № 3. В этом цехе над изделиями, входящими в эл. № 3, последовательно выполняется 15 операций осмотра и ремонта деталей $MTXO_{3.0} : MTXO_{3.14}$). Отремонтированные детали элементов (эл. № 3 и эл. № 4) с помощью операции $SBORA_2$ снова объединяются в один блок № 2. Затем в цехе № 4 с помощью двух операций ($MTXO_{4.11} : MTXO_{4.12}$) завершается ремонт эл. № 4. Далее эти операции позволяют присоединить к этим элементам, отремонтированный в цехе № 4 элемент № 6 с помощью операции $SBORKA_8$. Наконец, операциями ($MTXO_{4.13} : MTXO_{4.15}$) завершается ремонт блока 2. После этого осуществляется с помощью операции $SBORKA_1$ присоединение блока № 2 к основе (т. е. к блоку № 1). С помощью операции $RAZBOR_2$ от блока № 2 отделяется эл. № 3, который поступает, затем в цех № 3. Здесь зарождается цепь ремонтов элемента № 3 с помощью операций ($MTXO_{3.1} : MTXO_{3.18}$). После завершения операции $MTXO_{1.9}$ ремонта детали $D_{1.9}$ от блока № 1 с помощью операции $RAZBOR_3$ от основы изделия отделяется деталь $D_{10.1}$, которая направляется в цех № 12 для ее ремонта, образуя таким образом ветвь ремонта № 10. Далее от основы изделия Д (блок № 1) с помощью операции $RAZBOR_4$ отделяется блок № 5, содержащий группу деталей ($D_{51} + D_{57}$), который направляется в цех № 5, образуя при этом ветвь ремонта № 15. После этого осуществляется ремонт двух деталей основы изделия ($D_{1.11} + D_{1.12}$) с помощью операций $MTXO_{1.11} : MTXO_{1.12}$. За ними следует операция $SBORKA_1$, обеспечивающая подсоединение детали $D_{10.1}$, отремонтированной в цехе № 10, к основе детали $D_{1.13}$ и $D_{1.14}$ изделия. С помощью двух операций ($MTXO_{1.13} : MTXO_{1.14}$) ремонтируются соответственно детали основы (блока № 1).

The diagram illustrates a complex industrial process flow, likely for chemical manufacturing. It features a series of process units (PM) and their interconnections, organized into several distinct sections separated by dashed lines, each labeled with a plant name (ЦЕХ №).

Key Components and Flow:

- Raw Materials/Inputs:** The process begins with 'ПРИЕМНИК' (Receiver) and 'ЦЕХ ГОТОВОЙ ПРОДУКЦИИ' (Finished Product Plant).
- Process Units (PM):** Numerous units are shown, including:
 - PM₁ through PM₅₁ (various numbers).
 - SBORKA₁ through SBORKA₉ (Assembly/Disassembly units).
 - РАЗБОР₁ through РАЗБОР₉ (Disassembly units).
 - SBORKA₁ through SBORKA₉ (Assembly units).
- Plant Sections (ЦЕХ №):**
 - ЦЕХ №1:** Includes units like PM₁, RAZBOR₁, PM₃, RAZBOR₂, PM₅, RAZBOR₃, PM₇, RAZBOR₄, PM₉, SBORKA₃, PM₁₁, SBORKA₄, PM₁₂, PM₁₃, RAZBOR₅, PM₁₄, PM₁₅, RAZBOR₆, PM₁₆, PM₁₇, SBORKA₅, PM₁₈, PM₁₉, RAZBOR₇, PM₂₀, PM₂₁, SBORKA₅, PM₂₂, PM₂₃, PM₂₄, SBORKA₂, PM₂₅, SBORKA₇, PM₂₆, PM₂₇, PM₂₈, SBORKA₁, PM₂₉, PM₃₀.
 - ЦЕХ №2:** Includes units like PM₄₂, PM₄₆, PM₄₇, PM₄₈.
 - ЦЕХ №3:** Includes units like PM₃₁, RAZBOR₈, PM₃₂, PM₃₃, RAZBOR₉, PM₃₄, PM₃₅, SBORKA₉, PM₃₆, PM₃₇, SBORKA₈, PM₃₈, PM₃₉, PM₄₀, PM₄₁.
 - ЦЕХ №4:** Includes units like PM₄₃, PM₄₅.
 - ЦЕХ №5:** Includes unit PM₅₀.
 - ЦЕХ №6:** Includes unit PM₄₃.
 - ЦЕХ №7:** Includes unit PM₅₂.
 - ЦЕХ №8:** Includes unit PM₅₃.
 - ЦЕХ №9:** Includes unit PM₅₁.
 - ЦЕХ №10:** Includes unit PM₄₉.
- Flow Direction:** Arrows indicate the sequence of operations, generally moving from left to right, with some feedback loops and cross-connections between sections.

42

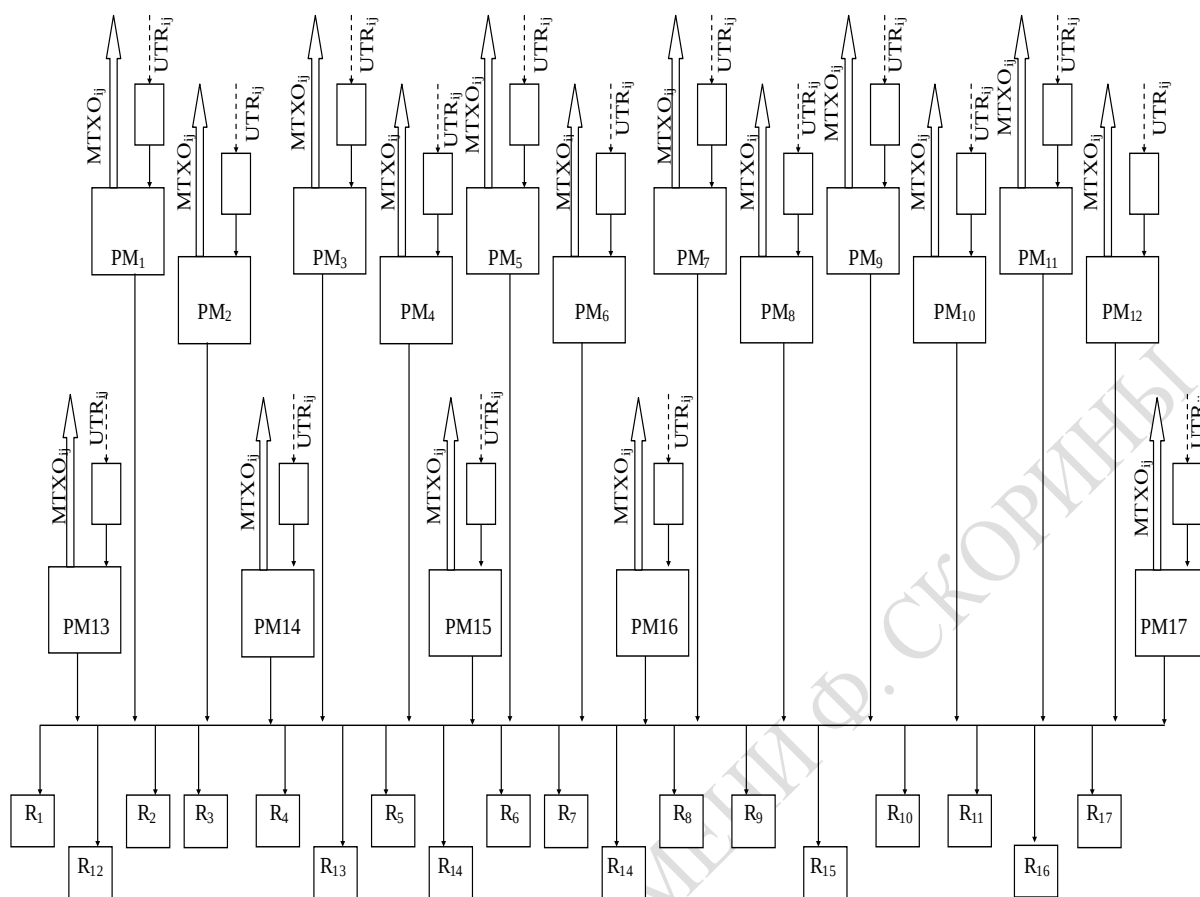


Рисунок 2.7 – Блок-схема реализации ТП СРП на рабочих местах

После этого осуществляются две операции $MTXO_{1.17} : MTXO_{1.18}$, соответственно ремонта деталей $D_{1.17}$ и $D_{1.18}$. Следующим шагом реализации технологической цепочки ремонта деталей является операция $RAZBOR_6$, обеспечивающая отделение от основы элемента № 9 и направление его в цех № 9 для последующего ремонта.

По окончании ремонта детали $D_{9.1}$ в цехе № 9 она поступает на место подсоединения ее к основе изделия с помощью операции $SBORKA_6$. Затем следуют три операции ремонта деталей основы ($MTXO_{1.19}...MTXO_{1.21}$). С помощью операции $RAZBOR_7$ от основы изделия отделяется деталь $D_{7.1}$, которая направляется для ремонта в цех № 7 с помощью операции $MTXO_{7.1}$. Далее над основой изделия (блок № 1) осуществляется ремонт детали $D_{1.22}$ с помощью операции $MTXO_{1.12}$. Используя операцию $SBORKA_5$, к основе изделия подсоединяют деталь $D_{8.1}$, отремонтированную в цехе № 8, что позволяет с помощью операции $MTXO_{1.23}$ осуществить ремонт детали $D_{1.23}$. К этому моменту реализации технологии ремонта изделия Д появляются условия для подсоединения к его основе элемента № 2, отремонтированного в цехе № 2. Это подсоединение осуществляется

с помощью операции $SBORKA_2$. Далее к основе изделия с помощью операции $SBORKA_7$ подсоединяется элемент № 7, отремонтированный в цехе № 7.

На следующем шаге технологии СРО осуществляется ремонт деталей $D_{1.25}$ и $D_{1.26}$ основы изделия с помощью операций $MTXO_{1.25}$ и $MTXO_{1.26}$. Таким образом, основа изделия D уже подготовлена для последней операции $SBORKA_1$, в ходе которой к основе подсоединяется блок № 2, отремонтированный в цехе № 4. Теперь уже все четыре элемента блока 2 (эл. № 2, эл. № 3, эл. № 6 и эл. № 4) отремонтированы соответственно в цехах 2, 3, 6 и 4. Завершается технология ремонта изделия D ремонтом № 8 деталей основы изделия ($D_{1.27} - D_{1.34}$) с помощью соответствующих микротехнологических операций ($MTXO_{1.27} \dots MTXO_{1.34}$).

Технолог предприятия составляет *блок-схему технологии* разборочных, ремонтных и сборочных операций согласно изложенной выше динамике выполнения операций в различных цехах предприятия. На рисунок 2.6 приведена блок-схема организации всех перечисленных операций согласно принятой технологии сборочно-разборочного производства изделия D , имеющего вложенную структуру компоновки (деталей в элементы, элементов в блоки и комбинация блоков и деталей в изделие сложной структуры).

Имея в своем распространении схему структуры изделия D , показанную на рисунок 2.6, технолог дополняет ее информацией о связи компонентов сборочно-разборочного производства ремонта изделия D . Пример дополнений к блок-схеме для нашего случая приведен на рисунок 2.7. Пунктирными линиями показан состав $\{MTXO_{ij}\}$, которые реализуются в каждом цехе предприятия. Далее технолог составляют списки операций $\{MTXO_{ij}\}$, $\{RAZBOB_k\}$ и $\{SBORKA_k\}$, выполняемых на рабочих местах (PM_k) в каждом цехе предприятия.

В качестве *исходной информации* для моделирования в распоряжении исследователя должна быть следующая информация:

- список выполняемых на PM_k операций (таблица 2.1);
- матрица вероятностей неисправности деталей в изделии D $MN = \|P_{ij}\|$; где i – номер ветви ремонта (№ цеха в данном случае); j -порядковый номер $MTXO_{ij}$ в i -й ветви ремонта (пример такой матрицы приведен в таблице 2.2);
- таблица запросов ресурсов предприятия, необходимых для выполнения $MTXO_{ij}$ при ремонте изделия D (пример таблицы приведен в таблице 2.3);
- таблица значений параметров функций распределения времени выполнения $MTXO_{ij}$ при осмотре детали на рабочем месте PM_k до

ее ремонта (пример таблицы приведен в таблице 2.4);

– таблица значений параметров функций распределения времени выполнения $MTXO_{ij}$ при ремонте детали на PM_k в случае неравности детали (пример таблицы приведен в таблице 2.5).

Таблица 2.1 – Списки выполняемых на РМ операций осмотра и ремонта деталей изделия Д в каждом цехе предприятия

№№ цехов	РМ	Тип операций	Список операций	№№ цехов	РМ	Тип операций	Список операций
1	1	OR	MTXO 1.1÷MTXO 1.6	1	27	SB	SBORKA 3
1	2	RZ	RAZBOR 1	1	28	OR	MTXO 1.29
1	3	OR	MTXO 1.7÷MTXO 1.8	1	29	SB	SBORKA 2
1	4	RZ	RAZBOR 2	1	30	OR	MTXO 1.30÷MTXO 1.34
1	5	OR	MTXO 1.9	1	31	SB	SBORKA 1
1	6	RZ	RAZBOR 3	3	32	OR	MTXO 3.1÷MTXO 3.5
1	7	OR	MTXO 1.10	4	33	OR	MTXO 4.1÷MTXO 4.6
1	8	RZ	RAZBOR 4	4	34	RZ	RAZBOR 8
1	9	OR	MTXO 1.11÷MTXO 1.12	4	35	OR	MTXO 4.7÷MTXO 4.10
1	10	SB	SBORKA 3	4	36	OR	MTXO 4.11÷MTXO 4.12
1	11	OR	MTXO 1.3÷MTXO 1.4	4	37	RZ	RAZBOR 9
1	12	SB	SBORKA 4	4	38	OR	MTXO 4.13÷MTXO 4.19
1	13	OR	MTXO 1.5÷MTXO 1.6	4	39	SB	SBORKA 8
1	14	RZ	RAZBOR 5	6	40	OR	MTXO 6.1÷MTXO 6.17
1	15	OR	MTXO 1.7	4	41	SB	SBORKA 9
1	16	RZ	RAZBOR 6	4	42	OR	MTXO 4.14
1	17	OR	MTXO 1.18	4	43	OR	MTXO 4.15
1	18	SB	SBORKA 6	2	44	OR	MTXO 2.1÷MTXO 2.5
1	19	OR	MTXO 1.19	2	45	OR	MTXO 2.6÷MTXO 2.9
1	20	RZ	RAZBOR 7	2	46	OR	MTXO 2.10÷MTXO 2.12
1	21	OR	MTXO 1.20	2	47	OR	MTXO 2.13÷MTXO 2.16
1	22	SB	SBORKA 5	10	48	OR	MTXO 10.1
1	23	OR	MTXO 1.21	5	49	OR	MTXO 5.1÷MTXO 2.16
1	24	OR	MTXO 1.22÷MTXO 1.24	8	50	OR	MTXO 8.1
1	25	SB	SBORKA 7	9	51	OR	MTXO 9.1
1	26	OR	MTXO 1.26÷MTXO 1.28	7	52	OR	MTXO 7.1

Поскольку некоторые $MTXO_{ij}$ используют общий ресурс ограниченного (Y_R), то необходимо иметь таблицу, в которой указаны значения параметров функций распределения требуемых размеров этого общего ресурса для выполнения $MTXO_{ij}$ на PM_k . Пример – таблица 2.6. Отметим, что общие ресурсы (ROB) используются только внутри цеха. Между цехами предприятия ресурсы не передаются, поскольку имеет место специализация ремонтных операций по PM_k , а значит, цехи предприятия также специализированы по выполняемым

в них ремонтным операциям. Очевидно, что существует жесткая конкуренция рабочих мест за ресурсы цеха, поскольку количество индивидуальных ресурсов (R_{im}) в цехе ограничено. Всего в каждом цехе расходуется до 5 типов индивидуальных ресурсов.

Таблица 2.2 – Матрица вероятностей неисправностей детали в изделии D

$i \backslash j$	1	2	3	4	5	6	7	8	9	10
1	0,2	0,2	0,2	0,2	0,3	0,3	0,7	0,5	0,8	0,5
2	0,3	0,5	0,2	0,1	0,3	0,2	-	-	-	-
3	0,2	0,5	0,7	0,2	0,4	0,1	-	-	-	-
4	0,5	0,5	0,5	0,2	0,8	0,8	-	-	-	-
5	0,2	0,8	0,7	0,5	0,2	0,8	-	-	-	-
6	0,2	0,8	0,7	0,5	0,8	0,9	-	-	-	-
7	0,8	0,5	0,7	0,8	0,2	0,5	-	-	-	-
8	0,2	0,5	0,5	0,5	-	0,4	-	-	-	-
9	0,2	0,3	0,5	0,5	-	0,3	-	-	-	-
10	0,5	0,5	0,2	0,9	-	0,2	-	-	-	-
11	0,3	0,5	0,2	0,1	-	0,2	-	-	-	-
12	0,2	0,4	0,2	0,1	-	0,2	-	-	-	-
13	0,2	0,4	0,3	0,2	-	0,3	-	-	-	-
14	0,2	0,3	0,5	0,2	-	0,2	-	-	-	-
15	0,3	0,3	0,5	0,3	-	0,5	-	-	-	-
16	0,3	0,5	-	-	-	0,5	-	-	-	-
17÷19	0,5	-	-	-	-	-	-	-	-	-
20	0,2	-	-	-	-	-	-	-	-	-
21÷23	0,8	-	-	-	-	-	-	-	-	-
24÷26	0,5	-	-	-	-	-	-	-	-	-
27÷28	0,2	-	-	-	-	-	-	-	-	-
29	0,5	-	-	-	-	-	-	-	-	-
30÷32	0,2	-	-	-	-	-	-	-	-	-
33	0,3	-	-	-	-	-	-	-	-	-
34	0,4	-	-	-	-	-	-	-	-	-

Итак, на реализацию технологии ремонтных работ в каждом цехе имеются: общий ресурс ограниченного объема (V_{0s}), s – номер цеха; пять типов ресурсов индивидуального пользования (RES_{rs}), отличающихся друг от друга номером (r) и количеством индивидуальных ресурсов (n_{rs}), которые могут использоваться рабочими местами PM_k . Ресурсы (индивидуальные и общие) выделяются PM_k на время выполнения на них $MTXO_{ij}$ и после окончания операции PM_k захваченные ресурсы возвращаются системе (предприятию) полностью.

Таблица 2.3 – Таблица запросов ресурсов предприятия операциями МТХО_{ij}

№ Цеха	РМк	№ МТХО _{ij}	Индивидуальные Ресурсы(г)					Используй- ов. ресурса	№ Цеха	РМк	№ МТХО _{ij}	Индивидуальные ресурсы(г)					Используй- ов. ресурса
			1	2	3	4	5					1	2	3	4	5	
1	1	1.1	1	-	-	1	1	-	2	46	2.12	-	-	-	-	-	+
1	1	1.2	1	-	-	-	-	+	2	47	2.13	-	-	-	-	-	+
1	1	1.3	1	-	-	1	1	+	2	47	2.14	-	-	-	-	-	+
1	1	1.4	-	-	1	-	-	-	2	47	2.15	-	-	-	-	-	+
1	1	1.5	-	1	-	-	-	+	2	47	2.16	-	-	-	-	-	+
1	1	1.6	-	-	1	-	-	+	3	32	3.1	-	-	1	-	-	+
1	3	1.7	-	1	-	-	-	-	3	32	3.2	-	-	1	-	-	+
1	3	1.8	-	-	-	-	1	+	3	32	3.3	-	-	-	-	-	-
1	5	1.9	-	-	-	-	1	-	3	32	3.4	-	-	1	-	-	-
1	7	1.10	-	-	-	-	1	+	3	32	3.5	-	-	-	-	1	-
1	9	1.11	-	-	2	-	-	-	3	32	3.6	-	-	-	-	1	-
1	9	1.12	-	-	-	-	1	-	3	32	3.7	-	-	-	-	1	-
1	11	1.13	-	-	-	-	1	-	3	32	3.8	-	-	-	-	1	-
1	11	1.14	-	-	-	-	1	-	3	32	3.9	-	-	-	-	1	-
1	13	1.15	-	-	2	-	-	+	3	32	3.10	-	-	-	-	1	-
1	13	1.16	-	1	-	-	1	+	3	32	3.11	-	-	-	-	1	-
1	15	1.17	-	1	-	-	1	-	3	32	3.12	-	-	-	-	1	-
1	17	1.18	-	1	1	-	-	+	3	32	3.13	-	-	-	-	1	-
1	19	1.19	-	-	1	-	1	-	3	32	3.14	-	-	1	-	-	+
1	21	1.20	-	-	1	-	-	+	3	32	3.15	-	-	1	-	-	+
1	23	1.21	-	1	1	-	-	+	4	33	4.1	-	-	-	-	1	-
1	24	1.22	-	1	-	-	-	+	4	33	4.2	-	1	-	-	-	+
1	24	1.23	-	1	-	-	-	+	4	33	4.3	-	1	-	-	-	+
1	26	1.24	-	1	-	-	1	+	4	33	4.4	-	-	-	-	1	-
1	26	1.25	-	1	1	-	1	-	4	33	4.5	-	-	-	-	1	-
1	26	1.26	-	-	-	-	1	+	4	33	4.6	-	1	-	-	-	-
1	26	1.27	-	-	1	-	1	-	4	35	4.7	-	-	-	-	1	-
1	26	1.28	-	1	-	-	-	-	4	35	4.8	-	-	-	-	2	-
1	28	1.29	-	-	-	-	1	+	4	35	4.9	-	-	-	-	2	-
1	30	1.30	-	-	-	-	1	+	4	35	4.10	-	-	-	-	2	-
1	30	1.31	-	-	-	-	1	-	4	36	4.11	-	-	-	-	2	-
1	30	1.32	-	-	-	1	1	-	4	36	4.12	-	-	-	-	2	-
1	30	1.33	1	-	-	-	1	-	4	38	4.13	-	-	-	-	2	-
1	30	1.34	1	-	-	-	1	-	4	42	4.14	-	1	-	-	-	+
2	44	2.1	-	-	-	-	-	+	4	43	4.15	-	-	-	-	1	-
2	44	2.2	1	-	-	-	-	-	5	49	5.1	-	-	2	-	-	+
2	44	2.3	-	-	1	-	-	+	5	49	5.2	-	-	2	-	-	+
2	44	2.4	-	-	-	-	-	+	5	49	5.3	-	-	1	-	-	+
2	44	2.5	-	-	-	-	-	+	5	49	5.4	-	-	1	-	-	-
2	45	2.6	-	-	-	-	-	+	5	49	5.5	-	-	1	-	-	+
2	45	2.7	-	-	-	-	-	+	5	49	5.6	-	1	-	-	-	+
2	45	2.8	-	-	-	-	-	+	6	49	6.1	-	-	-	-	-	+
2	45	2.9	-	-	-	-	-	+	7	52	7.1	-	-	-	-	-	+
2	46	2.10	-	-	-	-	-	+	8	50	8.1	-	-	-	-	-	+
2	46	2.11	-	-	-	-	-	+	9	51	9.1	-	-	-	-	-	+
									10	48	10.1	-	-	-	-	-	+

Таблица 2.4 – Значение параметров функции распределения времени выполнения МТХО_{ij} при осмотре детали на РМк F1k(t) до ремонта деталей

1-ая группа МТХО _{ij}				2-ая группа МТХО _{ij}				3-ая группа МТХО _{ij}			
Индексы МТХО _{ij}	Тип распр.	Мат. ожид.	Дисперсия	Индексы МТХО _{ij}	Тип распр.	Мат. ожид.	Дисперсия	Индексы МТХО _{ij}	Тип распр.	Мат. ожид.	Дисперсия
1.1	RAND	30,0	1,0	2.3	NORM	16,0	1,0	4.8	EXPO	25,0	1,0
1.2	RAND	15,0	1,0	2.4	NORM	15,0	1,0	4.9	EXPO	10,0	1,0
1.3	RAND	30,0	1,0	2.5	NORM	15,0	1,0	4.10	EXPO	10,0	1,0
1.4	RAND	9,0	1,0	2.6	NORM	12,0	1,0	4.11	EXPO	20,0	1,0
1.5	RAND	15,0	1,0	2.7	NORM	10,0	1,0	4.12	EXPO	10,0	1,0
1.6	RAND	17,0	1,0	2.8	NORM	6,0	1,0	4.13	EXPO	15,0	1,0
1.7	RAND	12,0	1,0	2.9	NORM	20,0	1,0	4.14	EXPO	20,0	1,0
1.8	RAND	10,0	1,0	2.10	NORM	6,0	1,0	4.15	EXPO	10,0	1,0
1.9	RAND	21,0	1,0	2.11	NORM	15,0	1,0	5.1	RAND	10,0	1,0
1.10	RAND	31,0	1,0	2.12	NORM	8,0	1,0	5.2	RAND	8,0	1,0
1.11	RAND	12,0	1,0	2.13	NORM	10,0	1,0	5.3	RAND	20,0	1,0
1.12	RAND	73,0	1,0	2.14	NORM	6,0	1,0	5.4	RAND	25,0	1,0
1.13	RAND	43,0	1,0	2.15	NORM	9,0	1,0	5.5	RAND	50,0	1,0
1.14	RAND	12,0	1,0	2.16	NORM	3,0	1,0	5.6	RAND	45,0	1,0
1.15	RAND	48,0	1,0	3.1	NORM	5,0	1,0	5.7	RAND	20,0	1,0
1.16	RAND	140,0	1,0	3.2	NORM	7,0	1,0	6.1	RAND	10,0	1,0
1.17	RAND	63,0	1,0	3.3	NORM	15,0	1,0	6.2	RAND	8,0	1,0
1.18	RAND	140,0	1,0	3.4	NORM	6,0	1,0	6.3	RAND	10,0	1,0
1.19	RAND	140,0	1,0	3.5	NORM	15,0	1,0	6.4	RAND	20,0	1,0
1.20	RAND	90,0	1,0	3.6	NORM	13,0	1,0	6.5	RAND	15,0	1,0
1.21	RAND	11,0	1,0	3.7	NORM	10,0	1,0	6.6	RAND	10,0	1,0
1.22	RAND	110,0	1,0	3.8	NORM	15,0	1,0	6.7	RAND	12,0	1,0
1.23	RAND	64,0	1,0	3.9	NORM	40,0	1,0	6.8	RAND	15,0	1,0
1.24	RAND	79,0	1,0	3.10	NORM	20,0	1,0	6.9	RAND	10,0	1,0
1.25	RAND	49,0	1,0	3.11	NORM	15,0	1,0	6.10	RAND	15,0	1,0
1.26	RAND	39,0	1,0	3.12	NORM	10,0	1,0	6.11	RAND	20,0	1,0
1.27	RAND	21,0	1,0	3.13	NORM	8,0	1,0	6.12	RAND	20,0	1,0
1.28	RAND	36,0	1,0	3.14	NORM	5,0	1,0	6.13	RAND	15,0	1,0
1.29	RAND	13,0	1,0	3.15	NORM	5,0	1,0	6.14	RAND	10,0	1,0
1.30	RAND	12,0	1,0	4.1	EXPO	15,0	1,0	6.15	RAND	8,0	1,0
1.31	RAND	26,0	1,0	4.2	EXPO	5,0	1,0	6.16	RAND	10,0	1,0
1.32	RAND	18,0	1,0	4.3	EXPO	25,0	1,0	6.17	RAND	10,0	1,0
1.33	RAND	12,0	1,0	4.4	EXPO	10,0	1,0	7.1	NORM	240,0	1,0
1.34	RAND	15,0	1,0	4.5	EXPO	20,0	1,0	8.1	NORM	300,0	1,0
2.1	NORM	5,0	1,0	4.6	EXPO	10,0	1,0	9.1	NORM	50,0	1,0
2.2	NORM	8,0	1,0	4.7	EXPO	20,0	1,0	10.1	NORM	120,0	1,0

Таблица 2.5 – Значение параметра функции распределения времени выполнения МТХО_{ij} при ремонте детали на РМ_kF_{2k}(τ) (в случае наличия неисправности детали)

1-ая группа МТХО _{ij}				2-ая группа МТХО _{ij}				3-ая группа МТХО _{ij}			
Индексы МТХО _{ij}	Тип распр.	Мат. ожид.	Дисперсия	Индексы МТХО _{ij}	Тип распр.	Мат. ожид.	Дисперсия	Индексы МТХО _{ij}	Тип распр.	Мат. ожид.	Дисперсия
1.1	NORM	30,0	1,0	2.3	REND	16,0	1,0	4.8	REND	25,0	1,0
1.2	NORM	15,0	1,0	2.4	REND	15,0	1,0	4.9	REND	10,0	1,0
1.3	NORM	30,0	1,0	2.5	REND	15,0	1,0	4.10	REND	10,0	1,0
1.4	NORM	9,0	1,0	2.6	REND	12,0	1,0	4.11	REND	20,0	1,0
1.5	NORM	15,0	1,0	2.7	REND	10,0	1,0	4.12	REND	10,0	1,0
1.6	NORM	17,0	1,0	2.8	REND	6,0	1,0	4.13	REND	15,0	1,0
1.7	NORM	12,0	1,0	2.9	REND	20,0	1,0	4.14	REND	20,0	1,0
1.8	NORM	100,0	9,0	2.10	REND	6,0	1,0	4.15	REND	10,0	1,0
1.9	NORM	21,0	1,0	2.11	REND	15,0	1,0	5.1	REND	10,0	1,0
1.10	NORM	30,0	1,0	2.12	REND	8,0	1,0	5.2	NORM	8,0	1,0
1.11	NORM	12,0	1,0	2.13	REND	10,0	1,0	5.3	NORM	20,0	1,0
1.12	NORM	70,0	9,0	2.14	REND	6,0	1,0	5.4	NORM	25,0	1,0
1.13	NORM	40,0	9,0	2.15	REND	9,0	1,0	5.5	NORM	50,0	1,0
1.14	NORM	12,0	1,0	2.16	REND	3,0	1,0	5.6	NORM	45,0	1,0
1.15	NORM	48,0	1,0	3.1	EXPO	5,0	1,0	5.7	NORM	20,0	1,0
1.16	NORM	140,0	9,0	3.2	EXPO	7,0	1,0	6.1	REND	10,0	1,0
1.17	NORM	60,0	1,0	3.3	EXPO	15,0	1,0	6.2	REND	8,0	1,0
1.18	NORM	140,0	9,0	3.4	EXPO	6,0	1,0	6.3	REND	10,0	1,0
1.19	NORM	140,0	9,0	3.5	EXPO	15,0	1,0	6.4	REND	20,0	1,0
1.20	NORM	90,0	9,0	3.6	EXPO	13,0	1,0	6.5	REND	15,0	1,0
1.21	NORM	12,0	1,0	3.7	EXPO	10,0	1,0	6.6	REND	10,0	1,0
1.22	NORM	100,0	9,0	3.8	EXPO	15,0	1,0	6.7	REND	12,0	1,0
1.23	NORM	64,0	9,0	3.9	EXPO	40,0	1,0	6.8	REND	15,0	1,0
1.24	NORM	79,0	9,0	3.10	EXPO	20,0	1,0	6.9	REND	10,0	1,0
1.25	NORM	49,0	9,0	3.11	EXPO	15,0	1,0	6.10	REND	15,0	1,0
1.26	NORM	39,0	1,0	3.12	EXPO	10,0	1,0	6.11	REND	20,0	1,0
1.27	NORM	21,0	9,0	3.13	EXPO	8,0	1,0	6.12	REND	20,0	1,0
1.28	NORM	36,0	9,0	3.14	EXPO	5,0	1,0	6.13	REND	15,0	1,0
1.29	NORM	13,0	1,0	3.15	EXPO	5,0	1,0	6.14	REND	10,0	1,0
1.30	NORM	12,0	1,0	4.1	REND	15,0	1,0	6.15	REND	8,0	1,0
1.31	NORM	26,0	1,0	4.2	REND	5,0	1,0	6.16	REND	10,0	1,0
1.32	NORM	18,0	1,0	4.3	REND	25,0	1,0	6.17	REND	10,0	1,0
1.33	NORM	12,0	1,0	4.4	REND	10,0	1,0	7.1	CONS	100,0	0,0
1.34	NORM	15,0	1,0	4.5	REND	20,0	1,0	8.1	CONS	300,0	0
2.01	REND	5,0	1,0	4.6	REND	10,0	1,0	9.1	CONS	50,0	0
2.02	REND	8,0	1,0	4.7	REND	20,0	1,0	10.1	CONS	100,0	0

Завершается задание *исходной информации* технологического процесса сборочно-разборочных операций ТП СРО приведением результатов мониторинга реального ТП СРО. Они представляют собой: количество индивидуальных ресурсов каждого типа (n_{rs}), имеющихся в каждом цехе предприятия, и начальный объем общих ресурсов цеха (V_{os}). Общие ресурсы используются частично многими PM_k и также по окончании выполнения $MTXO_{ij}$ возвращаются этим самым выполнения тех $MTXO_{ij}$, которые на PM_k ожидают освобождения ресурсов с целью последующего их захвата на время выполнения операций. Отметим, что операции разборки и сборки компонентов изделия используют общие ресурсы цеха предприятия и также обеспечены постоянным составом исполнителей, имеющих необходимую квалификацию (Пример задания исходной информации приведен в таблице 2.7).

Таблица 2.6 – Значение параметров функций распределения $F3(VR)$ размеров общего ресурса (VR) цеха, необходимого для выполнения $MTXO_{ij}$ на PM_k

№ Цеха	PM_k	Номера $MTXO_{ij}$	Тип распр.	Мат. ожид.	Дисперсия	№ Цеха	PM_k	Номера $MTXO_{ij}$	Тип распр.	Мат. ожид.	Дисперсия
1	1	1.2	RAND	10	3	2	46	2.10	NORM	8	1
1	1	1.3	NORM	12	4	2	46	2.11	NORM	7	2
1	1	1.5	NORM	9	3	2	46	2.12	NORM	8	3
1	1	1.6	RAND	10	12	2	47	2.13	NORM	6	2
1	3	1.8	NORM	6	2	2	47	2.14	NORM	7	1
1	7	1.10	NORM	8	3	2	47	2.15	NORM	8	2
1	13	1.15	NORM	6	2	2	47	2.16	NORM	7	1
1	13	1.16	RAND	12	4	3	32	3.1	NORM	9	2
1	15	1.18	RAND	5	10	3	32	3.2	NORM	7	1
1	19	1.20	NORM	7	2	3	32	3.14	NORM	8	3
1	21	1.21	RAND	2	9	3	32	3.15	NORM	7	3
1	24	1.22	RAND	7	10	4	35	4.2	RAND	2	9
1	24	1.23	RAND	8	12	4	35	4.3	RAND	3	12
1	24	1.24	RAND	6	10	4	36	4.14	RAND	10	17
1	26	1.26	NORM	7	2	5	49	5.1	RAND	3	8
1	29	1.29	NORM	8	2	5	49	5.2	RAND	3	6
1	30	1.30	NORM	6	2	5	49	5.3	NORM	7	1
2	44	2.1	NORM	10	3	5	49	5.5	NORM	7	2
2	44	2.3	NORM	7	2	5	49	5.6	NORM	8	3
2	44	2.4	NORM	6	2	6	40	6.2	RAND	3	7
2	44	2.5	NORM	10	3	6	40	6.3	NORM	8	3
2	45	2.6	NORM	5	1	6	40	6.6	NORM	7	2
2	45	2.7	NORM	10	4	6	40	6.9	NORM	4	1
2	45	2.8	RAND	6	10	6	40	6.10	RAND	3	7
2	45	2.9	RAND	3	8	6	40	6.12	RAND	4	9

Итак первая часть содержательного описания содержит в себе:

- рисунки 2.2, 2.3 и текст описания динамики взаимодействия трех типов операций, которые ранее были приведены;
- таблицы 2.1–2.7, в которых указываются запросы $MTXO_{ij}$ ресурсов предприятия, характеристики структуры операций;
- схемы разборки изделий, осмотра состояния компонентов изделия; матрицу вероятностей неисправности деталей изделия, таблицу времени выполнения операций ремонта изделий (и используемых при этом ресурсов предприятия).

Как видим, все эти таблицы представляют собой *описание воздействия внешней среды* на ТП СРП реального предприятия. Поскольку времена выполнения операций $MTXO_{ij}$ и размеры используемого ресурса являются случайными величинами, то в таблицах 2.4–2.6 указываются: тип функции распределения, математическое ожидание и дисперсия случайной величины, которые определяются либо в ходе мониторинга реального ТП СРО, либо задаются на основе априорной информации.

Вторую часть содержательного описания СРО содержит список параметров СРО статистик имитации и откликов имитации.

Параметром СРО является интенсивность поступления деталей для ремонта на предприятии (λ).

Статистиками имитации будут коэффициенты использования $MTXO_{ij}$ (η_{ij}); значения среднего времени ожидания деталей для ремонта с помощью операций $MTXO_{ij}$.

Из-за конкуренции за ресурсы цеха предприятия появляются статистики:

- средние времена ожидания в очереди ($\bar{t}_{ожij}$);
- средние длины очередей деталей в очередях к $MTXO_{ij}$ ($\bar{l}_{очij}$);
- коэффициенты использования $MTXO_{ij}$ рабочих мест (η_k), средние времена ожиданий $MTXO_{ij}$ освобождение PM_k ($\bar{t}_{ожk}$); средние длины очередей $MTXO_{ij}$ к PM_k ($\bar{l}_{очk}$);
- коэффициенты использования PM_k r -го типа ресурсов предприятия (η_{rs}): среднее время ожидания $MTXO_{ij}$ PM_k этих ресурсов ($\bar{t}_{ожrs}$); средние длины этих очередей ($\bar{l}_{очrs}$).

Откликами имитации будут: пропускная способность СРО v_h при имеющемся составе ресурсов предприятия (таблица 2.7) при интенсивности поступления изделий D (λ_h) на предприятия для их ремонта.

На имитационное моделирование СРО поставлены следующие задачи проектного моделирования:

1. Оценка пропускной способности технологического процесса СРО ремонтных работ при различной интенсивности поступления изделий на предприятия, т. е. нужно найти зависимость $v_h = \varphi(\lambda_h)$ при:

- заданной структуре изделия D (рисунок 2.4);
- составе ресурсов на предприятии (таблица 2.7);
- характеристиках ремонта деталей (таблицы 2.2–2.6).

2. Определение «узких мест» в существующей технологии СРО на предприятии при имеющемся составе ресурсов предприятия.

Как видим, с усложнением объекта имитации существенно возрастают состав и уровень детализации содержательного описания объекта имитации. Два примера, приведенные выше, показывают, насколько усложняется проблема имитации при увеличении уровня детализации СС.

Таблица 2.7 – Начальное количество индивидуальных и общих ресурсов, имеющихся в цехах предприятия (измеренные при мониторинге предприятия)

№№ res (ч)	Цех № 1			Цех № 2			Цех № 3			Цех № 4			Цех № 5			Цех № 6			Размер общего ресурса в других цехах
	Тип res	n_1	V_1	Тип res	n_2	V_2	Тип res	n_3	V_3	Тип res	n_4	V_4	Тип res	n_5	V_5	Тип res	n_6	V_6	
1	Rin	2	-	Rin	2	-	Rin	2	-	Rin	2	-	Rin	1	-	Rin	1	-	$V_{\text{общ}}=20$ в цехе №7
2	Rin	4	-	Rin	4	-	Rin	2	-	Rin	2	-	Rin	1	-	Rin	1	-	$V_{\text{общ}}=20$ в цехе №8
3	Rin	8	-	Rin	6	-	Rin	2	-	Rin	2	-	Rin	1	-	Rin	2	-	$V_{\text{общ}}=20$ в цехе №9
4	Rin	2	-	Rin	2	-	Rin	2	-	Rin	2	-	Rin	2	-	Rin	2	-	$V_{\text{общ}}=20$ в цехе №10
5	Rin	10	-	Rin	6	-	Rin	4	-	Rin	4	-	Rin	2	-	Rin	2	-	-
общ	Rob		50	Rob	-	30	Rob	-	30	Rob	-	30	Rob	-	20	Rob	-	20	-

Составить содержательное описание технологии ремонта изделия A, представленной на рисунке 2.8

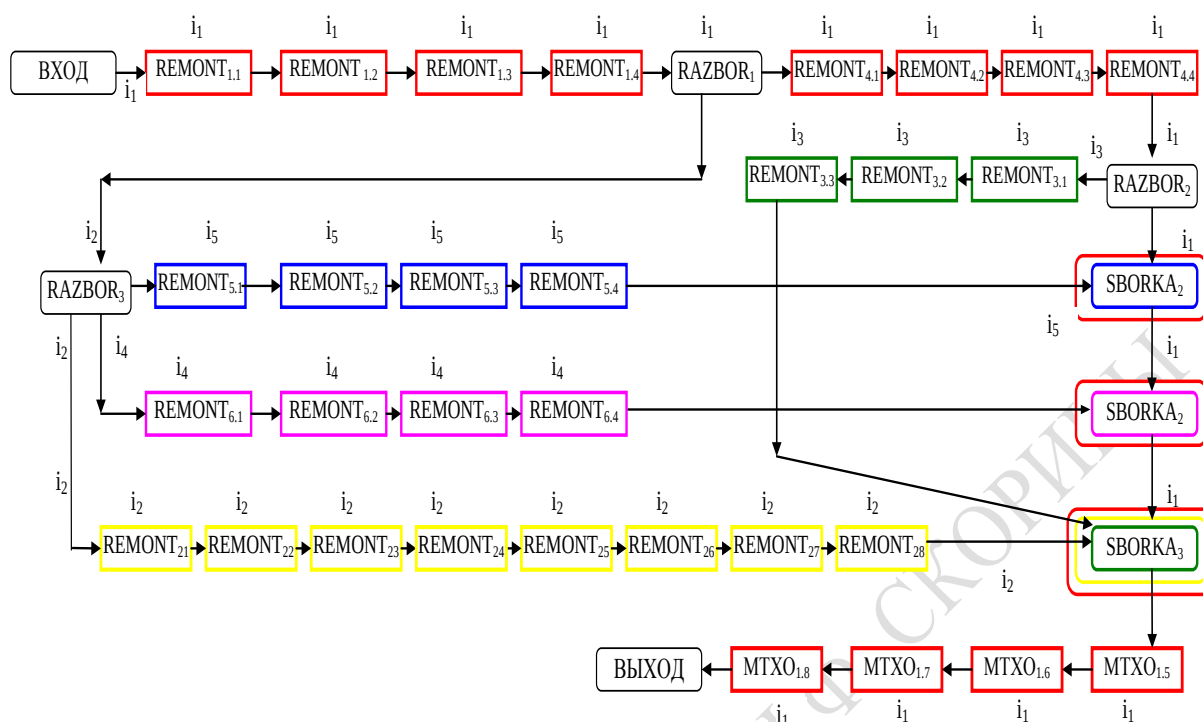


Рисунок 2.8 – Технология ремонта изделия А

2.4 Содержательное описание вероятностных технологических процессов производства

При решении задач управления, проектирования и исследования динамики функционирования таких сложных систем, какими являются технологические процессы производства (ТПП), используются математические модели. Зачастую исследования ТПП ориентированы на то, что эти процессы являются детерминированными по своей структуре, а параметры компонент ТПП – постоянными величинами. Создан специальный математический аппарат исследования детерминированных ТПП, использующий для анализа динамики развития ТПП во времени модели сетевого планирования [26] в виде сетевых графиков (СГР). С помощью СГР можно решить ряд задач исследования ТПП. Однако на практике времена выполнения являются случайными величинами. Кроме того, сами зависимости $MTXO_{ij}$ друг от друга могут быть случайными. Поэтому известные методики расчета примеров СГР [12] не позволяют адекватным образом отобразить динамику развития ТПП. В таких случаях для изучения вероятностных технологических процессов (ВТПП) используются имитационные модели (ИМ). С их помощью исследователи изучают не только времена

реализации ВТПП, но и расход во времени ресурсов предприятия, степень загрузки оборудования, изменение стоимостных показателей реализации ВТПП, расход материалов и комплектующих для реализации множества $\{МТХО_{ij}\}$. Существенной трудностью исследования ВТПП является либо параллельно последовательный характер выполнения $МТХО_{ij}$, либо строго последовательный характер пребывания ВТПП в различных состояниях. Каждый из этих случаев исследования ВТПП предлагает использование разных типов ИМ. Ниже рассматривается два примера ВТПП, на которых иллюстрируется использование разных типов вероятностных имитационных моделей, построенных на основе двух различных математических аппаратов исследования.

Описание последовательных технологических процессов производства.

Из-за недостаточной изученности ВТПП на начальных этапах проектного моделирования зачастую бывает трудно выделить $МТХО_{ij}$, и определить порядок следования их друг за другом. В силу сложной многофункциональной зависимости между ними трудно определить порядок следования друг за другом $МТХО_{ij}$. В таких ситуациях интегральные состояния ВТПП легче представить как результат взаимодействия $МТХО_{ij}$, поэтому удастся представить ВТПП как последовательность строго изменяющихся друг за другом состояний ВТПП. Такие ВТПП обычно называются последовательными, а сами переходы определяются матрицей вероятностей перехода ВТПП из i -го в j -е состояние $M_n = \|P_{ij}\|$.

Длительность нахождения ВТПП в каждом из состояний также представляет собой структурную величину τ_{ij} , функция вероятностей значений которой $F_{ij}(\tau)$ обычно известна исследователю. Динамика смены состояний ВТПП полностью определяется матрицей M_n (см. таблицу 2.8). На рисунке 2.9 приведен пример графа переходов последовательного ВТПП из i -го состояния в j -е состояние ($i, j = \overline{1,10}$). Такими состояниями ВТПП являются:

$j = 1$ – нормальные выполнения ВТПП без использования ресурсов предприятия и без использования оборудования ($SOST_1$);

$j = 2$ – нормальное выполнение ВТПП с использованием всех ресурсов предприятия и без использования оборудования ($SOST_2$);

$j = 3$ – нормальное выполнения ВТПП без использования ресурсов, но с использованием оборудования ($SOST_3$);

$j = 4$ – нормальное выполнение ВТПП с использованием оборудования и с использованием ресурсов ($SOST_4$);

$j = 5$ – отказ оборудования, без использования ресурсов (SOST₅);

$j = 6$ – отказ оборудования, но с использованием ресурсов (SOST₆);

$j = 7$ – одиночное резервирование оборудования (SOST₇);

$j = 8$ – групповое резервирование оборудования (SOST₈);

$j = 9$ – состояние профилактики оборудования (SOST₉);

$j = 10$ – ликвидация сложной аварии (SOST₁₀).

Таблица 2.8 – Параметры расхода ресурсов последовательного ВТПП₁

Вектор вероятностей начального состояния i_n	i_n	1	2	3	4	5	6	7
	P_n	0,60	0,10	0,05	0,05	0,05	0,05	0,10
Функция распределения цикла смен состояний $F_0(v)$	P_k	0,10	0,20	0,30	0,40	0,50	0,60	1,00
	v_k	70	80	90	100	110	120	130
Вектор вероятностей конечного состояния i_k	i_k	1	2	3	4	5	6	7
	P_k	0,10	0,05	0,05	0,10	0,10	0,10	0,50
Матрица вероятностей перехода из состояния i в состояние j $\ p_{ij} \ $	j	1	2	3	4	5	6	7
	i							
	1	0,05	0,30	0,20	0,10	0,05	0,20	0,10
	2	0,30	0,05	0,10	0,05	0,05	0,05	0,40
	3	0,20	0,10	0,05	0,10	0,15	0,20	0,20
	4	0,05	0,30	0,30	0,05	0,10	0,10	0,10
	5	0,05	0,05	0,05	0,05	0,05	0,70	0,05
	6	0,10	0,10	0,10	0,20	0,25	0,05	0,20
	7	0,20	0,10	0,05	0,05	0,50	0,05	0,05

Как видно из рисунка 2.9, граф переходов состоит из десяти узлов и десяти состояний ВТПП. Возможен любой переход ВТПП из состояния в состояние. Поскольку функции узлов (UZFL_j) этого графа одинаковы, то операцию перехода ВТПП из состояния i в состояние j можно представить одним и тем же процессом PR. UZEL. Каждое из состояний ВТПП нужно моделировать соответствующим процессом PRSOST_j. На рисунке 2.10 приведена схема взаимодействия процессов ВТПП с помощью процесса PR.UZEL. На схеме показано использование процессами PR. SOST_j оборудования (OBORUD) и ресурсов (RESURS). В таблице 2.9 приведены функции условных распределений времени и стоимости нахождения последовательного ВТПП, в различных состояниях.

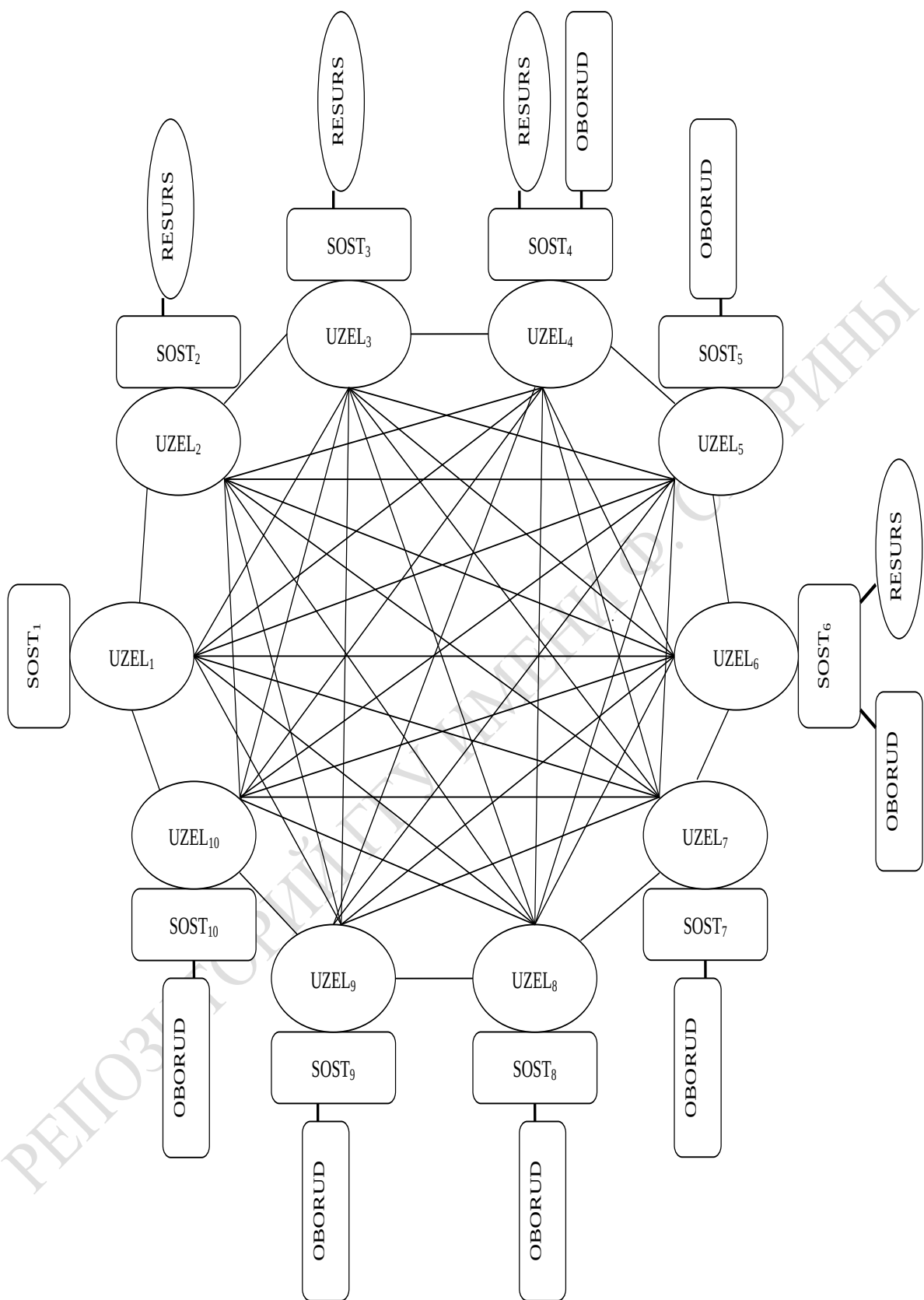


Рисунок 2.9 – Граф переходов последовательного ВТПП₁
из состояния i в состояние j

Таблица 2.9 – Функции условных распределений времени и стоимости нахождения ВТПП в различных состояниях

$i \backslash j$	Функции распределения времени нахождения в состоянии						
	1	2	3	4	5	6	7
1	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(20,3)$	$N(5,3)$	$N(6,3)$	$N(10,3)$
2	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(15,3)$	$N(10,3)$	$N(8,3)$	$N(6,3)$
3	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$
4	$N(5,3)$	$N(5,3)$	$N(5,3)$	$N(10,3)$	$N(10,3)$	$N(15,3)$	$N(12,3)$
5	$N(6,3)$	$N(7,3)$	$N(10,3)$	$N(10,3)$	$N(7,3)$	$N(6,3)$	$N(10,3)$
6	$N(6,3)$	$N(5,3)$	$N(10,3)$	$N(3,3)$	$N(4,3)$	$N(10,3)$	$N(8,3)$
7	$N(6,3)$	$N(5,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(6,3)$	$N(4,3)$
$i \backslash j$	Функция распределения стоимости выполнения						
	1	2	3	4	5	6	7
1	$N(10,3)$	$N(20,3)$	$N(16,3)$	$N(10,3)$	$N(20,3)$	$N(16,3)$	$N(10,3)$
2	$N(20,3)$	$N(10,3)$	$N(8,3)$	$N(10,3)$	$N(10,3)$	$N(6,3)$	$N(10,3)$
3	$N(16,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(8,3)$
4	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(8,3)$	$N(6,3)$	$N(4,3)$
5	$N(20,3)$	$N(10,3)$	$N(10,3)$	$N(8,3)$	$N(8,3)$	$N(12,3)$	$N(10,3)$
6	$N(16,3)$	$N(10,3)$	$N(6,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$
7	$N(10,3)$	$N(8,3)$	$N(6,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$	$N(10,3)$

Целью имитации последовательного ВТПП является решение следующих задач:

- оценка времени реализации ВТПП (T_{pp}) при заданных множествах времени нахождения ВТПП в каждом состоянии $\{\tau_{ij}\}$;
- определение коэффициентов загрузки оборудования $\{\eta_{OBoj}\}$ и использования ресурсов предприятия $\{\eta_{RESr}\}$ при заданной матрице переходов ВТПП из состояния в состояние (M_n);
- определение состава и структуры «узких мест» в ВТПП при имеющемся составе ресурсов и характеристиках надежности оборудования ВТПП.

Параметрами ВТПП являются:

- матрица вероятностей переходов ВТПП из состояния в состояние $M_n = \|P_{ij}\|$ (таблица 2.8);
- состав и размеры используемых ресурсов (n_{RESr} и V_{ORESr});
- надежность характеристики оборудования N_{OB} (функция распределения длительности безотказной работы $F_{BOj}(\tau)$ оборудования в j -м состоянии ВТПП, функция распределения длительности

интервала восстановления работоспособности оборудования $F_{BOj}(\tau)$; вероятность отказа оборудования ВТПП в j -м состоянии ВТПП (P_{OTj}); вероятность того, что отказ приведет систему в аварийное состояние, ($P_{ЛВj}$); функция распределения длительности ликвидации аварии $F_{ABj}(\tau)$ в j -м состоянии ВТПП);

– вектор вероятностей того, что ВТПП находятся в начальном состоянии равно i $\{P_{начij}\}$ (таблица 2.8);

– вектор вероятностей того, что j -е состояние ВТПП окажется завершающим $\{P_{кочj}\}$ или же функция вероятностей числа смен состояний ВТПП $F_{см}(v)$ (для тех случаев, когда вероятность завершения имитации окажется слишком маленькой, и поэтому завершение имитации может оказаться недостижимым) (таблица 2.8).

Таблица 2.10 – Фрагмент таблицы $TZRES_{ij}$ запросов $MTXO_{ij}$ ресурсов системы

№ п/п	ij	α_{ij}	R	n_r	$\pi_{зак}$	f_2	$\pi_{и}$	Имя распред.	Нач. цик. (1 пар)	Кон. цик. (2 пар)	M_x	σ_x	Обозн. парам.
25	5.12		10	1	0	0	1	nor	4	1	4	1	$F_{1ij}(\tau)$
			9	1	0	0	1	nor	8	2	8	2	$F_{2ij}(c)$
			8	1	0	0	1	nor	6	1	6	1	$F_{4ij}(KO)$
			7	1	0	0	1	nor	10	2	10	2	$F_{3ij}(mt)$
			6	3	1	1	3	1	-	-	-	-	Бриг.
			5	3	0	0	2	1	-	-	-	-	Исполн.
			4	1	1	1	1	nor	7	2	7	2	$F_{6ij}(V)$
			3	1	0	0	3	1	-	-	-	-	индресур
			2	1	1	1	1	nor	15	3	15	3	$F_{5ij}(V)$
			1	1	0	0	0	1	-	-	-	-	Индобор.
26	5.6		10	1	0	0	1	nor	12	3	12	3	$F_{1ij}(\tau)$
			9	1	0	0	1	nor	16	4	16	4	$F_{2ij}(c)$
			8	1	0	0	1	nor	4	1	4	1	$F_{4ij}(KO)$
			7	1	0	0	1	nor	8	2	8	2	$F_{3ij}(mt)$
			6	1	1	1	3	1	-	-	-	-	Бриг.
			5	2	0	0	2	1	-	-	-	-	Исполн.
			4	1	1	1	1	nor	6	2	6	2	$F_{6ij}(V)$
			3	1	0	0	2	1	-	-	-	-	индресур
			2	1	1	1	1	nor	12	3	12	3	$F_{5ij}(V)$
			1	1	0	0	0	1	-	-	-	-	Индобор.

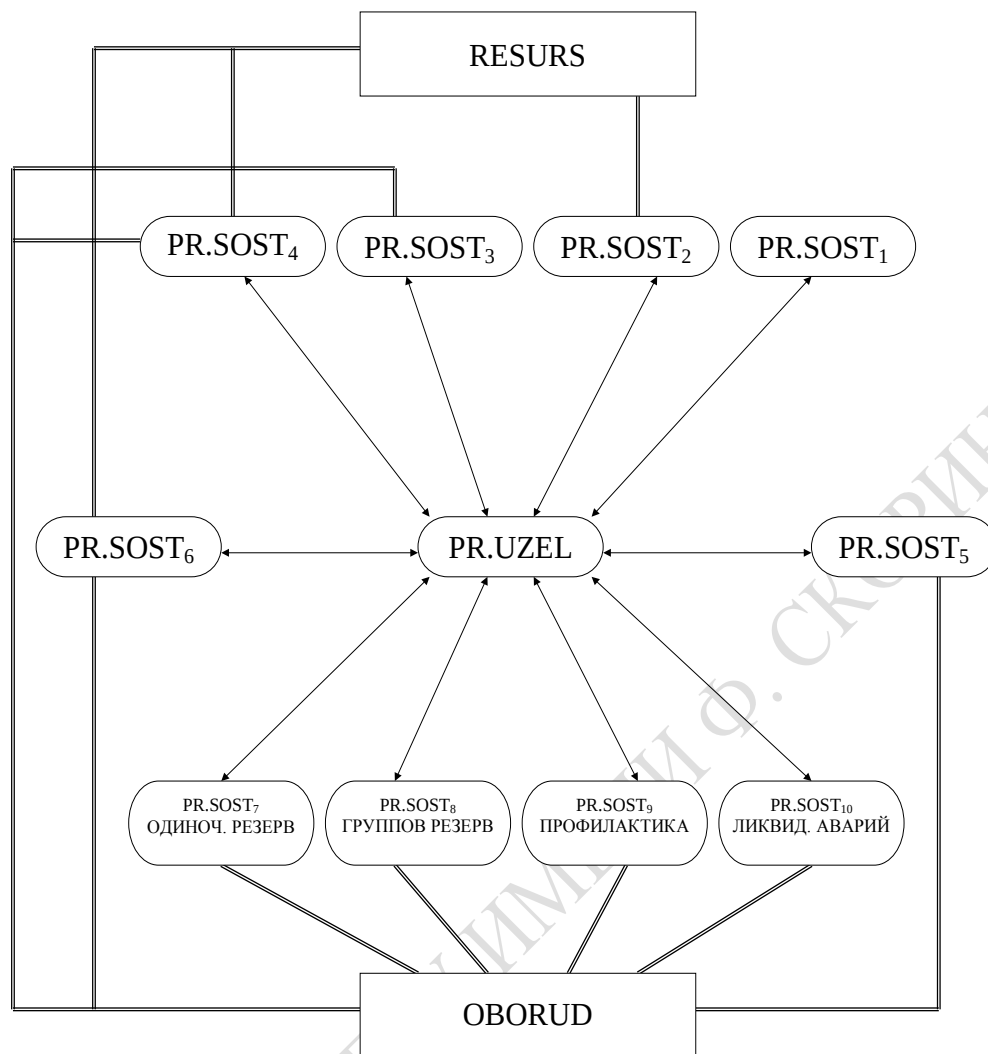


Рисунок 2.10 – Схема взаимодействия процессов PR.SOST_j в ВТПП₁ с помощью процесса PR.UZEL

Статистиками имитации, позволяющими разобраться в деталях реализации ВТПП при критичных значениях надежности функционирования оборудования, являются:

- графики использования ресурсов ВТПП в процессе имитационного эксперимента ($\eta_{\text{RES}}(t_0)$);
- временные диаграммы функционирования оборудования ВД.ОВ(t_0).

Откликами ВТПП являются:

- время реализации ВТПП (T_{pp});
- средние значения и выборочные дисперсии времени нахождения ВТПП в каждом состоянии (с учетом наличия отказов оборудования, восстановлений его работоспособности и ликвидации аварий) ($\bar{T}_{\text{SOSj}}$ S_{TSOSj}^2);

- средние значения и дисперсии коэффициентов использования оборудования в j -м состоянии ($\bar{\eta}_{OBoj}$, S_{OBoj}^2);
- средние значения и дисперсии коэффициентов использования ресурсов ВТПП в j -м состоянии ($\bar{\eta}_{RESj}$, S_{RESj}^2).

Таблица 2.11 – Таблица структуры сигналов и коммуникации агрегатов TSTRSG

АМТХО _{ij}	Адрес отправителя в режиме «прямо»					Адрес получателя в режиме «прямо»	
	Адрес получателя в режиме «инверсия»					Адрес отправителя в режиме «инверсия»	
x	i	k	L_k	l	d_k	j	r
0,1	0	2	1	1	1,0	1	1
0,2	0	1	1	1	1,0	2	1
1,2	1	1	1	1	1,0	2	2
1,3	1	2	3	1	1,0	3	1
1,4	1	2	3	2	1,0	4	1
1,5	1	2	3	3	1,0	5	1
2,9	2	1	2	1	1,0	9	1
2,10	2	1	2	2	1,0	10	1
3,10	3	1	1	1	1,0	10	2
3,8	3	2	2	1	1,0	8	1
3,11	3	2	2	2	1,0	11	1
4,11	4	1	1	1	1,0	11	2
4,7	4	2	3	1	1,0	7	1
4,12	4	2	3	2	1,0	12	1
4,6	4	2	3	3	1,0	6	1
5,12	5	1	3	1	1,0	12	2
5,13	5	1	3	2	1,0	13	3
5,6	5	1	3	3	1,0	6	2
9,14	9	1	1	1	1,0	14	1
10,14	10	1	1	1	1,0	14	2
8,14	8	1	1	1	1,0	14	3
11,14	11	1	1	1	1,0	14	4
7,13	7	1	1	1	1,0	13	1
12,13	12	1	1	1	1,0	13	2
6,13	6	1	1	1	1,0	13	4
13,14	13	1	1	1	1,0	14	5

Таблица 2.12 – Фрагмент коммутации входов событий TSOIN

Номер события i	Входы						
	n_{ex}	r	$AMTXO_{Si}$	$t_{ПРИХ}$	$Mun\ sig$	$Han\ sig$	π_{AB}
0	0	0	0	0	0	0	0
		-	-	-	-	-	-
1	1	1	0.1	-	-	-	-
		-	-	-	-	-	-
		-	-	-	-	-	-
		-	-	-	-	-	-
2	2	1	0.2	-	-	-	-
		2	1.2	-	-	-	-
3	1	1	1.3	-	-	-	-
		-	-	-	-	-	-
		-	-	-	-	-	-
4	1	1	1.4	-	-	-	-
		-	-	-	-	-	-
		-	-	-	-	-	-
		-	-	-	-	-	-
5	1	1	1.5	-	-	-	-
		-	-	-	-	-	-
		-	-	-	-	-	-
6	2	1	4.6	-	-	-	-
		2	5.6	-	-	-	-
7	1	1	4.7	-	-	-	-
8	1	1	3.8	-	-	-	-
9	1	1	2.9	-	-	-	-
10	2	1	2.10	-	-	-	-
		2	3.10	-	-	-	-

Задачей проектного моделирования является:

– нахождение типа и параметров регрессионной зависимости среднего времени реализации ВТПП от характеристик состояний, состава ресурсов и характеристик надежности оборудования $T_{pp} = \varphi(M_{II}, n_{RES}, V_{RES}, H_{OB})$;

– определение диапазона возможного изменения $T_{ПП}$ из-за вероятностного характера технологического процесса производства $\Delta \bar{T}_{ПП} = (\bar{T}_{ПП\ max} - T_{ПП\ min})$ как функции от надежностных характеристик оборудования $\Delta \bar{T}_{ПП} = \Psi(H_{OB})$.

– определение граничных значений состава ресурсов (n_{RES}^* и V_{ORES}^*) и допустимых вероятностей отказа оборудования (P_{OTj}^* и P_{ABj}^*), при которых для данной структуры имеет смысл организация ВТПП.

Описание дискретного параллельно-последовательного вероятного технологического процесса производства.

Объектом исследования является сложный дискретный параллельно-последовательный вероятностный технологический процесс производства (ВТПП) с непредсказуемыми заранее временами выполнения (τ_{ij}) составляющих – это процесс микротехнологических операций (МТХО_{ij}), где i – номера предыдущего и j – номера последующего события в ВТПП. При этом связи МТХО_{ij} друг с другом при реализации ВТПП таковы, что одни МТХО_{ij} могут начаться только после выполнения предшествующих МТХО_{ij}. Это означает, что *первой особенностью* ВТПП является *графовая структура* технологии его реализации. *Второй особенностью* ВТПП является *вероятностный характер запросов* ресурсов предприятия на выполнение МТХО_{ij}. При этом ресурсы предоставляются в распоряжение МТХО_{ij} только на время их выполнения и затем возвращаются ВТПП. Кроме времён выполнения МТХО_{ij} (τ_{ij}), которые являются случайными величинами, для выполнения МТХО_{ij} требуются следующие ресурсы: исполнители, оборудование, материалы и комплектующие изделия. *Третьей особенностью* ВТПП является наличие на предприятии группы *общих ресурсов*, совместно используемых различными группами МТХО_{ij}. При совместном использовании для МТХО_{ij} выделяется часть общего ресурса (V_{rij}) на время выполнения МТХО_{ij}, которая также возвращается обратно в распоряжение системы распределения ресурсов (SRR) ВТПП. *Четвертой особенностью* ВТПП является безвозмездный расход финансовых средств (C_{ij}), материалов (mt_{ij}) комплектующих изделий (КО_{ij}).

Причем величина остатка общего ресурса номера r (V_{OSr}) во времени реализации ВТПП зависит от операционной обстановки в ВТПП и от параллельно-последовательного характера выполнения МТХО_{ij}. *Пятой особенностью* ВТПП является *вероятностный характер выполнения* той или иной группы {МТХО_{ij}}, что обуславливает непредсказуемое их развитие во времени. Это означает, что одна группа {МТХО_{ij}} выполняется в строго детерминированном порядке друг за другом, а другая группа {МТХО_{ij}} выполняется вероятностным образом. Технология выполнения МТХО_{ij} состоит в реализации следующих действий:

- заказ ресурсов предприятия на выполнение $\{MTXO_{ij}\}$;
- их закрепление за $MTXO_{ij}$, на выполнение операций длительностью τ_{ij} ;
- возврат ресурсов по окончании $MTXO_{ij}$;
- выбор следующей группы $MTXO_{ij}$, которые выполняются по завершении $MTXO_{ij}$.

Динамику параллельно-последовательного взаимодействия $\{MTXO_{ij}\}$ друг с другом при выполнении дискретного ВТПП можно проследить на примере ВТПП, приведенном в работе [26], с помощью рисунка 2.11. В реализации ВТПП принимают участие 27 микротехнологических операций $MTXO_{ij}$ (i - и j -номера событий соответственно на входе и выходе $MTXO_{ij}$).

Всего свершается 15 событий. Входы и выходы событий пронумерованы. События на ВСГР представлен в виде кружков, разделенных на четыре сектора. В верхнем секторе кружков показаны номера событий ($i = \overline{0,15}$). В нижних секторах указаны резервы свершения событий R_i . Ранние и поздние сроки свершения (t_{pi} и t_{ni}) расположены соответственно в левых и правых секторах кружков, отображающих i -е события, а расчет резервов ведется по формуле $R_i = t_{ni} - t_{pi}$.

У начального события SOB_0 выходы 1 и 2 действительные, одинаковые, что обеспечивает одновременный запуск $MTXO_{01}$ и $MTXO_{02}$. У SOB_1 выход 1 действительный, а выход 2 является выходом четвертого типа с тремя разветвлениями ($l = \overline{1,3}$). У SOB_2 выход 1 второго типа (вероятностный, на два разветвления с вероятностями ($P_{2.9} + P_{2.10} = 1$)). У SOB_3 выход 1 действительный, а выход 2 (четвертого типа) имеет два разветвления. У SOB_4 выход 1 действительный, а у выхода 2 (третьего типа) имеет место три разветвления ($l = \overline{1,3}$), разыгрываемые по вектору вероятностей ($P_{4.7} + P_{4.12} + P_{4.6} = 1$).

У SOB_5 выход 1 третьего типа с тремя разветвлениями ($l = \overline{1,3}$) по вектору вероятностей ($P_{5.7} + P_{5.12} + P_{5.6} = 1$). У $SOB_6, SOB_7, SOB_8, SOB_9, SOB_{13}$ имеется один действительный выход. По одному входу имеют события $SOB_1, SOB_3, SOB_4, SOB_5, SOB_9, SOB_8, SOB_7$. По два входа имеется у событий $SOB_2, SOB_{10}, SOB_{11}, SOB_{12}, SOB_6$. У SOB_{13} и SOB_{14} имеется соответственно 4 и 5 выходов.

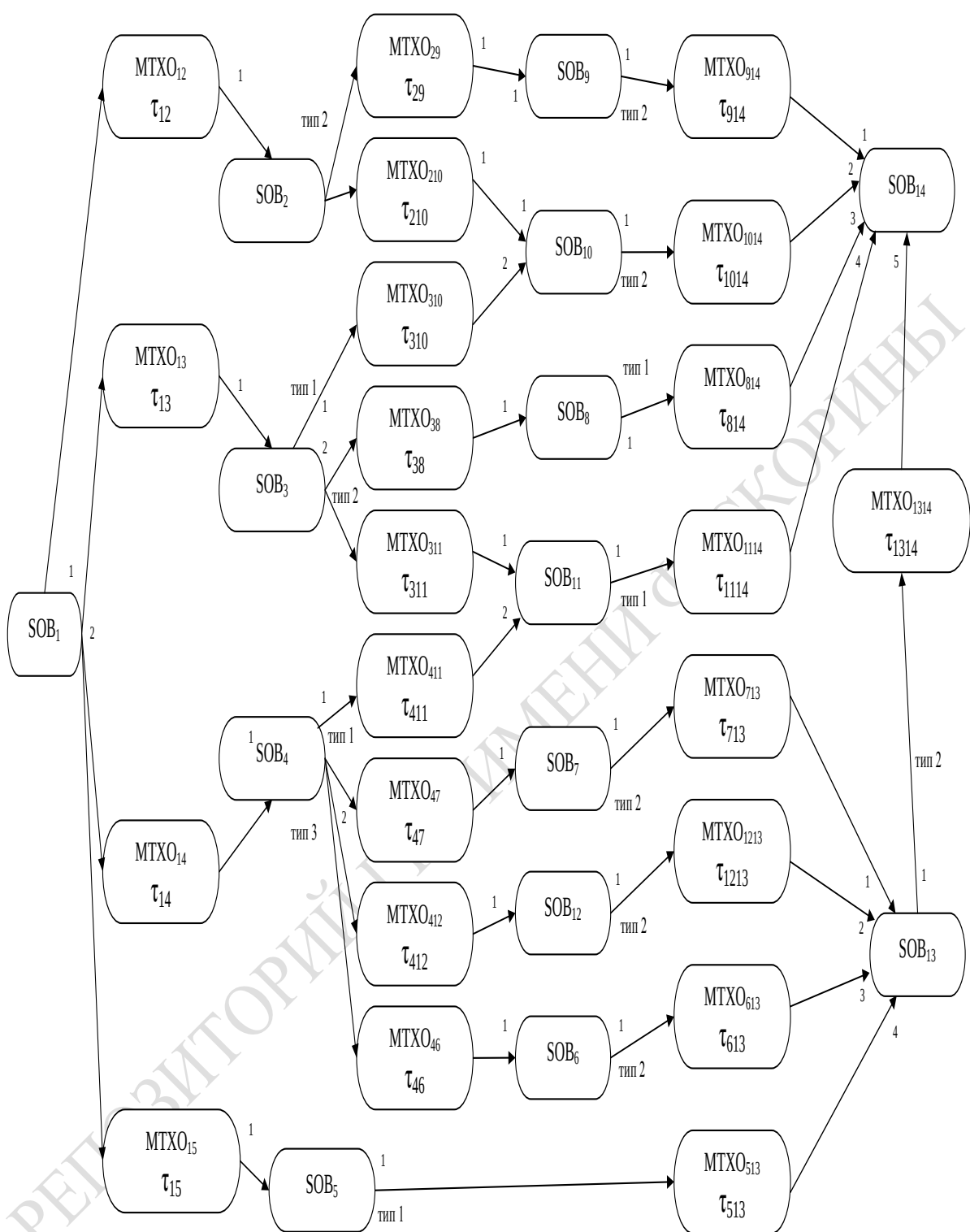


Рисунок 2.11 – Структура связей MTXO_{ij} и событий SOB_i в ТПП₃

Рассматривается ВТПП, при выполнении которого имеют место несколько вариантов реализации технологических операций в ходе его развития. Во-первых, при выполнении MTXO₀₁ происходит выход

значений вектора управляющих параметров U за допустимые пределы, поэтому одновременно формируются действительные сигналы на всех трех разветвлениях выхода 2 у SOB_1 . Во-вторых, при выполнении $MTXO_{15}$ также происходит выход значений вектора управляющих параметров U за допустимые пределы. Поэтому и с выхода 2 SOB_5 одновременно формируются действительные сигналы по обоим разветвлениям (поступающим соответственно на SOB_8 SOB_{11}). В-третьих, при выполнении $MTXO_{14}$ нет отказов оборудования. Поэтому на выходе 2 у SOB_4 по вектору вероятностей формируются один действительный сигнал (по $l = 1$ на $MTXO_{4,7}$) и два фиктивных сигнала (по $l = 2$ на $MTXO_{4,12}$ и по $l = 3$ на $MTXO_{4,6}$).

Таблица 2.13 – Фрагмент таблицы выходов событий TSTRUS

Номер собы- тия <i>i</i>	Выходы											
	$AMTO_{ij}$	$n_{вых}$	k	$тип$ $вых.$	n_{PB}	l	P_{il}	Матрица $\ \gamma_{ij}\ $				$t_{вых0}$
								$L_{стр}$	γ_{i1}	γ_{i2}	γ_{i3}	
0	0	2	1	1	1	1	0	0	0	0	0	0
	-	-	2	1	1	1	0	0	0	0	0	0
1	1.2	2	1	1	1	1	0	0	0	0	0	0
	1.3	-	2	4	3	1	0	3	1	1	1	-
	1.4	-	-	-	-	2	0	-	0	0	0	-
	1.5	-	-	-	-	3	0	-	0	0	0	-
2	2.9	1	1	2	2	1	0,7	-	-	-	-	-
	2.10	-	-	-	-	2	0,3	-	-	-	-	-
3	3.10	2	1	1	1	1	0	0	0	0	0	0
	3.8	2	2	4	2	1	0	3	1	1	1	0
	3.11	-	-	-	-	2	0	-	0	0	0	0
4	4.11	2	1	1	1	1	1	0	0	0	0	0
	4.7	-	2	3	3	1	0,5	0	0	0	0	0
	4.12	-	-	-	-	2	0,3	0	0	0	0	0
	4.6	-	-	-	-	3	0,2	0	0	0	0	0
5	5.12	1	1	3	3	1	0,2	0	0	0	0	0
	5.13	-	-	-	-	2	0,5	0	0	0	0	0
	5.16	-	-	-	-	3	0,3	0	0	0	0	0
6	6.13	1	1	1	1	1	0	0	0	0	0	0
	-	-	-	-	-	-	-	-	-	-	-	-

Таблица 2.14 – Фрагмент таблицы коммуникации МТХО_{ij} в тестовом примере

№ событий слева	Кол. вых. m_{1h}	Ном. вых. k_{1h}	Тип вых. тип ₁	Кол. разв. L_{1h}	Ном. разв. l_{1h}	Слева АМТХО _{hi}	Ном. события i	Кол. входов n_{1i}	Ном. входа Γ_{1i}	Адрес информ. β_i	Тип сигн. тип	Напр. сигн. нап.	Кол. выходов m_{2h}	Ном. вых. k_i	Тип вых. тип ₂	Ткол. разв. L_{2i}	Ном. разв. l_{2i}	Справа АМТХО _{ij}	Ном. события j	Кол. входов n_{2i}	Ном. входа Γ_{2j}
0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	1	1	1	0.2	2	2	1
Н а ч.	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	1	1	0.1	1	1	1
1	2	2	1	1	1	0.1	1	1	1	-	-	-	2	1	1	1	1	1.2	2	2	2
	-	-	-	-	-	-	-	-	-	-	-	-	-	2	4	3	1	1.3	3	1	1
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1.4	4	1	1
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	1.5	5	1	1
2	2	1	1	1	1	0.2	2	2	1	-	-	-	2	1	2	2	1	2.9	9	1	1
	-	-	-	-	-	1.2	2	2	2	-	-	-	2	1	2	2	1	2.10	10	2	1
3	1	1	1	1	1	1.3	3	1	1	-	-	-	2	1	1	1	1	3.10	10	2	2
	-	-	-	-	-	-	-	-	-	-	-	-	-	2	4	2	1	3.8	8	1	1
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	3.11	11	2	1
4	1	1	1	1	1	1.4	4	1	1	-	-	-	2	1	1	1	1	4.11	11	2	2
	-	-	-	-	-	-	-	-	-	-	-	-	-	2	3	3	1	4.7	7	1	1
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	4.12	12	2	1
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	4.6	6	2	1
5	1	1	1	1	1	1.5	5	1	1	-	-	-	1	1	3	3	1	5.12	12	2	2
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	5.13	13	4	3
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	5.6	6	2	2
6	2	1	3	3	1	4.6	6	2	1	-	-	-	1	1	1	1	1	6.13	13	4	4
	-	-	-	-	2	5.6	6	2	2	-	-	-	-	-	-	-	-	-	-	-	-
7	1	1	1	1	1	4.7	7	1	1	-	-	-	-	1	1	1	1	7.13	13	4	1
8	1	1	1	1	1	3.8	8	1	1	-	-	-	1	1	1	1	1	8.14	14	5	3
9	1	1	1	1	1	2.9	9	1	1	-	-	-	1	1	1	1	1	9.14	14	5	1
10	2	1	1	1	1	3.10	10	2	2	-	-	-	1	1	1	1	1	10.14	14	5	2
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	2	1	4	2	2	3.11	11	2	1	-	-	-	1	1	1	1	1	11.14	14	5	4
	2	2	3	3	1	4.11	11	2	2	-	-	-	-	-	-	-	-	-	-	-	-
12	1	1	3	3	2	4.12	12	2	1	-	-	-	1	1	1	1	1	12.13	13	5	2
	1	1	3	3	1	5.12	12	2	2	-	-	-	-	-	-	-	-	-	-	-	-
13	1	1	1	1	1	7.13	13	4	1	-	-	-	1	1	1	1	1	13.14	14	5	5
	1	1	1	1	1	12.13	13	-	2	-	-	-	-	-	-	-	-	-	-	-	-
	3	1	3	3	2	5.13	13	-	3	-	-	-	-	-	-	-	-	-	-	-	-
	1	1	1	1	1	6.13	13	-	4	-	-	-	-	-	-	-	-	-	-	-	-
14	5	1	1	1	1	9.14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	2	1	1	2	10.14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	3	1	1	3	8.14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	4	1	1	4	11.14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
К О н.	-	5	1	1	5	13.14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

В четвертых, при выполнении МТХО₁₅ осуществляется выход значений вектора управляющих параметров U за допустимые пределы, и поэтому на выходе l у SOB₅ формируются действительные сигналы по всем трем разветвлениям (МТХО_{5.12}, МТХО_{5.13}, МТХО_{5.6}). На выходе 2 у SOB₂ по $l = 1$ формируется действительный сигнал (МТХО_{2.9}), а при $l = 2$ формируется фиктивный сигнал. Итак, наличие фактов выхода значений параметров вектора U за допустимые пределы при выполнении МТХО₀₁, МТХО₁₅, МТХО₁₃ и наличие аварии при выполнении МТХО₁₄ предопределили структуру первого вероятностного сетевого графика ВСГР₁. Далее рассматриваются 5 вариантов значений параметров МТХО_{ij} для структуры ВСГР₁.

Запросы ресурсов ВТПП для выполнения МТХО_{ij} отображаются в таблице запросов TZRE_{ij}, фрагмент которой приведен в таблице 2.10. Каждая МТХО_{ij} для своего выполнения требует предоставления ресурсов десяти типов $R = \overline{1,10}$. (см. таблицу 2.10). Все распределения вероятностных запросов ресурсов нормального вида $N(M, S)$. Для выполнения любой МТХО_{ij} необходимы исследователи и бригада исполнителей. В таблице 2.11 приведен фрагмент структуры коммутации МТХО_{ij} и SOB_j. Фрагменты коммутации входов и выходов событий приведены в таблицах 2.12 и 2.14. Вход 2 у SOB₁ и выход 2-у SOB₃ соединены согласно матрицы коммутации событий с МТХО_{ij} ($\|\gamma_{ij}\|$ и $\|\beta_{ij}\|$). У выходов 2 событий SOB₄ и SOB₃ задано по три разветвления коммутаций с помощью векторов 2.14 $\{P_{ij}\}$. Фрагмент таблицы коммутаций МТХО_{ij} и SOB_j приведен в таблице 2.14, который определяет в совокупности структуру ВТПП.

Для представления надежностных характеристик оборудования и активизации процедур ликвидации аварий на оборудовании составлена таблица 2.15, где указаны функции распределения времен безотказной работы k -го устройства оборудования ($\tau_{БOK}$), времен восстановления работоспособности (τ_{VOK}), времен ликвидации аварий (τ_{ABK}), стоимости ликвидации аварий (C_{ABK}), вероятности появления аварии (P_{ABK}) при отказах на k -х устройствах. Начальные размеры ресурсов, выделяемых в распоряжение $\{МТХО_{ij}\}$ на время их реализации, даны в таблице 2.17; фрагмент структуры запросов ВТПП множеством $\{МТХО_{ij}\}$ приведен в таблице 2.16.

Таблица 2.15 – Таблица надежностных характеристик оборудования и параметров процедур восстановления поставарийной ситуации

№ п/ п	R	Тип параметра распределения		№ Вар. обозн.	$1(\bar{x} - S/2)$		$2(\bar{x} - S/4)$		$3(\bar{x})$		$4(\bar{x} + S/4)$		$5(\bar{x} + S/2)$	
		по τ	по c		τ	c	τ	c	τ	c	τ	c	τ	c
1	1	$F_{81}(\tau)$	-	БОТ	8	-	10	-	12	-	14	-	16	-
2	1	$F_{101}(\tau)$	$F_{102}(c)$	ВОС	2	2	4	4	6	6	8	8	10	10
3	1	$F_{121}(\tau)$	$F_{122}(c)$	АВР	6	4	8	6	10	8	12	10	14	12
4	2	$F_{71}(\tau)$	-	БОТ	12	-	14	-	16	-	18	-	20	-
5	2	$F_{91}(\tau)$	$F_{92}(c)$	ВОС	4	4	6	8	18	12	10	16	12	20
6	2	$F_{111}(\tau)$	$F_{112}(c)$	АВР	16	8	18	12	20	16	22	20	24	24
7	2	$F_{141}(\tau)$	$F_{142}(c)$	$APROC_k$	8	10	10	12	12	14	14	16	16	18
8	1	$F_{131}(\tau)$	$F_{132}(c)$	$APROC_k$	2	16	4	18	6	20	8	22	10	24

Таблица 2.16 – Таблица начальных размеров ресурсов ВТПП₂

№ п / п	Название ресурса	№ Вар. обоз.	$1(\bar{x} - S/2)$		$2(\bar{x} - S/4)$		$3(\bar{x})$		$4(\bar{x} + S/4)$		$5(\bar{x} + S/2)$	
			Кол.	V_{OH}	Кол.	V_{OH}	Кол.	V_{OH}	Кол.	V_{OH}	Кол.	V_{OH}
8	Комплектующ.	n_8	2	-	2	-	2	-	2	-	2	-
7	Материалы	n_7	2	-	2	-	2	-	2	-	2	-
6	Бригады	n_6	14	-	14	-	14	-	14	-	14	-
5	Исполнители	n_5	18	-	18	-	18	-	18	-	18	-
4	Общ. ресурсы	n_4	2	85	2	85	2	85	2	85	2	85
3	Инд. ресурсы	n_3	20	-	20	-	20	-	20	-	20	-
2	Общ. оборуд.	n_2	2	105	2	105	2	105	2	105	2	105
1	Инд. оборуд.	n_1	14	-	14	-	14	-	14	-	14	-

Таблица 2.17 – Фрагмент структуры запросов ресурсов ВТП₂ множеством {МТХО_{ij}}

п/п	Название ресурса	Но мер	Обоз наче ние	№ № МТХО (ij)																									
				0,1	0,2	1,2	1,3	1,4	1,5	2,9	2,10	3,10	3,8	3,11	4,11	4,7	4,12	4,6	5,12	5,13	5,6	9,14	10,14	8,14	11,14	7,13	12,13	6,13	13,14
1	Время	10	τ_{ij}	8,0	2,0	9,0	3,0	3,0	4,0	7,0	6,0	3,0	4,0	4,0	6,0	5,0	6,0	7,0	2,0	6,0	8,0	6,0	9,0	4,0	8,0	4,0	7,0	5,0	6,0
2	Стоим.	9	c_{ij}	10,0	4,0	12,0	6,0	6,0	8,0	7,0	7,0	5,0	4,0	4,0	5,0	2,0	4,0	5,0	4,0	3,0	10,0	4,0	8,0	4,0	6,0	5,0	2,0	3,0	5,0
3	Компл.	8	KO_{ij}	2,0	6,0	3,0	7,0	5,0	6,0	5,0	3,0	5,0	8,0	4,0	3,0	7,0	2,0	4,0	4,0	3,0	2,0	6,0	7,0	6,0	6,0	5,0	3,0	4,0	4,0
4	Матер 1	8	KO_{2ij}	0	4,0	4,0	0	0	8,0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2,0
5	Матер 2	7	mt_{ij}	12,0	4,0	12,0	6,0	8,0	11,0	6,0	6,0	5,0	7,0	7,0	8,0	4,0	10,0	5,0	6,0	9,0	4,0	12,0	6,0	5,0	7,0	3,0	4,0	3,0	6,0
6	Бригад	7	mt_{2ij}	0	0	2,0	0	0	6,0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Бригад.	6	n_6	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
8	Кл.общ. ресур.	5	n_5	2,0	2,0	3,0	2,0	3,0	2,0	3,0	2,0	2,0	1,0	1,0	1,0	1,0	2,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	2,0	1,0
9	Кл.общ. ресур.	4	n_4	2,0	2,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
10	Размер	4	V_{61}	4,0	4,0	6,0	4,0	5,0	3,0	4,0	5,0	4,0	8,0	8,0	8,0	9,0	7,0	5,0	3,0	6,0	3,0	8,0	6,0	6,0	6,0	7,0	3,0	5,0	2,0
11	Размер	4	V_{62}	6,0	8,0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	Колич.	3	n_3	2,0	2,0	1,0	2,0	2,0	3,0	2,0	3,0	2,0	2,0	2,8	2,0	2,0	2,0	2,0	2,0	2,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
13	Колич.	2	n_2	1,0	1,0	1,0	1,0	2,0	1,0	1,0	2,0	1,0	1,0	1,0	1,0	1,0	2,0	1,0	1,0	1,0	1,0	1,0	1,0	2,0	1,0	1,0	2,0	1,0	1,0
14	Размер ресурса	2	V_{51}	6,0	5,0	9,0	12,0	6,0	6,0	4,0	6,0	6,0	7,0	7,0	9,0	6,0	5,0	5,0	3,0	6,0	3,0	6,0	8,0	6,0	6,0	5,0	4,0	7,0	6,0
15	Размер	2	V_{52}	0	0	0	0	6,0	0	0	2,0	0	0	0	0	0	0	0	0	0	0	0	0	4,0	0	0	6,0	0	0
16	Отказ	2	T_2	0	0	0	0,5	0,3	0	0,4	0,5	0	0	0	0	0	0	0	0,2	0	0	0	0	0	0	0	0	0	0
17	Авар.	2	D_2	0	0	0	0,5	0,2	0	0,2	0	0,2	0	0	0	0	0	0	0,5	0	0	0	0	0	0	0	0	0	0
18	Колич.	1	n_1	1,0	1,0	1,0	2,0	3,0	1,0	1,0	1,0	1,0	1,0	2,0	2,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
19	Отк.	1	T_1	0	0	0	0	0	0,6	0	0	0	0,3	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Отк.	1	AB_1	0	0	0	0	0	0,2	0	0	0	0,1	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2.5 Итоги

Подведем итоги главы 2. Вначале в 2.1 формулируется методика составления содержательного описания СС. Она реализуется четырьмя действиями исследователя: выбор показателей качества СС; определение управляющих переменных ИМ СС; детализация описания режимов функционирования СС; составление ожидания воздействия на СС внешней среды.

Общая методика составления СОП в 2.2 иллюстрируется при описании ВП в ЛВС. Рассматривается пять уровней детализации ВП и РН на ЛВС: топология ЛВС, динамика обработки информации в узле ЛВС, реализации процессов набором функций ЛВС. Демонстрируется составление СОП ВП и РН в ЛВС: составление СОП ВП ЛВС первого уровня детализации для построения ИМ ТОПОЛОГИЯ ЛВС; составление СОП ВП в узле ЛВС на третьем уровне детализации для построения ИМ «ЗАДАЧИ ЛВС». При формулировке СОП исследователю необходимо выполнять следующую последовательность действий при определении: состава параметров РН и ВП; откликов и статистик имитации. Определяется состав задач проектного моделирования: оценка пропускной способности, оценка эффективности ЛВС; выбор структуры РН на ЛВС.

Состав и структура СОП поточного производства изготовления деталей и ремонта изделий сложной структуры, излагаемые в 2.3, иллюстрируют общую методику для другой предметной области исследований. Рассматривается технология составления СОП:

- составление СОП динамики обработки деталей двух типов четырьмя станками производственного участка;
- составление СОП технологии сборочно-разборочного производства ремонта изделий сложной структуры.

При составлении СОП иллюстрируются следующие действия исследователя для данных предметных областей: определение цели имитации; выделение особенностей области исследования; выделение особенностей ремонтируемых изделий; описание исходной информации для исследования; выбор состава параметров, откликов и статистик имитации СС. Далее демонстрируется составление СОП последовательного ВТПП. Формулируются цель имитации, состав параметров, статистик и откликов имитации ВТПП₁. Определяются следующие задачи имитации: нахождение типа и параметров регрессионной зависимости; определение диапазона переходного режима ($T_{пп}$), поиск граничных значений ресурсов предприятия

и характеристик надежности оборудования. Также демонстрируется составление СОП параллельно-последовательного процесса ВТПП₂, имеющего графовую структуру. Формулируются особенности ВТПП₂, перечисляются этапы технологии исследования ВТПП₂. Выделяются основные операции имитации ВТПП₂: заказ ресурсов, закрепление их за каждой МТХО_{ij}, возврат ресурсов обратно, выбор следующей группы МТХО_{ij}.

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

3 РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ОБЪЕКТА МОДЕЛИРОВАНИЯ

3.1 Методика составления концептуальной модели СС

Имея содержательное описание объекта моделирования, исследователь приступает к разработке его концептуальной модели. При построении простых имитационных моделей содержательное описание практически представляет собой концептуальную модель. Для построения более сложных ИМ появляется необходимость в выборе способа формализации объекта моделирования. Конечно, когда исследователь располагает средствами автоматизации моделирования, ориентированными на вполне определенный способ формализации, привык к ним (иногда разработал сам эти средства), то проблемы выбора у него не существует и он пытается строить ИМ только этим способом. Однако с появлением различных средств моделирования и усложнением ИМ эта проблема становится весьма актуальной. Отсюда, меняются цель и состав работ на этапе разработки концептуальной модели объекта моделирования. На основе анализа содержательного описания определяется общий замысел модели, выдвигаются основные гипотезы, фиксируются сделанные допущения. Уточняется задача моделирования, и если содержательное описание не дает необходимой информации для последующих этапов разработки ИМ, то имеет место возврат на этап составления содержательного описания. Обычно концептуальная модель сложной системы представляет собой упрощенное алгоритмическое отображение реальной системы. С учетом рекомендаций Н.П. Бусленко [3] сложная система расчленяется на конечное число частей (декомпозиция системы), сохраняя при этом связи, обеспечивающие их взаимодействие. Полученные части при необходимости вновь расчленяются до тех пор, пока не получают элементы, удобные для математического или алгоритмического описания. В результате этого сложная система представляется в виде многоуровневой конструкции взаимосвязанных элементов, объединяемых в подсистемы (подмодели) различных уровней. При этом стремятся к тому, чтобы получаемые подмодели отвечали реально существующим фрагментам системы.

В зависимости от целей моделирования используются различные виды концептуальных моделей систем. В ИМ, предназначенной для отработки вопросов оптимизации архитектуры сложной системы,

основной удельный вес в выполняемых функциях модели занимают стохастические элементы при достаточно простом алгоритме функционирования компонент системы. Основным математическим методом, используемым в ней, является метод статических испытаний. Если же при проектировании сложной системы на моделях осуществляется поиск «узких мест» в будущей системе, то основным является представление алгоритмических аспектов ее функционирования. Под «узким местом» в системе будем понимать такую ситуацию, когда скорость поступления запросов одних компонент системы на некоторый общий ресурс системы превышает скорость их удовлетворения со стороны управляющей части системы. В результате к этим ресурсам в системе выстраиваются длинные очереди заказов от компонент системы; в то же время другие ресурсы могут оставаться неиспользованными. Под ресурсом понимают в одних случаях наличие механизмов и приспособлений общего пользования, а в других случаях ресурсом является место, где выполняются некоторые типовые операции многими компонентами системы. Иногда в качестве ресурсов выступают персонал или ремонтные бригады, которые последовательно обслуживают различные компоненты сложной технической системы. По нашему мнению, основным содержанием этапа составления концептуальной модели является выбор способа формализации объекта моделирования. С этой точки зрения для всех применяемых способов формализации исследователь должен выполнить следующую последовательность работ: провести декомпозицию системы; выбрать параметры и переменные; уточнить критерии эффективности функционирования системы; аппроксимировать реальные процессы математическими выражениями; продумать вопросы задания исходной информации; учесть выдвигаемые гипотезы и предположения; установить структуру модели. Сравнивая получаемые при каждом способе формализации результаты, исследователь делает окончательный выбор. Результаты работ на этапе фиксируются в документации. Как видим, этот этап представляет собой как бы «черновую» (прикидочную) формализацию объекта моделирования. Ниже рассматриваются перечисленные виды работ исследователя при различных способах формализации. Для простоты изложения будем считать, что способы формализации эквиваленты соответствующему способу организации квазипараллелизма.

Декомпозиция сложной системы. Для выполнения этой работы можно воспользоваться рекомендациями, изложенными в [15]. Предлагая читателю ознакомиться с этими рекомендациями, рассмотрим особенности декомпозиции, связанные с различными способами имитации.

При организации квазипараллелизма с помощью *активностей* сложную систему принципиально можно представить множеством разнотипных активностей AK_{ij} и подпрограмм проверки выполнимости условий инициализации AK_{ij} . Декомпозиция системы на AK_{ij} должна быть такова, чтобы за заданные интервалы времени τ_{ij} ошибка аппроксимации ими функциональных действий $ФД_{ij}$ в системе координат «функциональные действия в модели $ФД_i$ – время функционирования компонент модели t_i » была бы незначительна.

Для *событийного* способа имитации возможна группировка $ФД_{ij}$ по видам событий C_{ij} , к которым они приводят систему. Поэтому необходимо стремиться к тому, чтобы активности AK_{ij} , аппроксимирующие эти группы $ФД_{ij}$, имели одинаковую точность аппроксимации, что означает одинаковые значения интервалов изменяя времени выполнения этих активностей τ_{ij} . Для такой декомпозиции объекта имитации на множество программ обслуживания событий C_{ij} весьма существенна слабая взаимосвязь между событиями. В этом случае свершение каждого из событий определяется своей подпрограммой проверки условий их выполнения.

Для *транзактного* способа имитации функциональные действия $ФД_{ij}$ аппроксимируются приборами массового обслуживания. Каждому типу $ФД_{ij}$ соответствует свой тип обслуживающего прибора, реализующего простейший алгоритм $АЛ_{ij}$ обслуживания транзактов. При этом наблюдается тесная связь между $АЛ_{ij}$, которую представляют в виде схемы. Существенным для декомпозиции системы при транзактном представлении ИМ является то, что все $ФД_{ij}$ аппроксимируются ограниченным стандартизованным набором приборов массового обслуживания.

При *агрегатном* способе имитации все $ФД_{ij}$ представляются универсальными математическими агрегативными схемами. В качестве определяющей характеристики декомпозиции системы агрегативными схемами также является величина ошибки аппроксимации $ФД_{ij}$ соответствующими активностями, реализуемыми этими агрегативными схемами. Существенна при этом возможность описания связей между компонентами системы K_i в ИМ с помощью сигналов (входных, управляющих, выходных).

При *процессном* способе организации квазипараллелизма декомпозиция системы определяется двумя соображениями. Во-первых, каждой компоненте K_i соответствует свой процесс. Таким способом достигается соответствие структуры ИМ и реальной системы. Во-вторых, все $ФД_{ij}$ аппроксимируются соответствующими активностями исходя из величины ошибки аппроксимации. Если ошибка аппроксимации некоторых $ФД_{ij}$ на интервале τ_{ij} велика, то этот интервал разбивается на несколько

более мелких интервалов, на каждом из которых часть ΦD_{ij} аппроксимируется своей активностью. Таким образом, компоненты модели K_i делят на последовательность активностей AK_{ij} , для которых ошибки аппроксимации ΦD_{ij} находятся в допустимом для данного исследования интервале. Важным для процессного способа имитации является то, что условия свершения событий C_{ij} индивидуальны для каждой компоненты реальной системы K_i и активности AK_{ij} тесно взаимосвязаны между собой.

При *транзактно-процессном* способе организации квазипараллелизма декомпозиция СС осуществляется в двух направлениях. С одной стороны, все аспекты влияния внешней среды на СС стараются отобразить в «теле» сложного транзакта, поступая при этом аналогично транзактному способу имитации. Структура «тела» сложного транзакта представляется в виде блок-схемы, отображающей основное использование компонентов «тела» транзакта процессами, моделирующими алгоритмы поведения СС. Как правило, при этом используется информационная «подкраска» сложного транзакта, которая сопровождается некоторой кодировкой информации для уменьшения размеров памяти сложного транзакта. По этой информации (составу и структуре кодировки структуры сложного транзакта) алгоритмы процессов СС выполняют специфические алгоритмические действия. Сами компоненты СС представляются в виде соответствующих версий процессов, структура которых аналогична той, которая имеет место при процессном способе имитации. Добавляются только те алгоритмические операции, которые связаны с распаковкой информации, заключенной в телах сложных транзактов, и выполнением соответствующих алгоритмических операций на основе анализа закодированной информации в «теле» транзакта. Поэтому, как правило, процессы при данном способе формализации имеют на порядок более сложную структуру и зачастую используют иерархическим образом организованную структуру алгоритма. Это обстоятельство позволяет расширить состав типов процессов. Некоторые процессы могут совмещать функции создания и уничтожения служебных транзактов (сложной структуры), используемых на отдельных участках функционирования СС. В данном случае условия свершения событий C_{ij} могут разделяться на несколько уровней и при этом также быть индивидуальными не только для компонентов реальной системы K_i и активностей AK_{ij} , но и тесным образом связаны со структурой сложных транзактов. Причем эти связи могут быть алгоритмическими.

При *агрегатно-процессном* способе имитации декомпозиция СС также осуществляется в двух направлениях. Все аспекты влияния

внешней среды на СС отображаются количеством и составом сигналов ее воздействия на компоненты СС. Кроме того, взаимодействие компонентов СС друг с другом также отображается с помощью сигналов. Компоненты СС отображаются соответствующими агрегатами. Количество агрегатов обычно ограничено. Они имеют универсальный алгоритм функционирования и стандартную схему взаимодействия с сигналами. По своей структуре агрегаты представляют собой универсальные процессы. Управляющая программа моделирования (УПМ) взаимодействует множеством активностей аналогично процессному способу имитации. Сами сигналы, в свою очередь, могут иметь сложную структуру (в том числе могут быть информационно «подкрашенными»). Информационная «подкраска» сигналов используется активностями алгоритмов агрегатов для выполнения соответствующих ветвей алгоритма агрегата. Как правило, агрегатно-процессный способ имитации используется для отображения специфики объекта имитации и поэтому данный способ имитации является специализированным для соответствующей предметной области исследований.

Просматривая на данном этапе трудности декомпозиции реальной системы при каждом из возможных способов имитации, исследователь ранжирует их по предпочтительности с точки зрения удобства декомпозиции объекта моделирования.

Выбор параметров и переменных. *Параметрами модели* будем считать те величины, значения которых исследователь может выбирать произвольно и изучение влияния изменения значений которых на исследуемую систему представляет цель имитации. Остальные характеристики модели могут принимать только вполне определенные значения, задаваемые перед началом имитации. Они являются переменными модели. Параметры входят в состав показателей качества моделируемой системы, выбранных на этапе составления ее содержательного описания, и являются варьируемыми характеристиками в функциях показателей качества. Второй группой характеристик, входящих в состав функций показателей качества, являются *статистики моделирования*. Они измеряются в ходе моделирования, и по окончании очередной имитации варианта системы их значения подставляются в функцию показателя качества моделируемой системы.

В общем случае описание параметра должно содержать в себе следующую информацию: определение и идентификатор параметра, указание единицы измерения, диапазона изменения, качественные характеристики (однозначный – многозначный, регулируемый – нерегулируемый), место применения в модели, источник получения значений параметра. Для каждого способа имитации исследователь

оценивает легкость определения значений квантов изменения временной координаты τ_{ij} для активностей, имитирующих соответствующие функциональные действия ΦD_{ij} . При имитации *активностями* и *процессами*, а также при событийном способе организации квазипараллелизма выделяют общие для всех алгоритмов активностей $АЛ_{ij}$ характеристики, которые называются глобальными. Поэтому подпрограммы проверки условий активизации событий $С_{ij}$ обычно используют именно эти характеристики модели. Для *агрегатного* способа имитации значения параметров и переменных модели определяют результат вычислений операторов перехода агрегатов в различные состояния, а также условия выдачи выходных сигналов другим агрегатам. В случае *транзактного* способа организации квазипараллелизма параметры модели обычно выступают в роли характеристик специальных блоков, обслуживающих транзакты. Например, это могут быть характеристики выделения ресурсов памяти транзактам в одних блоках и освобождения этих ресурсов в других блоках программы модели.

При *транзактно-процессном* способе имитации параметры имитации делятся на две группы: параметры внешней среды, определяющие интенсивности их создания и использования, а также характеристики структуры тела сложного транзакта; параметры самой СС, определяющие состав и типы процессов, их ресурсы, количество и состав и оборудования системы. При *агрегатно-процессном* способе имитации параметрами внешней среды обычно являются интенсивности поступления из вне в систему сложных сигналов и характеристики состава и структуры сложных сигналов. Параметрами самой СС являются: количественный состав агрегатов-имитаторов оборудования, состав и размеры ресурсов системы.

Прорабатывая на данном шаге вопросы параметризации объекта моделирования, исследователь определяет для себя степень удобства каждого способа имитации с точки зрения задания параметров моделирования.

Уточнение критериев эффективности функционирования системы. Исследователю приходится оценивать, при каком способе формализации выражение эффективности системы как функции от параметров и переменных системы окажется наиболее удобным и легко интерпретируемым. Оценивается удобство сбора статистики моделирования. Так, при имитации *активностями*, *процессами*, а также при *событийном* способе организации квазипараллелизма в качестве статистик моделирования могут выступать частоты их использования или удельный вес времени выполнения в общем времени имитации соответственно активностей, процессов, подпрограмм обслуживания

событий. Для *транзактного* способа имитации в качестве статистик моделирования обычно используются типовые характеристики обслуживания транзактов: средние времена жизни транзактов в системе (от момента создания в соответствующих источниках до момента их уничтожения в поглотителях); коэффициенты использования блоков транзактами (отношение времени обслуживания блоками транзактов к общему времени имитации); характеристики обслуживания очередей транзактов к блокам (средняя длина очереди, среднее время нахождения транзакта в очереди, количество транзактов, побывавших в очереди); вероятности завершения обслуживания транзактов к заданному сроку. Для *агрегатного* способа имитации к указанным ранее статистикам моделирования зачастую добавляются частоты поступления управляющих или выходных сигналов от одного агрегата к другому, которые могут иметь самую различную интерпретацию в зависимости от объекта и целей моделирования.

При *транзактно-процессном* способе имитации статистики моделирования представляют собой сочетание статистик использования транзактов сложной структуры со статистиками, отображающими динамику взаимодействия процессов друг с другом и степень использования транзактов. Поэтому наряду с характеристиками обслуживания очередей транзактов сложной структуры фиксируются статистики использования процессами ресурсов системы и коэффициентов загрузки самих процессов. Важной статистикой обслуживания внешней среды являются времена жизни транзактов (интервал времени от момента создания транзакта до момента его уничтожения).

При *агрегатно-процессном* способе имитации также имеет место сочетание статистик использования агрегатов-имитаторов технологических операций и агрегатов-имитаторов оборудования со статистиками использования ресурсов СС. Важной статистикой при данном способе формализации являются времена выполнения технологических процессов или отдельных их составляющих. В качестве критериев эффективности СС зачастую используются характеристики динамики информационно-логической взаимосвязи компонентов СС.

При *транзактно-процессном* способе имитации исследователю, согласно методам, приходится сочетать аппроксимацию процессов СС, процессного способа имитации с аппроксимацией функций внешней среды и кодировкой их в «теле» сложного транзакта.

В случае *агрегатно-процессного* способа имитации состав агрегатов типовой. Структура алгоритмических операций также типовая. Поэтому само исследование ресурсов и состав переменных моделирования, используемых агрегатами, отличаются друг от друга количеством

и вероятностными значениями запросов ресурсов и устройств СС. Благодаря типовой структуре алгоритмов агрегатов-имитаторов компонентов СС существенно сокращается объем работ при составлении данного шага этапа составления концептуальной модели СС.

В результате работы на данном шаге исследователь уточняет зависимость критериев эффективности моделируемой системы от накапливаемой статистики моделирования и определяет место и моменты включения процедур для вычисления статистик моделирования.

Аппроксимация реальных процессов математическими выражениями. Возможны два случая: исследователь знает вид функциональной зависимости поведения процессов от параметров и переменных моделирования, и когда реализация процессов зависит от сложившейся ситуации в системе. На данном шаге оценивается удобство аппроксимации этих процессов при каждом способе имитации.

При аппроксимации реальных процессов *событийным* способом имитации необходимо обратить внимание на идентичность алгоритмов, обслуживающих однотипные события в совпавших активностях. Аппроксимация реальных явлений обслуживанием *транзактов* требует представления взаимодействия компонент системы в виде схемы. Для агрегатного способа имитации аппроксимацию реальных процессов облегчает наличие хорошо изученных математических методов аппроксимации агрегатами. Для способа имитации *активностями* или *процессами* точность аппроксимации функциональных действий $ФД_{ij}$ активностями определяется значениями интервалов изменения модельного времени τ_{ij} . Иногда для уменьшения ошибки такой аппроксимации приходится разбивать интервалы τ_{ij} на несколько более мелких, создавая при этом новые активности.

При *транзактно-процессном* способе имитации исследователю приходится сочетать аппроксимацию процессов СС, согласно методике процессного способа имитации, с аппроксимацией функций внешней среды и кодировкой их в «теле» сложного транзакта. В случае *агрегатно-процессного* способа имитации состав агрегатов типовой. Структура алгоритмических операций также типовая. Поэтому само использование ресурсов и состав переменных моделирования, используемых агрегатами, отличаются друг от друга количеством и вероятностными значениями запросов ресурсов и устройств СС. Благодаря типовой структуре алгоритмов агрегатов-имитаторов компонентов СС существенно сокращается объем работ при выполнении данного шага этапа составления концептуальной модели СС.

Обычно исследователь сопоставляет величину ошибки от аппроксимации функциональных действий в реальных процессах при каждом

из способов имитации и оценивает сложность выполнения такой аппроксимации. Под сложностью выполнения аппроксимации функциональных действий ΦD_{ij} обычно понимают количество активностей, число взаимосвязей между ними, трудность написания и наглядность представления алгоритмов активностей AL_{ij} . Весьма существенна для выбора способа формализации также возможность реализации алгоритмов AL_{ij} на базовом языке программирования, имеющемся в распоряжении исследователя.

Задание исходной информации. Часто для задания исходной информации необходимо провести натурные эксперименты на моделируемой системе или на прототипах (при проектировании системы). Сложность проведения натурных экспериментов может существенно повлиять на выбор способа имитации. Может оказаться, что для обеспечения необходимой точности при аппроксимации некоторых явлений одним способом требуется проведение множества дорогостоящих экспериментов, тогда как при другом способе имитации эти эксперименты можно существенно сократить. Оцениваются объем работ по сбору статистики и степень доступности прототипов реальных компонент системы для организации соответствующих измерений.

Независимо от способа имитации зачастую определение функций распределения значений τ_{ij} квантов изменения временных координат $F_{ij}(\tau)$ является составной частью процедуры задания исходной информации моделирования. Индексы i и j означают, что возможно, для каждой активности необходимо находить свой закон распределения значений квантов изменения модельного времени.

Порядок проведения имитационных экспериментов чаще всего излагается в виде сценариев проведения эксперимента. При этом необходимо оценить трудозатраты на реализацию выбираемых сценариев. Может оказаться, что концептуальная модель объекта выбрана неверно, так как требует сложных сценариев для проведения имитационного эксперимента с большим расходом материальных и людских ресурсов. Как показано в [15], в ходе работ на данном этапе разработчик модели должен получить ответ на следующие вопросы: какую исходную информацию следует считать необходимой для построения модели; где можно получить эту информацию и какими методами следует производить обработку исходной информации. Весьма важно предусмотреть либо возможность замены труднодоступной информации более доступной, либо другие варианты решения построения модели. В ряде случаев для получения недостающей исходной информации полезно использовать аналитические модели (чаще всего регрессионные или дисперсионные). Р. Шеннон [30], например,

рекомендует провести следующую последовательность работ: «... просмотреть литературу, составить библиографические справки, проанализировать документы и отчеты, провести консультации со специалистами и экспертами».

Выдвижение гипотез и предположений. В каждом конкретном случае при выдвижении гипотез для описания функциональных действий $ФД_{ij}$ необходимо учитывать следующие факторы: объем информации, имеющейся для решения задачи; состав задач, для решения которых этой информации недостаточно; наличные ресурсы времени и памяти ЭВМ, на которой будет осуществляться имитация; ожидаемые результаты моделирования. На данном шаге исследователю приходится проводить серию натуральных и имитационных экспериментов, в ходе которых проверяются выдвинутые гипотезы об алгоритмическом представлении функциональных действий. Уточняются алгоритмы активностей $АЛ_{ij}$ и определяются формулы для расчета интервалов изменения временных координат компонент модели τ_{ij} . Так, при имитации активностями, событиями и процессами зачастую выдвигаются гипотезы о том или ином виде регрессионной зависимости при оценке интервалов изменения модельного времени τ_{ij} . Принятие подобных гипотез облегчает расчеты интервалов изменения модельного времени в ходе имитационного эксперимента. Для транзактного способа имитации чаще всего выдвигаются предположения об экспоненциальном характере функций распределения параметров τ_{ij} и используется допущение о пуассоновском характере входного потока транзактов. Для агрегатного способа имитации иногда приходится принимать гипотезы о конкретном виде операторов H – перехода агрегата из состояния в состояние – и операторов G – формирования выходных сигналов. Подтверждение экспериментом выдвинутых предположений и гипотез может повлиять на выбор способа формализации реальных явлений.

Исследователь также использует гипотезы и предположения при составлении самих алгоритмов $АЛ_{ij}$, аппроксимирующих функциональные действия $ФД_{ij}$. Может оказаться, что некоторые из этих предположений чрезвычайно трудно реализовать одним из способов имитации и очень просто учесть при другом способе организации квазипараллелизма. Это обстоятельство иногда оказывается решающим для выбора способа имитации.

Установление основной структуры модели. При решении этого вопроса необходимо учитывать сложившуюся обстановку к моменту начала проектирования сложной системы: состояние среды, в которой будет функционировать система; состав функций, для выполнения

которых предназначена система; наличие взаимосвязей между человеком и системой. Прежде всего нужно дать ответы на следующие вопросы: какие функции выполняет система и каковы способы их реализации в модели, каков состав детерминированных и недетерминированных функций системы и какова аппроксимация этих функций в модели, каково влияние факторов внешней среды на работу системы, какие используются способы взаимосвязи человека с системой и каким образом они аппроксимируются в модели.

Характер ответов на эти вопросы является определяющим при выборе способа формализации. Так, при *имитации активностями* исследователь проводит группировку активностей по условиям их выполнения, составляет списки взаимосвязанных друг с другом активностей и выбирает окончательный вариант схемы взаимодействия активностей с УПМ. Для *событийного* способа имитации события C_{ij} группируются по условиям их выполнения. Затем составляются списки взаимосвязей между событиями и уточняется схема связей с УПМ. При *имитации процессами* исследователь, наоборот, объединяет активности по принадлежности к одной реальной компоненте системы и составляет списки взаимосвязанных функциональных действий FD_{ij} внутри отдельных компонент сложной системы. Для *агрегатного* или *транзактного* способа имитации по схеме уточняется соответствие входов и выходов агрегатов или блоков, на которые поступают сигналы или транзакты.

Таким образом, при установлении основной структуры модели исследователь должен сделать окончательный выбор способа формализации. При этом необходимо провести следующую последовательность действий: оценить целесообразность построения имитационной модели и удобство аппроксимации результатов при каждом из способов имитации; проанализировать точность аппроксимации функциональных действий в системе и соответствие принятых гипотез и предположений; рассмотреть возможные режимы функционирования компонент системы; исследовать влияние возмущающих факторов на поведение системы; проверить достоверность исходной информации для моделирования и источников ее получения; заново пересмотреть постановку задачи на моделирование и оценить удобство способов имитации для ее решения. Указанная последовательность действий представляет собой проверку соответствия концепций имитационной модели реальной системе.

Завершается этап составления концептуальной модели проведением оценок будущих точностных характеристик имитации реальной системы при каждом из способов имитации. Возможно, что некоторые

из способов имитации исследователь забраковал на предыдущих шагах данного этапа. Для случая, когда некоторые способы имитации равнозначны, оценка точностных характеристик позволяет сделать окончательный выбор способа имитации (а значит, и способа формализации) объекта моделирования.

Документирование результатов. Последним действием при формулировке концептуальной модели является документирование результатов. Составляется итоговый документ, который должен содержать следующую информацию: подробную постановку задачи, список параметров, переменных и критериев эффективности системы, перечисление гипотез и предложений, обоснование целесообразности построения модели, описание концептуальной модели согласно принятому способу формализации.

Итак, рассмотренные выше действия привели исследователя к обоснованному выбору способа формализации. Можно утверждать, что к данному моменту разработки имитационной модели исследователь построил концептуальную модель, ориентированную на вполне определенный способ организации имитации.

3.2 Примеры составления концептуальных моделей СС

Проиллюстрируем изложенную выше методику составления концептуальных моделей СС [7] на основании содержательного описания (СОДОП) тех примеров СС, которые были рассмотрены нами ранее в главе 2. В примере 1 приведено СОДОП вычислительного процесса в ЛВС на первом уровне его детализации. Пример 2 представляет собой СОДОП вычислительного процесса в узле ЛВС при адаптации рабочей нагрузки к составу ресурсов узла ЛВС. СОДОП потока обработки деталей на производственном участке приведено в примере 3. Поточное производство разборочных, ремонтных и сборочных работ более сложно организовано, и поэтому СОДОП такого производства представлено в примере 4. Составление СОДОП последовательного ТПП приведено в примере 5. Наконец, методика составления СОДОП для дискретного параллельно-последовательного вероятностного технологического процесса производства иллюстрируется на примере 6. Для каждого из этих примеров методика составления концептуальной модели (КОМ) предполагает свою последовательность шагов и состав документов, составляющих КОМ.

Концептуальная модель первого уровня детализации ВП в ЛВС. Используем кибернетический подход к построению КОМ₁ первого уровня детализации ВП в ЛВС. На входе ВП узла ЛВС задана РН с помощью вектора ее состава X_{PH} и характеристик структуры РН (G_{ph}) с помощью графов GR_{f1i} для i -го типа запросов ZP_i пользователей ЛВС. На УД «Топология» РН представляет собой последовательность запросов $\{ZP_i\}$, которые обрабатываются ЛВС в течение определенного промежутка времени и формируют множество $\{X_{PH1}\}$. Различаем пять классов РН на узлах ЛВС: фоновые задачи (Fon_i), фоновые сетевые задачи ($FonNET_i$), диалоговая обработка информации ($DialNEt$), удаленный счет ($Calc_i$), транзитные запросы через узлы ЛВС ($TRahZ_i$). Структура РН определяется матрицей адресации ZP_i .

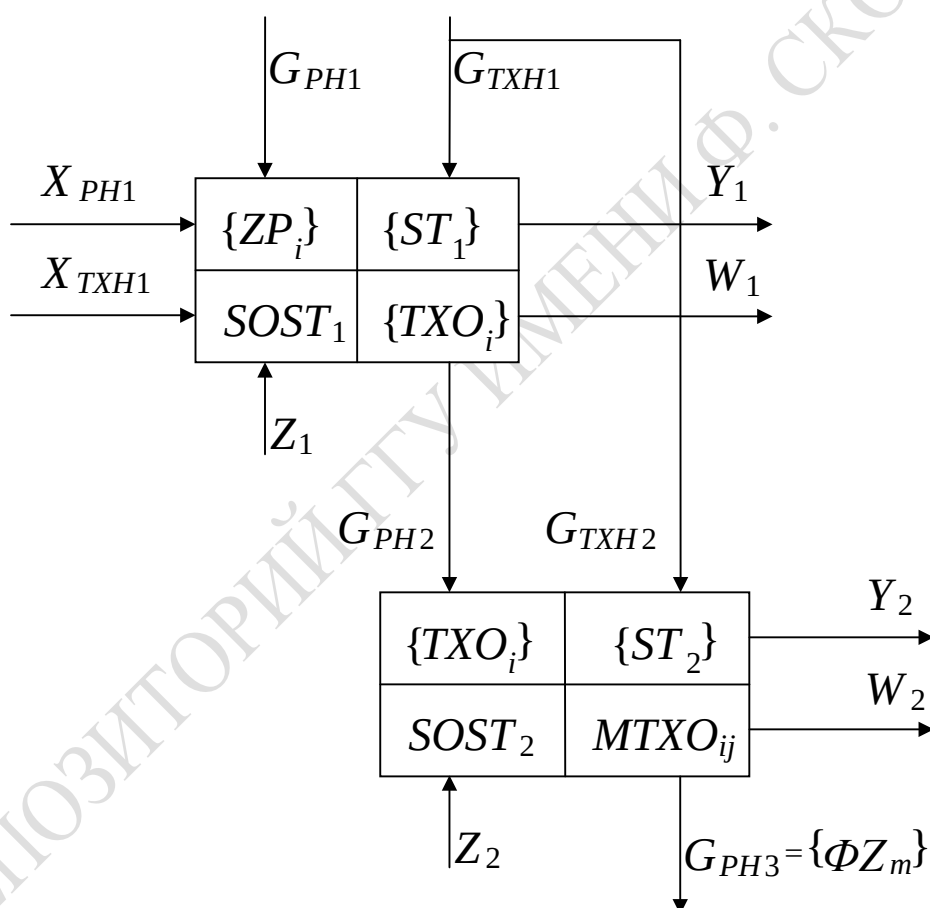


Рисунок 3.1 – Блок-схема связи концептуальных моделей ВП уровней детализации УД₁ и УД₂ согласно технологии исследования «сверху – вниз».

При исследовании динамики обслуживания ВП в ЛВС множества запросов $\{ZP_i\}$ фиксируются: текущие значения компонент вектора статистик $\{ST_{1k}\}$; последовательность смен состояний ВП $\{SOST_k\}$,

вектор откликов ВП (Y_1). Основным допущением будущей формализации ВП и РН на всех УД является полумарковская природа процессов использования ресурсов ЛВС. На рисунке 3.1 приведена блок-схема концептуальных моделей ВП при использовании технологии проектного моделирования «сверху вниз». Чтобы результаты исследования ВП на уровне УД₁ можно было использовать в дальнейшем, на рисунке 3.1 показан пример применения результатов исследований УД₁ на уровне детализации УД₂ и затем на УД₃. КОМ₁ ВП в узле ЛВС (УД₂ ВП в ЛВС) задается значениями компонент векторов параметров ВП: управления ВП ($X_{ТХН2}$); надежности функционирования устройств узла ЛВС (Z_2); характеристики структуры ВП ($G_{ТХЧ2}$). При моделировании ВП УД₂ фиксируются: текущие значения компонент вектора статистик имитации $\{ST_{2k}\}$; последовательность смен состояний ВП $\{SOST_2\}$, вектор откликов ВП (Y_2). Поскольку выполнение каждой МТХО_{ij} требует ресурсов узла ЛВС, то для информационной стыковки системы моделей на более нижних уровнях детализации ВП необходимо задавать последовательность функциональных задач $\{\Phi Z_m\}$, отображающих структуру запросов к ЛВС ($G_{РН3}$) на следующем УД₃. Кроме значений векторов откликов Y_1 и Y_2 для каждого УД определяются значения обобщенных показателей качества (W_1 и W_2), формируемых по значениям компонент соответственно векторов Y_1 и Y_2 .

Структура $G_{РН1}$, определяемая множеством запросов пользователей УД₁ $\{ZP_i\}$, отображается графом G_{r1} . Аналогичным образом структура $G_{РН2}$, определяемая множеством $\{МТХО_{ij}\}$, в свою очередь, отображается графом G_{r2} (см. рис. 3.1). В ходе имитационного эксперимента с ИМ «Топология ЛВС» на УД₁ фиксируется последовательность $\{ТХО\}$, которая уже затем на УД₂ в ИМ «узла ЛВС» задается в виде множества $G_{РН2}$. Аналогичным образом в ИМ «узла ЛВС» фиксируется статистика выполнения $\{МТХО_{ij}\}$, которая далее на УД₃ представляется в виде множества $G_{РН3}$, образуя в совокупности множество запросов ресурсов ЛВС задачами $\{\Phi Z_m\}$.

Вероятностный характер запросов ресурсов ЛВС КОМ₁ ВП УД₁ и УД₂ отображен на блок-схеме (рисунок 3.1) когда внешними воздействиями являются Z_1 и Z_2 .

Концептуальная модель ВП узла ЛВС при адаптации РН к составу ресурсов сервера ЛВС. Согласно кибернетическому подходу к построению КОМ₂ третьего уровня детализации ВП в ЛВС имеет вид, представленный на рисунке 3.2. При этом решается задача адаптации порядка выполнения последовательности задач $\{\Phi Z_m\}$ отложенного счета к имеющемуся составу ресурсов сервера ЛВС (RES_3), который определяется множеством $G_{ТХ3}$. Определяется вектор

состава рабочей нагрузки $X_{PH3} = (\lambda_D, \lambda_T)$, где λ_D и λ_T соответственно интенсивности поступления диалоговых и транзитных запросов на ресурсы сервера ЛВС. Сама же структура запросов задана в виде множеств G_{PH1} и G_{PH2} и не меняется в ходе имитации.

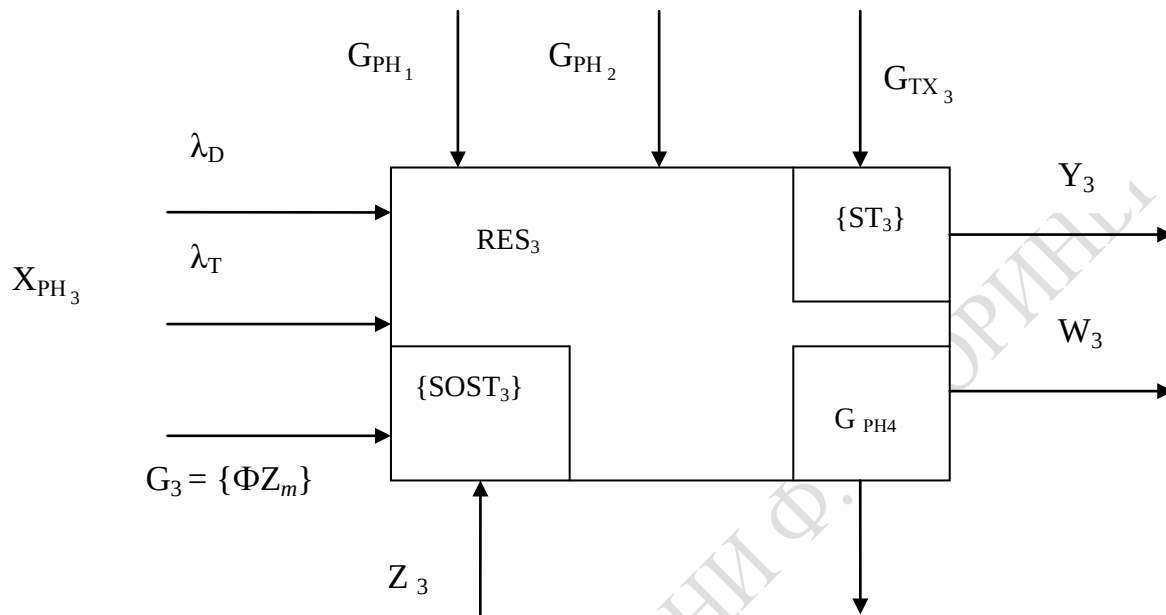


Рисунок 3.2 – Блок-схема концептуальной модели ВП УД₃ узла ЛВС при адаптации порядка пропуска к составу ресурсов узла (RES_3)

Управляемым параметром имитации является порядок следования компонентов из последовательности $\{\Phi Z_m\}$. При моделировании динамики решения последовательности задач $\{\Phi Z_m\}$ (на фоне приоритетного использования RES_3 узла ЛВС диалоговыми и транзитными запросами) фиксируется: статистика использования ресурсов сервера $\{ST_3\}$, последовательность смен состояний ВП $\{SOST_3\}$, вектор откликов ВП (Y_3). На рис. 3.2 приведена блок-схема концептуальной модели ВП узла ЛВС. Результатом имитации, кроме получения распределения значений компонент вектора откликов Y_3 , является определение обобщенного показателя качества организации ВП (W_3) при заданном порядке следования задач $\{\Phi Z_m\}$.

Кроме того, для моделирования ВП на более высоком уровне детализации фиксируется статистика запросов ресурсов сервера $\{G_{PH4}\}$, которая используется в качестве исходной информации. Анализ динамики расхода ресурсов сервера запросами задач отложенного счета при различном порядке их выполнения осуществляется с помощью множества состояний ресурсов узла ЛВС $\{SOST_3\}$.

Вероятностный характер запросов ресурсов узла ЛВС определяет наличие внешнего воздействия Z_3 в КОМ₂.

Концептуальная модель исследования динамики функционирования производственного участка обработки деталей. Декомпозиция системы осуществляется просто. Отдельными элементами данной системы (производственного участка) будут обрабатывающие станки (см. рисунок 2.5). В качестве параметров системы выступают интенсивности поступления деталей первого и второго типов (λ_1 и λ_2). Для простоты допустим, что поступление деталей обоих типов подчиняется пуассоновскому закону. Переменными модели системы являются функции распределения длительностей обслуживания i -й детали j -м обрабатывающим станком $F_{ij}(\tau)$. В качестве статистик моделирования будут выступать: коэффициенты загрузки станков OCT_j (η_j), количество обработанных j -м станком деталей v_j , размеры очередей к каждому из станков l_j , общее время обработки i -й детали на производственном участке T_i . Поскольку моделируется достаточно простая структура, то для данного случая в качестве критериев эффективности могут выступать коэффициенты загрузки станков η_j и средние значения времен обработки деталей T_i .

В результате имитационного моделирования необходимо найти следующие функциональные зависимости: $\eta_j = \phi_{ij}(\lambda_i)$; $T_i = \psi_{ij}(\lambda_i)$. При задании функций распределения $F_{ij}(\tau)$ длительностей обслуживания i -й детали на j -м станке достаточно аппроксимация ступенчатыми функциями. Выдвигаем гипотезу, что ϕ_{ij} и ψ_{ij} имеют вид полиномов, порядок и значения коэффициентов которых необходимо определить в ходе имитационных экспериментов. С помощью таких эмпирических зависимостей можно предсказывать характеристики загрузки оборудования (η_j) и времена обслуживания деталей на производственном участке (T_i) в зависимости от изменяющихся характеристик входного потока деталей λ_i при заданных $F_{ij}(\tau)$.

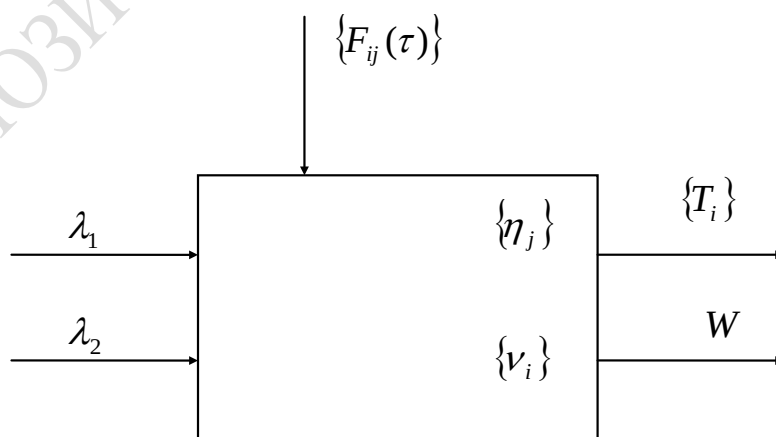


Рисунок 3.3 – Блок-схема концептуальной модели производственного участка обработки деталей

Структура модели достаточно проста и имеет блочный характер. Проверка достоверности концептуальной модели в силу ее простоты и близости к реальной системе не производится. Документацией концептуальной модели является: схема, приведенная на рисунке 3.3; текст содержательного описания; список параметров и переменных (λ_i ; $F_{ij}(\tau)$); состав статистики моделирования (η_j , v_i , l_i , T_i); соотношения для критериев эффективности ϕ_{ij} , ψ_{ij} ; аппроксимирующие выражения для задаваемых переменных $F_{ij}(\tau)$.

Концептуальная модель исследования динамики сборочно-разборочных операций ремонта изделий. Используя кибернетический подход к исследованию СРО ремонта изделий на предприятии (для примера 4), КОМ₄ исследования СРО имеет вид, представленный на рисунке 3.4. На время имитации h -го варианта W СРП в качестве начальной информации задаются:

- TK_i – технологическая карта ремонта изделия i -го типа;
- G_{Zi} – характеристики изношенности деталей, включающие в себя:
- матрицу неисправностей деталей MP_{ij} (i – номер ветви ремонта деталей, j – номер детали в блоке);
- список запросов ресурсов предприятия (ZPR_i), необходимых для выполнения множества технологических операций $\{MTXO_{ij}\}$.

В свою очередь, ZPR_i , включает в себя:

$\{SPVOR_{ij}\}$ – список функций вероятностей распределения требуемых объемов общего ресурса (V_{Rij}) для выполнения $\{MTXO_{ij}\}$;

$\{MFFOS_{ij}\}$ – список функций вероятностей распределения времени осмотра деталей при выполнении $\{MTXO_{ij}\}$;

$\{MFREM_{ij}\}$ – список функций вероятностей распределения времени ремонта изделий при реализации $\{MTXO_{ij}\}$.

На выходе КОМ₄ определяются следующие отклики имитации:

$Y_{1b} = v_{mh}$ – пропускная способность ТПСРО;

$Y_{2h} = \sum_{ij} \eta_{rijh}$ – суммарный коэффициент использования множества технологических операций ($MTXO_{ij}$);

$Y_{3h} = \sum_k \eta_{khh}$ – суммарный коэффициент использования множества рабочих мест;

$Y_{4h} = \sum_r \eta_{rhh}$ – суммарный коэффициент использования множества ресурсов;

Y_{sh} – величина дохода предприятия за счет ремонта потока изделий.

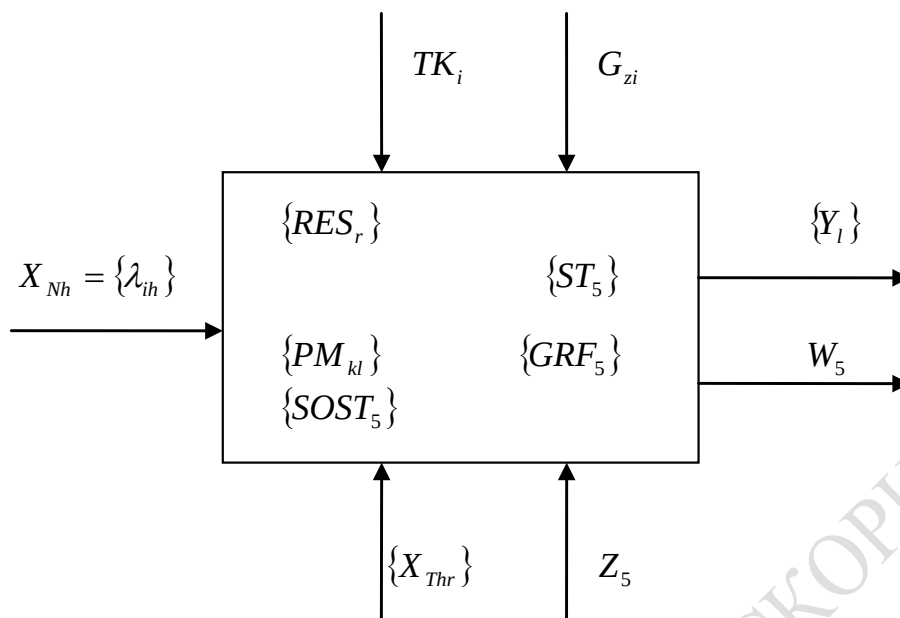


Рисунок 3.4 – Блок-схема концептуальной модели сборочно-разборочного производства ремонта изделий.

Обобщенный отклик h -го варианта технологии ремонта изделий определяется из выражения:

$$W = \sum_{l=1}^s \delta_l \cdot Y_l^*, \quad (3.1)$$

где $0 \leq \delta_l \leq 1,0$ – весовые коэффициенты важности отклика для исследователя $\left(\sum_{l=1}^s \delta_l = 1 \right)$,

Y_l^* – нормированные значения компонент вектора откликов $\{Y_l\}$.

До начала имитационного эксперимента (ИЭ) задается начальный состав ресурсов $\{RES_{0r}\}$, который используется на рабочих местах $\{RM_{kh}\}$ на время выполнения $\{MTXO_{ij}\}$, и затем они возвращаются системе распределения ресурсов предприятия (SRRP). Количество рабочих мест и их распределение по цехам номера l $\{PM_{klh}\}$ неизменно в ходе ИЭ.

В ходе имитации выполнения множества $\{MTXO_{ij}\}$ на множестве рабочих мест $\{PM_{klh}\}$ фиксируется статистика (ST_s) . Статистиками имитации СРО (компонентами множества $\{ST_{sn}\}$) являются:

– коэффициенты использования операций $MTXO_{ij}$ (η_{Tijh}), рабочих мест PM_k (η_{kh}), ресурсов предприятия (η_{rh});

– средние значения длин очередей $\bar{l}_{ors}$ и времени нахождения в очереди $(\bar{t}_{ors})$ к s -му элементу ИМ технологии СРО;

– T_{Rli} – общее время ремонта изделия i на l -й технологической ветви ремонта изделия D .

Параметрами являются:

$X_{Nh} = \{\lambda_{ih}\}$ – вектор интенсивностей поступления деталей i -го типа в h -м варианте ТП СРП;

X_{Thr} – вектор имеющихся ресурсов предприятия для организации СРО (r – номер ресурса, h – номер варианта технологии СРО).

Для изучения переходов ТП СРО из состояния в состояние используется множество характеристик состояний $\{SOST_s\}$. В ходе имитации ТП СРП фиксируется множество $\{GRF_s\}$, компонентами которого являются графики и временные диаграммы использования ресурсов предприятия на организацию ремонтов согласно технологической карте ремонта i -го изделия (TK_i).

По статистике и значениям откликов модели формируются *эмпирические зависимости* изменения обобщенного отклика от параметров моделирования:

$$W_h = \varphi_s(X_{Nh}, X_{Thr}). \quad (3.2)$$

Ищется такой вариант организации (h) ремонта изделий, который обеспечивает максимум функционала (3.2) при фиксированных значениях TK_i и G_{zi} . Дополнительной информацией для поиска функциональна (3.2) является статистика $\{RES_z\}$, $\{ST_s\}$, $\{SOST_s\}$ и $\{GRR_s\}$. Внешнее воздействие на систему (Z_s) в КОМ₄ отображает наличие в ТП СРО стохастических компонентов.

Концептуальная модель последовательного вероятностного ТПП. Согласно кибернетическому подходу переходим к построению КОМ₅ последовательного вероятностного ТПП вида, представленного на рисунке 3.5. На время имитации h -го варианта ВТПП₁ в качестве *исходной информации* задается:

– MP_{ij} – матрица вероятностей перехода ВТПП₁ из состояния i в состояние j ;

– $\{G_{zh}\}$ – множество характеристик запросов ресурсов (RES_i) предприятия в j -ом состоянии (таблица 3.1) в h -м варианте последовательного ТПП;

– $\{H_{rh}\}$ – множество характеристик надежности функционирования оборудования r -го типа (времена перехода r -го оборудования из состояния в состояние), вероятности отказа (P_{OTr}) и возникновения при этом аварии (P_{avr}) на r -м оборудовании (таблица 3.2).

Параметрами h -го варианта ВТПП являются компоненты множества ресурсов предприятия $\{X_{Rh}\}$.

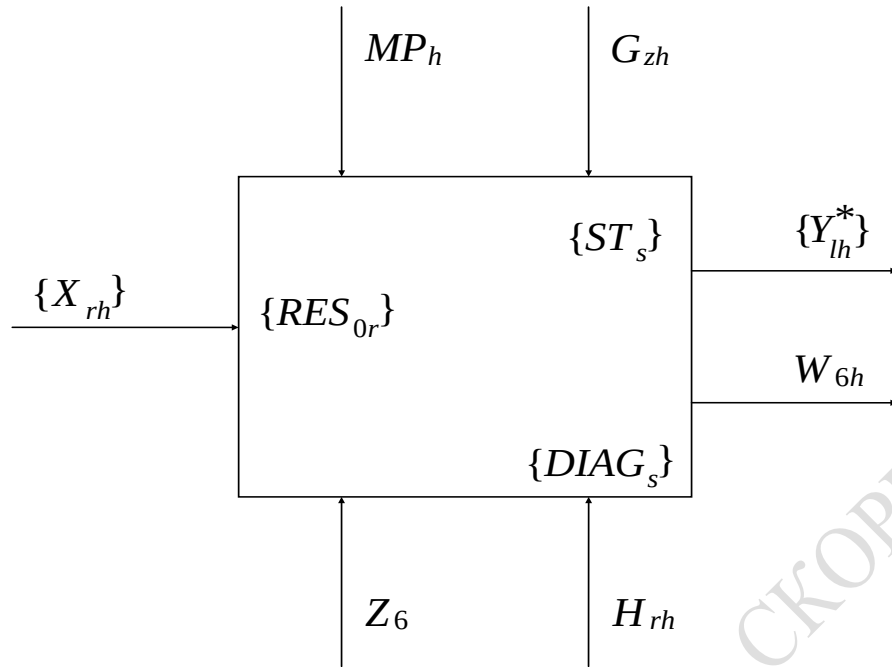


Рисунок 3.5 – Блок-схема концептуальной модели последовательного вероятностного ВТПП₁

Откликами при имитации h -го варианта ВТПП₁ являются следующие компоненты вектора $\{Y_{lh}\}$:

Y_{1h}^* – нормированное время реализации ВТПП₁ при имеющемся составе ресурсов, запросах запросов предприятия $\{G_{zh}\}$ и надежность характеристик оборудования $\{H_{rh}\}$.

$\{Y_{2lh}^*\} = \left\{ \sum_{r=1}^R \eta_{irh} \right\}$ – суммарный коэффициент использования ресурсов r -го типа;

Y_{3h}^* – приведенные значения фиксированных затрат предприятия на реализацию h -го варианта ВТПП₁.

Обобщенный отклик h -го варианта последовательного ВТПП₁ определяется из выражения:

$$W_h = \delta_1 Y_{1h}^* + \delta_2 \sum_{r=1}^R Y_{2rh}^* + \delta_3 \cdot Y_{3h}^*, \quad (3.3)$$

где $\delta_1, \delta_2, \delta_3$ – весовые коэффициенты важности отклика ВТПП₁ для исследователя.

В ходе ИЭ фиксируется множество: статистик изменений переменных во времени (ST_s) и диаграмма смен состояний процесса $\{DIAG_s\}$.

По статистикам и значениям откликов модели формируются эмпирические зависимости обобщенного отклика от параметров моделирования:

$$W_h = \varphi_s(X_{rh}). \quad (3.4)$$

Целью имитации является поиск рационального варианта (h_0) ВТПП₁ при фиксированных значениях MP_h , G_{zh} , H_{rh} и RES_{or} .

Внешнее воздействие Z_5 в КОМ₅ означает наличие в ВТПП₁ запросов на ресурсы, имеющих вероятностный характер, и возможность стохастических связей компонентов в ВТПП₁.

Концептуальная модель параллельно-последовательного технологического **процесса производства**. Используя кибернетический подход, составим КОМ₆ параллельно последовательного ВТПП. На рисунке 3.6 приведена блок-схема КОМ₆ вероятностного ТПП, в котором допускается одновременное выполнение МТХО_{ij}. Зависимость выполнения одной группы $\{MTXO_{ij}\}$ от завершения другой группы $\{MTXO_{si}\}$ требует рассмотрения технологии на более высоком уровне детализации. Поскольку ВТПП имеет графовую структуру, то перед имитацией необходимо задать структуру графовой зависимости операций (GRF_h). Кроме того, в качестве *исходной информации* необходимо задать множество запросов $\{G_{zh}\}$ ресурсов каждой МТХО_{ij} в h -го варианте ВСГР (таблицы 2.16, 2.10).

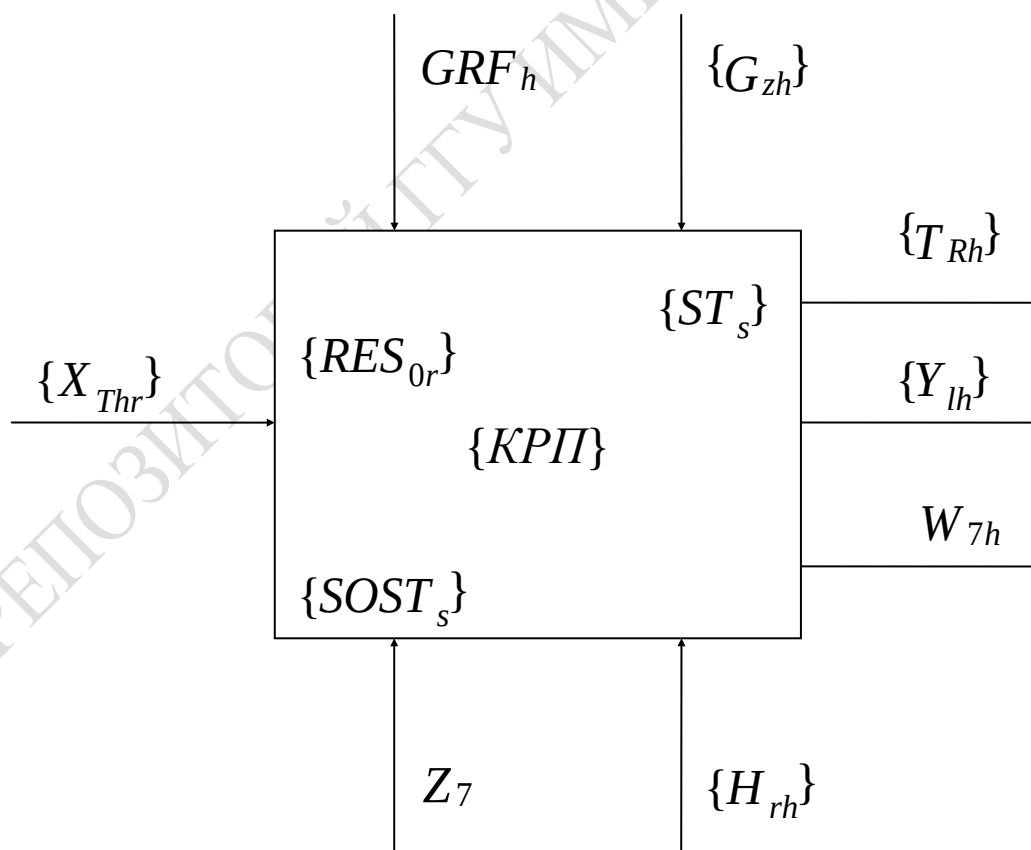


Рисунок 3.6 – Блок-схема концептуальной модели вероятностного параллельно-последовательного ВТПП₂

Параметрами h -го варианта ВТПП₂ являются компоненты множества ресурсов предприятия $\{X_{Thr}\}$ (таблица 2.15) и компоненты множества характеристик надежности функционирования оборудования r -го типа (времена выполнения МТХО_{ij}, вероятности отказа (P_{OTr}) и вероятности возникновения при этом аварии (P_{abz}) на r -м оборудовании (таблица 2.17).

Откликами при имитации h -го варианта ВТПП₂ являются:

$Y_{1h} = T_{kh}$ – критическое время реализации ВТПП₂ при имеющемся составе ресурсов, запросах МТХО_{ij} на эти ресурсы $\{G_{zh}\}$ и надежность характеристик оборудования $\{H_{rh}\}$;

$\{Y_{zrh}^*\} = \sum_{r=1}^R \eta_{rh}$ – суммарный коэффициент использования ресурсов r -го типа;

Y_{3h}^* – приведенные значения финансовых затрат предприятия на реализацию h -го варианта ВТПП₂.

Обобщенный отклик h -го варианта ВТПП₂ вычисляется по формуле:

$$W_h = \delta_1 Y_{1h}^* + \delta_2 \sum_{r=1}^R Y_{zrh}^* + \delta_3 \cdot Y_{3h}^*, \quad (3.5)$$

где $\delta_1, \delta_2, \delta_3$ – весовые коэффициенты важности отклика ВТПП₂ для исследователя.

Во время ИЭ фиксируется множество следующих статистик $\{ST_s\}$:

- множество использования ресурсов ВТПП₂ $\{\eta_{rh}\}$;
- множество диаграмм использования ресурсов ВТПП₂ $\{VD_z(t)\}$;
- множество расхода во времени имитации ВТПП₂ материалов, комплектующих и финансовых ресурсов $\{GRAF_r(t)\}$. По статистике и значениям откликов модели формируются эмпирические зависимости обобщенного отклика от параметров моделирования:

$$W_h = \varphi_7(X_{Thr}). \quad (3.6)$$

Целью имитационного моделирования является поиск рационального варианта (h_0) ВТПП₂ при фиксированных значениях GRF_h , $\{G_{zh}\}$, $\{RES_{0r}\}$ и $\{H_{rh}\}$. Внешнее воздействие Z_6 в КОМ₆ отображает наличие стохастического характера запросов МТХО_{ij} ресурсов предприятия и вероятностный характер самой структуры ВТПП₂.

3.3 Итоги

Подведем итоги демонстрации методики составления концептуальной модели исследования СС. В 3.1 излагается методика составления

КОМ путем реализации следующих действий исследователя: декомпозиция СС для каждого способа формализации СС; выбор параметров и переменных моделирования; уточнение критериев эффективности функционирования СС; аппроксимация реальных процессов математическими выражениями; выдвижение гипотез и предположений; установление структуры ИМ; документация результатов этапа. В 3.2 общая методика составления КОМ демонстрируется на шести примерах из разных предметных областей:

- составление КОМ исследования ВП в ЛВС (3.1);
- исследование ВП в узлах ЛВС при адаптации РН к составу ресурсов сервера ЛВС (3.2);
- исследование динамики функционирования производственного участка обработки деталей (3.3);
- исследование динамики функционирования сборочно-разборочных операций ремонта изделий сложной природы (3.4);
- изучение последовательного вероятностного технологического процесса производства (3.5);
- исследование технологии параллельно-последовательного ВТП₂ (3.6).

4 ФОРМАЛИЗАЦИЯ СЛОЖНОЙ СИСТЕМЫ

4.1 Методика формализации объекта моделирования

При составлении формального описания системы исследователю рекомендуется следующая последовательность действий: уточнение декомпозиции системы; алгоритмизация компонент модели; информационная стыковка алгоритмов модели; установление управляющих связей между компонентами модели; уточнение взаимодействия с УПМ; документация этапа.

Уточнение декомпозиции системы. В соответствии с показателями качества функционирования системы на предыдущем этапе был установлен уровень детализации моделирования. Исследователь также выполнил декомпозицию системы на составные компоненты, выделил места взаимодействия компонент системы друг с другом. Поэтому основные усилия исследователя на данном этапе направляются на уточнение схемы взаимодействия компонент системы между собой и с УПМ.

При имитации активностями уточняется состав активностей, имитирующих функциональные действия $ФД_{ij}$, и подпрограмм проверки условий инициализации активностей $АК_{ij}$. Обычно уточнение сводится к делению некоторых активностей на составляющие, соответствующие меньшим значениям τ_{ij} их временной координаты, если при оценке ошибки аппроксимации $ФД_{ij}$ активностями $АК_{ij}$ ее значение оказалось слишком большим. Кроме того, возможно введение ряда служебных активностей, в функции которых входит, например, сбор статистики моделирования или анализ границ появления конфликтных ситуаций в модели системы.

Для *событийного способа* имитации весьма существенна проверка совместимости выполнения условий появления событий реальной системы и обслуживания их одной и той же подпрограммой модели.

В случае *транзактного способа* имитации важно, например, определить условия запуска блоков на обслуживание транзактов, имитирующих реальные функциональные действия $ФД_{ij}$, а также достаточность состава обслуживающих блоков и дисциплин обслуживания транзактов блоками для имитации всего многообразия явлений в реальной системе. Под дисциплиной обслуживания транзактов понимают порядок поступления и выбора транзактов из очереди к данному блоку.

Для *агрегатного способа* формализации исследователь обязан определить состав операторов H и G , а также проверить возможность

и достаточность описания с их помощью алгоритма $АЛ_{ij}$ активности $АК_{ij}$.

В случае *процессного способа* имитации важно проверить состав и связи активностей, объединяемых в процессы. Иногда может оказаться, что некоторые функциональные действия $ФД_{ij}$ одной и той же компоненты модели могут выполняться одновременно в силу ряда допущений при построении модели. Поэтому имитацию таких функциональных действий необходимо выполнять двумя процессами. Это означает, что одна реальная компонента системы представлена двумя процессами. Причем эти процессы будут состоять в основном из идентичных активностей и отличаться по составу друг от друга одной или двумя активностями.

Для *транзактно-процессного способа* имитации приходится сочетать *декомпозицию* взаимодействия сложных транзактов с процессами и с декомпозицией самих процессов. Здесь транзакты, обладающие «телом» (иногда сложной структуры) представляют собой динамические элементы ИМ. В структуре таких *динамических* элементов ИМ можно выделить: заголовок и информационно-логическую часть транзакта. В отличие от обычно транзактного способа имитации, где используется только заголовок, который перемещается внутри ИМ от одной очереди к прибору массового обслуживания, (ПМО) при смешанном способе имитации заголовок дополняется адресом нахождения информационно-логической части транзакта сложной структуры (α_i). Наряду с идентификатором (i) и приоритетом (π_i) этот адрес и составляет триаду $TR(i, \pi_i, \alpha_i)$, которая перемещается ИМ от одного процесса к другому. Эти процессы представляют собой *статические* элементы ИМ. Они также имеют сложную структуру, похожую на случай процессного способа имитации. При этом новым является то, что процессы могут являться источниками транзактов сложной структуры и находиться внутри самой ИМ и вне ИМ, отображая таким образом воздействие внешней среды на объект имитации. Все замечания по структуре процессов, сделанные нами выше при описании процессного способа имитации, в полной мере можно отнести к статическим элементам ИМ, имеющим в своей структуре более сложные алгоритмы активностей. Важно только то, что эти активности выполняются в модельном времени t_0 строго последовательно, что позволяет говорить о состояниях статического элемента ИМ сложной системы.

При *агрегатно-процессном* способе имитации функции динамических элементов ИМ выполняют сигналы. Но, в отличие от агрегатного способа имитации сигналы (S_g) в ИМ при данном способе имитации имеют более сложную структуру. Во-первых, различают сигналы двух

типов: действительные (sgd) и фиктивные (sgt). Для каждого типа сигнала в агрегатах (также сложной структуры) выполняются различные ветви алгоритма агрегатов. Во-вторых, сигналы могут нести информационную нагрузку, указывая на факт совершения аварии в виде признака ($\pi_{ab} = 1$), имевший место при выполнении предыдущего агрегата. В-третьих, сигналы могут иметь направление движение от агрегата-источника к агрегату-адресату движения сигнала. Сигналы, передаваемые от агрегата к агрегату, несут только информационную нагрузку, указывая только факт окончания выполнения агрегата. Агрегаты, являясь *статическими* элементами ИМ (в отличие от агрегатов в агрегатном способе имитации) имеют более сложный состав и структуру. По своей сути это процессы, но с более универсальным алгоритмом их выполнения, обеспечивающим имитацию выполнения в модельном времени $MTXO_{ij}$. При выполнении алгоритм агрегатов может использовать ресурсы ИМ. Под *ресурсом* (RES_r) в ИМ понимают информационно логические структуры, которые выделяются агрегатом, имитируя при этом использование устройств, имитацию расхода материалов и финансовых средств и времени выполнения.

Алгоритмизация компонент модели. Уточнив декомпозицию системы в соответствии с выбранным способом имитации, исследователь переходит к уточнению алгоритмов, аппроксимирующих функциональные действия $ФД_{ij}$ компонент реальной системы. Для этой цели составляются временные диаграммы (ВД) функционирования этих компонент.

При *имитации активностями* (или процессами) с помощью ВД изучаются места и времена взаимодействия активностей (или процессов) друг с другом. Для устранения конфликтных ситуаций при обслуживании активностей (или процессов) в УПМ исследователь устанавливает их приоритеты.

При *событийном способе имитации* с помощью ВД определяются общие по времени и условиям появления события в системе и выделяются места взаимодействия компонент при появлении типовых событий.

В случае *транзактного способа* имитации ВД возможных конфликтных ситуаций в обслуживании транзактов им (транзактам) устанавливаются приоритеты обслуживания и указываются дисциплины обслуживания транзактов этими блоками.

При *агрегатном способе* имитации также иногда возможно применение ВД. С их помощью исследуются некоторые случаи изменения координат агрегатов и возможность одновременного поступления

на вход агрегата нескольких сигналов. ВД при агрегатном способе имитации позволяет установить приоритеты обслуживания сигналов и наметить последовательность действий УПМ при возникновении конфликтных ситуаций в модели системы.

При *транзактно-процессном* способе имитации ВД кроме установления приоритетов и дисциплин обслуживания используются указатели, которые служат для проверки своевременности доступности информации в структуре сложного транзакта для процессов их использующих. Поскольку динамические элементы могут иметь в своей структуре алгоритмическую составляющую, необходимо проверить с помощью ВД возможность появления конфликта между алгоритмами динамического и статического элементов ИМ. Безусловно, исследователю зачастую приходится выполнять дополнительную работу по построению и анализу ВД. Однако в сложных ситуациях, при суперпозиции взаимодействий нескольких процессов и транзактов, ВД являются единственным средством для отработки правил разрешения конфликтных ситуаций в ИМ.

При *агрегатно-процессном* способе имитации роль ВД возрастает. Из-за квазипараллелизма использования ресурсов ИМ несколькими агрегатами в ряде случаев для синхронизации параллельных взаимодействий групп агрегатов приходится выделять специальные типы агрегатов, которые могут выполнять функции логических схем («И» и «ИЛИ»). В таких случаях ВД функционирования агрегата позволяет определить режимы нахождения агрегатов, установить условия перехода с одного режима на другой режим, фиксировать моменты таких переходов в модельном времени t_0 . Как правило, эти моменты определяются приходом или посылкой сигналов от одного агрегата к другому. С помощью ВД удастся установить возможность возникновения конфликтов между агрегатами при использовании ресурсов СС, определить приоритеты и порядок разрешения этих конфликтов.

Информационная стыковка алгоритмов модели. Следующим шагом формализации является определение мест использования и переработки информации компонентами модели. Уточняется состав входной и выходной информации модели. Для каждой компоненты модели уточняется состав управляющих параметров, влияющих на выполнение алгоритмов ($АЛ_{ij}$). При различных способах имитации различен и порядок действий исследователя по информационной стыковке $АЛ_{ij}$.

Так, при *имитации активностями* выделяются активности, связанные информационно друг с другом, и составляется информационная схема взаимодействия активностей. Весьма существенная увязка

всех мест взаимодействия активностей по информации с приоритетами активностей и возможным составом операций взаимодействия активностей. Увязка обычно состоит в согласовании по времени появления информации для активностей, имеющих различные приоритеты обслуживания УПМ. Необходимо предусмотреть способы задержки передачи информации от активностей, имеющих больший приоритет, к активностям с меньшим приоритетом обслуживания УПМ.

Для *событийного способа имитации* наличие информационной взаимосвязи событий друг с другом может существенно повлиять на принятие решений об объединении или разнесении событий, о применении общих или различных подпрограмм обслуживания событий в модели. Если при одних и тех же событиях в разные моменты поведения реальной системы используется различная информация для выработки управляющих воздействий между компонентами системы, то в модели они должны быть представлены различными программами обслуживания событий.

При *транзактном способе* имитации не моделируются информационные связи между компонентами модели. Поэтому наличие таких связей ставит под сомнение возможность транзактного представления модели системы. Иногда выйти из этого затруднения помогает выделение двух типов транзактов (информационных управляющих) и использование блоков с управляющими входами [15]. Подобное ограничение существует и для агрегатного способа имитации.

Легче всего организуется имитация информационного взаимодействия при *процессном способе формализации*. В каждом процессе выделяются активности, информационно связанные с активностями других процессов. Составляется схема информационного взаимодействия между процессами и определяются приоритеты процессов при информационной увязке процессов друг с другом.

При *транзактно-процессном способе* имитации роль информационной стыковки компонентов ИМ возрастает. Поскольку в ИМ необходимо отобразить взаимодействие динамических элементов со статическими элементами, то исследователю необходимо отображать в ИМ информационные связи. Общих рекомендаций не существует, поскольку каждый объект имитации уникален и сама имитация, как правило, ведется на высоком уровне детализации процессов, происходящих в СС. Для выявления «узких мест» взаимодействия компонентов ИМ осуществляется согласование времени появления информации и использования ее динамическими и статическими элементами ИМ. Определяются приоритеты обслуживания транзактов и процессов управляющей программой моделирования. В сложных случаях для появления

информации в необходимое время используются специальные управляющие транзакты (задержки появления информации). В таких ситуациях информация для процессов недоступна до прихода такого управляющего транзакта. Поэтому в ИМ используются два типа транзактов: управляющие (UTR) и информационные (ITR), а также специальные управляющие входы в процессах обслуживания транзактов с помощью комбинации сигналов «открыть» (OT_j), «закрыть» (ZK_j), «прервать» (PR_{kj}), здесь k и j – соответствующие номера входов процессов.

При *агрегатно-процессном способе* имитации информационная стыковка агрегатов осуществляется в самих алгоритмах агрегатов. Весьма существенная увязка «узких мест» информационного взаимодействия агрегатов. Как правило, синхронизация передачи информации между агрегатами осуществляется с помощью сигналов. Уточняется структура сигналов, которые несут в себе информацию о появлении аварии оборудования при выполнении предыдущей группы агрегатов. В ряде случаев для задержки передачи информации, доступной агрегатам, используются специальные типы агрегатов. Алгоритм функционирования таких агрегатов сводится к проверке условий доступности информации группе агрегатов и формированию соответствующих управляющих сигналов.

Уточняется схема информационных связей между компонентами модели. Для способов имитации активностями, процессами и событиями устанавливаются места рождения и потребления информации. В ряде случаев приходится строить модели массивов информации. Особенно важна эта процедура для процессного способа имитации, поскольку процессы могут обмениваться информацией большого объема, а в силу ограничений на объем модели трудно организовать информационный интерфейс между процессами через глобальные переменные. Для транзактного способа, например при использовании в качестве средства формализации МК АСИМ (например, в [19]), можно организовать специальные логические схемы по обслуживанию информационных транзактов отдельно от управляющих. При агрегатном способе имитации некоторые сигналы могут нести часть информационной нагрузки, в то время как другие сигналы определяют только управляющие воздействия между агрегатами.

Установление управляющих связей между компонентами модели. Следующим шагом переработки концептуальной модели содержательного описания в формальное описание является составление схемы управляющих связей между компонентами модели. Такая схема обычно представляет собой ориентированный граф, в котором

вершинами являются элементы модели, а связи показывают направление воздействия элементов друг на друга, условий появления этого воздействия. Объем работы по составлению схемы зависит от принятого способа имитации.

Например, для *транзактного способа имитации* схему управляющих связей вообще составлять не требуется, поскольку развязка блоков обслуживания транзактов осуществляется через очереди к блокам. Для *агрегатного способа* достаточно составить матрицу коммутации агрегатов друг с другом, и исследователю гарантируется реализация управляющих связей между агрегатами.

В случае *имитации активностями, событиями и процессами* исследователю приходится уточнять схемы управляющих связей между компонентами модели, которые им были ранее составлены на этапе создания концептуальной модели. Как правило, необходимость в уточнении схемы управляющих связей возникает из-за того, что на этапе формализации выясняется необходимость размножения компонент или разделения отдельных функциональных действий $ФД_{ij}$ на несколько более мелких. Как следствие, появляются новые компоненты модели. В тех местах, где имеет место взаимодействие одного процесса с другим, на схеме управляющих связей устанавливаются «сигнальные линии» (ребра ориентированного графа). Начало такой «сигнальной линии» находится в конце той активности, которая формирует воздействие на другую активность. Конец «сигнальной линии», обозначаемый стрелкой, входит в начало алгоритма выполнения соответствующей компоненты модели (либо активности, либо подпрограммы обслуживания событий, либо процесса).

При *транзактно-процессном способе* имитации роль схемы управляющих связей между компонентами весьма существенна. Граф связей представляет собой блок-схему передачи динамических элементов от одного статического элемента в очередь к другому статическому элементу ИМ. Условиями этого информационного воздействия обычно являются приоритеты выбора транзактов из очередей. В качестве *сигнальных линий* используются специальные процессы. Причем аналогом таких линий являются активности процессов, осуществляющих посылку транзактов сложной структуры в очередь к процессу, из которой они выбираются, той активностью, которая и является фактическим концом сигнальной линии. Таким способом имитируется заранее не прогнозируемое время задержки в сигнальной линии.

При *агрегатно-процессном способе* имитации граф связей агрегатов обычно выполняет функции установления управляющих связей между агрегатами с помощью соответствующих сигналов.

Уточнение взаимодействия с УПМ. Далее исследователь обязан уточнить вопросы взаимодействия компонент модели с УПМ. В соответствии с выбранным способом имитации устанавливается свой набор операторов взаимодействия компонент модели с УПМ. Уточняется структура операторов, с помощью которых организуются возврат на УПМ и модификация временной координаты компонент модели системы. Для агрегатного и транзактного способов имитации этот шаг формализации необязателен, поскольку структура взаимодействий элементов модели с УПМ унифицирована.

Документация этапа. В результате уточнения концептуальной модели исследователь получает строгое описание алгоритмов $АЛ_{ij}$ на выбранном способе формализации и операторов взаимодействия активностей как с УПМ, так и друг с другом. По завершении этапа формализации предоставляется следующая документация:

- временная диаграмма функционирования элементов модели;
- схема функционального взаимодействия компонент друг с другом и с УПМ;
- схема информационного взаимодействия между компонентами;
- описание текста алгоритмов $АЛ_{ij}$ на языке формализации.

Для агрегатного способа имитации требуются еще матрица коммутаций агрегатов друг с другом и описание начального состояния моделируемой системы. При транзактном способе необходимо также описание источников и поглотителей транзактов.

Результатом работ является формальное описание сложной системы, свободное от второстепенной информации (имеющейся в содержательном описании) и устанавливающее структуру алгоритмического представления объекта моделирования. Может оказаться, что информации, имеющейся в содержательном описании, недостаточно для формализации объекта моделирования. В этом случае следует вернуться к этапу составления содержательного описания и дополнить его данными, необходимость в которых обнаруживается при формализации объектов моделирования. На практике таких возвратов может быть несколько.

4.2 Формализация вычислительного процесса в ЛВС аппаратом полумарковских процессов

Рассмотрим формализацию ВП и РН ЛВС на двух уровнях детализации. Для первых двух уровней детализации («топология ЛВС» и «узел ЛВС») используем транзактно-процессный способ имитации (со сложной структурой транзакта).

Для отображения алгоритмов взаимодействия компонентов ВП и РН используем аппарат полумарковских процессов [7]. Графовая структура ВП на первых двух УД позволяет использовать полумарковские модели ВП: вначале рассматривается ВП всей ЛВС (ПММ₁), а затем рассматривается реализация каждого состояния ВП в ЛВС (узла ПММ₁). В свою очередь, первая представляет собой полумарковскую модель более высокого уровня представления использования ресурсов ЛВС (ПММ₂), в которой состояниями являются выполнения МТХО_{ij}. Для третьего уровня детализации ВП и РН на ЛВС («Задачи ЛВС») формализуется динамика расхода ресурсов сервера ЛВС запросами пользователей, которая также использует транзактно-процессный способ имитации. Согласно содержательному описанию (СОП) узла ЛВС не выполняются предпосылки для представления ВП в виде полумарковской модели. Поэтому для описания ВП и РН используем аппарат прямой имитации РН с помощью временных диаграмм использования ресурсов j -й функциональной задачи ΦZ_j в виде транзакта сложной структуры (STCC), а сами компоненты ВП представим в виде приборов систем массового обслуживания с управляющими входами [19].

Формализация ВП уровней детализации «топология ЛВС» и «узла ЛВС». На основании содержательного описания ВП, приведенного в 2.1, и концептуальной модели, представленной в 3.1, используем методику формализации СС, представленную в 4.1, для построения формального описания первых двух уровней детализации ВП в ЛВС. Предполагаем, что выполняются предпосылки для построения ПММ₁ и ПММ₂. Используем способ описания СС аппаратом полумарковских процессов. Считаем, что на вход k -й узла ЛВС подается последовательность технологических операций $\{ТХО_{ij}\}$, используемых программно-технологическое обеспечение (ПТО _{k}) и ресурсы узлов ЛВС. Технология реализации любой ТХО _{ki} на ресурсах ЛВС определяется графом $\{GR2_{ik}\}$. Все ТХО _{ki} поступают в k -й узел ЛВС от i -го пользователя с интенсивностью (λ_{ki}). Пользователи i -го типа реагируют на результаты вычислений на k -м узле ЛВС с временной задержкой $\{\tau_{OBi}\}$. При этом известны также вероятности использования i -й пользователем ресурсов ЛВС (q_{1i}). От ПММ₁ на вход ПММ_{2 k} , отображающих ВП в узел k , поступают запросы функциональных задач $\{\Phi Z_k\}$ на ресурсы ЛВС. Состав вектора РН $ХРН_2$ включает: вероятности поступления функциональных задач для их выполнения в узле ЛВС $\{q_{2i}\}$; интенсивности их поступления $\{\lambda_{2i}\}$; средние времена реакции компонентов узла ЛВС на выполнение МТХО _{lki} (τ_{REli}). В таблице 4.1 приведен состав параметров РН на уровне «топология ЛВС». Поскольку

изначально был определен полумарковский характер процесса преобразования $\{MTXO_{lki}\}$ в последовательность $\{MXO_{ki}\}$, то возможно произвести свертку $\{TXO_{ki}\}$ в запрос ZP_i . Аналогичным образом определяем параметры и отклики ВП на уровне детализации «узел ЛВС». Результаты подобной формализации приведены в таблице 4.2.

Таблица 4.1 – Состав параметров и откликов ВП уровня детализации «ТОПОЛОГИЯ ЛВС»

Название вектора	Обозначение компонента	Назначение компонента вектора
1	2	3
X(TXH ₁)	K, l	Общее количество узлов и порядковые номера в графе топологий G_{r1}
	$\ NET\ _{ij}$	Матрица связей узлов сети
Структуры ВП (G _{TXH1})	v_{SYSPk}	Скорость выполнения запросов на SVP в k -м узле ЛВС
	v_{SPDK}	Скорость выполнения запросов в СПД ЛВС
	q_{rs}, C_{rs}	Пропускная способность и стоимость использования канала
	τ_{OBDi}	Среднее время обдумывания i -го запроса в ЛВС
	V_{TPKi}	Средний объем транзитного сообщения через узел k i
	V_{TPOKi}	Средний объем транзитного ответа ЛВС i -му запросу
Надежности функционирования Z_l	P_{OTY}	Вероятность отказа оборудования узла ЛВС
	P_{OTCB}	Вероятность отказа канала связи
	P_{BOCY}	Вероятность восстановления работы узла ЛВС за время Δt
	P_{BOCCB}	Вероятность восстановления канала связи
	τ_{BCY}	Среднее время восстановления узла
	τ_{BCK}	Среднее время восстановления канала связи
Ограничений $\{G_l\}$	C_{li}	Стоимость услуг l -го узла i -му запросу
	G_k	Пропускная способность k -го канала связи
	T_k	Задержка запросов в k -м канале
	T_m	Максимально доступное время задержки канала

Окончание таблицы 4.1

1	2	3
Статистик $\{ST_l\}$	$\Sigma \tau_{svsp}$	Общее время использования SVSP
	$\Sigma \tau_i$	Суммарное время прохождения через ЛВС запросов i -го типа
	l_{oul}	Средняя длина очереди запросов к l -му узлу ЛВС
	t_{oul}	Среднее время ожиданий запросов в узлах l -го номера
Состояний $\{SOS_l\}$	$v_{Обt0}$	Число узлов ЛВС, занятых обслуживанием в моменты t_0
	v_{CBt0}	Число свободных линий связи в моменты t_0
	$v_{Отt0}$	Число отказавших линий связи на момент t_0
Откликов $\{Y_l\}$	η_{svspil}	Коэффициент занятости l -го узла обслуживанием запросов i -го типа
	$\eta_{ТPl}$	Коэффициент занятости l -го узла транзитными запросами
	T_{ki}	Среднее время жизни в ЛВС запросов i -го типа
	KN_i	Количество промежуточных узлов на трафике в модели для запросов

Таблица 4.2 – Состав параметров и откликов ВП уровня детализации «УЗЕЛ ЛВС»

Название вектора или множества	Обозначение параметров	Название параметра, отклика и статистик ВП ЛВС
1	2	3
Управление ВП ($X_{ТХН2}$)	v_{SYSk}	Скорость выполнения запросов SUS в k -м узле ЛВС
	v_{disk}	Скорость диспетчера k -го узла ЛВС
Структуры ВП $G_{ТХН2}$	T_{un}	Тип узла ЛВС (рабочая станция или сервис)
	v_{SVSO}	Исходная скорость SUS в k -м узле ЛВС
	$\tau_{Обpi}$	Время обработки i -й ФЗ на узле ЛВС
	$V_{Трki}$	Объем событий транзитного сообщения i -го типа через узел k

Окончание таблицы 4.2

1	2	3
Надежности функционирования компонентов (Z_2)	Z_1	Вектор надежности оборудования узла ЛВС
	$P_{\text{ОТВЗ}}$	Вероятность отказа диспетчером отправки задач
	$P_{\text{ВОСДЗ}}$	Вероятность восстановления работоспособности диспетчера задач
	$P_{\text{ОТПЗ}}$	Вероятность отказа диспетчера приема задач
	$P_{\text{ВОСПЗ}}$	Вероятность восстановления диспетчера приема задач
	$\tau_{\text{ВОСДЗ}}$	Среднее время восстановления диспетчера отправки задач
	$\tau_{\text{ВОСПЗ}}$	Среднее время восстановления диспетчера приема задач
Ограничений (G_z)	C_2	Стоимость услуг при выполнении $MTXO_i$
	G_{2k}	Пропускная способность k -го канала при передаче результатов в сеть
	T_{2k}	Задержка передачи информации в k -м канале сети
	T_m	Максимально доступное время задержки результатов в канале
Статистик $\{ST_2\}$	$\Sigma \tau_{\text{SVSi}}$	Общее время использования SUS запросом i -го типа
	$l_{\text{ОЧДЗ}}$	Длина очереди ΦZ_m к диспетчеру задач
	$l_{\text{ОЧПЗ}}$	Длина очереди ΦZ_m к диспетчеру приема задач
Состояний $\{SOS\}$	i	Номер решаемой ΦZ_m
	$l_{\text{ОЧ}j}$	Длина очереди ΦZ_m к j -му узлу GR_{2i}
Откликов $\{Y_2\}$	KN_i	Количество промежуточных узлов ЛВС на графике i -го типа
	$t_{\text{ОБС}j}$	Среднее время выполнения ΦZ_m j -го типа
	$V_{\text{HDD}i}$	Максимально использованный объем ресурса HDD в ΦZ_m номера j
	η_{SVS}	Суммарный коэффициент занятости SUS в узел ЛВС

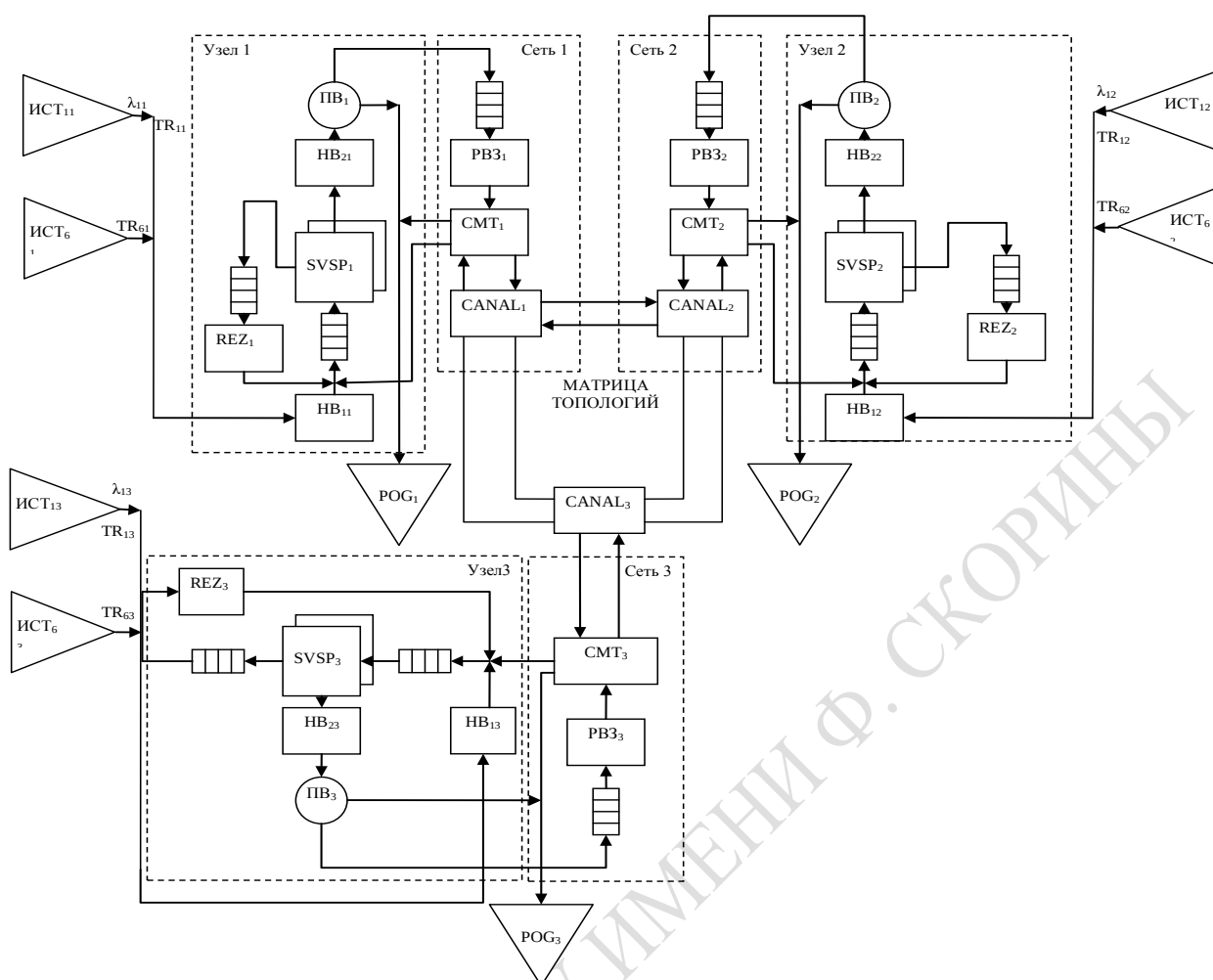


Рисунок 4.1 – Блок-схема алгоритма ВП уровня детализации
«ТОПОЛОГИЯ ЛВС»

На рисунке 4.1 приведена блок-схема алгоритма ВП уровня детализации «топология ЛВС». На входе ВП ЛВС отображаются запросы пользователями узлов ЛВС, которые мы представим в виде источников транзактов (TR_{ik}) сложной структуры, которые мы обозначили на рисунке 4.1 в виде $УСТ_{ik}$ ($i = \overline{1,6}$ – номера пользователей, $k = \overline{1,3}$ – номера узлов ЛВС) двух типов: синхронные (TR_k) и асинхронные (TR_k). Асинхронные источники (нечетные номера) формируют новые транзакты с интенсивностями λ_{ik} , а синхронные транзакты формируют новый транзакт только после получения ответа от ЛВС на предыдущий запрос на ресурсы узла ЛВС. Различаем следующие типы транзактов: локальный диалог (TR_{1k}), формируемый источником $УСТ_{1k}$; короткий сетевой диалог (TR_{1k}), поступающий от $УСТ_{4k}$; продолжительный сетевой диалог с сопоставлением ответа ЛВС (TR_{6k}), формируемый источниками $УСТ_{6k}$; фоновые запросы, выполняемые на узле (TR_{1k}); запросы к удаленному узлу ЛВС без возврата результатов (TR_{3k}), поступающие от $УСТ_{3k}$; заказ на удаленный счет (TR_{6k}),

формируемый UCT_{5k} . На блок-схеме алгоритма ВП (рисунок 4.1) изображены связи компонентов ВП ЛВС, состоящей из трех узлов. Такими компонентами ЛВС являются: узел k , где расположены ресурсы k -го узла, сетевые средства связи узлов друг с другом (SET_k), асинхронные источники 1-го типа (UCT_{ik}) для $i = 1, 3, 5$ и синхронные источники запросов 2-го типа (UCT_{ik}) для $i = 2, 4, 6$.

Связь между узлами осуществляются с помощью устройств $CANA_k$ согласно матрице топологии ЛВС. На рисунке 4.1 отображены составляющие компонентов ВП ЛВС. Назначение компонентов следующее.

$SVSP_k$ – блок ресурсов k -го узла ЛВС, состоящий из частей выделения ресурсов CPU для: обслуживания транзактных запросов (ihnet, фоновых задач, диалогов всех типов, удаленного счета). Каждая часть CPU_k обеспечивает и обслуживает запросы определенного типа (т. е. транзактов TR_{ik}).

REZ_k – алгоритм, отображающий длительный сетевой диалог пользователей, обеспечивая при этом хранение ответа на запрос до получения результатов и помещая TR_{ik} в очередь нужного узла ЛВС.

$СМТ_k$ – алгоритм коммутатора k -го узла ЛВС, отображающий контроль отправки запросов в канал связи и прием запросов или же ответов от других узлов ЛВС. Кроме того, он организует обработку и передачу транзитных запросов через k -й узел ЛВС.

$CANAL_k$ – алгоритм сетевого диспетчера, осуществляющий передачу TR_{ik} на другие узлы ЛВС согласно матрице топологии.

$НВ_{ik}$ – та часть алгоритма диспетчера узла ЛВС, которая анализирует содержимое TR_{ik} и осуществляет их постановку в очередь запросов на ресурсы узла ЛВС.

$НВ_{2k}$ – та часть алгоритма диспетчера узла ЛВС, которая с помощью $ПВ_k$ отбирает транзакт из общего потока или S_0 .

TR_{ik} – обслуживает транзитные запросы и посылает их в очередь к $РЕЗ_k$, либо требует выделения внутренних ресурсов, отсылая TR_{ik} в очередь к $SVSP_k$, либо посылает TR_{ik} (нечетных номеров типов i узлов ЛВС) на поглотитель транзактов (POG_k) по окончании сеанса взаимодействия пользователя с ЛВС.

Возможны следующие маршруты движения TR_{ik} в ВП уровня детализации «топология ЛВС»:

- для TR_{1k} (фоновые заказы ресурсов k -го узла), когда имеет место последовательность 1-го типа ($UCT_{1k}, SVSP_{1k}, POG_k$);
- для TR_{2k} (локальный диалог), когда последовательно используется тройка компонентов ВП ($UCT_{2k}, SVSP_k, POG_k$);
- для TR_{3k} (посылка сообщений удаленному пользователю без возврата результатов) и для TR_{5k} (заказ на удаленный счет), когда

имеет место последовательность 3-го типа (UCT_{3k} , $SYSP_k$, CMT_k , $CANAL_k$, CMT_{k+1} , $SVSP_{k+1}$, $CANAL_{k+1}$, CMT_{k+1} , $SVSP_{k+1}$, POG_{k+1});

– для TR_{4r} (короткий диалог пользователя узла k с пользователем $k+1$ узла), когда имеет место последовательность 4-го типа (UCT_{4k} , $SYSP_k$, CMT_k , $CANAL_k$, $CAWAL_{k+1}$, CMT_{kH} , $SVSP_{k+1}$, $CANAL_{k+1}$, CMT_{k+1} , $CANAL_k$, CMT_k , $SVSP_k$, POG_1);

– для TR_{6k} (длительный диалог) имеет место последовательность 6-го типа (UCT_{6k} , $SYSP_k$, CMT_k , $CANAL_k$, $CANAL_{k+1}$, CMT_{k+1} , $SVSP_{k+1}$, $CANAL_k$, CMT_k , цикл ($SVSP_k$, REZ_k , $CANAL_k$, $CANAL_{k+1}$, CMT_{k+1} , цикл ($SPSP_{k+1}$, RES_{k+1}), $CANAL_{k+1}$, $CANAL_k$, CMT_k , $SPSP_k$, POG_k).

На рисунке 4.2 приведена блок-схема алгоритма ВП уровня детализации «узел ЛВС». На данном УД вычислительного процесса необходимо представить динамику использования ресурса CPU k -го узла. Для этой цели используются приборы СМО с управляющими входами «открыть» (0) и «закрыть» (3). При отображении расхода ресурса CPU _{k} в любой момент выполняется только тот прибор СМО, который подготовлен для его выполнения сигналом, поступившим на вход «открыть» (0). Если же поступил сигнал «закрыть» на вход (3), то прибор СМО не готов для отображения расхода ресурса CPU.

На рисунке 4.2 показаны следующие компоненты алгоритма ВП уровня детализации «узел ЛВС».

ДИСП – отображает работу той части алгоритма диспетчера ЛВС, которая обеспечивает выделение ресурса CPU _{k} и организацию последовательно обслуживания очереди TR_{ik} к CPU _{k} .

SYSP – представляет собой группу алгоритмов ВП использования CPU _{k} и состоит из следующих частей расхода ресурса CPU _{k} диалоговыми запросами внутреннего использования или запросами удаленного доступа к файловой системе узла k .

SERV – часть алгоритма ОС, которая отвечает за предоставление ресурсов CPU _{k} и услуг в общее пользование (обрабатываются диалоговые запросы и запросы удаленного пользования).

USP – часть алгоритма ОС, отвечающая за обработку локального пользовательского запроса или местного диалога.

LRS – часть алгоритма ОС, обрабатывающая фоновые, сетевые и транзитные запросы, которые используют ресурсы узла ЛВС.

При формализации ВП и РН на уровне детализации «топология ЛВС» использован следующий технологический прием. Отображение совместного использования одного и того же ресурса CPU запросами различной природы. С помощью прибора ДИСП ресурс CPU предоставляется прибором SERV, URS и LRS, отображающим различные стороны организации ВП в узле ЛВС. С помощью системы управляющих

сигналов «открыть» и «закрыть» обслуживание TR_i обеспечивается чередование выполнения приборов ДИСП и одного из приборов SERV, URS и LRS.

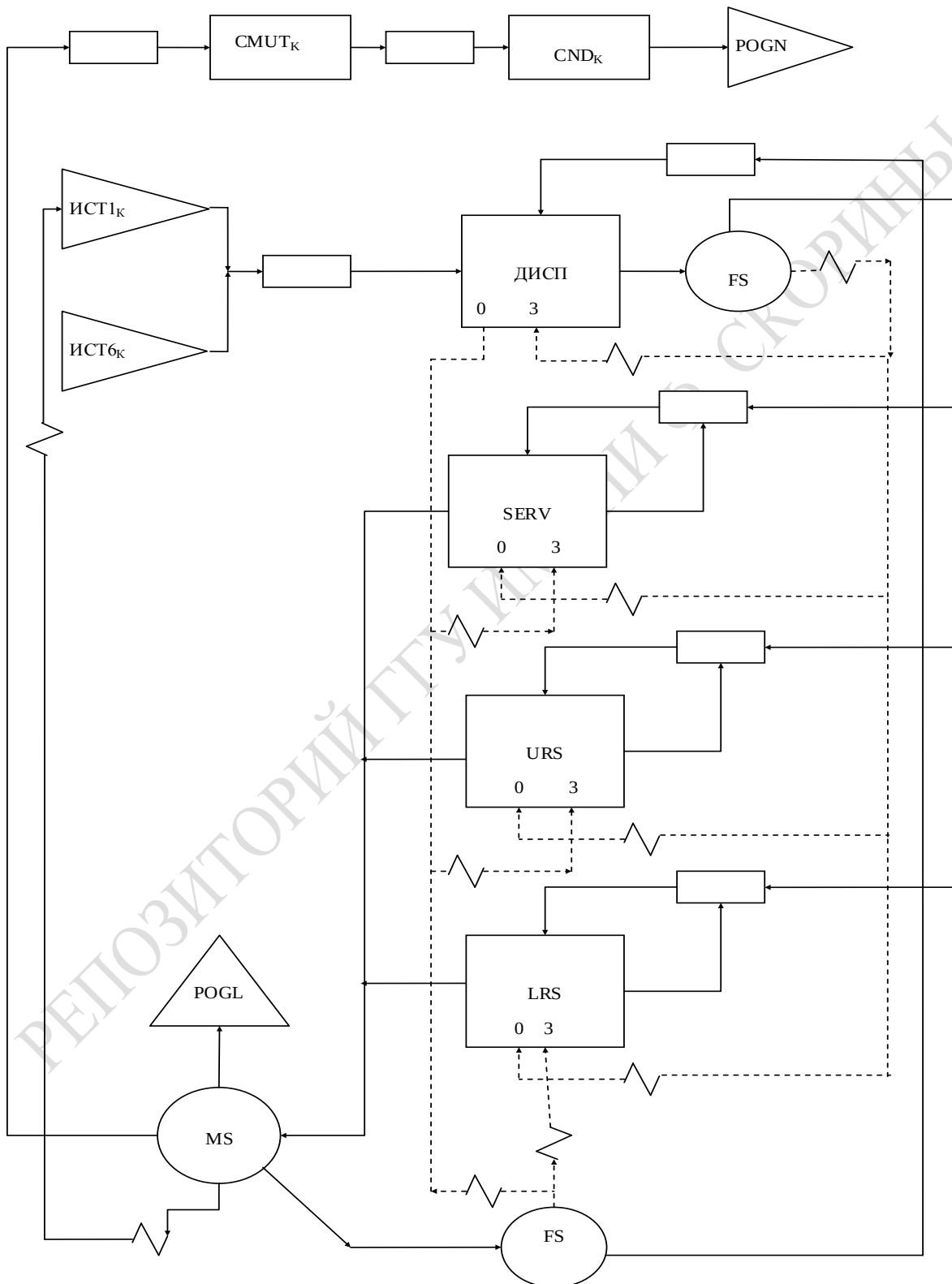


Рисунок 4.2 – Блок-схема алгоритма ВП уровня детализации «узел ЛВС»

Траектория движения TR_i следующая: UCT_i , ДИСП (SERV или LRS), MR, ДИСП, (SERV или URS или LRS), MP и т. д.

После MR транзакт направляется по одному из направлений: ДИСП (внутренние заказы); $СМТ_1$ (обработка запросов на других узлах ЛВС); POCL (окончание выполнения заказа пользователя на ресурсы ЛВС).

Как видно из рисунка 4.2 в любой момент времени только одно из устройств – ДИСП, SERV, URS и LRS – может выполняться на одном и том же $СРU_k$. Другими компонентами алгоритма ВП уровня детализации «узел ЛВС» являются (рисунок 4.2):

СМУТ – алгоритм отображения внешнего окружения узла ЛВС (по своей сути он является имитатором коммутатора узла ЛВС).

СПД – отображает алгоритм выполнения остальных узлов ЛВС.

$РОВL_k$ – поглотитель транзактов узлом k в момент окончания обслуживания ЛВС запросов.

UCT_{ik} – источник транзактов, имитирующий запросы i -го типа в виде транзактов сложной структуры.

Классификация источников UCT_{ki} и транзактов TR_{ik} совпадает с классификацией источников для ВП уровня детализации «топология ЛВС».

РН уровня «топология ЛВС» представляется следующим образом на узле k : источники UCT_{ik} первого типа ($i = 1, 3, 5$) формируют с интенсивностью λ_{ik} TR_{ik} сложной структуры, обладающих «телом» вида:

$$TR_{ik} = (i, \pi_i, \{q_{ik}\}; \{\alpha_{ик}\});$$

где i, π_i – тип и приоритет транзакта;

$\{q_{ik}\}$ – вектор вероятностей поступления i -го типа запросов $МТХO_{ij}$ на их выполнение в k -м узле последовательностью $\{МТХO_{ij}\}$;

$\{\alpha_{ик}\}$ – вектор адресов нахождения в БД информации о структуре РН.

По адресу $\alpha_{ик}$ указывается следующая информация о структуре графа G_{r1} заказов i -го курса и их реализации на k -м узле ЛВС:

P_{olk} – вектор вероятностей начала полумарковского процесса с l -го состояния.

$MP_{ilk} = \|P_{ilj}\|$ – матрица вероятностей перехода $ПММ_l$, представленной графом G_{r1} , из l состояния в состояние j для i -го запроса ресурса $СРU_k$;

$F_{1ik}(v)$ – функция распределения числа переходов (v_1) i -го запроса в $ПММ_1$ на k -м узле ЛВС из состояния в состояние;

$MF_{2lj}(\tau)$ – матрица условных распределений времен использования $СРU_k$ (при условии перехода $ПММ$ из состояния l в состояние j).

Для UCT_{ik} 2-го типа ($i = 2, 4, 6$) по сигналу возврата от ЛВС имитируется реакция пользователей через интервалы τ_{R1ik} . При этом формируется TR_{ik} , обладающий «телом» вида (4.1).

РН уровня «узел ЛВС» представляется следующим образом. На узле k источники UCT_{ik} 1-го типа ($i = 1, 3, 5$) формируют с интенсивностью λ_{ik} TR_{ik} сложной структуры вида:

$$TR_{ik} = (i, \pi_i, \{q'_{ik}\}, \{\beta_{ik}\}), \quad (4.1)$$

где i, π_i – тип и приоритет транзакта;

$\{q'_{ik}\}$ – вектор вероятностей поступления функциональных задач $\{\Phi Z_k\}$ на их выполнение в k -м узле ЛВС последовательностью $\{MTXO_{lk}\}$;

$\{\beta_{ik}\}$ – вектор адресов нахождения в базе данных информации о структуре ΦZ_k .

По адресу α_{ik} указываются информация о структуре графа G_{r2} заказов ресурсов k -го узла ЛВС:

P_{1lk} – вектор вероятностей начала полумарковского процесса, представленного графом G_{r2} , с l -го состояния на k -м узле ЛВС;

$MP_{2lk} = \|P'_{ij}\|$ – матрица вероятностей перехода ПММ₂, представленной графом G_{z2} , из состояния l в состояние j для i -го запроса ресурса CPU_k ;

$MF_{3ik}(v)$ – функция распределения числа переходов (v_2) i -го запроса в ПММ₂ на k -м узле ЛВС из состояния в состояние;

$MF_{4lj}(\tau)$ – матрица условных распределений времен использования CPU_k (при условии перехода ПММ из состояния l в состояние j).

Для UCT_{ik} 2-го типа ($i = 2, 4, 6$) по сигналу возврата ЛВС имитируется реакция пользователей через интервала τ_{R2ik} .

При этом формируется TR_{ik} , обладающий «телом» вида (4.1).

Приборы REZ_k, PBZ_k на уровне топологии ЛВС представляют собой обычные устройства СМО с дисциплиной FIFO выбора TR_{ik} из очередей. На приборе HV_{1k} также осуществляется выбор TR_{ik} в соответствии с их приоритетами (π_{ik}). На приборе HV_{2k} осуществляется вероятностный переход TR_{ik} либо на прибор $CETh_k$, либо на поглотитель POG_k .

Прибор $CMTh_k$ является коммутатором TR_{ik} либо в очередь к прибору $SVSP_k$, либо на прибор $CANAL_k$ в случае выполнения запроса в сети согласно матрице топологии (направляется TR_{ik} на другой узел ЛВС). Алгоритмы перечисленных приборов достаточно просты и их реализация зависит от языка выбранной системы моделирования. Поэтому ниже мы только рассмотрим особенности алгоритмов

прибора $SVSP_k$, блок-схема которого представлена на рисунке 4.3. Прибор $SVSP_k$ реализует алгоритм $ПММ_1$.

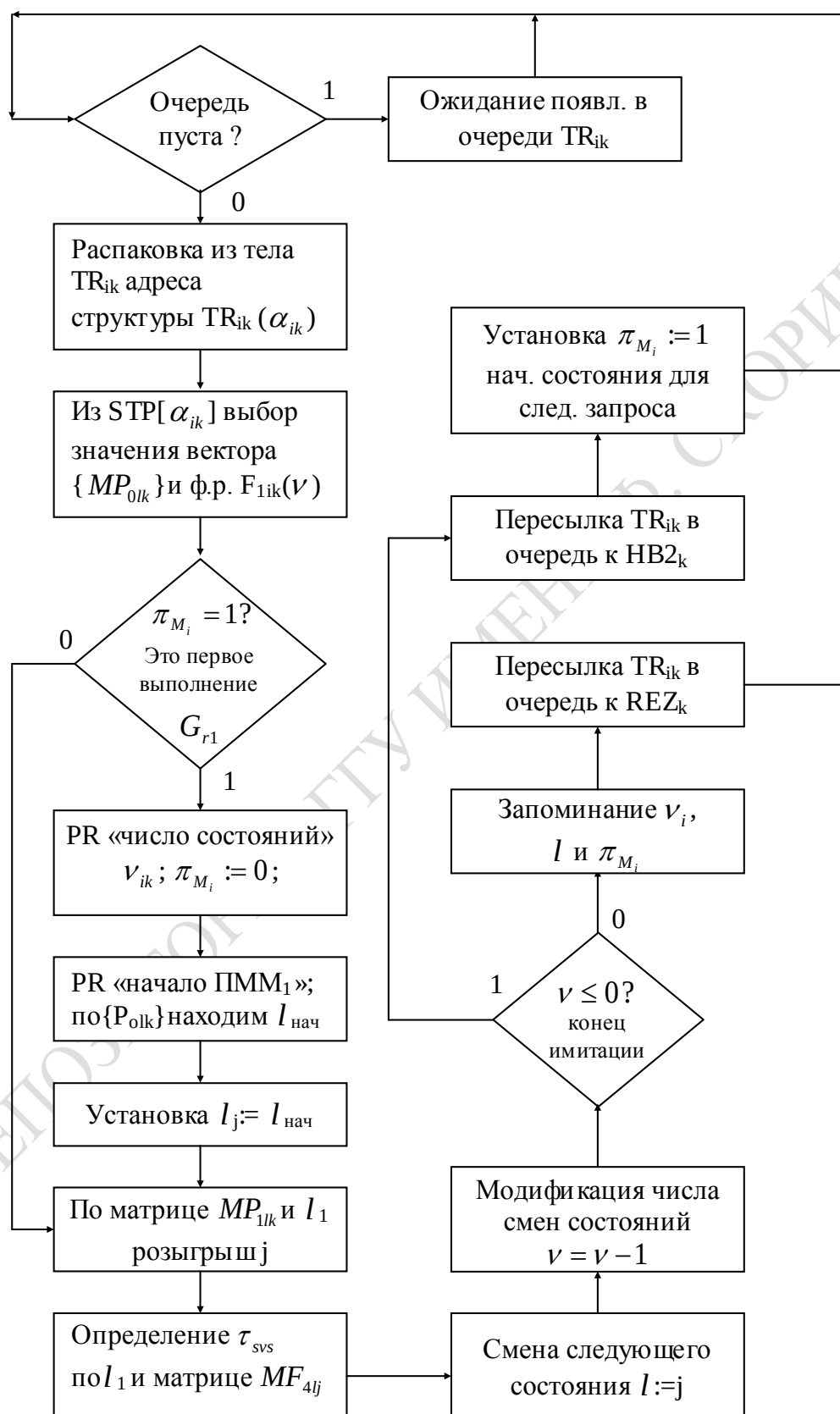


Рисунок 4.3 – Алгоритм реализации графа G_{r1}

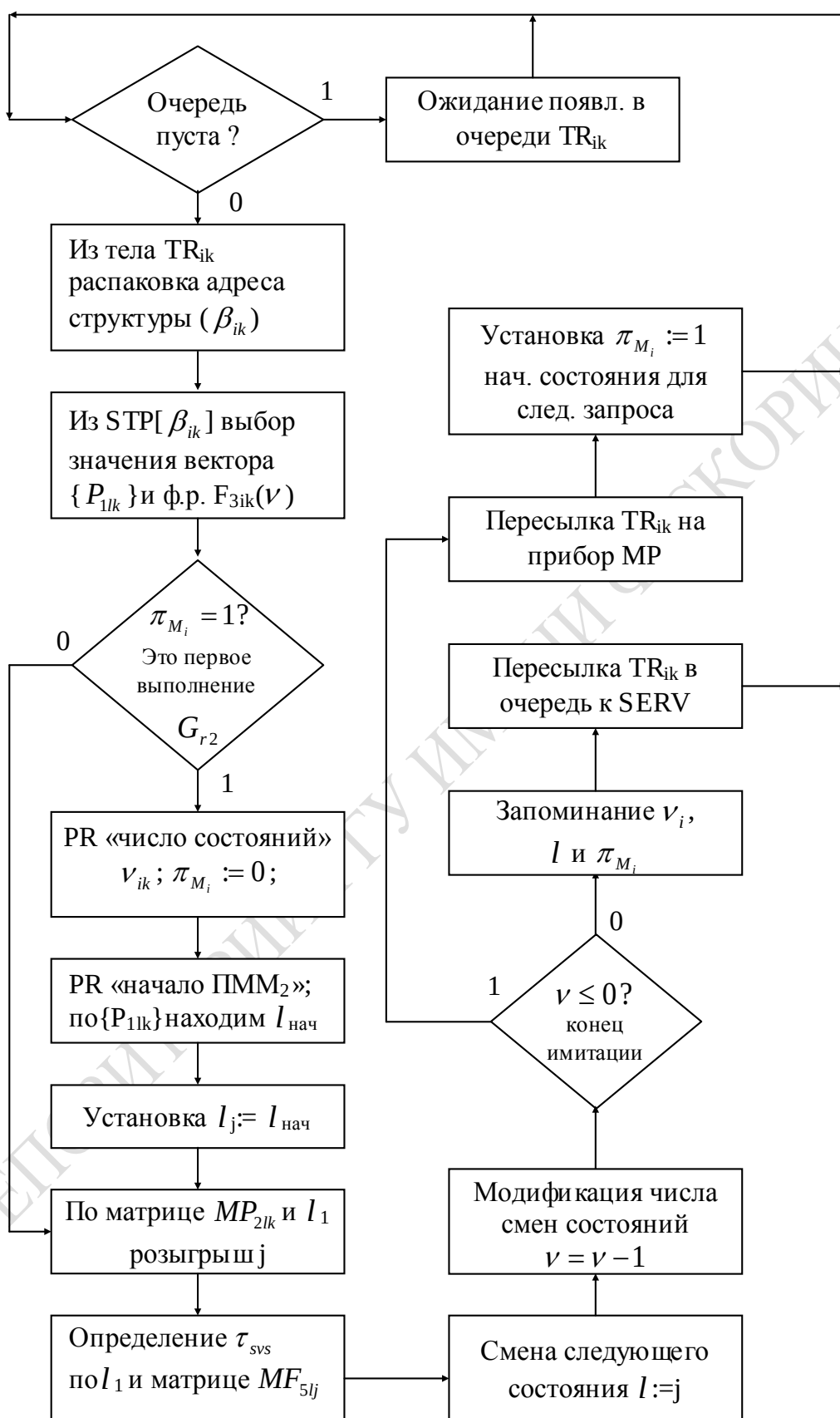


Рисунок 4.4 – Алгоритм реализации графа G_{r2}

Вначале проверяется условие «очередь» TR_{ik} к прибору $SVSP_k$ пуста. Если транзактов TR_{ik} нет в очереди, то необходимо отразить ожидание поступления транзакта. Осуществляется распаковка «тела» TR_{ik} и определяется адрес (α_{ik}) структуры G_{r1} в базе данных. Из нее извлекается функция распределения $F_{1ik}(v)$ числа квантов, разыгрывается его конкретное значение (v) и определяется вектор вероятностей начального состояния ПММ₁ $\{P_{olk}\}$. В случае первичного выполнения G_{z1} определяется начальное состояние l . Далее по матрице вероятностей перехода из состояния в состояние MP_{1lk} и предыдущему состоянию l определяется номер следующего состояния j . По номерам i и j и матрице условных распределений $MF_{4lj}(\tau)$ разыгрывается длительность (τ_{iRk}) использования $SVPK_k$, транзактом TR_{ik} осуществляется подготовка к следующему состоянию G_{r1} , модифицируется число состояний ($v := v - 1$) и проверяется условие окончания имитации по графу G_{r1} . Если условие не выполнено, то TR_{ik} пересылается в очередь к прибору REZ_k . При окончании имитации TR_{ik} с помощью G_{r1} он пересылается в очередь к прибору HB_{2k} .

Блок-схема алгоритма реализации графа G_{21} приведена на рисунке 4.4. Приборы $CMUT_k$ и $СПД_k$ на уровне «узел ЛВС» также представляют собой обычные СМО с дисциплиной FIFO выбора TR_{ik} из очередей к этим устройствам. На приборе ДИСП осуществляется выбор TR_{ik} в соответствии с их приоритетами (π_{ik}). На приборе FS осуществляется пересылка TR_{ik} в одну из очередей к приборам $SERV$, URS и LRS . Прибор ВИСП с интервалом времени Δt_1 пересылает транзакт на прибор FS. Алгоритм выполнения приборов $SERV$, URS и LRS представляет собой реализацию графа G_{r2} . Различаются они только значением характеристик G_{r2} для каждого типа транзактов (i), формируемых источниками UCT_{ik} . Поэтому ниже мы только рассмотрим алгоритм реализации ПММ₂ графом G_{r2} , блок-схема которого представлена на рисунке 4.4. Вначале проверяется условие очередь TR_{ik} к прибору $SERV_k$ пуста. Если транзактов TR_{ik} нет в очереди, то необходимо отразить ожидание его поступления. Осуществляется распаковка «тела» TR_{ik} и определяется адрес (β_{ik}) структуры G_{r2} в базе данных. Из нее извлекаются функции распределения $F_{3ik}(v)$ числа квантов и разыгрывается конкретное его значение v и определяется вектор вероятностей начального состояния ПММ₂ $\{P_{1lk}\}$. В случае первичного выполнения G_{r2} определяется начальное состояние l . Далее по матрице вероятностей перехода из состояния в состояние MP_{1lk} и предыдущему состоянию определяется номер следующего состояния j . По номерам i и j и матрице условных распределений $MF_{5lj}(\tau)$ разыгрывается длительность (τ_{ilk}) использования CPU прибором

$SERV_k$ при обслуживании TR_{ik} . Наконец, осуществляется подготовка к следующему состоянию G_{r2} , моделируется число смен состояний и проверяется условие окончания имитации по графу G_{r2} . Если условие не выполнено, то TR_{il} снова пересылается к прибору $SERV_k$. В противном случае TR_{ik} пересылается в очередь к прибору MR. Этот прибор пересылает в одну из трех очередей: если i означает заказы на ресурсы других узлов ЛВС, то TR_{ij} поступает в очередь к прибору $CMUT_k$. Во втором случае TR_{ik} через прибор FS поступает в очередь к прибору ДИСП_k. В функции прибора FS входит формирование из TR_{ik} сигнала «открыть» ДИСП_k и «закрыть» все остальные приборы ($SEKV_k$, URS_k и LRS_k).

При окончании обслуживания TR_{ik} на приборе ДИСП_k с помощью прибора формируется сигнал закрытия прибора ДИСП_k и открытия того из приборов ($SERW_k$, URS_k и LRS_k), в очередь выполнения которого поступает TR_{ik} для имитации расхода ресурсов CPU_k.

Отметим, что часть источников UCT_{ik} транзактов TR_{ik} для алгоритмов ВП обоих уровней детализации одинаковы. Как видим, формальное описание ВП в ЛВС первых двух уровней детализации по размерам существенно больше чем содержательное описание ВП в ЛВС. Оно включает в себя: таблицы 4.1 и 4.2; блок-схемы алгоритма ВП, представленные на рисунке 4.1 и рисунке 4.2; блок-схемы реализации G_{r1} и G_{r2} , приведенные на рисунке 4.3 и рисунке 4.4 и изложенный выше текст, сопровождающий описание блок-схем на этих рисунках.

4.3. Формализация ВП в узле ЛВС с помощью информационной подкраски запросов ресурсов пользователей

Поскольку объектом исследования является ВП на узле ЛВС, являющимся сервером, то динамику использования ресурсов запросами пользователей трудно представить с помощью полумарковской модели. Поэтому поток запросов пользователей на ресурсы F сервер ЛВС в режимах DR и DIS представим с помощью соответствующих временных диаграмм (ВД DR_j и ВД DIS_j) использования CPU_j и HDD_j (рисунок 4.5).

Формализация РН для адаптации к существующему составу ресурсов узла ЛВС. С помощью ВД DR_j определим следующие параметры имитации режимов DR и DIS:

- времена обслуживания сервером k -х запросов пользователей (τ_{Djk});
- интервалы использования CPU_{Lj} и HDD_{Lj} соответственно (Δt_{11jk} и (Dt_{12jk}) во время k -го цикла обслуживания запросов режима DR;
- количество использований $CPU_{1j}(n_{11j})$ и $HDD_{1j}(n_{12j})$ в режиме DR;
- интенсивность поступления запросов пользователей режима DIS(λT_j) на сервер;
- интервалы использования CPU_{2j} и HDD_{2j} соответственно (Δt_{21jk} и (Dt_{22jk}) во время обслуживания запросов режима DIS;
- количество использований $CPU_{2j}(n_{21j})$ и $HDD_{2j}(n_{22j})$ и в режиме DIS.

Интервалы времени и количество использования CPU_j и HDD_j , а также длительности цикла использования сервера в режимах DR и DIS (Δt_{11jk} и (Dt_{12jk}) являются случайными величинами. Поэтому динамику использования ресурсов сервера ЛВС представим в виде:

$F_{11j}(\Delta t) - F_{12j}(Dt)$ – функций распределения длительности использования соответственно CPU_j и HDD_j в режиме DR;

$F_{21j}(\Delta t) - F_{22j}(Dt)$ – функций распределения длительности использования CPU_j и HDD_j в режиме DIS;

$\bar{\tau}_{Dj}$ – среднее значение цикла взаимодействия пользователя в диалоговом режиме;

$\lambda_{1j} = 1 / \bar{\tau}_{Dj}$ – интенсивность поступления диалоговых запросов пользователей (предполагаем, что запросы на ресурсы поступают через равные интервалы времени) в режиме DR;

$p_{dj} = \frac{n_{12j}}{n_{11j}}$ – вероятность использования HDD_j в цикле взаимодействия диалогового запроса пользователей;

$\bar{\tau}_{Tj}$ – среднее значение цикла выполнения транзитных запросов;

$\lambda_{2j} = 1 / \bar{\tau}_{Tj}$ – интенсивность поступления транзактных запросов пользователей в режиме DIS;

$p_{Tj} = \frac{n_{22j}}{n_{21j}}$ – вероятность использования HDD_j в цикле взаимодействия транзактного запроса пользователем.

Таким образом, формальное описание режимов DR и DIS имеет вид:

$$\{\lambda_{Dj}; n_{11j}; P_{Dj}; F_{11j}(\Delta t); F_{12j}(\Delta t)\} - \text{для режима DR}; \quad (4.2)$$

$$\{\lambda_{Tj}; n_{22j}; P_{Tj}; F_{21j}(\Delta t); F_{22j}(\Delta t)\} - \text{для режима DIS}. \quad (4.3)$$

В режиме отложенного счета (FON) на сервере выполняется в виде пакета состояний из (BDZAD_s) $n_{\text{пак}j}$. Временная диаграмма выполнения n -й задачи представлена на рисунке 4.5. В пакете могут выполняться строго последовательно $n_{\text{пак}j}$ различных типов задач с наименьшим приоритетом. Без учета предложений в использовании CPU_{3j} и HDD_{3j} и выполнение S-й задачи согласно BDZAD_{sj} представляет собой перемещающуюся последовательность квантов использования ресурсов и CPU_{3j} и HDD_{3j}:

Δt_{31jsk} – время k -го кванта использования CPU_{3j} задачами ПАК S-го типа;

Δt_{32jsk} – время k -го кванта использования HDD_{3j} задачами ПАК S-го типа;

Из BDZAD_{sj} следует, что S-ю задачу можно решить двумя последовательностями $\{\Delta t_{31jsk}\}$ и $\{\Delta t_{32jsk}\}$, и в которых число элементов различно:

n_{31sj} – количество квантов использования CPU_{3j};

n_{32sj} – количество квантов использования HDD_{3j}.

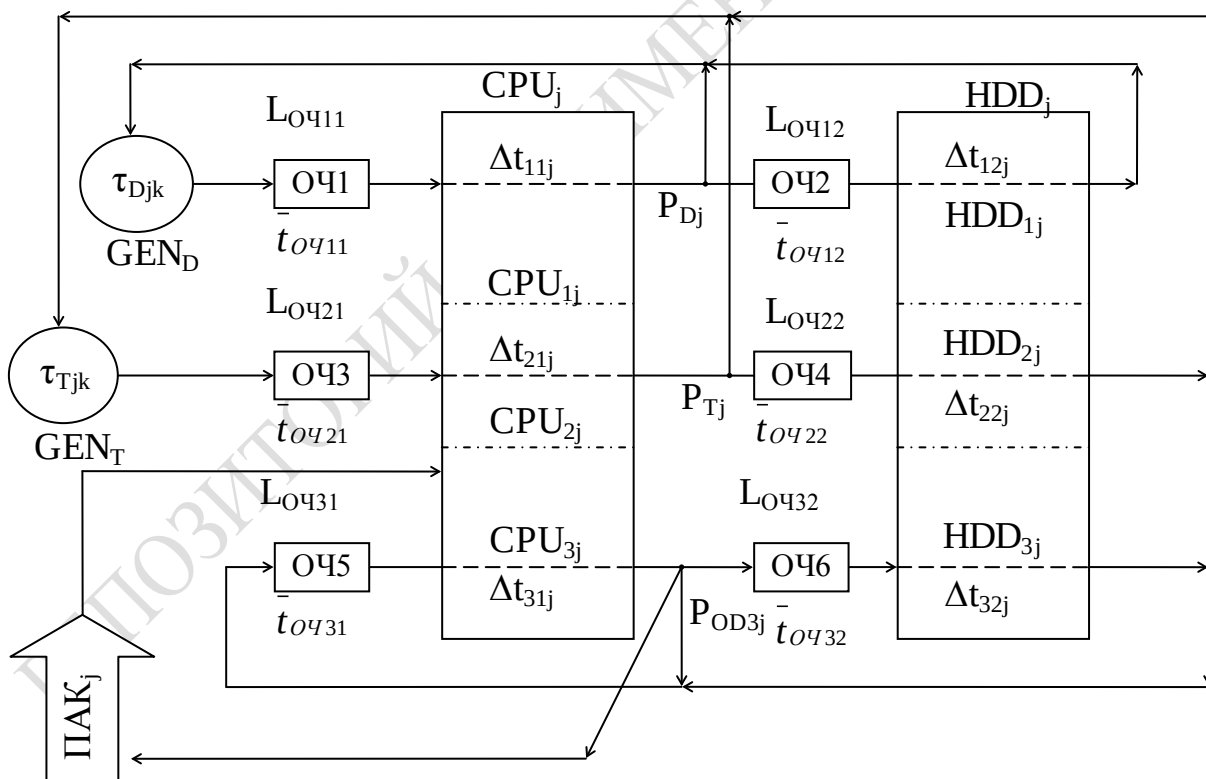


Рисунок 4.5 – Блок-схема использования ресурсов j -го узла ЛВС запросами пользователей в трех режимах (DR, DIS, FON)

Из-за вероятностного характера времен использования S-й задачей ресурсов CPU_{3j} и HDD_{3j}, используя содержимое этих

последовательностей, S-я задача представляется функциями распределения длительности в режиме использования ресурсов сервера ЛВС $F_{31j}(\Delta t)$ и $F_{32j}(\Delta t)$ и вероятностью обмена в S-й задаче

$$p_{OBSj} = \frac{n_{31sj}}{n_{32sj}}.$$

Проведя подобные преобразования $BDZAD_{sj}$ для технологического моделирования, пакет задач представим множеством запросов ресурсов сервера ЛВС:

$$\{n_{пакj}, F_{31sj}(\Delta t), F_{32sj}(\Delta t), p_{OBSj}\}. \quad (4.4)$$

Итак, набором зависимостей (4.2)–(4.4) будем характеризовать РН на узел ЛВС.

Использование ресурсов j -го узла ЛВС (сервера) запросами вида (4.2)–(4.4) в трех режимах представим в виде блок-схемы (рисунк 4.6). Как видно из рисунка 4.6, выполнение пакета задач ПAK_j в режиме FON имеет самый низкий приоритет при выборе ресурсов узла ЛВС и осуществляется при наличии обслуживания ОС узла ЛВС длительностью (Δt_{ock}) более приоритетных запросов в режиме DR и DIS. ВП на сервере ЛВС представляется взаимодействием запросов на ресурсы узла (транзакты) TR_{Dj} , TR_{Tj} , TR_{Fj} с обслуживающими устройствами (CPU_j и HDD_j) с управляющими входами [19]. На рисунке 4.6 не показано устройство ОС, поскольку имеется ввиду, что запрос в любом режиме обслуживания получает в свое распоряжение ресурс CPU_j или HDD_j на время Δt_{ocj} . По окончании кванта использования ресурса Δt_{ocj} происходит прерывание выполнения запроса, и далее срабатывает система приоритетов режимов при выборе следующего использования ресурсов сервера ЛВС.

Устройства CPU_j и HDD_j развязаны друг от друга соответствующими очередями (рисунк 4.6). Использование очередей фиксируется следующими статистиками:

– длина очереди и среднее время нахождения в очередях к CPU_{1j} и HDD_{1j} ($L_{оч11}; \bar{t}_{оч11}$) и ($L_{оч12}; \bar{t}_{12}$); (4.5)

– длина очереди и среднее время нахождения TR_{Tj} в очередях к TR_{Tj} CPU_{2j} и HDD_{2j} ($L_{оч21}; \bar{t}_{оч21}$) и ($L_{оч22}; \bar{t}_{оч22}$); (4.6)

– длина очереди и среднее время нахождения TR_{Fj} в очередях к CPU_{3j} и HDD_{3j} ($L_{оч31}; \bar{t}_{оч31}$) и ($L_{оч32}; \bar{t}_{оч32}$). (4.7)

Предполагается, что измерения параметров запросов осуществляются в ходе натурных экспериментов (ИЭ) на реальном сервере

ЛВС при интенсивности поступления запросов λ_{Djo} и λ_{Tjo} и длительности кванта работы ОС узла Δt_{oc} . Значения параметров, измеренные в ходе НЭ, назовем базовыми. Для дальнейшего моделирования пересчитаем интервалы времен использования ресурсов узла ЛВС путем деления на величину Δt_{oc} , что равносильно переходу к относительным характеристикам.

В результате подобных пересчетов появляется возможность моделирования при скоростях CPU_j и HDD_j , отличных от базовых (на случай замены сервера с другой элементной базой). Таким образом, формальные описания РН во всех режимах выполнения запросов пользователей представляют собой следующие множества:

$$\text{для режима DR } \{\lambda_{Dj}; n_{11j}; P_{Dj}; \bar{v}_{11j}; \bar{v}_{12j}\}; \quad (4.8)$$

$$\text{для режима DIS } \{\lambda_{Tj}; n_{21j}; P_{Tj}; \bar{v}_{21j}; \bar{v}_{22j}\}; \quad (4.9)$$

$$\text{для режима FON } \{n_{ПАКj}, \{F_{31sj}(v); F_{32sj}(v); P_{Обsj}\}\}; s = 1, n_{пакj}. \quad (4.10)$$

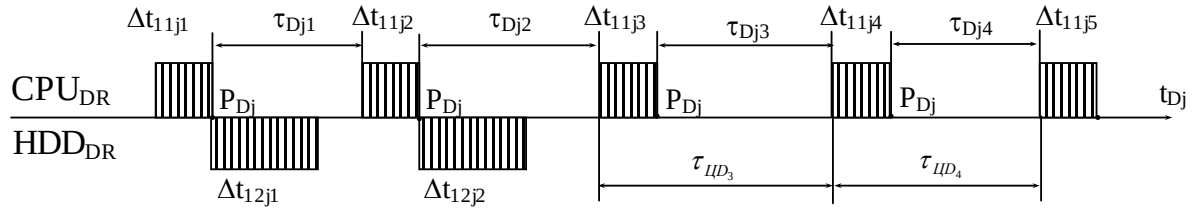
На рисунке 4.6 показано, что транзакты TR_{Dj} и TR_{Tj} моделируются на входе ИМ ВП ЛВС с помощью генераторов $GEND_k$ и $GENT_k$ соответственно с интенсивностями λ_{Dj} и λ_{Tj} . Транзакты обладают «телом», в котором содержатся множества соответственно (4.8) и (4.9). Фоновый режим обработки FON_j из множества $\{CRON_j\}$ из задач отложенного счета формируется в пакет $ПАК_j$, которые иницируются в моменты окончания предыдущей задачи. Каждая задача из $ПАК_j$ представляется транзактом TR_{Fj} , также обладающим «телом» состоящим из множества (4.10). Генераторы $GEND$ и $GENP$ через $\bar{\tau}_{uD}$ и $\bar{\tau}_{uT}$ моделируют TR_{Dj} или TR_{Tj} , обладающих «телом». Согласно транзактно-процессному способу формализации в состав «тел» TR_{Dj} и TR_{Tj} , составляющих собой первую часть РН на «узел» ЛВС, входят триады:

$$(S_1, \bar{v}_{11sj}, P_{1sj}, \bar{v}_{12sj}) \text{ и } (S_2, \bar{v}_{21sj}, P_{2sj}, \bar{v}_{22sj}), \quad (4.11)$$

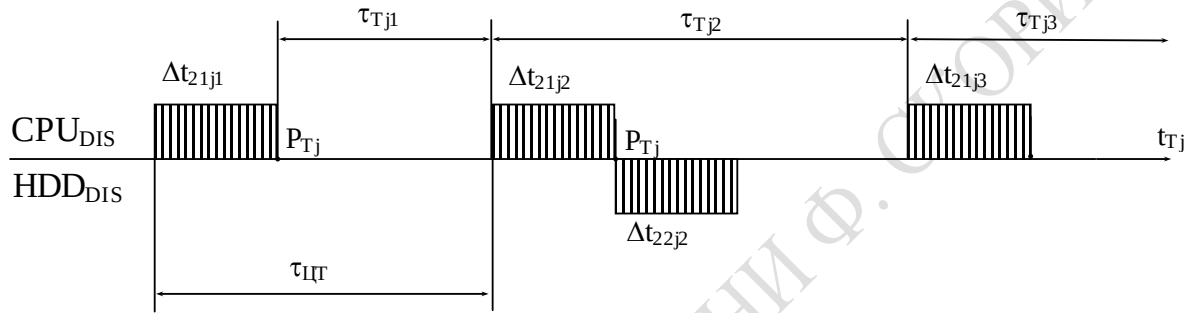
где S_1 и S_2 – номера диалоговых и транзактных пользователей.

Вторую часть РН на узел ЛВС составляет последовательность «тел» TR_{Fj} вида (4.10). Формальное описание состава ресурсов узла ЛВС состоит из алгоритмов функционирования трех процессов-имитаторов расхода ресурсов узла ЛВС.

а) режим DR



б) режим DIS



в) режим FON

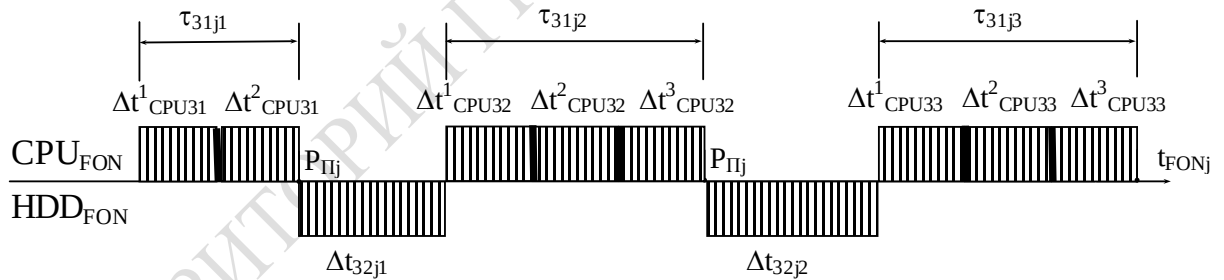


Рисунок 4.6 – Временная диаграмма инвариантов использования CPU_j и HDD_j j -го узла ЛВС в трех режимах использования запросов ресурсов узла ЛВС (без учета прерываний CPU и работы ОС узла ЛВС)

Первый тип имитаторов обладает следующим алгоритмом функционирования. Генераторы ($GEND_k$ и $GENT_k$) с интенсивностями λ_{Dj} и λ_{Tj} рожают, соответственно TR_{Dj} и TR_{Tj} , фиксируя моменты рождения транзактов (t_{PDj} и t_{PTj}). В моменты окончания обслуживания CPU_j предыдущего транзакта по вероятностям P_{Dj} и P_{Tj} и транзакты λ_{TRj} и TR_{Tj} возвращаются на генераторы. По окончании обслуживания

транзактов устройством HDD_j предыдущие транзакты возвращаются на генераторы $GEND_k$ и $GENT_k$, где они уничтожаются. В момент уничтожения транзактов фиксируется статистика интервала обслуживания транзакта злом ЛВС соответственно (t_{OBCDjk} и t_{OBSJjk}).

Второй тип имитаторов обладает следующим алгоритмом функционирования CPU_j очередей к CPU_j : из очередей к CPU_j , согласно приоритету этих очередей, в таком порядке: TR_{Dj} , TR_{Tj} и TR_{Fj} . После выбора транзакта из очереди на CPU_j осуществляется обслуживание транзакта длительностью Δt_{oc} . По истечении этого интервала времени проверяется условие: имеются ли в очередях более приоритетные транзакты. При наличии приоритетных транзактов PR_{Tj} возвращается в начало очереди $ОЧ_1$, а TR_{Fj} поступает в начало очереди $ОЧ_3$. На внешнее устройство CPU_j по вероятностям P_{Dj} , P_{Tj} , P_{Fj} , транзакты TR_{Dj} , TR_{Tj} , TR_{Fj} поступают в очереди $ОЧ_4$, $ОЧ_5$, $ОЧ_6$ соответственно к устройству HDD_j .

Третий тип имитаторов обладает следующим алгоритмом функционирования процесса-имитатора HDD_j : из очередей к HDD_j выбираются (согласно приоритетам этих очередей) в порядке транзакты TR_{Dj} , TR_{Tj} и TR_{Tj} ; после выбора транзакта из очереди на HDD_j осуществляется имитация обслуживания транзакта длительностью Δt_{oc} . Отметим, что в отличие от имитатора CPU_j обслуживание транзактов TK_j , TR_{Tj} и TR_P осуществляется без прерываний HDD_j . По окончании обслуживания на HDD_j числа квантов Δt_{oc} (n_{12} и n_{22}) транзакты TR_{Dj} и TR_{Tj} возвращаются соответственно на $GEND_k$ и $GENT_k$.

В моменты окончаний обслуживания TR_{Fj} модифицируется количество обслуженных «тел» ($\Omega_{sj} = \Omega_{sj} - 1$) и проверяется условие, обслужен ли последний элемент задачи S (конец задачи). По окончании выполнения последнего кванта использования устройств CPU_j и HDD_j инициируется выбор из пакета следующей задачи. При этом проверяется условие окончания имитации всего пакета задач.

Итак, формальное описание ВП и РН на узле ЛВС представляет собой:

- временную диаграмму (рисунок 4.5);
- блок-схему ВП и РН в ЛВС (рисунок 4.6);
- формальное описание РН режимов DR_j и DS_j вида (4.8) и (4.9);
- формальное описание РН режима FON_j вида (4.10);
- изложенное выше описание алгоритмов функционирования генераторов $GEND_k$ и $GENT_k$;
- описание алгоритма функционирования устройство-имитатора CPU_j ;
- описание алгоритма функционирования устройство-имитатора HDD_j .

Устройства-имитаторы CPU_j и HDD_j представляют собой процессы типа приборов массового обслуживания с прерыванием и управлением, использующие приоритетную дисциплину выбора транзактов сложной природы (кроме заголовка транзакта они имеют еще «тело», в котором содержится информация для алгоритма выполнения устройства). В силу простоты алгоритмов-имитаторов этих устройств блок-схемы реализации их алгоритмов не приводятся. Отметим, что из зависимостей (4.8)–(4.10) формируются: вектор структуры рабочей нагрузки X_{PH3} и множества, характеризующие рабочую нагрузку на узел ЛВС G_{PH1} , G_{PH2} , G_{PH3} .

4.4 Формализация технологии поточного производства

Проиллюстрируем методику формализации СС на двух примерах технологии поточного производства (ТПП). В примере 3 формализуется динамика взаимодействия компонентов участка обработки деталей двух типов на трех станках (рисунок 2.4). Затем в примере 4 рассматривается более сложный ТПП, а именно технология сборочно-разборочного производства ремонта деталей сложной структуры. Для отображения алгоритмов взаимодействия компонентов участка обработки деталей, приведенного на рисунке 2.4, в примере 3 используем транзактно-процессный способ имитации. Запросы на обработку деталей будем хранить в телах транзактов, определяющих дальнейший путь деталей в цехе от одного станка к другому станку. Сами же станки представим процессами-имитаторами операций обработки деталей. В примере 4 иллюстрируется технология составления формального описания иерархически построенного ТПП ремонта изделий сложной структуры. Здесь также используется транзактно-процессный способ формализации ТПП. Но транзакты имеют более сложную структуру «тела». Процессы позволяют описать алгоритмы выполнения микро-технологических операций, а связи между операциями устанавливаются технологом с помощью соответствующих блок-схем. Процессы-имитаторы $MTXO_{ij}$ используют содержимое «тела» транзакта-имитатора компонентов изделия сложной структуры. В зависимости от информации, находящейся в «теле» транзакта сложной структуры, процесс-имитатор $MTXO_{ij}$ выполняет различные алгоритмические действия, отображая их выполнение при этом в модельном времени t_0 .

Формализация динамики функционирования участка обработки деталей несколькими станками. Декомпозиция

124

В качестве входной информации используются функции распределения длительностей квантов обслуживания деталей i -го типа j -м станком $F_{ij}(\tau)$ и вероятность P_1 поступления на первый станок детали 1-го типа. Эта вероятность выполняется по интенсивности поступления деталей i -го типа (λ_i) по формуле:

$$p_1 = \varphi_1(\zeta_1 + \zeta_2). \quad (4.11)$$

Ранее были выбраны в качестве критерия эффективности системы коэффициенты загрузки станков η_j и средние значения времен обработки деталей T_i .

Вычисления значений этих откликов модели производим по формулам:

$$\eta_j = \sum_{i=1}^2 \sum_{h=1}^{w_{ij}} \frac{t_{rijh} - t_{Hijh}}{T_n}, \quad (4.12)$$

где η_j – коэффициенты загрузки ОСТ $_j$;

w_{ij} – количество обработанных деталей i -го типа j -м станком (в расчет берутся лишь детали, прошедшие обработку на j -м станке);

T_n – календарный период исследования производительности участка;

$$T_1 = \frac{1}{W_{13}} \sum_{h=1}^{W_{13}} (t_{k13h} - t_{H13p}); \quad T_2 = \frac{1}{W_{24}} \sum_{h=1}^{W_{24}} (t_{k24h} - t_{H24p}), \quad (4.13)$$

где W_{13} и W_{24} – число деталей, обработанных соответственно третьим и четвертым станками.

В ходе ИЭ необходимо найти зависимости:

$$\eta_j = \varphi_{ij}(\lambda_i); \quad T_i = \Psi_{ij}(\lambda_i), \quad (4.14)$$

которые согласно нашей гипотезе имеют вид полиномов. Порядок этих полиномов и значения коэффициентов необходимо определить в ходе ИЭ. Для начала предполагаем, что целевые функции (4.14) имеют вид нелинейных регрессионных зависимостей откликов от параметров модели:

$$\begin{aligned} \eta_j &= \beta_{0j} + \beta_{1j}\lambda_1 + \beta_{2j}\lambda_2; \\ T_i &= \gamma_{0i} + \gamma_{1i}\lambda_1 + \gamma_{2i}\lambda_2, \end{aligned} \quad (4.15)$$

где β_{kj} и γ_{kj} – коэффициенты регрессионных зависимостей (4.15), значение которых необходимо определить с помощью ИЭ. При составлении схем информационных связей между ОСТ_j используется временная диаграмма (рисунок 4.7). Для этой цели выделим глобальные переменные, отображающие связи между станками:

π_j – переменная, принимающая значение 1, когда j -й станок занят обработкой детали, и равная 0 при простоях j -го станка;

L_{0rq} – счетчик числа деталей, находящихся в очереди на обработку к q -му станку;

$T_{очj}$ – среднее время нахождения деталей в очереди j .

Дальнейшая формализация производственного участка обработки деталей зависит от принятого способа имитации. Далее в 5.2 изложен алгоритм ИМ производственного участка. Воспользуемся способом имитации активностями.

Переход станка от простоя к обработке детали и обратно будем отображать соответствующей активностью АК_{ij}. Схема формализации объекта моделирования для этого случая представлена на рисунке 4.8. Индекс i соответствует стадии обработки деталей (i – начало обработки, $i = 2$ – конец обработки), а j соответствует номеру ОСТ_j. Через УЗ_{ij} на рисунке 4.8 обозначим условия инициализации. АК_{ij} выполняется соответствующими блоками модели, описываемыми отдельно. На схеме формализации рисунка 4.8 изобразим эти блоки рассосредоточенными по соответствующим алгоритмам активностей АК_{ij} (верхние участки подпрограмм на рисунке 4.9). Если УЗ_{ij} выполняются, что УПМ передает управление на соответствующую АК_{ij} в список ij инициализируемых активностей. При невыполнении УЗ_{ij} УПМ данную активность игнорирует и рассматривает следующую из списка активностей.

Активности АК_{i0} моделируют процесс поступления деталей i -го типа извне на обработку к первому станку. УПМ активизирует алгоритм АК_{i0} при выполнении условия УЗ_{i0}, которое состоит в наступлении момента поступления детали i -го типа (t_{ia}) в очередь ОЧ_i на обработку ($t_{ia} \leq t_0$). Операторы В_{1i} записывают в ОЧ_j к ОСТ_j очередной номер детали типа i . Операторы увеличивают значения счетчика числа деталей в ОЧ₁ к ОСТ₁ ($L_{оч1} = L_{оч1} + 1$). Операторы В_{3i} формируют моменты t_{ia} очередной активизации АК_{i0}, представляющие собой сумму текущего значения модельного времени t_0 , после которого новая деталь i -го типа поступает в систему:

$$t_{ia} = t_0 + 1/\lambda_i, \quad (4.16)$$

где λ_i – значение интенсивностей поступления деталей i -го типа для данного варианта имитации.

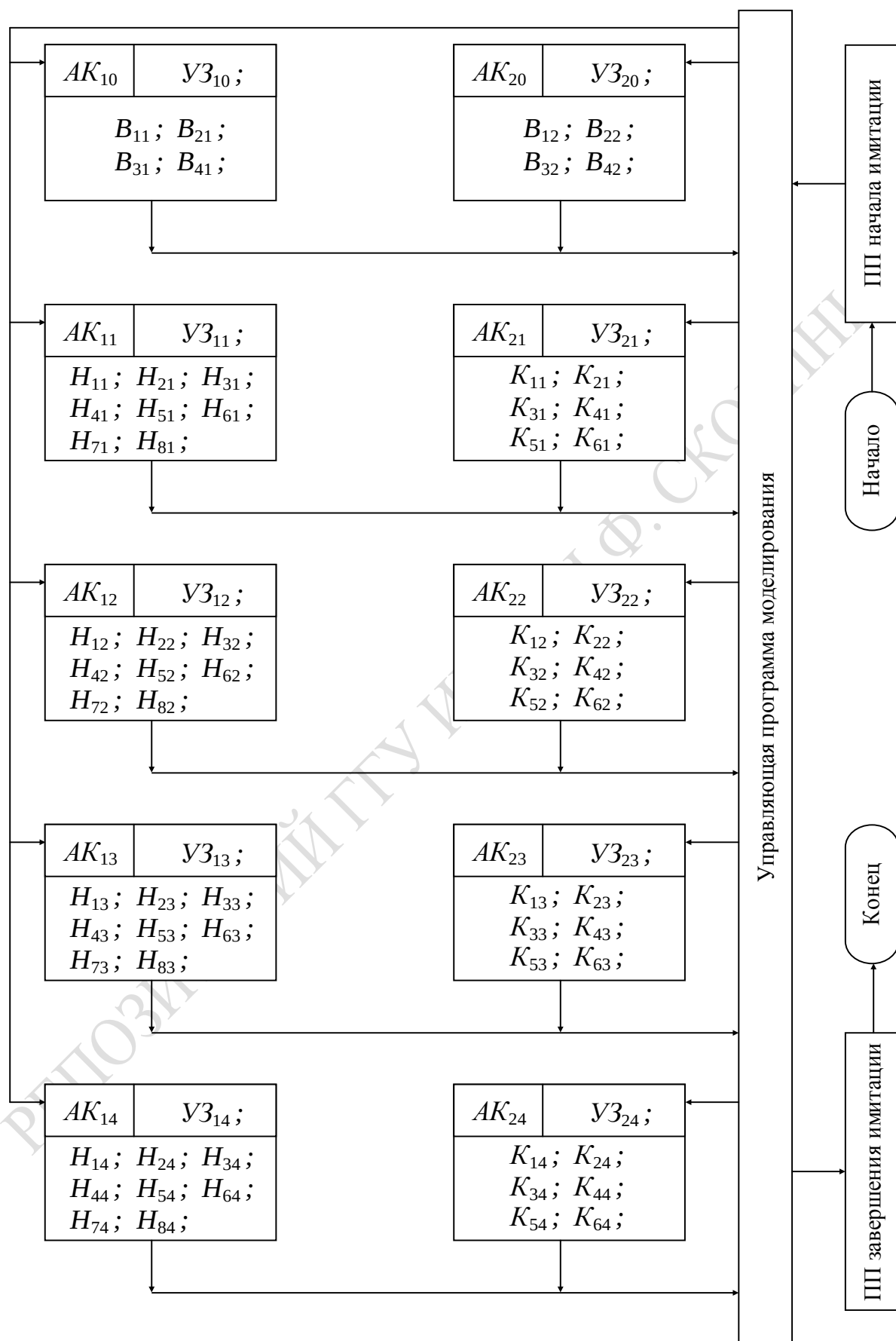


Рисунок 4.8 – Схема моделирования производственного участка с помощью активностей

Операторы B_{4i} возвращают управление УПМ, которая $F_{2k}(\tau)$, заносит значения t_{ia} в упорядоченный по величине список моментов инициализации активностей.

Алгоритмы выполнения активностей AK_{1j} и AK_{2j} состоят из операторов, соответственно H_{aj} K_{ej} , которые возвращают управление УПМ, которая заносит очередные значения в упорядоченный по возрастанию список инициализации активностей. Алгоритмы выполнения активностей AK_{1j} и AK_{2j} состоят из операторов соответственно. Активности AK_{1j} инициализируются УПМ при совместном наступлении двух условий: j -й станок свободен и имеются в $ОЧ_j$ к нему детали. Активности инициализируются УПМ при наступлении момента окончания обработки детали j -м станком. Операторы обеспечивают фиксацию моментов соответственно начала и конца обработки деталей.

Алгоритм выполнения активностей AK_{1j} состоит в следующем. С помощью оператора H_{2j} устанавливаются признаки занятости j -го станка обработкой детали ($\pi_j = 1$). Операторы H_{3j} организуют выбор из $ОЧ_{1j}$ номера детали i ; номер очередной детали i обрабатывается j -м станком в данный момент модельного времени t_0 , содержащегося в глобальной переменной. Запоминание номера детали v обеспечивается оператором H_{4j} . Уменьшение на единицу числа деталей, находящихся в очереди к j -му станку, выполняется оператором H_{5j} ($L_{ОЧj} := L_{ОЧj-1}$). Для формирования моментов очередной инициализации активностей используются функции распределения длительностей обслуживания деталей $F_{ij}(\tau)$. Операторы H_{0j} по номеру детали i и $F_{ij}(\tau)$ формируют очередное значение интервала обработки детали i станком j (τ_{ij}). С помощью оператора H_{7j} формируются запланированные моменты очередной инициализации активностей AK_{ij} по формуле:

$$t_{ja} := t_0 + \tau_{ij}. \quad (4.17)$$

Наконец, операторы H_{0j} возвращают управление УПМ. При выполнении алгоритмов активностей AK_{ij} реализуется следующая последовательность операторов. С помощью операторов K_{ij} сбрасывается признак занятости j -го станка ($\pi_j = 0$). Операторы K_{3j} обеспечивают извлечение номера детали i и формирование по нему $ОЧ_1$, куда поступают детали после обработки на $ОСТ_j$. Возврат ЧПМ обеспечивается оператором K_{6j} .

Формальное описание функционирования производственного участка содержит в себе:

- приведенный выше текст;
- схему обработки деталей в цехе (рисунок 2.4);

- блок-схему концептуальной модели производственного участка обработки деталей (рисунок 4.6).
- временную диаграмму функционирования (рисунок 4.7);
- схему моделирования (рисунок 4.9).

Как видим, состав документации накапливается, пополняясь на каждом из этапов технологии исследования производственного участка.

Формализация технологического процесса ремонта изделия D сложной структуры. На основании содержательного описания ТП СРО (пример 4) о концептуальной модели ранее мы установили возможность описания процесса с помощью графа $GTxO_{jh}$. Структуру этого графа пометим в «теле» транзакта TR_i в виде технологической карты $GTxO_{jh}$. Для формализации ТП СРО используем транзактно-процессный способ имитации СС. Технологическая карта TK_{ih} , расположенная в «теле» транзакта TR_i сложной структуры, описывает множество $\{MTxO_{ijh}\}$ с помощью трех типов описателей операций:

- обслуживания запросов на обработку деталей, узлов и компонентов изделия сложной структуры (согласно, $GTxO_{jh}$, которые ранее были определены как операции ремонта или замены);

- разработки изделия и его компонентов для ремонта как их замены, изменяющих структуру ТП СРО и обозначаемых в дальнейшем как операции «расщепления» участков ТП СРО_i на параллельно выполняемые ветви ($RACH_{jh}$);

- сборки компонентов изделия после их ремонта, также изменяющих структуру ТП СРО_i и обозначаемых как операции «объединения» участков ТП СРО_{ih} в одну сеть ($OBED_{jh}$) и $OBED_{jh}$, приводящих к ликвидации параллельных ветвей процесса. В примере 4 операции $RACH_{jh}$ и состоят только из одной $MTxO_{ijh}$, а операции 1-го типа образуют упорядоченное множество $\{MTxO_{ijh}\}$, порядок выполнения которых задается графом $GTxO_{jh}$. Каждая $MTxO_{ijh}$ для своей реализации требует использования следующих ресурсов предприятия: время выполнения операции (τ_{ijh}), количество материалов и комплектующих изделий (mt_{ijh}), состав рабочей силы (IS_{ijh}), состав оборудования (OB_{ijh}), стоимость ее выполнения (CT_{ijh}). При этом состав и структура $MTxO_{ijh}$ в $GTxO_{jh}$ фиксированы и задаются в виде TK_i .

Все ТП СРО исследуемого предприятия реализуются на множестве узлов обработки $\{УЗО_k\}$, представляющих собой рабочие места (PM_k) с закрепленными за ними оборудованием, оснасткой, составом исполнителей, обслуживающих это оборудование. Таким образом, каждая $MTxO_{ijh}$ характеризуется вектором ресурсов ТП СРО_h (τ_{ijh} , mt_{ijh} , IS_{ijh} , OB_{ijh} , CT_{ijh}), одни компоненты которого являются

детерминированными (IS_{ijh} , OB_{ijh}), а остальные являются вероятностными, задаваемыми с помощью соответствующих функций распределения:

$$(F_{1ijh}(\tau), F_{2ijh}(mt), F_{3ijh}(CT)). \quad (4.18)$$

Списки исполнителей IS_{ijh} и оборудования OB_{ijh} определяют ресурсы индивидуального использования, необходимые для выполнения $MTXO_{ijh}$. Исследование динамики взаимодействия $\{MTXO_{ijh}\}$ на одном и том же составе рабочих площадей предприятия и узлов обработки изделий $\{УЗО_l\}$, предполагает решение задач, изложенных при описании содержательного описания. При реализации графа $G_{MTXO_{ijh}}$ отобразим динамику использования $\{УЗО_l\}$ в ходе имитации выполнения $\{MTXO_{ijh}\}$ с учетом появления операций $RAON_{ij}$ и $OBED_{jh}$. Для этой цели используется TK_i в теле транзакта TR_i и копий управляющих транзактов (UTR_{il}). При появлении в графе $G_{TXO_{ih}}$ тех $MTXO_{ij}$, которые требуют замены $УЗО_l$ на $УЗО_{l+1}$ из TR_i , рождается новый UTR_{il+1} , который в дальнейшем поступает в очередь к $УЗО_{l+1}$ и затем существует в ИМ до завершения его обслуживания устройством $УЗО_{l+1}$. Таким образом, любое завершение обслуживания транзактов на $УЗО_l$ формирует переход TK_i в новое состояние. Сами же операции обработки UTR_l реализуются во время его обслуживания алгоритмом ИМ устройства $УЗО_l$.

Поскольку $УЗО_l$ являются базовыми компонентами ИМ ТП $СРО_h$, они также параметризованы и специализируются на обслуживании определенных групп $\{MTXO_{ijh}\}$. Алгоритм функционирования процессов на $УЗО_l$ реализует: выбор очередного UTR_{il} из общей очереди согласно их приоритету (Π_{il}); формирование списка запросов требуемых ресурсов каждого типа для выполнения $MTXO_{ijh}$. Поэтому реализация алгоритмов $УЗО_l$ начинается только после захвата требуемого состава ресурсов предприятия. Имитация $MTXO_{ijh}$ сводится к временной задержке UTR_{il} на интервале τ_{ijhl} , формируемых в l -й реализации ИМ ВТПО по соответствующим функциям распределения (4.18). С помощью TK_{ih} ТП $СРО_h$ отображается на трех уровнях детализации, на верхнем уровне ТП $СРО_h$ представляется графом G ТП $СРО_h$, в котором связи между узлами этого графа детерминированы. Вероятностными могут быть только параметры расхода ресурсов предприятия при реализации $\{MTXO_{ij}\}$. Второй уровень описания ТП $СРО$ позволяет задавать динамику реализации $\{MTXO_{ij}\}$ с помощью графа TXO_{ih} . Особенность этого графа – то, что ветви его являются одноуровневыми вероятностными подграфами с детерминированным

порядком составления $MTXO_{ij}$ и вероятностным характером использования ресурсов предприятия. На третьем уровне представления этого графа отображается процесс захвата и освобождения ресурсов предприятия на рабочих местах. Ресурсы описываются следующим набором параметров: число элементов, списки параметров (r), сам список наименований ресурса ($SP_1, \dots, SP_r$), характеристика типа (тип) распределения захватываемого объема ресурса (Y_r). Концептуальная модель **примера 4** приведена ранее на рисунке 3.4. При формализации ТП CPO_h будем придерживаться следующих принципов [15]. **Во-первых**, это объединение в одном тексте модели декларативного и алгоритмического способов описания компонент ИМ ТП CPO . В качестве элементарных единиц ТП CRP используем приборы систем массового обслуживания с управлением (СМОУ). Количество этих приборов, связи между ними с помощью системы очередей и дисциплины обслуживания транзактов составляют декларативную часть описания ИМ. Функционирование приборов СМОУ моделируется реализацией сложного алгоритма. Алгоритмы СМОУ содержат операторы: планирования длительности обработки транзактов TR_i ; изменения параметров ИМ; расчета статистик имитации; алгоритмические примитивы общего назначения (ветвления, циклы, выбор адресата).

Во вторых, применяем транзактно-процессный способ имитации. В качестве базовых единиц ИМ ТП CRP используем приборы СМОУ. Время функционирования приборов определяется по стандартному алгоритму, используя возможности языка программирования СИ. **В-третьих**, организуются связи между компонентами процесса по управлению. Каждая активность ИМ представляется «си-функцией» $SM\ MICIC$ [14], в теле которой имеется указание на следующую активность, которая должна быть инициализирована. Взаимодействие между $\{MTXO_{ij}\}$ осуществляется с помощью транзактов сложной природы, «тело» которых служит информацией для выполнения алгоритма приборов СМОУ. Перевод приборов и транзактов из состояния в состояние осуществляется специальными функциями из библиотеки $SM\ MICIC$. **В-четвертых**, в ИМ ТП CPO используем параметризацию и размножаемость компонентов модели. Каждый TR_i и прибор СМО (в данном случае имитатор $MTXO_{ij}$) имеют набор параметров, который отделен от программ компонента модели или от «тела» транзакта сложной природы. Параметризованность TR_i дает возможность управления обслуживанием транзактов и сбора комплексной статистики имитации (о жизненном цикле, маршруте и времени TR_i пребывания в очередях к СМОУ). **В-пятых**, ИМ ТП CPO описывается

на высоком уровне детализации для каждого имитатора $MTXO_{ij}$. При этом компоненты ИМ объединяются в комплексе на принципах итеративной имитации. Все подмодели объединяются по параметрам и потокам запросов $MTXO_{ij}$ на рабочих местах PM_k . **В-шестых**, в ИМ реализуется «наследование» TR_i по информации. В информации, расположенной в «теле» TR_i , определен порядок выполнения операций «сборки», разборки и ремонта изделия, что позволяет автоматически организовать связи $MTXO_{ij}$ на разных уровнях детализации графа G ТП CPO_i формируются информационные транзакты ($TRINF_i$) с интенсивностью λ_i . Затем приборы имитаторы $MTXO_{ij}$ по информации, находящейся в «теле» транзакта $TRINR_i$, (соответствующие операторы TK_i [14]), формируются управляющие транзакты (UTR_{ij}), являющиеся запросами на обслуживание их в моделях PM_k . Сама $TRINF_j$ поступают согласно G ТП CPO_i последовательно от одной $MTXO_{ij}$ к другой $MTXO_{ij}$ (соответственно к устройствам с управлением). Каждый имитатор $MTXO_{ij}$ по информации в «теле» $TRINF_i$ формирует новые управляющие транзакты UTR_{ij} , которые затем отслуживаются на PM_k . При этом времена обслуживания UTR_{ij} на PM_k задаются либо в виде функций распределения, либо соответственно рассчитываются в ходе имитации.

В седьмых, с помощью операторов ожидания (ЖДАТЬ (τ_k)) осуществляется синхронизация компонентов модели в модельном времени t_0 согласно стандартному алгоритму управляющей программы моделирования (УПМ). Кроме того, формируются статистики имитации с помощью соответствующих наблюдений за имитацией ($STATIST_r$).

В примере 4 видна многоуровневая структура компоновки деталей D_{ij} в единое изделие (рисунок 2.6). В соответствии со сформулированным в содержательном описании изделия D в технологии его ремонта можно выделить 10 ветвей технологических операций, в которых число $MTXO_{ij}$ колеблется от 1 до 34 операций ремонта. Наличие неисправностей в изделии D задается матрицей вероятностей этих неисправностей (таблица 2.2), которая может быть использована для розыгрыша в ИМ жребиев 1-го типа [17]. Срабатывание этих жребиев приводит к имитации ремонта деталей D_{ij} изделия D . С помощью рисунка 2.5 по структуре компоновки изделия деталями технолог предприятия формирует блок-схему технологии ремонта (рисунок 2.6). Далее технолог предприятия перерабатывает (рисунок 2.7) и формирует блок-схему связей $\{MTXO_{ij}\}$ со множеством рабочих мест $\{PM_k\}$ и используем множеством ресурсов R_r (рисунок 2.7). Каждая ветвь ТП CPO ремонта изделия D описывается последовательностью $\{MTXO_{ij}\}$ и содержит от 1 до 34 операций ремонта деталей D_{ij} . Отметим, что дочерние ветви формируются операциями разборки изделия ($RAZBOR_r$)

и завершаются операциями сборки изделия с его составными компонентами ($RAZBOR_r$). На этой блок-схеме показано, что каждая $MTXO_{ij}$ выполняется на соответствующем рабочем месте PM_k и требует для своего выполнения подмножества ресурсов предприятия (R_r). Список запросов ресурсов предприятия, необходимых для выполнения $MTXO_{ij}$ при ремонте изделия D , приведен в таблице 2.3. Начальное значение ресурсов каждого типа, имеющихся у предприятия перед началом имитации, приведено в таблице 2.7. При выполнении $MTXO_{ij}$ на устройстве $RAZBOR_r$ ремонт приводится не во всех случаях, и его необходимость определяется таблицей 2.2. Однако при выполнении каждой $MTXO_{ij}$ на PM_k проводится операция осмотра детали. Время выполнения этой операции является случайной величиной и задается до имитации с помощью соответствующей функции распределения $F_{1k}(\tau)$ в таблице 2.4. Сама же операции ремонта деталей осуществляется на PM_k только для срабатывания жребия «нужен ремонт», который моделируется согласно таблице 2.5, и его выполнение имитируется в течение интервала времени по функции распределения $F_{2k}(\tau)$, представленной в таблице 2.5. Кроме того, для выполнения операции ремонта может понадобиться некий общий ресурс предприятия номера g , на котором PM_k захватывает место размером (V_R) на время выполнения операции и который PM_k затем освобождает по окончании выполнения $MTXO_{ij}$. Размер этого ресурса также является случайной величиной, которая в таблице 2.6 задается соответствующей функцией распределения $F_{3kr}(V_R)$ до начала имитации. Кроме того, каждая $MTXO_{ij}$ для своего выполнения на PM_k требует индивидуальных ресурсов предприятия, список которых также задается до имитации таблицей 2.3. При построении ИМ ТП СРО достаточно заменить операции $MTXO_{ij}$ $RAZBOR_r$ и $SBORKA_s$ соответствующими процессами, являющимися приборами СМОУ. $PR.REMONT_{ij}$ управляющий транзакт UTR_{ij} , представляющий собой запрос на выполнение $MTXO_{ij}$ на PM_k . Все эти UTR_{ij} поступают в общую очередь к PH_k (рисунок 2.7). После имитации выполнения $MTXO_{ij}$ на PM_k транзакт UTR_{ij} возвращается на $PR.REMONT_{ij}$, что будет означать конец имитации $MTXO_{ij}$ на PM_k . При этом приборы PM_k возвращают ресурсы, необходимые для выполнения $MTXO_{ij}$. Эти ресурсы прибор PM_k запрашивает у системы распределения ресурсов (SRR_k) в ИМ ТП СРО согласно списку, представленному в таблицах 2.3, 2.6. Если нет свободного места на ресурсы, то прибор PM_k ждет его освобождения. Времена имитации (в l -й реализации процедуры Монте-Карло) разыгрываются соответственно по функциям распределения $F_{1k}(\tau)$ и $F_{2k}(\tau)$, представленным в таблицах 2.4, 2.5. Сам же транзакт

TRINF_i, имитирующий запрос ресурсов предприятия при выполнении MTXO_{ij}, переходит во входную очередь следующего прибора REMONT, имитируя таким образом запрос ресурсов выполнения следующей MTXO_{ij} по технологической цепочке выполнения операции (рисунок 2.6).

Операции «разборки» и «сборки» компонентов изделия D также осуществляются на PM_к. На рисунке 2.7 пунктирными линиями обозначены запросы PM_к общих ресурсов R_г, которые изображаются в виде прямоугольника (RESURS). Аналогичным образом на рисунке 2.6 обозначены границы цехов предприятия. Операции «разборки» или «сборки» на PM_к обозначены соответственно RC_г и SB_г. Линии связи обозначают траектории движения TRINF_i основных и дочерних технологических линии ремонта изделия.

Отметим, что оба типа транзактов (TRINF_i и UTR_{ij}) являются сложными транзактами и обладают «телом». Здесь V – номер основной технологической цепочки ремонта изделия D; j – номер прибора REMONT_{ij} внутри технологической ветви ремонта изделия. Структуры «тел» обоих типов транзактов различны. В «теле» TRINF_i содержится технологическая карта ТК_i, фрагмент состава и структуры которой приведен на рисунке 4.9. Как видно из рисунка 4.9, в ТК_i содержится 129 строк. Технолог предприятия разрабатывает, используя структуру изделия (рисунок 2.5). Строки типа RE используются алгоритмом процесса REMONT_{ij} и обеспечивают по номеру i выполнение технологической ветви ремонта (i) при номере прибора (j) на PM_к. В этой строке также имеется: номер следующей строки ТК_i (K_i), адрес поля статистики жизни транзакта (St_{ij}) и номер ветви (ind = 1,6). Строка RC_г активизирует алгоритм процесса RAZBOR_г и обеспечивает выполнение r-й ветви операции «разборки». В ней указывается: номер операции «разборки» (j), номер PM_к, где выполняется операция (к), количество разветвлений технологической ветви; (s) номер «дочерней» технологической ветви (v₁) и адрес строки ТК_i (f₁). Строки SB_г инициируют процесс SBORKA_г и обеспечивают выполнение i-й ветви операции «сборки» ветвей в базовую ветвь. Номер операции «сборка» (j), номер PM_к, где выполняется операция (к), количество объединяемых дочерних ветвей (s) технологии, номер очередной технологической ветви (i₁), адрес строки ТК_i(f₁) той ветви, с которой данная ветвь технологии объединяется и для которой потом TRINF_{i1} уничтожается.

Формальное описание ТП ремонта изделия D сложной структуры содержит в себе:

- приведенный выше текст описания технологии ремонта изделия D;
- структурную схему изделия D (рисунок 2.5);

K _i	ТИП	i	j	k	k _i	S	i ₁	f ₁	ПП	K _i	ТИП	i	j	k	k _i	S	i ₁	f ₁	ПП
1	GEN	-	-	-	2	-	-	-	-	63	RE	2	12	46	64	-	-	-	-
2	RE	1	1	1	3	-	-	-	-	64	RE	2	13	46	65	-	-	-	-
3	RE	1	2	1	4	-	-	-	-	65	RE	2	14	47	66	-	-	-	-
4	RE	1	3	1	5	-	-	-	-	66	RE	2	15	47	67	-	-	-	-
5	RE	1	4	1	6	-	-	-	-	67	RE	2	16	48	68	-	-	-	←
6	RE	1	5	1	7	-	-	-	-	68	RE	3	1	45	69	-	-	-	←
7	RE	1	6	1	8	-	-	-	-	69	RE	3	1	45	70	-	-	-	-
8	RC ₁	1	1	2	9	1	4	83	→	70	RE	3	2	45	71	-	-	-	-
9	RE	1	7	3	10	-	-	-	-	71	RE	3	3	45	72	-	-	-	-
10	RE	1	8	3	11	-	-	-	-	72	RE	3	4	45	73	-	-	-	-
11	RC ₂	1	2	4	12	1	2	52	→	73	RE	3	5	45	74	-	-	-	-
12	RE	1	9	5	13	-	-	-	-	74	RE	3	6	45	75	-	-	-	-
13	RC ₃	1	3	6	14	1	5	129	→	75	RE	3	7	45	76	-	-	-	-
14	RE	1	10	7	15	-	-	-	-	76	RE	3	8	45	77	-	-	-	-
15	RC ₄	1	4	8	16	1	5	102	→	77	RE	3	9	45	78	-	-	-	-
16	RE	1	11	9	17	-	-	-	-	78	RE	3	10	45	79	-	-	-	-
17	RE	1	12	9	18	-	-	-	-	79	RE	3	11	45	80	-	-	-	-
18	SB ₃	1	3	10	19	1	10	129	←	80	RE	3	12	45	81	-	-	-	-
19	RE	1	13	11	20	-	-	-	-	81	RE	3	13	45	82	-	-	-	-
20	RE	1	14	11	21	-	-	-	-	82	RE	3	14	45	83	-	-	-	←
21	SB ₄	1	4	12	22	1	5	108	←	83	RE	4	1	31	84	-	-	-	-
22	RE	1	15	13	23	-	-	-	-	84	RE	4	2	31	85	-	-	-	←
23	RE	1	16	13	24	-	-	-	-	85	RE	4	3	31	86	-	-	-	-
24	RC ₅	1	5	14	25	1	8	127	→	86	RE	4	4	31	87	-	-	-	-
25	RE	1	17	15	26	-	-	-	-	87	RE	4	5	31	88	-	-	-	-
26	RE	1	18	15	27	-	-	-	-	88	RE	4	6	31	89	-	-	-	-
27	RC ₆	1	6	16	28	1	9	128	→	89	RE	4	8	32	90	1	6	109	←
28	RE	1	1	17	29	-	-	-	-	90	RE	4	7	33	91	-	-	-	-
29	SB ₆	1	6	18	30	1	9	128	-	91	RE	4	8	33	92	-	-	-	-
30	RE	1	19	19	31	-	-	-	-	92	RE	4	9	34	93	1	3	68	←
31	RE	1	20	19	32	-	-	-	-	93	RE	4	9	35	94	-	-	-	-
32	RE	1	21	19	33	-	-	-	-	94	RE	4	10	35	95	-	-	-	-
33	RE ₇	1	7	20	34	1	7	126	→	95	RE	4	9	36	96	1	3	82	←
34	RE	1	22	21	35	-	-	-	-	96	RE	4	11	37	97	-	-	-	-
35	SB ₅	1	5	22	36	1	8	127	←	97	RE	4	12	37	98	-	-	-	-
36	RE	1	23	23	37	-	-	-	-	102	RE	5	1	50	103	-	-	-	←
37	SB ₂	1	2	24	38	1	2	67	←	103	RE	5	2	50	104	-	-	-	-
38	RE	1	24	25	39	-	-	-	-	104	RE	5	3	50	105	-	-	-	-
39	SB ₇	1	7	26	40	1	7	126	←	105	RE	5	4	50	106	-	-	-	-
40	RE	1	25	27	41	-	-	-	-	106	RE	5	5	50	107	-	-	-	-
41	RE	1	26	28	42	-	-	-	-	107	RE	5	6	50	108	-	-	-	-
42	RE	1	1	29	43	1	4	101	←	108	RE	5	7	50	21	-	-	-	→
43	RE	1	27	30	44	-	-	-	-	109	RE	6	1	43	110	-	-	-	←
44	RE	1	28	30	45	-	-	-	-	110	RE	6	2	43	111	-	-	-	-
45	RE	1	29	30	46	-	-	-	-	111	RE	6	3	43	112	-	-	-	-
46	RE	1	30	30	47	-	-	-	-	112	RE	6	4	43	113	-	-	-	-
47	RE	1	31	30	48	-	-	-	-	113	RE	6	5	43	414	-	-	-	-
48	RE	1	32	30	49	-	-	-	-	114	RE	6	6	43	115	-	-	-	-
49	RE	1	33	30	50	-	-	-	-	115	RE	6	7	43	116	-	-	-	-
50	RE	1	34	30	51	-	-	-	-	116	RE	6	8	43	117	-	-	-	-
51	RE	1	1	30	1	-	-	-	-	117	RE	6	9	43	118	-	-	-	-
52	RE	2	1	42	53	-	-	-	-	118	RE	6	10	43	119	-	-	-	-
53	RE	2	2	42	54	-	-	-	-	119	RE	6	11	43	120	-	-	-	-
54	RE	2	3	42	55	-	-	-	-	120	RE	6	12	43	121	-	-	-	-
55	RE	2	4	42	56	-	-	-	-	121	RE	6	13	43	122	-	-	-	-
56	RE	2	5	42	57	-	-	-	-	122	RE	6	14	43	123	-	-	-	-
57	RE	2	6	42	58	-	-	-	-	123	RE	6	15	43	124	-	-	-	-
58	RE	2	7	42	59	-	-	-	-	124	RE	6	16	43	125	-	-	-	-
59	RE	2	8	42	60	-	-	-	-	125	RE	6	17	44	98	-	-	-	→
60	RE	2	9	42	61	-	-	-	-	126	RE	7	1	52	39	-	-	-	→
61	RE	2	10	42	62					127	RE	8	1	51	25	-	-	-	→
62	RE	2	11	42	63	-	-	-	-	128	RE	9	1	53	29	-	-	-	→
										129	RE	10	1	49	18	-	-	-	→

Рисунок 4.9 – Состав и структура технологической карты ТК_г

- блок-схему связи компонентов сборочно-разборочных операций ремонта изделия D (рисунок 2.6);
- блок-схему концептуальной модели сборочно-разборочного производства ремонта изделия D (3.4);
- состав и структуру технологической карты (ТК_i) ремонта изделия (рисунок 4.9);
- таблицы исходной информации о выполнении МТХО_{ij} на рабочих местах (таблицы 2.1–2.7).

Здесь также наблюдается рост числа документов в виде технологического обеспечения проектного моделирования ТП ремонта изделия D при переходе на следующие типы исследования СС.

4.5 Формализация вероятностных технологических процессов производства

Методику формализации СС проиллюстрируем на двух примерах технологии дискретных процессов производства. В **примере 5** формализуется динамика функционирования последовательных технологических процессов производства, имеющего графовую структуру вида, представленную на рисунках 2.9, 2.10. Затем в **примере 6** рассматривается вероятностный технологический процесс с параллельно-последовательным характером выполнения МТХО_{ij} также графовой структуры, представленной на рисунке 2.11. Для отображения алгоритмов взаимодействия компонентов последовательного ВТПП используем процессный способ имитации в комбинации с полумарковским представлением переходов ВТПП из одного состояния в другое состояние. При разработке алгоритмов взаимодействия параллельно-последовательного ВТПП используем агрегатно-процессный способ имитации в комбинации с процедурой Монте-Карло. Графовый характер связи вероятностных {МТХО_{ij}}, составляющих ВТПП, позволяет для формализации использовать модификацию аппарата сетевого планирования с процедурой Монте-Карло [17]. ВТПП представляется вероятностным сетевым графиком (ВСПР), а операции МТХО_{ij} и АСОВ_i. В отличие от известного агрегатного способа имитации [15], агрегаты АМТХО_{ij} и АСОВ_j имеют более сложную структуру и по своей сути являются специализированными процессами, обладающими универсальным алгоритмом имитации технологии взаимодействия в модельном времени t_0 с другими операциями ВТПП. Сигналы также имеют сложную структуру из-за информационной «подкраски» сигналов, имеющей место при появлении аварий во время имитации

операций агрегатами $AMTXO_{ij}$. Агрегаты позволяют стандартным образом описать алгоритм выполнения $MTXO_{ij}$, а связи между агрегатами устанавливаются технологом предприятия с помощью ВСГР. Агрегаты-имитаторы используют содержимое «тела» сигнала и базу данных модели ВТПП при выполнении соответствующих алгоритмических действий в процессе имитации ВСГР.

Декомпозиция последовательно ВТПП₁ не приводится, поскольку технологию реализации ВТПП₁ можно отобразить в обобщенном виде как переходы ВТПП₁ из одного состояния в другое. Поэтому на этапе составления содержательного описания уже было проведено разбиение ВТПП₁ на различные состояния, представляющие собой процессы $PR.SOST_{ij}$; $i, j = \overline{1,10}$ – номера состояний ВТПП₁. Переходы из состояния в состояние организуют процесс PRUZTL.

Для каждого из состояний ВТПП, отображаемого соответствующим процессом $PR.SOST_{ij}$, необходимо разработать алгоритм агрегата-имитатора в виде реентерабельной программы. Взаимодействие $PR.SOST_s$ осуществляется согласно блок-схеме, приведенной на рисунке 2.10. $PR.SOST_{ij}$ используют оборудование (OBORUD) и ресурс (RESURS) предприятия (рисунок 2.10). В состояниях ВТПП₁, соответствующих восстановлению работоспособности оборудования (рисунок 2.10) после отказов функционирования, или ликвидации отказов оборудования ($PR.SOST_{10}$) используются процессы-имитаторы процедур ликвидации аварий ($PR.PROC_k$).

Процессы $PR.SOST_{i1}$ и $PR.SOST_{i2}$ обладают алгоритмом, который затем реализуется одним алгоритмом, который затем реализуется одной и той же реентерабельной программой. Также можно объединить в одну реентерабельную программу процессы-имитаторы состояний $PR.SOST_{i3}$ и $PR.SOST_{i6}$. Для остальных процессов ($PR.SOST_{i7}$ и $PR.SOST_{i10}$) необходимо разработать на следующем этапе технологии имитации соответствующие программы. С помощью справочника состояний полумарковской модели (SPMM) определяется начальное состояние полумарковской модели (ПММ) ВТПП₁. В этом состоянии определяется:

- вектор начального состояния ПММ (ϑ_H), с помощью которого находится номер начального состояния ПММ (ϑ_k), с помощью которого определяется номер конечного состояния ПММ (i_k),
- по функции распределения числа смен состояний в ПММ $F_0(x)$ определяется число смен состояний в l -й реализации ПММ (ϑ_{kl}).

Далее разрабатывается алгоритм процесса PR.UZEL, который функционирует следующим образом. Восстанавливаются: номер

предыдущего состояния ПММ (x), указатель (УКАЗ) на возможность перехода ПММ в нестандартные состояния, минуя розыгрыш по матрице переходных операций. На этом завершается имитация начального выполнения ПММ и начинается розыгрыш номера следующего состояния ПММ по матрице $\|p_{ij}\|$. Затем многократно осуществляется розыгрыш номера следующего состояния (i), который запоминается в справочнике SPMM. С помощью процедуры «ВЫБОР» осуществляется либо стандартный запуск процессов при $0 \leq j \leq 6$ (согласно матрицы $\|p_{ij}\|$, либо нестандартный запуск процессов $j \geq 7$. Нестандартно иницируемый запуск этих процессов определяется как результат анализа пары указателя: предыдущего состояния (УКАЗ) и наличия аварийной ситуации ($p_{ab} = 1$). Далее следуют проверки условия завершения l -й реализации ИМ ВТПП₁ ($l \leq N_n$, где N_n – количество реализация ПММ, номер реализации ПММ). Если условия завершения l -й реализации имитации ПММ согласно процедуре Монте-Карло не выполняются, инициализируется процесс PR. UZEL_j переходит в состояние ожидания повторного запуска соответствующим процессом PR.SOST_{ij}. Отметим, что при имитации выполнения процессов ИМ ВТПП₁ накапливается в ТБ следующая статистика: времени жизни ($T_{жj}$) PR.SOST_{xj}, число переходов ПММ в различные состояния ($n_{ПМ}$).

Множество процессов {OBORUD} состоит из двух групп имитаторов функционирования оборудования:

- процессов-имитаторов оборудования общего пользования номера f_r ({PROBOR_{f2}});
- процессов-имитаторов оборудования индивидуального пользования f_1 ({PR.OBIN_{f1}}). Множество результатов {RESURS} включает в себя две группы ресурсов:
 - ресурсы индивидуального пользования номера r_1 {RESIND_{r1}};
 - ресурсы общего пользования номера r_2 {RESOP_{r2}}.

Идея процедуры «ВЫБОР» состоит в следующем. Если один из процессов-имитаторов оборудования с помощью переменной связи процессов установит, что авария на оборудовании произошла ($p_{AB} = 1$), то значение этой глобальной переменной станет доступно процессу PR.SOST_{ij}. Используется вторая глобальная переменная УКАЗ. Если $p_{AB} = 0$ и эта переменная связи УКАЗ = 1, то это указывает, что последующий выбор следующего состояния ИМ ВТПП₁ осуществляется процессом PR.UZEL на основе матрицы переходных вероятностей $\|p_{ij}\|$. Во всех остальных случаях в глобальной переменной связи УКАЗ

содержится заказ на переход к нестандартным состояниям ПММ: ликвидация аварии $PR.SOST_{i7}$; переход на общую профилактику ($PR.SOST_{i8}$); одиночное резервирование оборудования ($PR.SOST_{i9}$); общее резервирование оборудования ($PR.SOST_{i10}$).

В базе данных процесса $PR.UZEL$ (БД) хранится информация о структуре полумарковского процесса: матрица переходов ПММ из состояния в состояние $(\|p_{ij}\|)$; функция распределения числа смен состояний ПММ $F_0(x)$; векторы вероятностей начального и конечного состояний ПММ соответственно V_H и V_R . В каждом ij -м состоянии ВТПП₁ используются следующие типы ресурсов: длительность нахождения ВТПП₁ в состоянии (ϕ_{ij}) стоимость реализации процесса (C_{ij}); количество комплектующих деталей и материалов соответственно (KO_{ij} и mt_{ij}); номер бригады исполнителей (n_6), номер дополнительного исполнителя (n_5), номер ресурса общего пользования (n_4), номер ресурса общего пользования (n_3), номер оборудования общего пользования (n_2), номер оборудования индивидуального пользования (n_1).

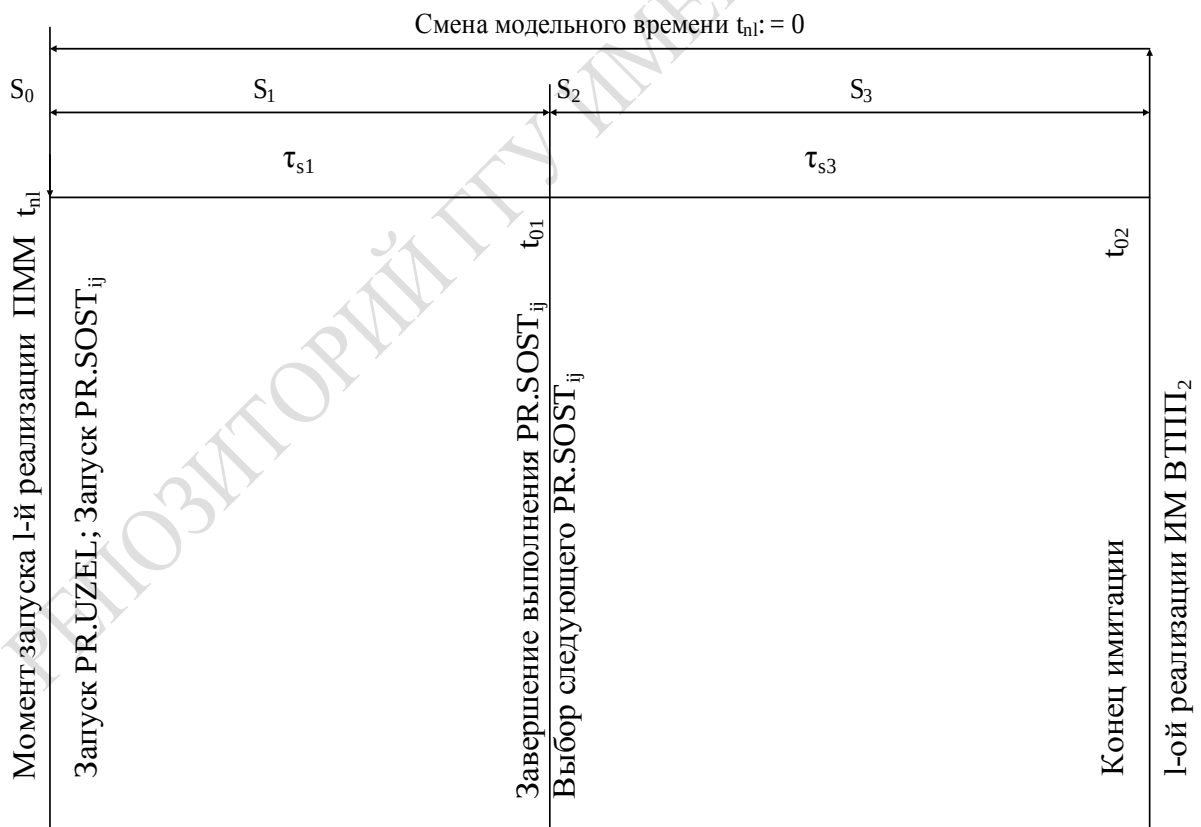


Рисунок 4.10 – Временная диаграмма алгоритма выполнения $PR.UZEL$

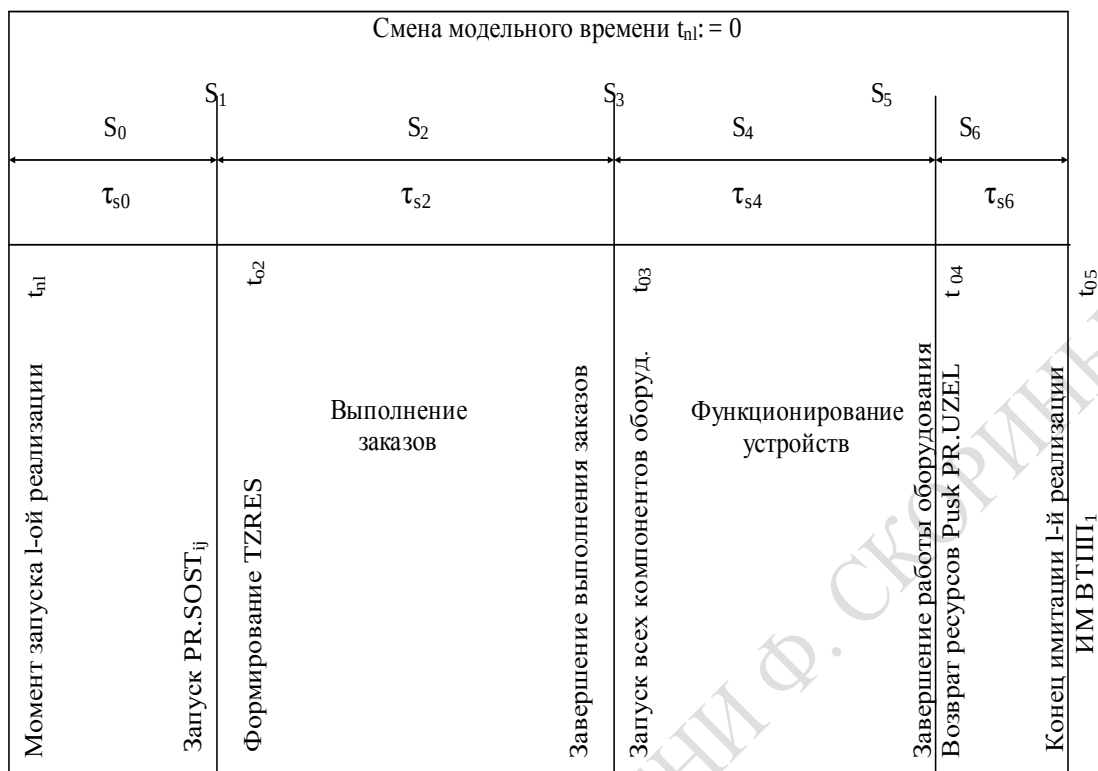


Рисунок 4.11 – Временная диаграмма алгоритма выполнения PR.SOST_{ij}

Первые четыре типа ресурсов являются случайными величинами, и поэтому для каждого PR.SOST_{ij} задаются соответствующие функции распределения:

$$F_{1ij}(\phi); F_{2j}(c); F_{3ij}(ko); F_{4ij}(mt). \quad (4.19)$$

Временная диаграмма алгоритма выполнения PR.UZEL приведена на рисунке 4.10. В моменты $t_0 = 0$ PR.UZEL переходит в начальное состояние (S_0). После выполнения описанных ранее алгоритмических операций PR.UZEL находится в состоянии (S_0) ожидания возврата управления от одного PR.SOST_{ij} длительностью $\tau_{\phi s1}$. По завершении кванта выполнения PR.SOST_{ij} управление возвращается на процесс PR.UZEL в момент t_{01} и процесс PR: UZEL мгновенно в модельном времени (в состоянии S_2) выбирает номер следующего процесса PR.SOST_{ij} и передает ему управление, а сам в состоянии S_3 длительностью $\tau_{\phi s3}$ ожидает следующей инициализации. В момент t_{02} PR.UZEL устанавливает конец l -й реализации ИМ ВТПП₁, модифицирует номер реализации ПИММ и переходит в состояние S_0 .

Для данного примера ВТПП₁ необходимо разработать диаграмму изменения во времени t_0 алгоритма процесса PR.SOST_{ij}. Алгоритм функционирования PR.SOST_{ij} представляет собой выполнение операций в следующих состояниях. При нормальной работе реализуются следующие ситуации:

- ресурсы предприятия не используются, и оборудование функционирует безотказно (PR.SOST_{i3});
- ресурсы используются, но оборудование функционирует с простыми отказами (PR.SOST_{i4});
- ресурсы используются, но при функционировании оборудования возможны простые аварии (PR.SOST_{i5});
- ресурсы используются, но при использовании оборудования возможны сложные аварии, для ликвидации которых необходимо выполнить последовательность процессов ликвидации аварий (PR.PROC_k) и затем уже происходит возврат на PR.SOST_{i6}.

Временная диаграмма алгоритма процесса PR.SOST_{ij} приведена на рисунке 4.11. В момент $t_{ж1}$ процесс PR.SOST_{ij} иницируется процедурой Монте-Карло, определяя начало l -й реализации ИМ ВТПП₁, и процесс PR.UZEL переводит PR.SOST_{ij} в состояние S_1 . В этом состоянии формируется множество запросов процессом ресурсов предприятия следующим образом. Согласно таблице заказов ресурсов TZRES_{ij} по функциям распределения (4.13) формируется для PR.SOST_{ij} множество вероятностных характеристик использования ресурсов: $(\phi_{ijl}, c_{ijl}, ko_{ijl}, mt_{ijl})$.

Из той же таблицы TZRES_{ij} определяются запросы на остальные ресурсы. В состоянии S_1 накапливается статистика расхода ресурсов предприятия. Кроме того, осуществляется учет расхода ресурсов, бригад исполнителей и исполнителей, реализующих все технологические операции в этом состоянии. По завершении выполнения всех запросов процесс PR.SOST_{ij} переходит в состояние S_3 . В этом состоянии процессом PR.SOST_{ij} активизируются также имитаторы устройств индивидуального (номера f_1) и коллективного (номера f_2) использования.

В состоянии S_1 процесс PR.SOST_{ij} ожидает выполнения на оборудовании заказов процесса. В момент t_{04} завершается функционирование оборудования, и возвращаются ресурсы предприятия системе распределения ресурс. После возврата ресурсов в состоянии S_5 иницируется PR.UZEL, и происходит переход в состояние S_6 ожидания окончания имитации l -й реализации до момента t_{05} . В этом случае начинается $(l + 1)$ реализация ИМ ВТПП и цикл имитации повторяется N раз согласно процедуре Монте-Карло.

Формальное описание последовательного дискретного ВТПП₁ должно включать:

- приведенный выше текст описания ВТПП₁;
- схему взаимодействия процессов PR.SOST_j в ВТПП₁ с помощью процесса PR.UZEL (рисунок 2.4);
- временную диаграмму алгоритма выполнения PR.UZEL (рисунок 4.10);
- временную диаграмму алгоритма выполнения PR.SOST_{ij} (рисунок 4.11).

Отметим, что устройства-имитаторы процессов PR.SOST_{ij} и PR.UZEL представляют собой процессы типа приборов массового обслуживания с управлением [14].

Формализация динамики взаимодействия компонентов вероятностного технологического процесса производства с параллельно-последовательным характером его реализации.

Поскольку ВТПП₂, приведенный в **примере 6**, имеет графовую структуру взаимодействия множества {MTXO_{ij}}, то его опишем с помощью вероятностных сетевых графиков (ВСГР). В ВСГР узлами являются события (SOB_i), а дугами представляются MTXO_{ij}, соединяющие SOB_i и SOB_j. В общем случае для выполнения MTXO_{ij} требуются следующие ресурсы ВТПП₂: время выполнения операции (τ_{φijl}), стоимость ее реализации (C_{ijl}), список количества материалов r₁-го типа {mt_{r1ijl}}, список количества комплектующих изделий r₂-го типа {ко_{r2ijl}}; список размеров индивидуальных ресурсов r₃-го типа {V_{r3xjl}}, список размеров общих ресурсов размеры {V_{r4xjl}}, коллективно используемых всеми MTXO_{ij}. Индекс *l* означает номер реализации процедуры Монте-Карло ($l = \overline{1, N}$, N_m – количество реализаций обычно ≥ 30). Эти запросы ресурсов ВТПП являются случайными величинами, задаваемыми до начала имитации с помощью соответствующих функций распределения:

$$F_{1ij}(\tau_\phi); F_{2ij}(C); F_{r1ij}(mt); F_{r2ij}(ko); F_{r3ij}(V_3); F_{r4ij}(V_4). \quad (4.20)$$

Кроме того, для выполнения MTXO_{ij} используются усредненные дифференцированные запросы ресурсов ВТПП, задаваемые с помощью следующих списков: индивидуально используемого оборудования {SP.OBIN_{kij}}; коллективно используемого оборудования общего пользования SP.OBOR_{kij}; индивидуально используемых ресурсов {SP.RES_{kij}}; коллективно используемых ресурсов общего пользования {SP.REOP_{kij}}. Исполнителей операций {SO.ISP_{kij}} бригад исполнителей {STBRIG_{k1j}}. Состав этих списков (иногда только их количества,

когда ресурсы не персонифицированы по $MTXO_{ij}$, определяется особенностями реализации $MTXO_{ij}$ в ВСГР, и поэтому они индивидуальны и детерминированы для каждой $MTXO_{ij}$.

В примере ВСГР, приведенном на рисунке 2.10, согласно содержательному описанию ВТПП₂ пользователь имеет в своем распоряжении все перечисленные функции распределения и детерминированные списки запросов ресурсов, представленные в таблицах 2.3–2.7. При представлении ВТПП₂ аппаратом ВСГР кроме описания $\{MTXO_{ij}\}$ необходимо дополнительно указать состав входов и выходов событий SOB_i и SOB_j , являющихся соответственно инициаторами и преемниками $MTXO_{ij}$. События SOB_i могут иметь a_i и b_i выходов. Как видно из рисунка 4.7, некоторые выходы SOB_i являются кустовыми:

- выход 1 у SOB_2 имеет 2 разветвления ($l = 1, 2$ – номера разветвлений);

- выход 2 у SOB_4 имеет 3 разветвления ($l = \overline{1,3}$);

- выход 1 у SOB_5 имеет 3 разветвления ($l = \overline{1,3}$);

- выход 2 у SOB_1 имеет 3 разветвления ($l = \overline{1,3}$);

- выход 2 у SOB_3 имеет 2 разветвления ($l = 1, 2$).

Остальные SOB_i имеют одиночные выходы. Кустовые выходы SOB_i по своей структуре могут быть 3 типов:

- действительные выходы, инициирующие выполнение всех $MTXO_{ij}$ ($SOB_0, SOB_6 \div SOB_{13}$);

- вероятностные выходы, когда только по одному из разветвлений по вектору вероятностей $\{P_{kl}\}$ инициируется соответствующая $MTXO_{ij}$, и по остальным разветвлениям на выходе формируются фиктивные сигналы, которые обходят выполнение $MTXO_{ij}$ и приходят непосредственно на один из входов SOB_j (вектора вероятностей P_{kl} приведены в таблице 2.5);

- резервные выходы 4-го типа используются для коррекции ВТПП₂ при появлении аварии на оборудовании и в ходе выполнения $MTXO_{sj}$, когда на вход этой $MTXO_{ij}$ приходит сигнал с признаком (рав = 1) «была авария от SOB_1 и SOB_3 на оборудовании используемом одной из $MTXO_{ij}$ до свершения SOB_1 и SOB_3 . В данном примере 6 предполагается, что авария на оборудовании ВТПП₂ происходит при выполнении $MTXO_{1.12}$ и $MTXO_{1.13}$. Поэтому у SOB_1 и SOB_3 инициируются выходы 2 (типа 3), обеспечивая таким образом включение резервных $MTXO_{ij}$ на случай аварии. Резервными $MTXO_{ij}$ в примере 6 (рисунок 4.12) являются: $MTXO_{1.3}$, $MTXO_{1.4}$, $MTXO_{1.12}$, $MTXO_{38}$, $MTXO_{31}$, $MTXO_{5.12}$; $MTXO_{56}$. Наличие при реализации в ВТПП

оборудования может существенно повлиять на динамику реализации последующего $\{MTXO_{ij}\}$. Если авария произошла, то выполняется одна группа $\{MTXO_{ij}\}$, а при отсутствии аварии может выполняться другой состав $MTXO_{ij}$. Переключение событиями SOB_1 и SOB_5 ветвей $MTXO_{ij}$ должно осуществляться оперативно в ходе имитации ВСГР и зависеть от операционной обстановки в системе. Эту функцию выполняют выходы 4-го типа у SOB_1 и SOB_5 с помощью комбинаций фиктивных и действительных сигналов на выходах 4-го типа. Только действительные сигналы активизируют соответствующие $MTXO_{ij}$. Механизм переключения действительных сигналов у кустовых выходов 4-го типа основан на использовании нулевой матрицы коммутации $\|\gamma_{nr}\|$. Наличие единицы на пересечении n -й строки с r -м столбцом этой матрицы означают необходимость включения резервной $\{MTXO_{ih}\}$ в ВСГР в случае, когда на n -й вход приходит сигнал от $MTXO_{ij}$ с признаком $MTXO_{ij}$ «была авария» ($rabr = 1$), который определяет необходимость ликвидации поставарийной ситуации в ВТПП2. Поэтому на r -м разветвлении кустового выхода у SOB_3 и SOB_5 формируется действительный сигнал, если станет истинной булева функция z ; которая вычисляется из выражения $z = grh \wedge rabn$, где n – номера входов, а r – номера разветвлений SOB_5 .

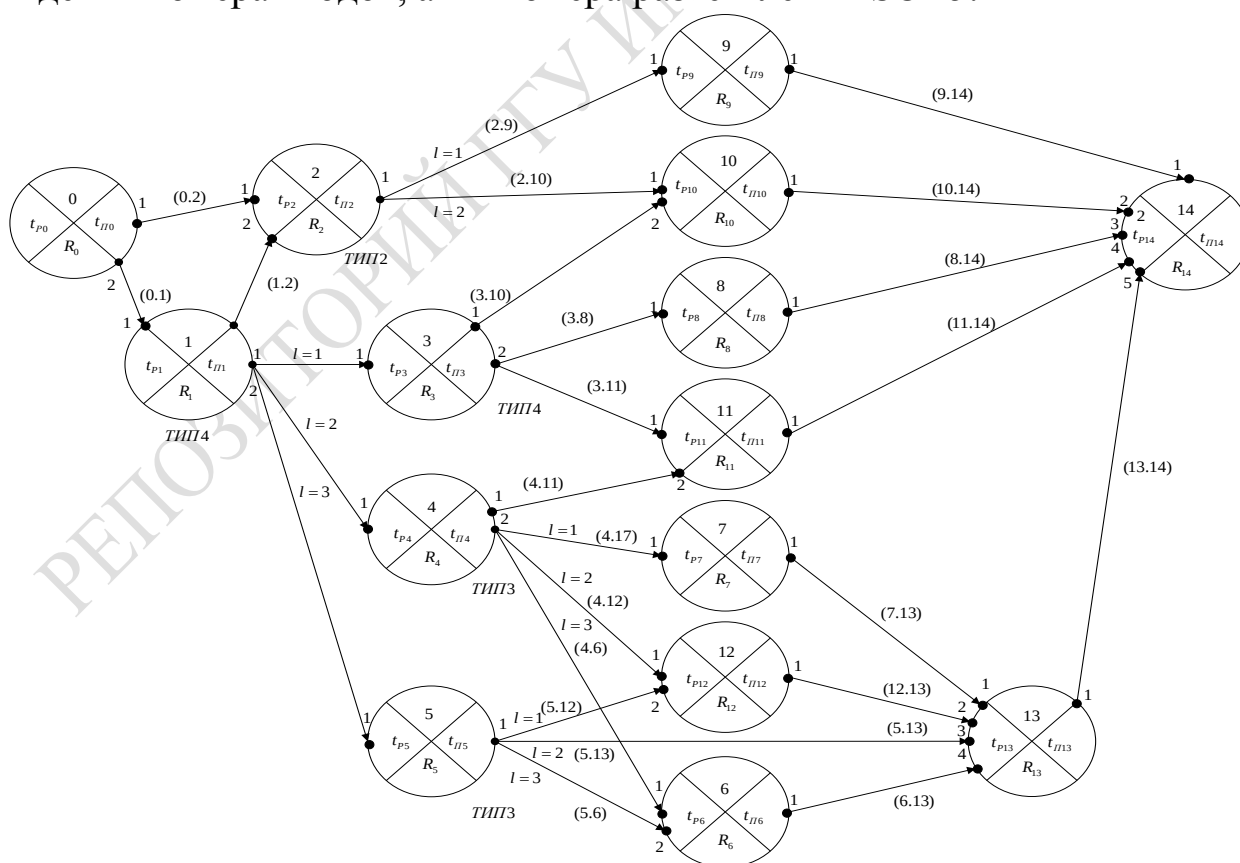


Рисунок 4.12 – Сетевой график вероятностного технологического процесса производства (тестовый пример)

Другую роль играют кустовые выходы 2-го типа у SOB_3 . До имитации были заданы векторы вероятностей формирования на одном из разветвлений действительных сигналов p_{jh} (где h – номер компоненты этого вектора). В l -й реализации ВСГР по жребию 2-го типа разыгрывается номер разветвления h_0 , на котором формируется действительный сигнал, а на остальных разветвлениях формируются фиктивные сигналы. В итоге в ходе l -й реализации ВСГР на выходе SOB_2 активизируется либо $MTXO_{29}$, либо $MTXO_{3,10}$. У выходов 4-го типа (у SOB_1 и SOB_5) формируется запланированная комбинация инициализации $MTXO_{ij}$ с помощью матрицы β_{nr} . Более детально алгоритм выбора разветвлений на выход SOB можно найти в [26].

Особенностью предлагаемой формализации аппаратом ВСГР является использование для определения моментов свершения событий SOB_i двух режимов инициализации ВСГР: прямой имитации, когда модельное время t_0 растет от 0 до свершения последнего события (T_{31}) в l -й реализации ВСГР; инверсной имитации, шага t_0 уменьшается от T_{31} до нуля. Здесь формализация ВСГР сочетается с использованием процедуры Монте-Карло. ВСГР при имитации выполнения $MTXO_{ij}$ по соответствующим функциям распределения разыгрываются фактические значения параметров $MTXO_{ij}$ в l -ой реализации $ВСГР_l$:

$$(\tau_{\phi ij l}, C_{ij l}, \{mt_{r1ij l}\}, \{ko_{r2ij l}\}, \{V_{r4ij l}\}, \{V_{r6ij l}\}). \quad (4.21)$$

По спискам запросов индивидуальных ресурсов, оборудования и исполнителей формируется заказ к системе распределения ресурсов ВТПП₂ для закрепления их за $MTXO_{ij}$ до окончания времени ее выполнения. По этому заказу в распоряжение $MTXO_{ij}$ либо немедленно выделяются свободные ресурсы, либо соответствующие запросы ожидают освобождения требуемого ресурса. По окончании прямой имитации выполнения l -й реализации ВСГР осуществляется переход на инверсную имитацию, когда процесс активизации $MTXO_{ij}$ выполняется в обратном порядке (от конечного события SOB_4 до начального события SOB_0). В этом режиме определяются поздние сроки свершения событий ($t_{\Pi il}$) в l -й реализации ВСГР по формулам:

$$t_{\Pi il} = \min (t_{\Pi jl} - \tau_{\phi ij l}). \quad (4.22)$$

Это означает, что от позднего срока всех j -х событий, следующих за i -м событием, вычитаются продолжительности работ, связывающих i -е событие с j -м, и выбирается минимальная величина. По свершению

исходного события SOB_0 происходит переход в режим прямой имитации ВСГР, но при в $(l + 1)$ -й реализации. Далее рассчитываются резервы времени выполнения событий по формуле:

$$R_{il} := t_{Пil} - t_{pil}. \quad (4.23)$$

Резервы показывают, за какой предельно допустимый период времени можно завершать свершение события i , не изменяя времени свершения завершающего события. Для каждой $MTXO_{ij}$ определяются статистики выполнения $MTXO_{ij}$:

$$\begin{aligned} t_{PHijl} &= t_{pil}; \quad t_{ПНij} = t_{Пjl} - \tau_{фijl}; \\ t_{POijl} &= t_{pil} + \tau_{фijl}; \quad t_{ПОijl} = t_{Пjl}. \end{aligned} \quad (4.24)$$

Когда все статистики свершения событий и выполнения $\{MTXO_{ij}\}$ осуществляются поэтапно, анализ параметров $ВСГР_l$ определяет критический путь KRP_l . Для этого выбираются все события i , для которых нет резервов ($R_{il} = 0$).

KRP_l представляет собой последовательность триад:

$$KRP_l = \{(SOB_i, MTXO_{ij}, SOB_j)\}. \quad (4.25)$$

Начинается эта последовательность триад событием SOB_0 и завершается событием SOB_{14} , в которых имеют место соотношения:

$$t_{pol} = t_{пол} = 0; \quad t_{p14l} = t_{п14l} = T_{kpl}; \quad (4.26)$$

где T_{kpl} – время выполнения l -й реализации ВСГР.

Кроме KRP_l в состав интегральной статистики l -й реализации $ВСГР_l$ входит выборка суммарного расхода R -го ресурса ВТПП l -й реализации $ВСГР$ $\{R_{ORl}\}$, где: $R = 7$ – материалы, $R = 8$ – комплектующие детали; $R = 9$ – стоимость $MTXO_{ij}$. По окончании имитации N_m – реализаций $ВСГР_l$ ($l = \overline{1, N_m}$), где N_m – количество реализаций согласно процедуре Монте-Карло формируется последовательность статистик имитации ВТПП₂ типа 1:

$$ZAP_1 = (\{KRP_l\}, \{T_{kpl}\}, \{R_{ozl}\}, \{R_{okl}\}, \{R_{oql}\}, BDOS_1, BDOS_2);$$

где $\{KRP_l\}$ – множество критических путей;

$\{T_{kpl}\}$ – выборка значений критических времен выполнения $ВСГР_l$;

$\{R_{okl}\}$ – графики расхода ресурсов R -го типа при реализации $ВСГР_l$;

BDOS_{1r4} и BDOS_{2r2} – временные диаграммы изменения свободного места собственно на общих ресурсах номера $r4$ и на оборудовании общего пользования номера $r2$.

Затем формируются следующие множества записей интегральной статистики имитации ВСГР:

типа 2 – $\{\text{STSOB}_{il}\}$; $i = \overline{0,14}$; статистика свершения событий (t_{pil} , $t_{\Pi il}$, R_i);

типа 3 – $\{\text{STMTXO}_{ijl}\}$; статистика выполнения операций (t_{pnijl} , $t_{\Pi nijl}$, $t_{poi jl}$, $t_{\Pi oi jl}$);

типа 4 – $\{\text{STINR}_{r1l}\}$ и $\{\text{STOBR}_{r2l}\}$ – статистика использования ресурсов индивидуального и общего использования;

типа 5 – $\{\text{STOBIN}_{r1l}\}$ и $\{\text{STOBIN}_{r2l}\}$ – статистика использования оборудования индивидуального пользования.

В этом файле статистики содержится необходимая информация для ее усреднения согласно процедуре Монте-Карло. По записям ZAP_1 формируется граф критических путей GKRP путем наложения друг на друга KPP_t . При этом определяются вероятности разветвлений (q_{ij}). Используя записи типа 2 и типа 3 и факт записи статистики путем усреднений, по окончании N_m реализаций определяют оценки математических ожиданий времен свершения событий $(\bar{t}_{pi}, \bar{t}_{\Pi i}, \bar{R}_i)$, времен реализации MTXO_{ij} , $(\bar{\tau}_{ij})$ и дисперсий $(S_{\tau ij}^2)$ времен выполнения MTXO_{ij} в GKRP . По записям типа 4 и типа 5 определяются интегральные статистики расхода ресурсов в моменты свершения событий SOB_i . Имея в своем распоряжении GKRP , состоящий из множеств $\{(\bar{t}_{pi}, \bar{t}_{\Pi i}, \bar{R}_i)\}$, $\{(\bar{\tau}_{ij})\}$ и $\{q_{ij}\}$ исследователь получает усредненное значение $\bar{T}_{kr}$ реализации ВСГР.

В итоге все множество статистик имитации заменяется либо средними значениями и их дисперсиями, либо соответствующими функциями распределения. Поскольку количество замеров временных диаграмм использования ресурсов в моменты свершения событий весьма большое, то при обработке в N реализациях диаграмм BDOS_{1r4} и BDOS_{2r2} , а также графиков расхода ресурсов R -го типа $\{R_{ORI}\}$ путем наложения этих графиков друг на друга формируются усредненные временные диаграммы изменения остатка места на общих ресурсах и усредненные графики расхода ресурсов при свершении событий на ВСГР.

После обработки файла статистики для параллельно-последовательного ВТПП_2 формальное описание включает в себя:

– приведенный выше текст формализации ВТПП_2 с помощью ВСГР;

- структуру связей $\{MTXO_{ij}\}$ и $\{SOB_i\}$ в ВТПП₂ (рисунок 2.10);
- блок-схему концептуальной модели вероятностного параллельно-последовательного ВТПП₂ (рисунок 3.6);
- пример вероятностного сетевого графика технологического процесса производства (рисунок 4.12);
- множество G_{z2} запросов $\{MTXO_{ij}\}$ ресурсов предприятия ВТПП₂ (таблица 3.4);
- состав параметров и откликов ИМ ВТПП₂ (таблица 3.5);
- множество характеристик надежности функционирования оборудования ВТПП₂ (таблица 3.6).

Отметим, что в качестве имитаторов $MTXO_{ij}$ и SOB_i используются соответствующие агрегаты. Алгоритм функционирования агрегатов, составляющих в совокупности ИМ ВСГР, разрабатывается на следующем этапе технологии имитационного моделирования ВТПП₂ (см. 5.6).

4.6 Итоги

Подведем итоги. В 4.1 формулируется общая методика формализации объекта моделирования, включающая выполнение следующих действий исследователя: уточнение декомпозиции СС для каждого способа имитации; алгоритмизация компонент ИМ; информационная стыковка алгоритмов компонент модели; установление управляющих связей между компонентами ИМ; уточнение взаимодействий с УПМ; документация ИМ. В 4.2 демонстрируется общая методика при формализации ВП в ЛВС алгоритмом полумарковского процесса. В лабораторной работе 4.1 формализуется ВП в ЛВС при создании ИМ «ТОПОЛОГИЯ ЛВС» и «УЗЛА ЛВС», когда объектом формализации ВП является полумарковская модель. Другой пример демонстрации общей методики является построение ИМ ВП узла ЛВС для случая, когда ИМ используется для адаптации РН к составу ресурсов узла ЛВС рассматривается в 4.3. Демонстрируется формализация ВП узла ЛВС путем использования транзактов сложной структуры, в «теле» которого содержится информационно-подкрашенная структура РН на узле ЛВС.

Для второго типа предметной области (поточные операции ремонта изделий) в 4.4 демонстрируется применение общей методики формализации. В 4.3 излагается технология формализации динамики функционирования участка обработки деталей несколькими станками. В 4.4 приводится общая методика формализации технологического процесса ремонта изделия D сложной структуры.

Следующий пример демонстрации общей методики формализации СС для случая, когда объектами имитации являются ВТПП, приведен в 4.5. В 4.5 формализуется последовательный ВТПП₁ аппаратом полумарковских процессов на основе результатов полученных в 2.5. В 4.6 формализуется динамика взаимодействия компонентов ВТПП₂ с параллельно-последовательным характером его реализации на основе результатов полученных в 2.6.

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

5 ПРЕОБРАЗОВАНИЕ ФОРМАЛЬНОГО ОПИСАНИЯ СЛОЖНОЙ СИСТЕМЫ В ИМИТАЦИОННУЮ МОДЕЛЬ

5.1 Методика преобразования формального описания СС в имитационную модель

На этом этапе исследователь обычно выполняет следующую последовательность действий: обеспечивает внутреннюю и внешнюю синхронизации компонент модели; синхронизирует моменты появления информации от одной компоненты модели к другой; описывает поведение компонент при появлении конфликтных ситуаций в модели; уточняет состав исходной информации; составляет описание программы окончания имитации; устанавливает контроль за ходом имитации; уточняет состав документации на моделирование. Объем и содержание работ при выполнении указанной последовательности действий зависят от способа имитации и сложности объекта моделирования. Тем не менее можно говорить об установившейся методике реализации указанных действий, к изложению которой мы переходим.

Внутренняя синхронизация компонент модели. Переход от формального описания к описанию имитационной модели системы также осуществляется на основе декомпозиции сложной системы на составные части. Для каждой компоненты сложной системы, кроме указания ее алгоритмических функций, необходимо установить временную координату функционирования. Добавление операторов преобразования временной координаты компоненты модели является первым шагом превращения формального описания объекта моделирования в его имитационную модель.

В ряде случаев из-за больших ошибок аппроксимации явлений отдельными активностями для процессного и событийного способов имитации, а также и при имитации активностями может появиться необходимость дополнительного разделения компонент (активностей, подпрограмм обслуживания событий, процессов) на несколько более мелких, которые выполняются с меньшим интервалом τ_{ij} изменения временной координаты. Очевидно, что при этом алгоритмы соответствующих активностей тоже как бы распадаются на несколько частей. Подобного рода работы и составляют основное содержание процедуры установления временных координат функционирования компонент модели.

Для транзактного и агрегатного способов имитации такая процедура является лишней, поскольку она выполнена уже на этапе формализации объекта моделирования и изменение временных координат агрегатов или блоков модели унифицировано. При этом весьма важно обеспечить внутреннюю и внешнюю синхронизации компонент модели в модельном времени.

Внутренняя синхронизация компонент обеспечивается самими компонентами с помощью семафоров. Под семафором обычно понимают некоторую глобальную переменную, значение которой либо разрешает, либо запрещает активизацию имитатора компоненты в модельном времени. В случае запрета активизации все компоненты модели выстраиваются в очередь на активизацию согласно их приоритетам. Сброс семафора выполняется одной или несколькими компонентами и приводит к немедленной активизации всех ждущих этого семафора компонент модели. Установка же семафора может немедленно породить очередь компонент на активизацию. Семафоры могут устанавливаться с различной целью. Например, при использовании общего ресурса сложной системы какой-либо компонентной выставляется семафор для всех конкурирующих компонент, а после освобождения этого ресурса семафор сбрасывается.

Отметим, что организация внутренней синхронизации компонент модели, например с помощью семафоров, обычно не используется при транзактном и агрегатном способах. Для транзактного способа имитации сами очереди к блокам являются хорошим средством внутренней синхронизации и семафоры не нужны. В случае агрегатного способа имитации внутренняя синхронизация выполнена уже на этапе составления формального описания, когда исследователь либо организует взаимодействие координат агрегата, либо использует управляющие и входные сигналы для имитации взаимодействия агрегатов. Зато при процессном способе имитации для обеспечения внутренней синхронизации зачастую кроме семафоров требуется установка операторов взаимодействия процессов друг с другом и выбор дисциплин обслуживания очередей к процессам для их информационного обмена. На данном этапе разработки имитационной модели поведение исследователя определяется выбранным ранее способом формализации объекта моделирования.

При агрегатном способе имитации новой модификации [26] в качестве элементов внутренней синхронизации используются агрегаты-синхронизаторы, которые формируют выходные сигналы на агрегаты-исполнители. Возможно использование двух типов элементов организации взаимодействия агрегатов:

– агрегатов-имитаторов $AGLAST_j$, которые активизируются в моменты прихода на их входы самого позднего сигнала от агрегатов-исполнителей функций систем управления;

– агрегатов-имитаторов $AGFIRST_j$, которые активизируются в моменты прихода на любой из их входов самого раннего по времени сигнала от агрегатов исполнителей функций $AGISPF_{ij}$ (здесь i и j – номера агрегатов-синхронизаторов).

В моменты активизации $AGLAST_j$ и $AGFIRST_j$ со всех выходов этих агрегатов формируются сигналы, активизирующие агрегаты $AGISPF_{ij}$. Агрегаты-синхронизаторы $AGLAST_j$ и $AGFIRST_j$ представляют собой сложным образом организованные логико-функциональные схемы управления, соответствующие логическим схемам «И» и «ИЛИ». На вход любого из агрегатов-синхронизаторов может поступать от агрегатов $AGISPF_{ij}$ любое число сигналов. Как видим, по сравнению с кусочно-линейными агрегатами Н. П. Бусленко [3], имеет место существенное усложнение функций агрегатов и, как следствие, $AGLAST_j$, $AGFIRST_j$, $AGISPF_{ij}$ гораздо более сложные по структуре и составу, чем кусочно-линейные агрегаты Н. П. Бусленко.

При агрегатно-процессном способе имитации, в случае использования вероятностных сетевых графиков (ВСПР) в качестве средств внутренней синхронизации имитационных моделей ВТПП, используются агрегаты-имитаторы событий ($ASOB_j$), на входы которых поступают сигналы от множества агрегатов-имитаторов микротехнологических операций ($АТОР_{ij}$). Как показано в [26], $ASOB_i$ могут иметь любое число входов, обеспечивая таким образом синхронизацию во времени и по готовности информации для дальнейшего развития ВТПП различных цепочек $\{АТОР_{ij}\}$. В моменты прихода самого позднего сигнала на входы $ASOB_j$ от $\{АТОР_{ij}\}$ срабатывает «спусковая функция», активизирующая агрегат $ASOR_j$. В моменты активизации $ASOB_j$ со всех своих выходов формирует два типа сигналов: действительные (Sgd) и фиктивные (Sgf). Действительные сигналы Sgd активизируют агрегаты $АТОР_j$, а Sgf обходят выполнение алгоритма $АТОР_{js}$ и поступают непосредственно на входы $ASOB_i$. В общем случае, согласно [26] все выходы нумеруются и могут быть кустовыми. На k -м выходе число разветвлений может быть равно dk и поэтому по каждому l -му разветвлению может формироваться либо Sgd , либо Sgf ($l = \overline{1, dk}$). Выходов может быть 4 типа: действительные (1-го типа) всегда формируют только Sgd на всех разветвлениях), вероятностные (2-го типа) формируются на одном из разветвлений Sgd , а с остальных выходов поступают Sgf , (3-го типа) условного управления. Тогда

состав сигналов на каждом разветвлении может представлять собой смесь Sgd и Sgt. Sgd формируется только в том случае, когда становится истинной функция:

$$Z = (\pi_{abm} \wedge \gamma_{mz}), \quad (5.1)$$

где $\pi_{abm} = 1$ – указатель «была авария оборудования ВТПП во время выполнения АТОР_{hj}, посылающего сигнал на m -й вход ASOB_j;

$\lambda_{mz} = 1$ – признак, означающий, что на r -м разветвлении k -го выхода ASOB_j нужно сформировать действительный сигнал Sgd.

Как видим, с помощью {ASOB_j} и механизма формирования Sgd на выходах и агрегатов появляются очень широкие возможности реализации внутренней синхронизации ветвей {АТОР_{ij}} с помощью агрегатов ASOB_j.

Внешняя синхронизация компонент модели. Такая синхронизация чаще всего реализуется системой моделирования, которую выбрал исследователь. Для этой цели используются операторы синхронизации компонент в системе дискретных событий. Например, в случае процессного представления различаем два типа операторов синхронизации: немедленного перевода процесса в ждущее состояние (типа WAIT (τ), где τ – длительность задержки процесса в ждущем состоянии) и перевода процесса в ждущее состояние до выполнения некоторого условия A (типа WAIT WHILE (A)). Состав, синтаксис и семантика операторов синхронизации зависят от используемой системы моделирования. Однако одно существенное свойство этих операторов является общим для любых моделей. Речь идет о делении этими операторами алгоритма функционирования компонент ИМ на кванты.

Выбор мест установки операторов синхронизации процессов модели является искусственным. В основу деления алгоритмов процессов на кванты необходимо положить точностные аспекты имитации явлений и процессов. Подобным же образом исследователь поступает и при имитации активностями или событийном способе имитации. Для транзактного способа имитации проводить внешнюю синхронизацию компонент модели не требуется, поскольку она уже выполнена на этапе формализации, когда осуществлялась стыковка источников и поглотителей транзактов с алгоритмом функционирования объекта по схеме модели объекта. Агрегатный способ имитации также не требует организации внешней синхронизации, поскольку эти работы выполнены на этапе формализации и, кроме того, все взаимодействия агрегатов осуществляются через УМП, и весьма унифицированно. При агрегатно-процессном способе имитации УПМ дополнительно использует набор операторов ожидания прихода сигналов на входы ASOB_j

или активизации $АТОР_{ij}$ [26]. Здесь также наблюдается высокая степень унификации взаимодействия ИМ типа ВСГР с УПМ.

Синхронизация моментов появления информации для компонент модели. Важным моментом при построении модели сложной системы является синхронизация появления информации, доступной для ее активных компонент. Иногда своевременное обновление информации, общей для компонент, имеет решающее значение. Например, важно, чтобы результаты выполнения алгоритма функционирования n -го кванта процесса m появились в общем поле информации, доступной для компоненты, а именно в момент t_1 . Появление этих результатов раньше или позже момента t_1 может существенно исказить динамику имитации взаимодействия компонент сложной системы. Чтобы обеспечить такую синхронизацию моментов доступности информации к компонентам модели (например, процессам), часто прибегают к следующему приему. Используют рабочий массив, который доступен только для данного процесса, и оператор условного ожидания момента времени t_1 (WAIT WHILE (t_1)). При наступлении момента t_1 происходит перезапись рабочего массива в общее поле информации процессов модели. Аналогично может осуществляться синхронизация моментов доступности информации компонентам модели при имитации активностями или событиями. В более простых случаях для синхронизации моментов доступности информации компонентам модели достаточно использовать глобальные переменные модели. Отметим, что агрегатный и транзактный способы имитации не обеспечивают подобной синхронизации взаимодействия между компонентами модели. Таким образом, после установки операторов внутренней и внешней синхронизаций алгоритмы компонент модели оказываются разделенными на кванты. Именно в этих местах в ходе имитации и будет происходить обращение к управляющей программе моделирования выбранной системы моделирования. В случае агрегатно-процессного способа имитации при использовании аппарата сетевого планирования сама структура ВСГР представляет собой оригинальный способ синхронизации моментов появления информации в ИМ ВТПП.

Отражение конфликтных ситуаций в модели. Следующим шагом при преобразовании формального описания сложной системы в описание имитационной модели является отражение в ней возможных конфликтных ситуаций, с которыми может столкнуться УПМ в ходе имитации. Мы уже рассматривали случай общего использования ресурсов системы различными компонентами модели и применения для этой цели семафоров. Не менее важным моментом является синхронизация процессов друг с другом с помощью операторов запуска

и останова процессов. Почти во всех системах моделирования алгоритмического типа, построенных на принципах процессного представления объекта моделирования, присутствуют операторы запуска и останова процессов. Очевидно, что в таких случаях в модели могут происходить одновременные запуски процессов. Как правило, отражение конфликтных ситуаций при этом берут на себя системы моделирования, и чаще всего они используют системы приоритетов самих процессов. Поэтому на шаге преобразования формального описания объекта в его имитационную модель исследователь уточняет приоритеты компонент модели (активностей, событий, процессов, транзактов, управляющих и входных сигналов агрегатов). При транзактном и процессном способах имитации в случае использования очередей транзактов к блокам или процессов к общим ресурсам системы исследователю приходится изменять дисциплины обслуживания этих очередей.

В ряде моделей сложных систем весьма существенно отражение информационных связей между компонентами в информационной базе модели. В тех случаях, когда обмен информацией между компонентами системы является средством подачи исходных данных для выполнения алгоритмов функционирования компонент модели, составляются модели массивов. Модель массива представляет собой набор характеристик размера массива и параметров, на которые реагируют алгоритмы компонент модели. Если необходима имитация передачи массивов между компонентами модели, то они оформляются в виде «подкрашенных» требований. Подкраской требований является набор характеристик модели массива, хранящийся в информационной базе. По модели в этом случае циркулирует только идентификатор (адрес) модели массива, который выбирается из очереди очередной компонентой модели. По этому идентификатору компоненты модели определяют местоположение информации в модели массива. Затем уже данные, находящиеся в модели по указанному адресу, компоненты модели используют при отражении алгоритма их функционирования.

В случае транзактно-процессного способа имитации [14] транзакты имеют сложную структуру и по своей сути зачастую являются динамическими элементами ИМ, заголовок которых движется от одного процесса в очередь к другому процессу. В «теле» такого транзакта могут быть размещены: программа выбора дальнейшего пути транзакта по ИМ; индивидуальные рабочие переменные, статистика имитации, программа оперативной обработки этой статистики. Как видим, добавление свойств информационной подкраски транзактов

существенно расширило возможности систем моделирования в предупреждении и разрешении конфликтных ситуаций в ИМ.

Уточнение исходной информации для моделирования. Следующим шагом преобразования формального описания в описание имитационной модели является уточнение исходной информации для моделирования. Чаще всего набор действий по заданию исходной информации для компонент модели оформляется в виде блока задания начальных условий имитации. Алгоритм блока задания начальных условий реализуется один раз в начале имитации мгновенно в модельном времени. В ходе задания начальных условий устанавливаются режимы моделирования. Состав действий зависит от способа имитации.

Например, при *процессном* подходе к имитации для этой цели используется специальный процесс, именуемый инициатором модели. Инициатор модели устанавливает каждый из процессов модели в начальное состояние перед очередным прогоном имитационного эксперимента на ЭВМ. В функции инициатора модели входит также выдача исследователю документальной информации о начальных условиях процесса имитации. В ряде случаев в функции инициатора входят также задание и модификация параметров моделирования. Обычно при реализации инициатора осуществляется замена формальных параметров модели фактическими значениями для данного варианта имитации функционирования большой системы. Поэтому исследователь в алгоритме инициатора должен предусмотреть вопросы подстановки фактических значений на место формальных параметров.

При имитации *активностями* также выделяются активности-лидеры, т. е. такие активности, в функции которых входит установка остальных активностей в начальное состояние. Для запуска таких служебных активностей обычно выделяется некоторая глобальная переменная модели, указывающая на режим «разгона» модели и изменяемая активностью-лидером по окончании установки остальных активностей в начальное состояние.

Для *событийного* способа имитации выделяется специальное начальное состояние модели, которое обслуживается соответствующей подпрограммой обслуживания событий. При реализации подпрограммы обслуживания начального события устанавливаются начальные состояния всех компонент модели и назначаются соответствующие условия их свершения.

При *транзактном* способе имитации устанавливаются начальные значения параметров источников транзактов, состояние очередей, а также занятость блоков обслуживания транзактов, с которых необходимо начать имитацию.

Такой режим работы модели до появления начального состояния и называют «разгоном» имитационной модели. В случае *агрегатного* способа имитации дело упрощается, поскольку для каждого агрегата описано начальное состояние. Система автоматизации моделирования, выполняющая агрегатный способ имитации, реализует операторы задания начальных условий. Причем от исследователя никакая дополнительная работа на данном шаге перехода к имитационной модели не требуется, поскольку он выполнил эту работу уже на этапе формального описания системы.

При *транзактно-процессном* способе имитации роль «разгона» ИМ в начальное состояние возрастает, либо следует предусмотреть ситуацию, когда по завершении переходного периода необходимо всю накопленную информацию уничтожить, но оставить состояние очередей и транзактов процессов, которые получены на момент стабилизации переходного процесса и обнулить все значения откликов ИМ. При *агрегатно-процессном* способе имитации очередей между процессами нет, сигналы мгновенно формируются в начале каждой реализации ИМ ВТПП, и поэтому период «разгона» вероятностного сетевого графика ВТПП отсутствует.

Организация окончания имитации. Следующим моментом в разработке модели является задание условий окончания имитации очередного варианта моделирования сложной системы в виде специального блока модели, контролирующего ход имитации и называемого финальным. Его алгоритм функционирования отражает последовательность действий по контролю за моментами окончания имитации и начала обработки результатов моделирования. В некоторых случаях в функции финального блока входит также планирование очередного имитационного эксперимента. Иногда контроль за моментом окончания имитации может вестись с заданным шагом изменения модельного времени. В эти моменты финальный блок активизируется и проверяет выполнение условий окончания имитации. При выполнении условий окончания моделирования процесс имитации завершается, модельное время фиксируется, и в дальнейшем алгоритм финального блока или процесса реализуется мгновенно в модельном времени.

В случае агрегатно-процессного способа имитации и использования ВСГР необходимо предусмотреть момент окончания реализаций ВСГР и перехода на начало 1-й реализации ИМ. Например, в системе автоматизации имитационного моделирования САИМ, эту функцию выполняет специальная подсистема MONTEK, обеспечивающая реализацию последовательности ИЭ согласно методу статистических испытаний [26].

Контроль за ходом имитации. В некоторых случаях исследователю необходимо следить за ходом имитации и иметь возможность принять решение о прекращении имитации. При процессном способе имитации для этой цели необходимо предусмотреть блок или процесс контроля за ходом имитации. Алгоритм блока или процесса контроля зависит от состава и структуры модели. В нем предусматривается его активизация с заданным шагом. В ходе активизации формируется осведомительная информация о состоянии модели, которая выдается либо на печать, либо на экран терминального устройства.

В тех случаях, когда предполагается длительная работа имитационной модели на ЭВМ, в функции блока контроля может входить копирование во внешней памяти ЭВМ состояния модели. Это обеспечивает возможность возобновления работы модели с прерванного места при появлении отказов в ЭВМ. Для этой цели используются либо стандартные средства математического обеспечения ЭВМ, на которой реализуется модель, либо специальные средства возобновления имитации, предоставляемые системой моделирования. Подобные же действия должен предусмотреть исследователь при имитации активностями или подпрограммами обслуживания событий. В этом случае разрабатываются процедуры проверки выполнимости условий наступления контролируемой ситуации в модели и фиксации экспресс-информации о ходе имитации. Для транзактного способа имитации оперативный контроль за ходом имитации организовать довольно трудно, да он в большинстве случаев и не нужен. При агрегатном способе имитации исследователь должен заранее побеспокоиться о контроле за поведением агрегатов с помощью контрольных координат изменения состояний агрегатов. Сама организация квазипараллельной работы агрегатов стандартизована. Для агрегатно-процессного способа имитации в случае использования ВСГР оперативный контроль имитации ВТПП является необходимым после каждой l -й реализации. Поскольку процесс перехода на очередную реализацию осуществляет процедура MONTEK [26], то необходимо предусмотреть промежуточную выдачу информации о динамике свершения событий после каждой реализации ИМ ВСГР. Для этой цели в составе языка моделирования системы САИМ имеются соответствующие управляющие операторы имитации ВСГР.

Организация сбора статистики. Завершается преобразование формального описания в описание имитационной модели выбором мест установки операторов сбора статистики. Выбор места фиксации поведения объекта моделирования также является искусством исследователя. Существует множество способов фиксации статистики

моделирования. Прежде всего это применение так называемых «наблюдателей» модели. Трассировка алгоритмов компонент модели с помощью наблюдателей позволяет отслеживать как развитие их функций во времени, так и их взаимодействие друг с другом. Каждый такой оператор-наблюдатель изменяет значения некоторого счетчика моментов своей активизации и записывает их во внешнюю память. Эта информация используется затем финальным блоком при обработке статистики моделирования. Иногда важно фиксировать времена жизни процесса в модели. Например, при процессном представлении для этой цели фиксируются момент создания (или первой активизации) процесса и момент его уничтожения (или последней пассивизации процесса). Разница между этими моментами и представляет собой время жизни объекта.

В некоторых системах моделирования в распоряжение исследователя предоставляется набор операторов сбора статистики. Эти операторы исследователь помещает в выбранные места в алгоритмах процессов, разрывая дополнительно временные интервалы функционирования процессов, образованные операторами синхронизации. Ряд систем моделирования не требует от исследователя установки операторов сбора статистики, обеспечивая ему сбор стандартной статистики о функционировании компонент модели. Чаще всего в состав такой статистики входят следующие характеристики: число взаимодействий компонент модели, общее время и коэффициент использования компоненты модели, число отказов функционирования компоненты системы и время, в течение которого она находилась в аварийном состоянии.

В ряде случаев стандартной статистики функционирования процессов исследователю недостаточно. Мало информации дает также и трассировка модели с помощью наблюдателей. Тогда при процессном и событийном способах формализации (а также при имитации активностями) исследователю приходится заводить специальные процессы, события, активности, фиксирующие статистическую информацию о моделируемом объекте. Такие специальные процессы (события или активности) могут активизироваться либо с заданным шагом модельного времени, либо при выполнении логических условий в ходе имитационного эксперимента на ЭВМ. В обоих случаях исследователь должен предусмотреть в алгоритме финального процесса набор процедур для вычисления статистик моделирования и на их основе определить отклики имитационной модели согласно выбранным критериям качества вариантов моделирования большой системы. Для агрегатного и транзактного способов сбор статистики стандартизован и данный

шаг перехода от формального описания к описанию имитационной модели отсутствует.

В случае агрегатно-процессного способа имитации при использовании аппарата описания ИМ в виде ВСГР сама система моделирования САИМ автоматически включает подсистему обработки статистики имитации для завершения усреднения статистик и откликов согласно процедуре Монте-Карло [26].

Документирование. Завершается этап перехода к описанию имитационной модели уточнением состава документации на модель. Результатом работы исследователя на данном этапе является описание имитационной модели, готовое к программированию на выбранном языке моделирования. Для событийного и процессного способов имитации, а также при использовании активностей размер имитационной модели существенно увеличивается по сравнению с формальным описанием за счет введения дополнительных блоков и операторов моделирования. При агрегатном и транзактном способах имитации этот этап практически вырождается и имитационная модель мало чем отличается от формального описания. В этом одно из достоинств данных методов и причина их большой оперативности в разработке имитационной модели.

5.2 Представление имитационной модели вычислительного процесса в ЛВС аппаратом полумарковских процессов

На этапе компоновки ИМ топология ЛВС из библиотеки элементов ПТКИ ЛВС LIB.COMLVS из универсальных параметризованных элементов ИМ вначале была сформирована в библиотеке LIB.TOPOL имитационная модель [8], состоящая из группы активностей, имитирующих обслуживание запросов пользователей, согласно графу верхнего уровня представления ЛВС (GR_1).

Построение полумарковской модели РН и ВП узла ЛВС. Кроме основных компонентов в состав ИМ комплексом ПТКИ ЛВС [24] из библиотеки элементов добавляются блоки начала и окончания имитации, блок сбора статистики, процедура PR.MONTEK и УПМ. На рисунке 5.1 приведена блок-схема программы ИМ Топология ЛВС. Связь активностей ИМ с управляющей программой моделирования осуществляется с помощью операторов ожидания WAIT(τ_{ij}), а также с помощью набора процедур и функций, являющихся имитаторами

устройств, функциональное назначение которых в составе модели приведено в таблице 5.1. Возможные пути движения транзактов по ИМ топологии ЛВС приведены в таблице 5.2. Поскольку для реализации одного варианта состава ресурсов ЛВС используется метод статистических испытаний (l -й реализации процедуры Монте-Карло), то по завершении имитации УМП перейдет управление на PR.MONTEK. Эта процедура повторяет эксперименты с ИМ топологии ЛВС, меняя только начальные значения базовых генераторов псевдослучайных величин ξ , заданные до начала имитации, N_m раз. Только при проведении N_m экспериментов при одних и тех же значениях параметров имитации процедуры PR.MONTEK передает управление на блок окончания имитации, который формирует результирующую запись в файл статистики. В ходе имитации всех реализаций блок сбора статистики (БСС) формирует в ходе l -й реализации статистику имитации l -й реализации ИМ и запоминает множество статистик и откликов, состав которых определен ранее на этапе формализации ЛВС (верхнего уровня детализации «Топология ЛВС»).

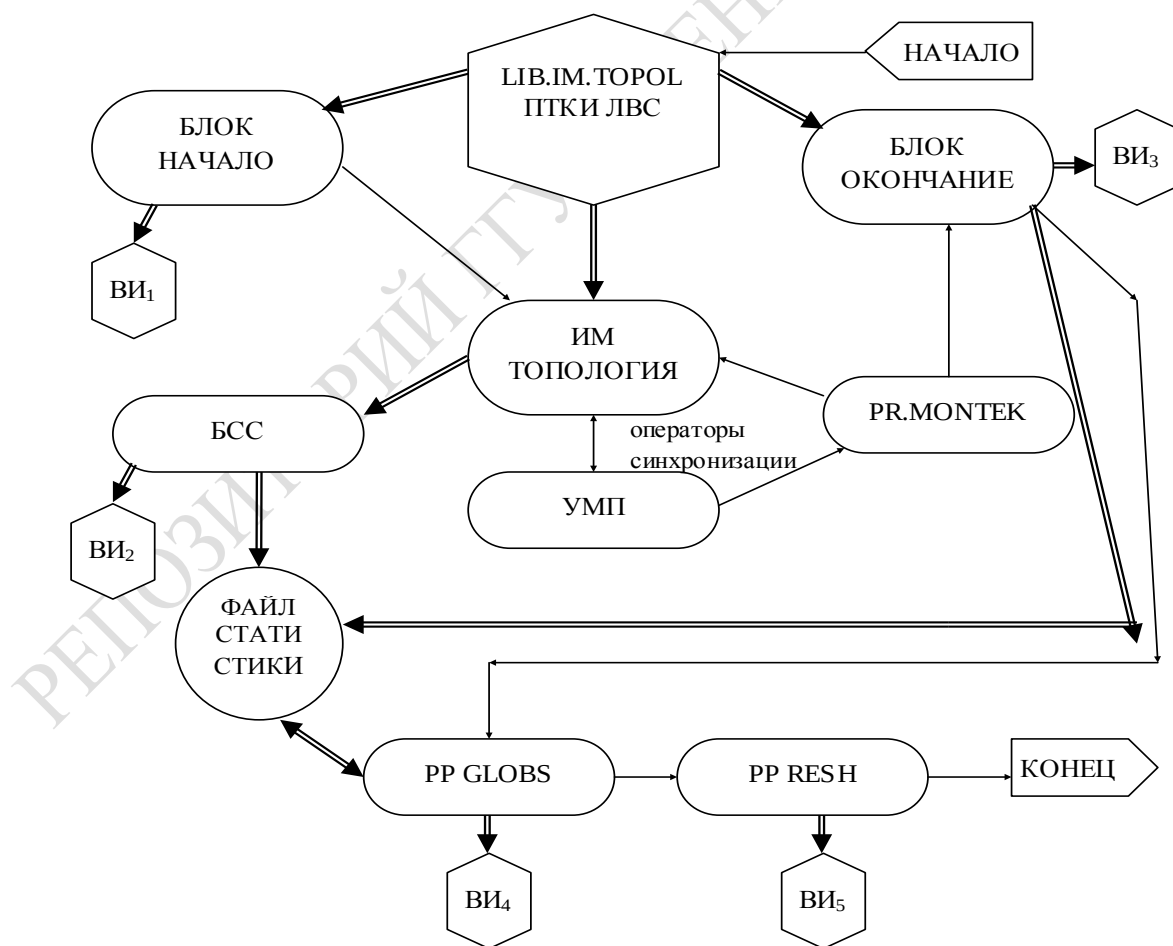


Рисунок 5.1 – Блок-схема программы имитационной модели «Топология ЛВС»

Таблица 5.1 – Состав библиотеки описаний компонентов ИМ ВП на ЛВС (LIB.COMLVS)

Уравнение детализ. ВП	Обозначение процедуры функций	Назначение процедур и функций
УД «ТОПОЛОГИЯ ЛВС»	IN.NET _i	Имитатор обработки на ЦП <i>i</i> -го узла запросов пользователей, уходящих с данного узла в сеть
	FON _i	Имитатор обработки на ЦП <i>i</i> -го узла фоновых запросов
	DIAL _i	Имитатор обработки на ЦП диалоговых запросов пользователей
	CALCL _i	Имитатор обработки на ЦП счетных заданий пользователей
	MIN; MOVT	Имитатор маршрутизаторов запросов пользователей внутри <i>i</i> -го узла
	SYSP _i	Имитатор работы УП <i>i</i> -го узла ЛВС
	DISP _i	Имитатор алгоритма диспетчера запросов пользователя <i>i</i> -го узла ЛВС
	DTR _i	Имитатор алгоритма диспетчера транзактных запросов пользователей <i>i</i> -го узла ЛВС
	REZ _i	Имитатор захвата запросов пользователей ресурса внешней памяти <i>i</i> -го узла ЛВС
	CMT _i	Имитатор алгоритма коммутатора сети
	CANAL _i	Имитатор передачи информации по каналом связи <i>i</i> -го узла ЛВС
УД «УЗЕЛ ЛВС»	SERV	Имитатор обработки на сервере ЛВС запросов диалогового пользователя
	USR	Имитатор обработки на УП локальных или диалоговых запросов пользователей
	LRS	Имитатор обработки на ЦП фоновых, сетевых и транзактных запросов, захватывающих локальные ресурсы узла ЛВС
	SYSP	Имитатор работы ЦП
	DISP	Имитатор диспетчера запросов пользователей узла ЛВС
	CMUT _i	Модель внешней среды <i>i</i> -го узла ЛВС
	СПД	Имитатор передачи данных в сеть

Таблица 5.2 – Возможные движения транзактов по устройствам ИМ «Топология ЛВС», закодированные в «ТЕЛЕ» TR_i для различных типов запросов

№ ша-га	Тип 1. Фоновые сообщения	Тип 2. Локаль- ный диалог	Тип 3. Посылка сообщений	Тип 4. Короткий диалог	Тип 5. Заказ на уда- ление системы	Тип 6. Длитель- ный диалог
1	ИСТ ₁	ИСТ ₂	ИСТ ₃	ИСТ ₄	ИСТ ₅	ИСТ ₆
2	SYSP ₁	SYSP ₂	SYSP ₁	SYSP ₁	SYSP ₁	SYSP ₁
3	ПОГЛ.	ПОГЛ.	CMT _{PD}	CMT _{PD}	CMT _{PD}	CMT _{PD}
4	—	—	CANAL _{PD}	CANAL _{PD}	CANAL _{PD}	CANAL _{PD}
5	—	—	CANAL _i	CANAL _i	CANAL _i	CANAL _i
6	—	—	CMT _i	CMT _i	CMT _i	CMT _i
7	—	—	SYSP _i	SYSP _i	SYSP _i	SYSP _i
8	—	—	CANAL _{адр}	CANAL _{адр}	CANAL _{адр}	CANAL _{адр}
9	—	—	CMT _{адр}	CMT _{адр}	CMT _{адр}	CMT _{адр}
10	—	—	SYS _{адр}	SYS _{адр}	SYS _{адр}	REZ
11	—	—	ПОГЛ	CANAL _{адр}	ПОГЛ	SYS _{адр}
12	—	—	—	CMT _{адр}	—	CANAL _{адр}
13	—	—	—	CANAL _j	—	CMT _{адр}
14	—	—	—	CANAL _{PD}	—	CANAL _j
15	—	—	—	CMT _{PD}	—	CANAL _{PD}
16	—	—	—	SYST _{PD}	—	CMT _{PD}
17	—	—	—	GJUK	—	SYST _{PD}
18						ПОГЛ

Моделирования с помощью последовательности программ обработки статистики имитации (PR.GLOBS) и принятия решений (PR.RESH). Это обычные подпрограммы, алгоритмы выполнения которых реализованы на языке базовой системы моделирования MICIC [16] комплекса ПТКИ ЛВС. CLOBS усредняет множества статистик $\{ST_{kl}\}$ и откликов $\{Y_{jl}\}$, $l = \overline{1, 10}$ м согласно процедуре PR.MONTEK. Дополнительные блоки программ выдают на дисплей исследователю информацию об ошибках имитации (ВИ₁), интегральных откликах (ВИ₃), контрольные записи в файл статистики (ВИ₂). После вторичной обработки статистики имитации исследователю выдаются интегральные статистики и отклики имитации ЛВС (выдачи ВИ₄ и ВИ₅).

ИМ второго уровня детализации ЛВС (ИМ УЗЕЛ ЛВС) легко создается с помощью компонент библиотеки LIB.IMMODESET комплекса имитации [19], блок-схема которой отражена на рисунке 5.2.

На этапе компоновки ИМ из библиотеки LIB.CAM.LVS с помощью универсальных параметризованных элементов (структуру новых мы рассматривали ранее в 4.2) была сформирована в библиотеке LIB.IMMODESET новая ИМ. Она состоит из группы активностей, имитирующих обслуживание запросов пользователей согласно графу нижнего уровня иерархии компонентов ЛВС (GR_2). Здесь также имеются алгоритмы основных компонентов в составе ИМ комплекса ПТКИ ЛВС [24]. Далее из библиотеки элементов в ИМ добавляются блоки НАЧАЛО и ОКОНЧАНИЕ имитации, блок сбора статистики (BCC_2), процедура PR.MONTEK в УПМ. На рисунке 5.2 приведена блок-схема программы ИМ УЗЕЛ ЛВС. Связь активностей ИМ с УПМ также осуществляется с помощью операторов ожидания $WAIT(\tau_{ij})$ и набора функций-имитаторов устройств ИМ, функциональное назначение которых приведено в таблице 5.1. Как и в предыдущей ИМ с помощью PR.MONTEK достигается повторение N_m раз реализаций ИМ УЗЕЛ ЛВС. Как видно из рисунка 5.2, выход на ПП «ОКОНЧАНИЕ» имитации осуществляет PR.MONTEK в моменты завершения N_m раз реализаций ИМ. Функции BCC_2 и ПП «ОКОНЧАНИЕ» имитации аналогичны предыдущей модели. Другое дело, что состав функций BCC_2 различный, поскольку состав статистики другой, чем у предыдущей модели. После завершения варианта имитации на ИМ узла ЛВС подпрограмма «ОКОНЧАНИЕ» имитации активизирует обработку интегральной статистики имитации варианта моделирования ВП узла ЛВС с помощью последовательности процедур PR.OBRABOT₂ и PR.RFCHEN₂. Это обычные процедуры, но имеющие другой алгоритм выполнения, реализованные на языке базовой системы моделирования MICIS [16] комплекса ПТКИ ЛВС.

5.3 Построение имитационной модели вычислительного процесса в узле ЛВС на основе информационной подкраски запросов ресурсов системы пользователями

При переходе от формального описания ВП в узле ЛВС к имитационной модели [10] исследователю необходимо было выполнить следующий объем работы:

– поделить алгоритмы процессов на несколько активностей таким образом, чтобы каждая активность оканчивалась операторами синхронизации $WAIT(\tau)$ и $WAIT.WHILE(A)$;

– определить места установки в завершающих процесс активностях операторов сбора статистики для функции статистик обслуживания транзактов;

– с помощью ПТКИ ЛВС из библиотеки компонентов LIB.COMLVS выбрать универсальные параметризованные элементы и подпрограммы организации имитации (начало имитации, конец имитации, блок-сбора статистики, процедура PR.MONTEK, выдачи оперативной информации).

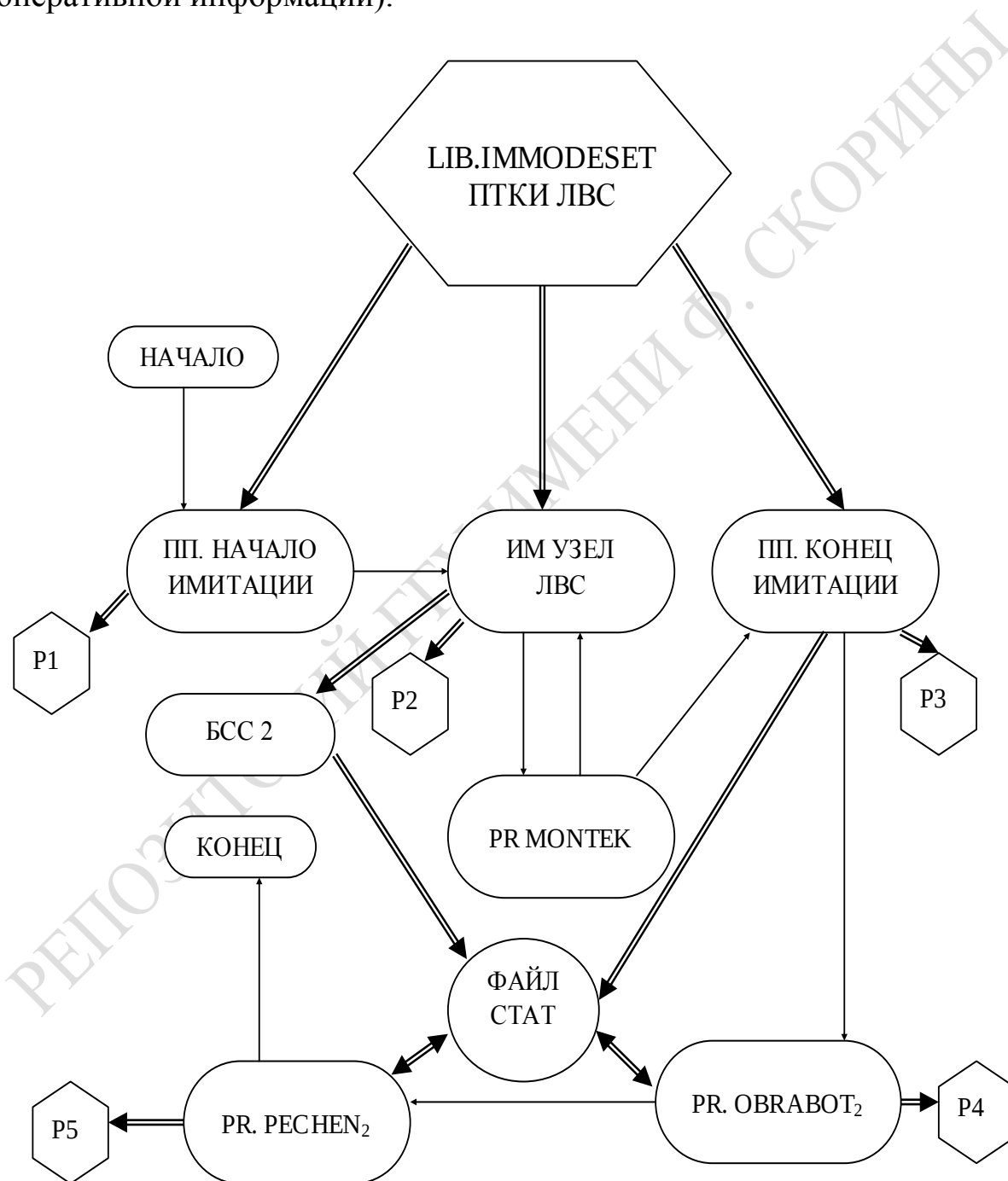


Рисунок 5.2 – Блок-схема программы имитационной модели «УЗЕЛ ЛВС»

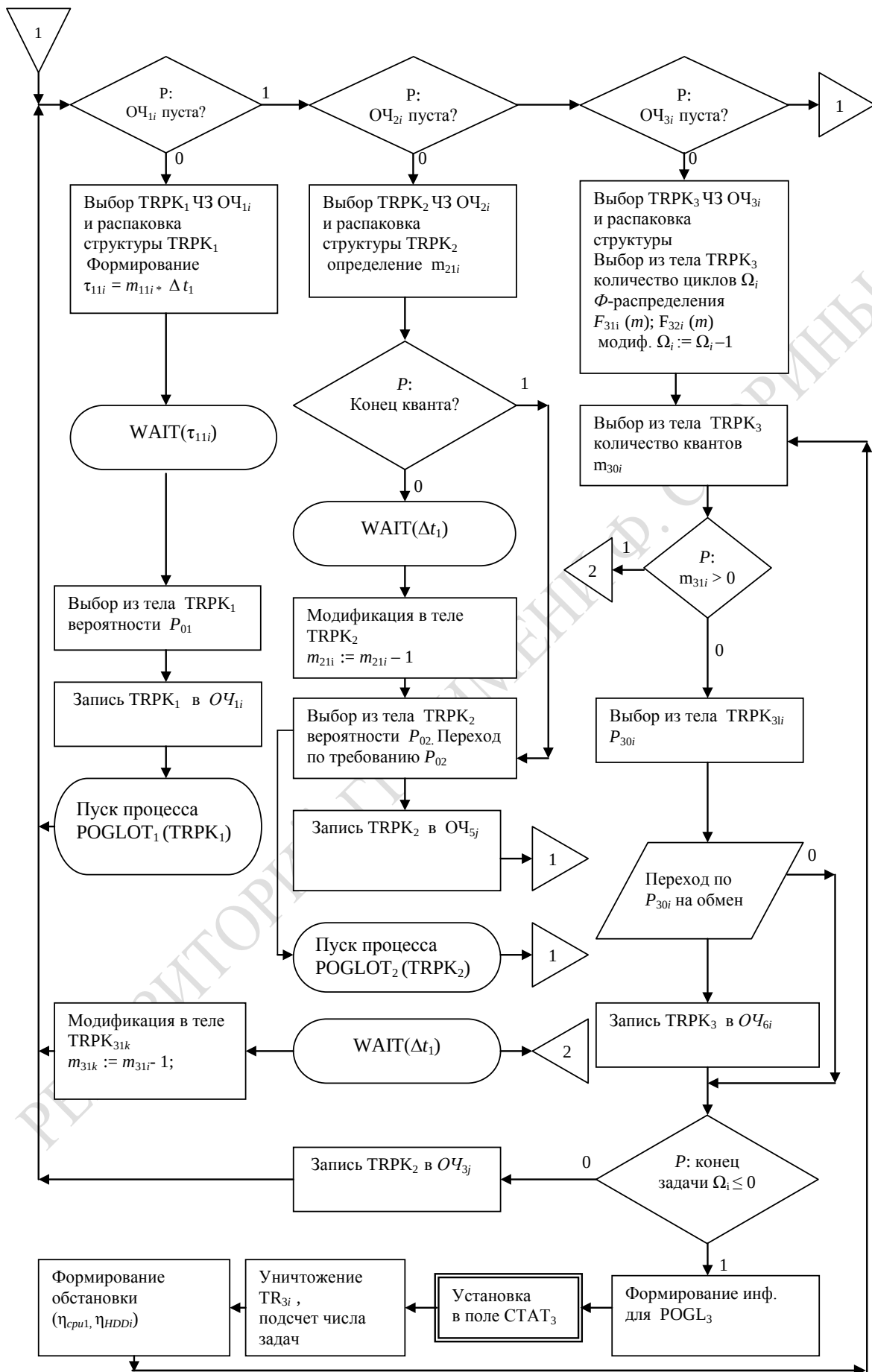


Рисунок 5.4 – Блок-схема алгоритма процесса PR.CPU_i

понимается приход сигналов, формируемых операторами уничтожения транзактов внутри алгоритма процесса-имитатора ДПУ_i (PR.CPU_i).

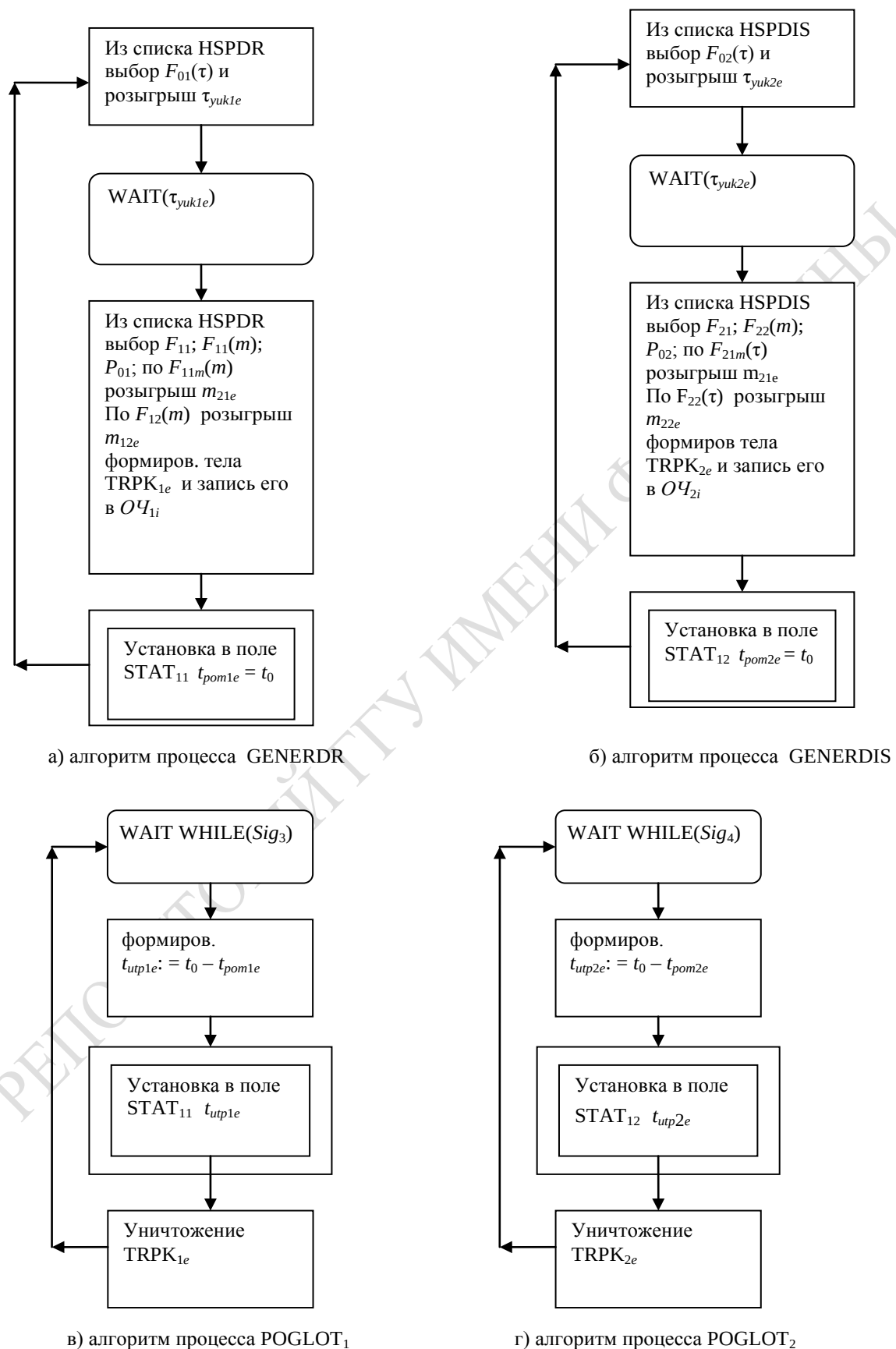


Рисунок 5.6 – Блок-схема алгоритмов вычислительных процессов

Как видно из блок-схем процессов, связь активностей процессов с УПМ осуществляется с помощью перечисленных выше операторов синхронизации. Поскольку для реализации ИМ ВП в ЛВС2 очередного варианта состава ресурсов необходимо, согласно методу статистических испытаний, повторение N_m раз «прогонов» модели, то в структуре ИМ используется процедура PR.MONTEK. Поэтому окончание имитации одного варианта моделирования ВП в ЛВС контролируется процедурой PR.MONTEK, которая имитирует выполнение программы «окончание» имитации. Эта программа заносит в файл имитации итоговую статистику имитации и тот кусочек итоговой статистики, который накоплен от момента предыдущей записи статистики до момента конца имитации ВП в ЛВС. Обработка статистики имитации, накопленной в файле статистики осуществляется с помощью последовательности подпрограмм PR.OBRABOT₃ и PR.RELHEN₃. Это обычные программы, реализованные на языке базовой системы моделирования MICIS [16] комплекса ПТКИ ЛВС.

Как видно из рисунка 5.3, подпрограмма PR. ПАКЕТ имитирует режим отложенного счета [4] на фоне использования ресурсов CPU_i и HDD_i более приоритетными запросами, формируемыми генераторами GENER.DR и GENER.DIS. В ходе имитационных экспериментов формируются выдачи программами:

- НАЧАЛО (БИ₁) для контроля вводимой информации;
- ОКОНЧАНИЕ (БИ₂) для контроля результатов l-й реализации ИМ;
- OBRABOT (БИ₃) для анализа результатов усреднения процедуры Монте-Карло;
- RESEN (БИ₄) для принятия решений.

5.4 Построение ИМ технологии поточного производства

Построение ИМ функционирования производственного участка обработки деталей. Требуется реализация следующей последовательности действий исследователя.

Отметим, что внутренней синхронизации активностей не требуется из-за простоты примера. В качестве внешней синхронизации используются операторы синхронизации, и поэтому возврат от активностей от всех операторов синхронизации происходит на одну и ту же метку УПМ. Конфликтные ситуации в модели разрешаются УПМ при проверке условий инициализации активностей. Поскольку активности взаимодействуют друг с другом посредством глобальных переменных,

то отражением конфликтных ситуаций (инициализации активностей в одно и то же модельное время) занимается УПМ. Информационное взаимодействие между активностями не требуется. Для указания момента или условий окончания моделирования используется специальный оператор в составе алгоритма начальной активности. Перед каждым изменением модельного времени t_0 УПМ проверяет выполнение этого условия и в случае его удовлетворения передает управление на выполнение финальной активности. В алгоритме финальной активности осуществляется обработка статистики моделирования. Использование активностей-контролеров в данной имитационной модели излишне.

Завершается перевод формального описания в модель выбором мест установки операторов сбора статистики $STATIST_{ij}(i, j)$ (индексации операторов ведется соответственно по порядку выполнения активностей и по номерам OCT_j). В данном примере алгоритм активностей позволяет установить операторы $STATIST_{ij}$ в начале его выполнения. Причем операторы $STATIST_{ij}$ фиксируют моменты начала работы $OCT_j(t_{nijh})$ и длительности простоя $OCT(\tau_{nij})$, а операторы $STATIST_{2j}$ определяют моменты окончания работы $OCT_j(t_{kijh})$ и длительность работы $OCT(\tau_{pij})$. Таким образом, за каждой активностью закреплены две глобальные статистики; t_{nijh} и t_{kijh} . По этим статистикам операторы сбора статистики формируют множества значений $\{\tau_{nij}\}$ и $\{\tau_{pij}\}$. Накоплением этой статистики обычно занимается УПМ.

Информационные связи между блоками осуществляются посредством транзактов. Транзакты состоят из одного атрибута (номера детали i). Блоки изменения маршрута используют значения этих атрибутов в алгоритме своей работы. Атрибут 1 формируется в момент создания транзактов. Задание начальных условий (длительность моделирования T_u) осуществляется специальным оператором, выполняющимся в начале моделирования. По этому оператору УПМ запоминает значение T_u . При каждом изменении модельного времени проверяется условие $t_0 \geq T_u$. При его выполнении процесс имитации завершается и управление передается по адресу, указанному оператором задания интервала времени моделирования. Вид этих операторов зависит от языка моделирования. В конце описания модели следует набор операторов обработки статистики.

Схема моделирования производственного участка, приведенная на рисунке 4.8, превращается в ИМ после установки-операторов синхронизации $WAIT(\tau)$ в конце каждой активности. По этим операторам происходит возврат на УПМ, а затем УПМ сама активизирует соответствующую активность по адресу AK_{ij} . В итоге модифицированная

таким образом ИМ в общей технологии принимает вид, представленный на рисунке 5.7.

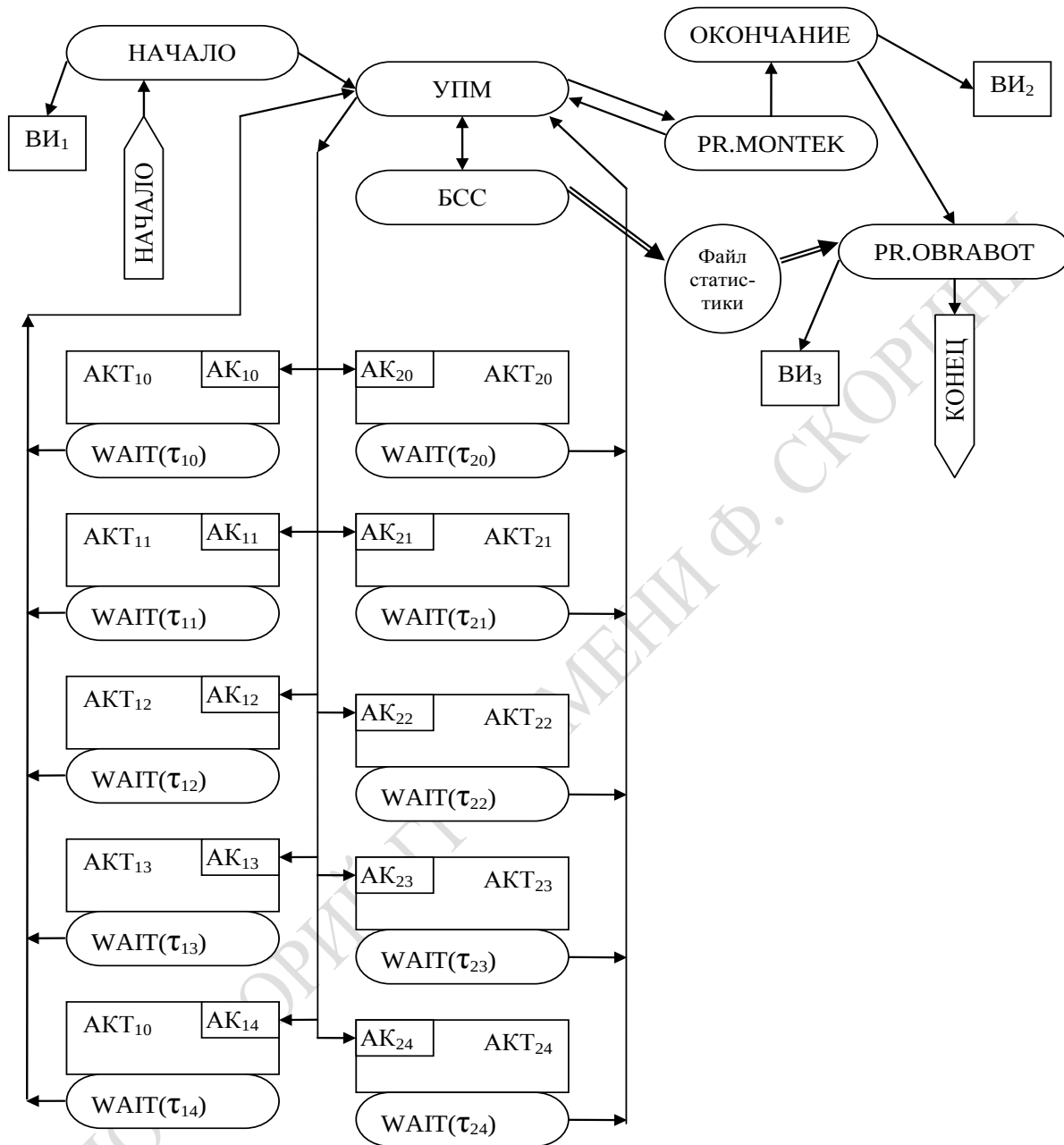


Рисунок 5.7–Блок-схема программы имитационной модели производственного участка

От установки специальных обслуживающих блоков, выполняющих функции сбора статистики, мы в данном примере отказались, поскольку УПМ, как правило, фиксирует следующую стандартную статистику: время жизни транзакта (от момента рождения до момента уничтожения транзакта), коэффициенты использования транзактами блоков, характеристики очередей (средняя и максимальная длина

очереди, число транзактов, побывавших в очереди, среднее время пребывания транзакта в очереди). Именно простота освоения обусловила широкое применение транзактного способа имитации, несмотря на ряд его очевидных недостатков (затраты ресурсов машинного времени и памяти ЭВМ на организацию имитации; множество дополнительных блоков, не имеющих прямого отношения к исследуемому объекту).

Далее процесс создания ИМ стандартный: вызывается программа «НАЧАЛО», в функции которой входит ввод начальной информации об интенсивности поступления деталей и «Разогрев» очередей в ИМ до установившегося состояния. С момента достижения стационарности процесса поступления деталей в цех начинается собственно имитация варианта ИМ (при заданном составе интенсивностей поступления деталей и функций распределения времени их обработки на станках) до выполнения условия окончания имитации, когда вновь иницируется процедура PR.MONTEK. Эта процедура проверяет условие окончания варианта имитации (когда имитация функционирования производственного участка при одних и тех же данных очередей транзактов повторяется N_M раз).

Только при выполнении этого условия PR.MONTEK иницирует выполнение программы OBRABOT₄. Эта программа осуществляет усреднение статистик имитации согласно методу статистических испытаний производственного участка обработки деталей. Усреднение и вычисление дисперсий имитации осуществляется по известным алгоритмам процедуры Монте-Карло, которые ранее были рассмотрены. Из рисунка 5.7 видно, что за время имитации производственного участка ИМ последовательных реализаций в три вида выдачи информации:

- ВИ₁ означает контрольный вывод на печать исходной информации каждого варианта параметров моделирования;
- ВИ₂ – интегральная статистика одного варианта имитации производственного участка;
- ВИ₃ – результирующие выводы с учетом усреднения с помощью процедуры Монте-Карло.

Отметим, что координация всех операторов сбора статистики также осуществляется централизованным образом путем периодического просмотра очередей, иницируемого УПМ. Очевидно, что из-за простоты ИМ программа RESHEN вырождается в ряд процедур вывода результатов ИЭ и сравнения результатов имитации по известным методикам. Эти программы уникальны для конкретного состава производственного участка. Но по своей сути это обычные программы, реализованные в среде системы моделирования MICIC [16].

Построение ИМ функционирования сборочно-разборочных операций ремонта сложного изделия. Выполняем следующую последовательность действий.

Во-первых, уточняется блок-схема связи компонентов имитационной модели сборочно-разборочных операций (ИМ СРО), разработанная на предыдущем этапе технологии разработки ИМ (при формализации ТП СРО в разделе 4.4). При этом определяются ошибки формализации ТП СРО, определяются, что ИМ используются: 28 раз операция $REMONT_{ij}$; 8 раз операция $RAZBOR_j$, 9 раз операция $SBORKA_j$, 57 раз выполняются операции на рабочем месте PM_k ($k = 1, 57$). На рисунке 5.8 приведена уточненная блок-схема связи компонентов ИМ СРО сложного изделия.

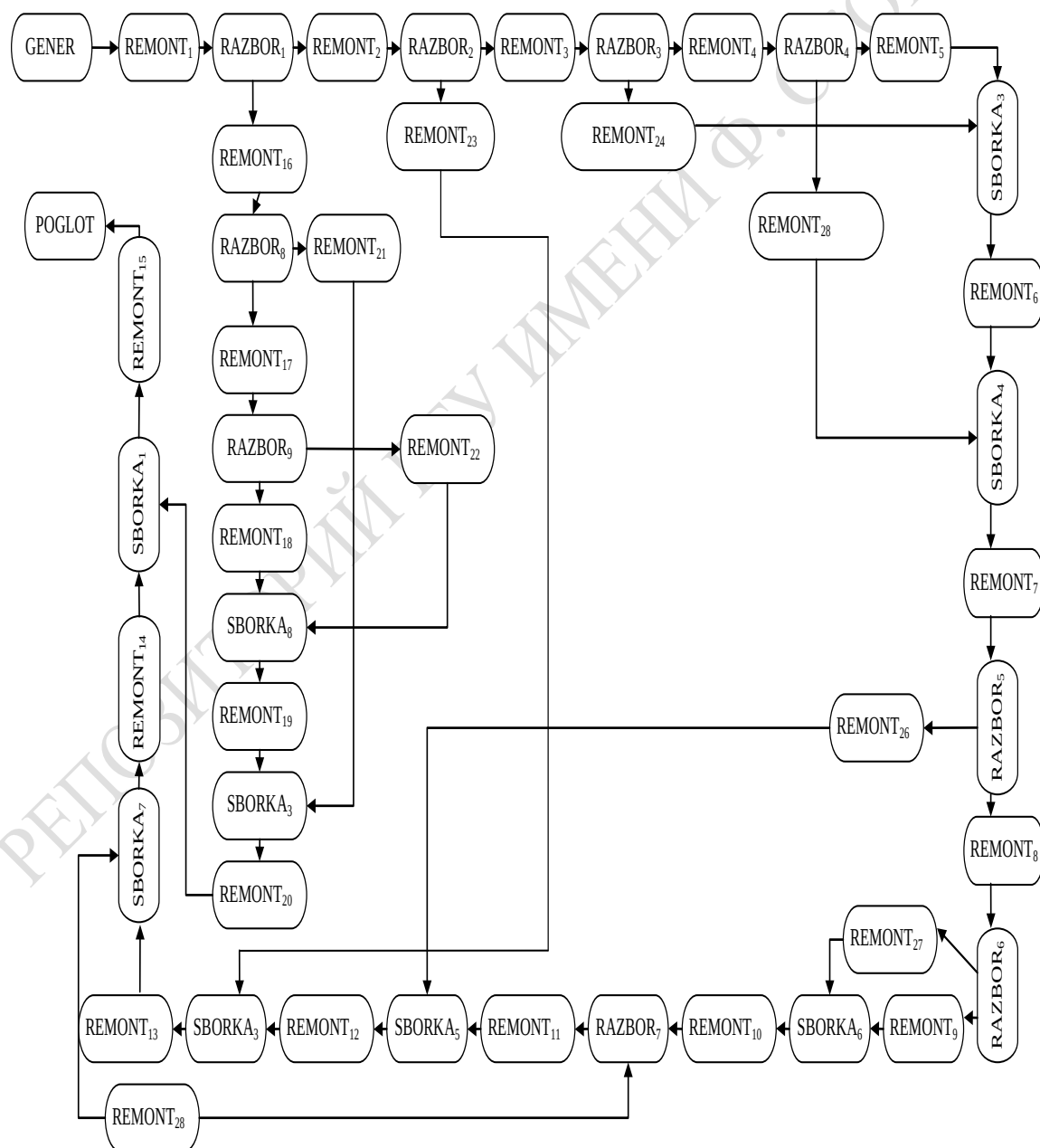


Рисунок. 5.8 – Блок-схема связи компонентов ИМ СРО изделия D

Во-вторых, уточняется структура технологической карты (TK_i) ТП СРО, хранящейся в теле информационного транзакта (TRINE) i -го технологического потока ремонта изделия. Фрагмент уточнений структуры TK_i транзакта $TRINF_i$ приведен в таблице 5.3.

Таблица 5.3 – Фрагмент структуры TK_i транзакта $TRINF$ в ИМ СРО

Тип PR	i	j	k	k_i	№ опер. ста- тист.	S_0	$S_i = 1$		$S_i = 2$		$S_i = 3$		N строки TK_i	
							i_1	f_1	i_2	f_2	f_3	f_4	КУ- ДА	ОТ- КУДА
GEN	1	–	–	2	St ₀₁	–	–	–	–	–	–	–	–	–
RE	1	1	1	3	St _{1.1}	–	–	–	–	–	–	–	–	–
RE	1	2	1	4	St _{1.2}	–	–	–	–	–	–	–	–	–
RE	1	3	1	5	St _{1.3}	–	–	–	–	–	–	–	–	–
RE	1	4	1	6	St _{1.4}	–	–	–	–	–	–	–	–	–
RE	1	5	1	7	St _{1.5}	–	–	–	–	–	–	–	–	–
RE	1	6	1	8	St _{1.6}	–	–	–	–	–	–	–	–	–
RAZ ₁	1	1	2	9	St _{i.1}	1	4	83	–	–	–	–	83	–
RE	1	7	3	10	St _{1.7}	–	–	–	–	–	–	–	–	–
RE	1	8	3	11	St _{1.8}	–	–	–	–	–	–	–	–	–
RAZ ₂	1	2	4	12	St _{i.1}	1	2	52	–	–	–	–	52	–
RE	1	9	5	13	St _{1.9}	–	–	–	–	–	–	–	–	–
RAZ ₃	1	3	6	14	St _{i.1}	1	5	129	–	–	–	–	129	–
RE	1	10	7	15	St _{1.10}	–	–	–	–	–	–	–	–	–
RAZ ₄	1	4	8	16	St _{i.1}	1	6	102	–	–	–	–	102	–
RE	1	11	9	17	St _{1.11}	–	–	–	–	–	–	–	–	–
RE	1	12	9	18	St _{1.12}	–	–	–	–	–	–	–	–	–
SBO ₃	1	3	10	19	St _{i.1}	1	10	129	–	–	–	–	–	12
RE	1	13	11	20	St _{1.13}	–	–	–	–	–	–	–	–	–
RE	1	14	11	21	St _{1.14}	–	–	–	–	–	–	–	–	–
SBO ₄	1	4	12	22	St _{i.1}	1	5	108	–	–	–	–	–	108

Здесь k – номер строки TK_i , которая используется процессами в данный момент модельного времени t_0 (в примере $k = 1, 129$). Выбрав k -ю строку TK_i , процессы получают всю информацию по адресации в транзакте в ИМ СРО, где указаны:

тип процесса (REMONТ – RE_{ij} ; RAZBOR _{j} – RAZ_j ; SBORKA – SBO_j , генератор IDFT _{R_i} – GEN);

ji – номер потока информационных транзактов;

j – текущий номер операции внутри потока i ;

k – номер процесса, имитирующего рабочее место PM_k ;

K_i – номер следующей строки TR_i при последовательном порядке выбора строк TK_i (когда нет изменения порядка выполнения операций в ТП СРО);

St_{ij} – номер места в БДИМ для накопления статистики выполнения операций (ij);

S_0 – количество разветвлений в операции $RAZBOR_j$ или количество объединяемых ветвей технологии ремонта во время операции $SBORKA_j$;

i_1 – номер нового технологического потока, создаваемого оператором $RAZBOR$ или уничтожаемого технологического потока оператором $SBORKA_j$;

f_i – номер строки TK_i , с которой начинается имитация ветви номера S_i .

В таблице 5.3 указано также, куда передается управление и как происходит размещение потока транзактов или откуда они пришли на операцию сборки. Далее передается управление на выбор следующей строки TK_i . Отметим, что процессы $PR.RAZBOR_i$, $PR.RAZBOR_j$, $PR.SBORKA_j$ расшифровывают стандартным образом информацию, содержащуюся в строке TK_i .

В таблице 5.4 приведено соответствие процессов операциям $MTXO_{ij}$ и указаны также номера рабочих мест, где реализуется множество операций $\{MTXO_{ij}\}$, $\{RAZBOR_j\}$ и $\{SBORKA_j\}$.

Таблица 5.4 – Таблица соответствия процессов операциям $MTXO_{ij}$ и рабочим местам PM_k

Состав $MTXO_{ij}$	Обознач. PM_k	Обознач. процессов	Список процессов разбора-сбора	Обоз. PM_k
1	2	3	4	5
$MTXO_{1.1} : MTXO_{1.6}$	PM_1	$RAZBOR_1$	Процессы разборки	PM_2
$MTXO_{1.7} : MTXO_{1.8}$	PM_3	$RAZBOR_2$	Процессы разборки	PM_4
$MTXO_{1.9}$	PM_5	$RAZBOR_3$	Процессы разборки	PM_6
$MTXO_{1.10}$	PM_7	$RAZBOR_4$	Процессы разборки	PM_8
$MTXO_{1.11} : MTXO_{1.12}$	PM_9	$RAZBOR_5$	Процессы разборки	PM_{14}
$MTXO_{1.13} : MTXO_{1.14}$	PM_{11}	$RAZBOR_6$	Процессы разборки	PM_{53}
$MTXO_{1.15} : MTXO_{1.16}$	PM_{13}	$RAZBOR_7$	Процессы разборки	PM_{32}
$MTXO_{1.17} : MTXO_{1.18}$	PM_{15}	$RAZBOR_8$	Процессы разборки	PM_{34}
$MTXO_{1.19}$	PM_{17}	$SBORKA_1$	Процессы сборки	PM_{29}
$MTXO_{1.20} : MTXO_{1.21}$	PM_{19}	$SBORKA_2$	Процессы сборки	PM_{24}
$MTXO_{1.22}$	PM_{21}	$SBORKA_3$	Процессы сборки	PM_{10}
$MTXO_{1.23}$	PM_{23}	$SBORKA_4$	Процессы сборки	PM_{12}
$MTXO_{1.24}$	PM_{25}	$SBORKA_5$	Процессы сборки	PM_{22}

Окончание таблицы 5.4

1	2	3	4	5
MTXO _{1.25} : MTXO _{1.26}	PM ₂₇	SBORKA ₆	Процессы сборки	PM ₁₈
MTXO _{1.27} : MTXO _{1.34}	PM ₃₀	SBORKA ₇	Процессы сборки	PM ₂₆
MTXO _{4.1} : MTXO _{4.6}	PM ₃₁	SBORKA ₈	Процессы сборки	PM ₃₆
MTXO _{4.7} : MTXO _{4.8}	PM ₃₃	SBORKA ₉	Процессы сборки	PM ₃₈
MTXO _{4.9} : MTXO _{4.10}	PM ₃₅	ОБСЛ MTXO	Процессы ремонта	
MTXO _{4.11} : MTXO _{4.12}	PM ₃₇	ОБСЛ MTXO	Процессы ремонта	
MTXO _{4.13} : MTXO _{4.15}	PM ₃₉	ОБСЛ MTXO	Процессы ремонта	
MTXO _{6.1} : MTXO _{6.17}	PM ₄₃	ОБСЛ MTXO	Процессы ремонта	
MTXO _{3.1} : MTXO _{3.14}	PM ₄₅	ОБСЛ MTXO	Процессы ремонта	
MTXO _{2.1} : MTXO _{2.16}	PM ₄₂	ОБСЛ MTXO	Процессы ремонта	
MTXO _{10.1}	PM ₄₁	ОБСЛ MTXO	Процессы ремонта	
MTXO _{5.1} : MTXO _{5.7}	PM ₅₀	ОБСЛ MTXO	Процессы ремонта	
MTXO _{8.1}	PM ₅₁	ОБСЛ MTXO	Процессы ремонта	
MTXO _{1.9}	PM ₅₅	ОБСЛ MTXO	Процессы ремонта	
MTXO _{7.1}	PM ₅₇	ОБСЛ MTXO	Процессы ремонта	

В-третьих, исследователь, используя формальное описание процессов, приступает к переводу этих описаний в имитационные модели. На рисунке 5.9 приведена блок-схема алгоритма процесса PR.REMONT_{ij}. В алгоритм PR.REMONT_{ij} добавляются операторы организации квазипараллелизма транзактно-процессного способа имитации:

– WAIT WHILE (A) – ожидание, пока не выполнится условие A; в данном случае, это активизация данного процесса другими процессами с помощью операторов Pusk и Contin (соответственно запуска и продолжения выполнения процесса);

– STOP(REMONT) – останов этого процесса после отправки управляющего транзакта UTR_{ij} на процесс PK_k и переход в ожидание продолжения процесса после имитации на PM_k предыдущего выполнения процесса;

– PUSK (SBORKA) и Pusk (RAZBOR) – по завершении очередной операции ремонта запуск алгоритмов соответственно процессов SBORKA и RAZBOR;

– WHITE(B) – ожидание, пока не выполнится условие, состоящее в окончании выполнения процессом PR.PM_k операции на рабочем месте.

Кроме того, в алгоритме PR.REMONT_{ij} имеются три места установки операторов функциям статистики перехода информационного транзакта из состояния в состояние.

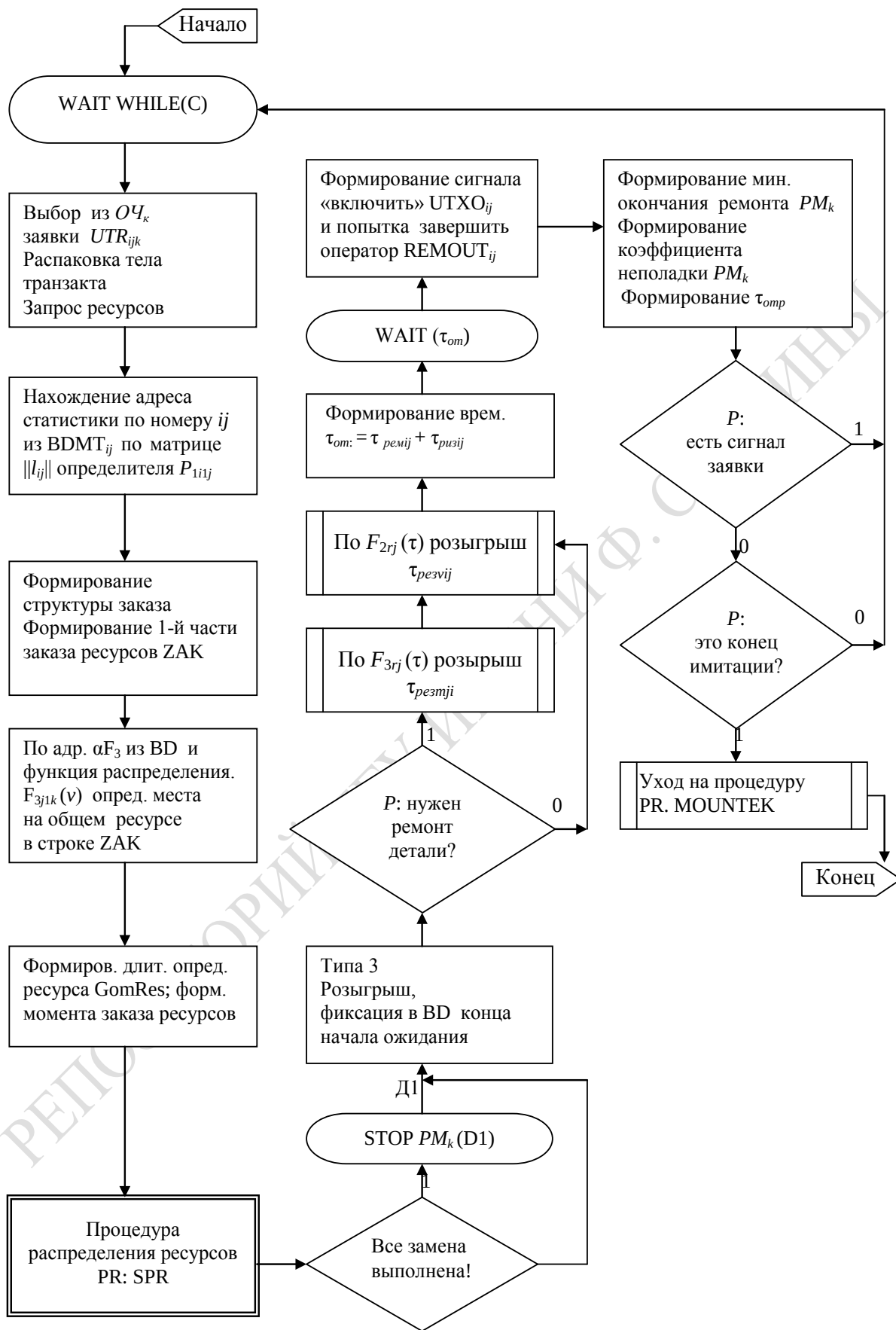


Рисунок 5.9 – Блок-схема алгоритма процесса PM_k

На рисунке 5.10 приведена блок-схема алгоритма процесса-имитатора выполнения $MTXO_{ij}$ на k -м рабочем месте.

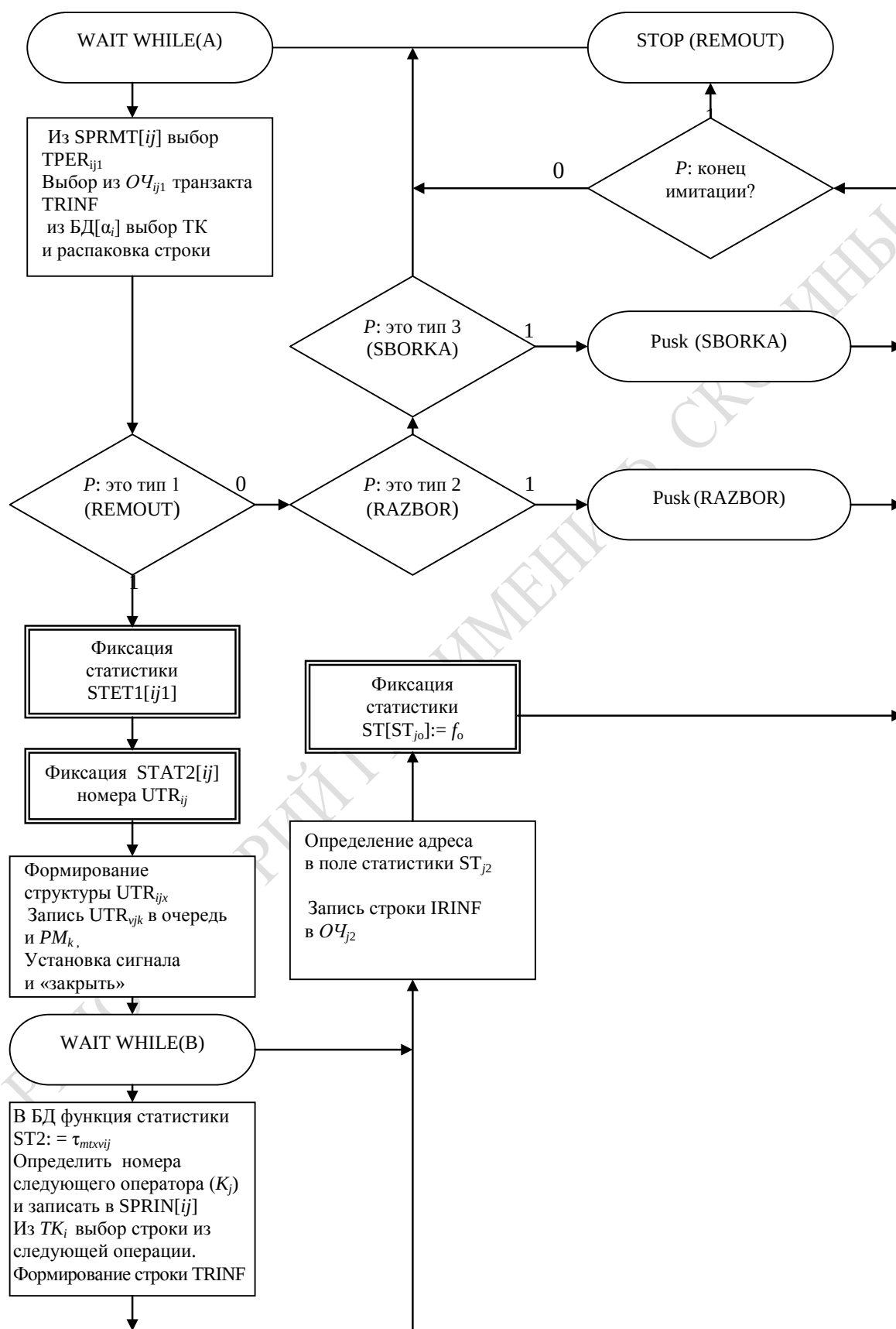


Рисунок 5.10 – Блок-схема алгоритма процесса PR REMOUNT_{ij1}

В алгоритме $PR.PM_k$ добавляются следующие операторы организации квазипараллелизма транзактно-процессного способа имитации:

- $WAIT\ WHILE(C)$ – ожидания, пока не выполнится условие C ; в нашем случае это активизация данного процесса процессом $PR.REMONT_{ij}$ путем послыки во входную очередь процесса управляющих транзактов UTR_{ij} .

- $WAIT(\tau_{om})$ – ожидания, равного сумме заказанного времени в транзакте $MTXO_{ij}$, определяющего длительность операции и времени ожидания предоставления ресурса предприятия для выполнения ij -й операции ремонта;

- $STOP\ PM_k(D_1)$ – остановка процесса в том случае, когда требуемый ресурс R занят и будет освобожден при завершении выполнения другой $MTXO_{ij}$;

Кроме того, в алгоритме процесса $PR.PM_k$ устанавливаются два обращения к процедурам формирования по функциям распределения времени разборки изделия $\tau_{РАЗБ\bar{I}}$ и собственно ремонта изделия $\tau_{РЕМ\bar{I}}$ в l -й реализации процедуры Монте-Карло, которые в сумме определяют заказанную величину времени имитации ij -й операции ремонта ($\tau_{zl} = \tau_{ремil} + \tau_{разil}$). Как видим, алгоритм $PR.PM_k$ ожидает появления во входной очереди транзактов UTR_{ij} , заказывает ресурсы для выполнения операции ремонта, используя для заказа системе распределения ресурсов (SRR) информацию, находящуюся в теле UTR_{ij} . Если все ресурсы имеются в наличии, то SRR закрепляет их за процессом $PR.PM_k$ и имитация операции ремонта выполняется немедленно. Если какой-либо ресурс занят, то $PR.PM_k$ ожидает его освобождения и затем имитирует выполнение операции ремонта на PM_k .

На рисунке 5.11 приведена блок-схема алгоритма модели процесса $PR.RAZBOR_{ji}$. К формальному описанию $PR.RAZBOR_{ji}$ добавляются следующие операторы организации квазипараллелизма транзактно-процессным способом имитации:

- $WAIT\ WHILE(D)$ – ожидания, пока не выполняются условия D , означающее выполнение операции запуска данного процесса процессом $PR.REMONT$ (как продолжение согласно TK_i следующей операции);

- $WAIT(\tau_{jl})$ – ожидания, имитирующего выполнение операции в l -й реализации процедуры $PR.MONTEK$;

- $ASTOP(Z_0)$ – автостановка по окончании операции разборки изделия и перехода в операцию ожидания запуска $PR.REMONT$;

- $ASTOP(D_3)$ – автостановка до выделения занятого ресурса процедурой распределения ресурсов $PR.KAZRES$;

- $GENER(i, f_i)$ – оператор генерации нового потока i при выполнении соответствующей строки TK_i и указание номера строки $TK_i (l_i)$

с которой начинается новая технологическая цепочка ремонта частей изделия.

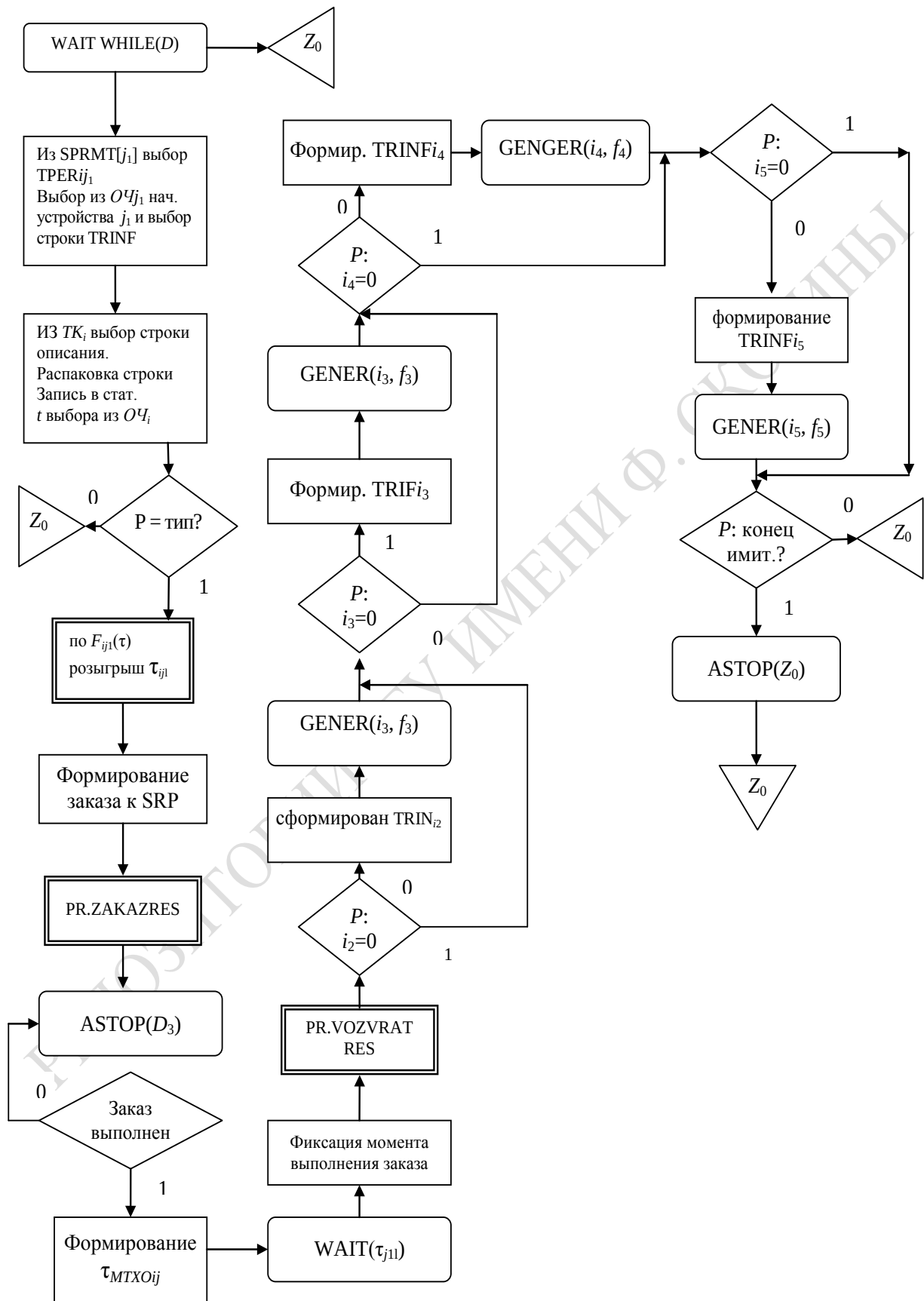


Рисунок 5.11 – Блок-схема алгоритма процесса PR.RAZBORj1

Отметим, что в алгоритме $PR.RAZBOR_{ji}$ предусмотрено до трех разветвлений основного потока и образование в результате четырех технологических потоков. В том случае, когда число разветвлений больше 3, в TK_i необходимо поставить подряд несколько операторов разборки изделия. Как видно из рисунка 5.11, в алгоритме процесса $PR.RAZBOR$ необходимо предусмотреть обращение к процедуре розыгрыша по функции распределения времени выполнения операций. Кроме того, в алгоритме процесса имеет место обращение к двум процедурам выделения и возврата ресурсов предприятия ($PR.ZAKAZREC$ и $PR.VOZVRATRES$) для выполнения операции разборки изделия (согласно k -й строки TK_i).

На рисунке 5.12 приведена блок-схема алгоритма модели $PR.SBORKA_{ji}$. К формальному описанию $PR.SBORKA$ добавляются следующие операторы организации квазипараллелизма транзактно-процессным способом имитации:

- $WAIT\ WHILE(E)$ – ожидания, пока не выполнится условие E , означающее выполнение оператора запуска данного процесса процессом $PR.REMONT$ (как продолжение согласно TK_i следующей операции);

- $POGLOT(i_1, f_1)$ – оператор запуска процесса уничтожения информационного транзакта согласно k -й строки TK_i номера (f_{i1}), которой завершается технологическая цепочка номера $i1$;

- $WAIT(\tau_{MTXOij})$ – ожидания, имитирующего выполнение операции в l -й реализации процедуры $PR.MONTEK$;

- в алгоритме ИМ $PR.SBORKA$ имеет место обращение к процедуре розыгрыша значений времени выполнения операции τ_{MTXOi} и заказанного места на общем ресурсе (V_{3j11});

- обращение к процедурам заказа и возврата ресурсов системе распределения ресурсов ($PR.ZAKAZSRR$ и $PR.VOZVRAT$).

Итак, после модификации алгоритмов процессов-имитаторов основных операций каждый процесс поделен на несколько активностей, универсальных для каждого типа имитаторов. Если не предпринимать никаких объединений процессов по таким активностям, то ИМ СРО тестового примера должны содержать: 28 подмоделей $PR.REMONT$; 8 подмоделей $PR.RAZBOR_j$; 9 подмоделей $PR.SBORKA_j$; 57 подмоделей $PR.PM_k$. Это означает, что в совокупности ИМ СРО будет иметь очень большие размеры, которые отладить чрезвычайно трудно. Это обстоятельство обусловило построение универсальной ИМ СРО, состоящей только из реентерабельных программ процессов. Для каждой копии процесса z -го типа $Z = 1,5$ в базе данных модели (БДМ) выделено место под рабочие массивы, статистики имитации и тело TK_i , которые используются и модифицируются одной реентерабельной программой ИМ процесса данного типа.

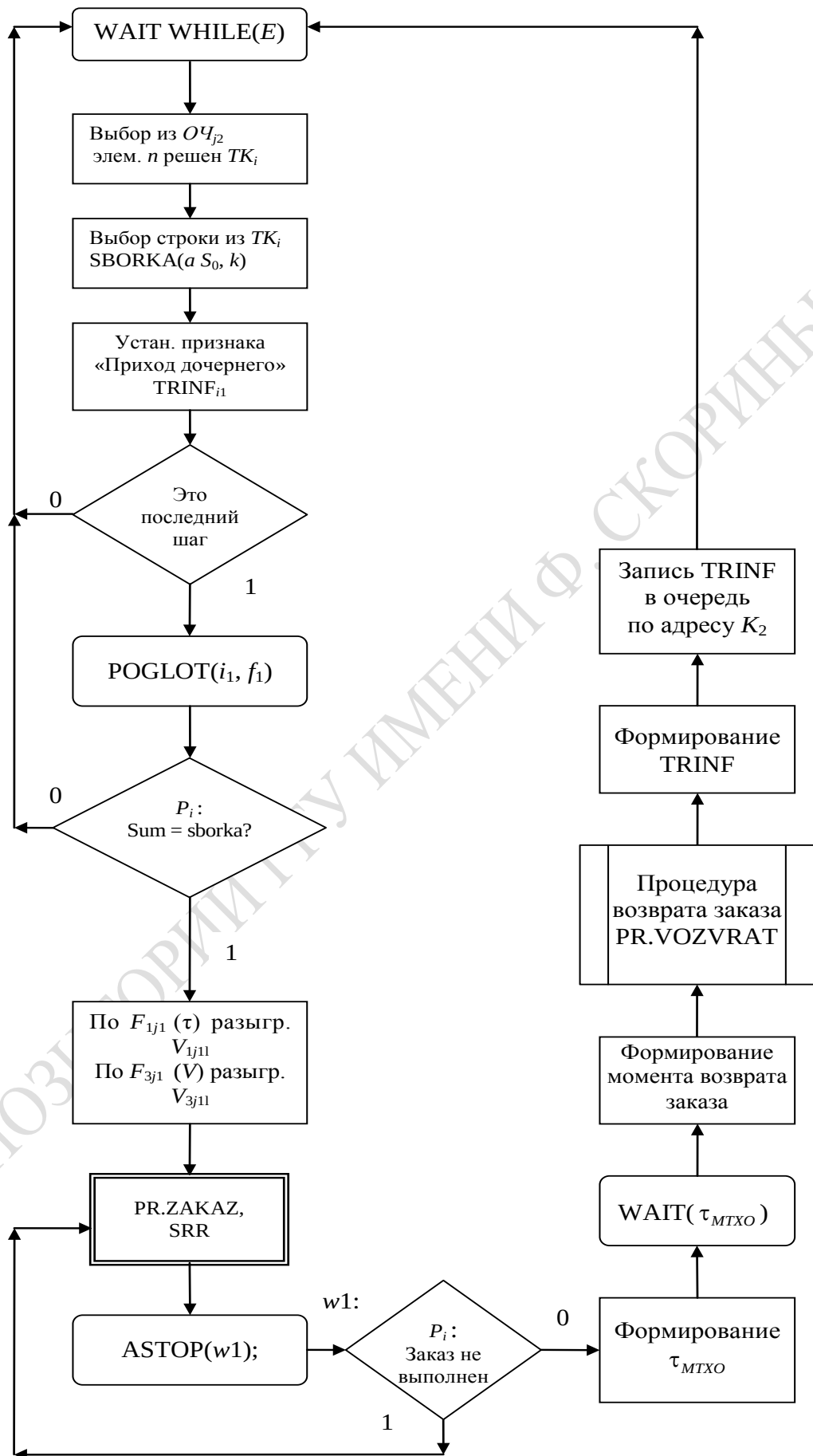


Рисунок 5.12 – Блок-схема алгоритма процесса PR.SBORKA_{j1}

На рисунке 5.13 приведена блок-схема связи процессов и программ таким образом преобразованной ИМ СРО.

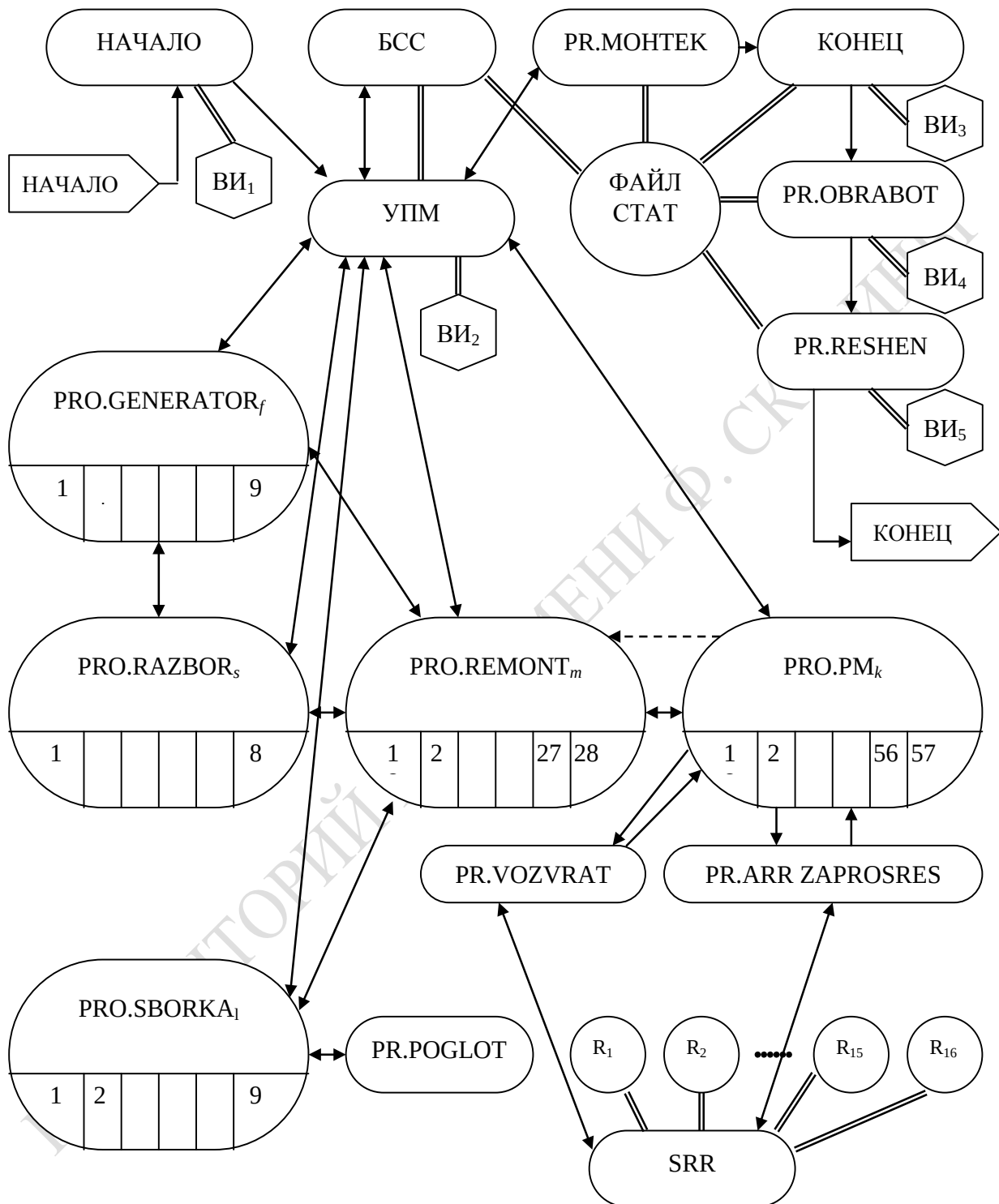


Рисунок 5.13 – Блок-схема связи процессов и программ ИМ СРО

Каждый тип реентерабельной программы обслуживает различное число копий процесса. Динамика взаимодействия процессов-имитаторов реализуется следующими копиями процессов:

- PR.GENERATOR _{i} , $i = \overline{1,9}$ – может создавать новые информационные транзакты (INTK _{i}) в «теле» которых находится TK _{i} ;
- PR.RAZBOR _{j_i} , $j_1 = \overline{1,8}$ – обслуживает до 8 копий PR.RAZBOR согласно выше описанному модифицированному алгоритму PR.RAZBOR;
- PR.SBORKA _{j_1} , $j_1 = \overline{1,9}$ – обслуживает 9 копий PR.SBORKA, согласно выше описанному алгоритму PR.SBORKA;
- PR.PM _{k} , $k = \overline{1,57}$ – обслуживает 57 копий PR.PM _{k} согласно рассмотренному выше алгоритму;
- PR.REMONT _{m} – обслуживает 28 копий PR.REMONT _{ij} согласно описанному выше алгоритму.

Кроме реентерабельных программ, перечисленных выше универсальных процессов, в состав ИМ СРО входят:

- НАЧАЛО – подпрограмма ввода исходной информации и «разогрева» ИМ в стационарный режим имитации;
- БСС – блок сбора статистики, фиксирующий все записи статистики на файле статистики для l -х реализаций ИМ;
- PR.MONTEK – процедура N_m реализаций одного варианта параметров модели (каждую реализацию начинает с установившегося состояния очередей к процессу, но при разных значениях базовых генераторов моделирования псевдослучайных чисел, используемых при розыгрыше жребиев 3-го типа для определения времен разборки, ремонта и сборки изделия);
- КОНЕЦ, обеспечивающая запись финальной статистики о состоянии ИМ на файл статистики и последующую активизацию программы PR.OBRABOT;
- PR.OBRABOT и PR.RECHEN организуют вторичную обработку статистики имитации l -х реализаций ИМ, усредняя согласно процедуре Монте-Карло статистики и отклики имитации. В ходе имитации организовано 5 типов выдачи информации об имитации СРО: контроль вводимой информации (ВИ₁), вывод УПМ оперативной информации о ходе имитации (ВИ₂); вывод интегральных откликов и статистик имитации (ВИ₃); результаты вторичной обработки статистики имитации (ВИ₄); результаты анализа вариантов имитации программой принятия решений (ВИ₅). Особенностью взаимодействия реентерабельных программ процессов является возможность одновременного обслуживания реентерабельной программой PRO REMONT _{m} нескольких копий процессов PR.REMONT, выполняющих заказы из разных строк TK _{i} , что означает квазипараллельное выполнение копий процессов PR.REMONT _{ij} , PR.RAZBOR _{j} , PR.SBORKA _{j} и PR.PM _{k} .

Как видим, в данной ИМ реализовано два уровня квазипараллелизма. На первом уровне УПМ организует квазипараллелизм, реализуя

операторы синхронизации, запуска и останова процессов. На втором уровне сами реентерабельные программы процессов организуют квазипараллельное обслуживание копий процессов на основе процессного способа имитации. На рисунке 5.14 показаны связи между реентерабельными программами-процессами, которые реализуются с помощью очередей между процессами.

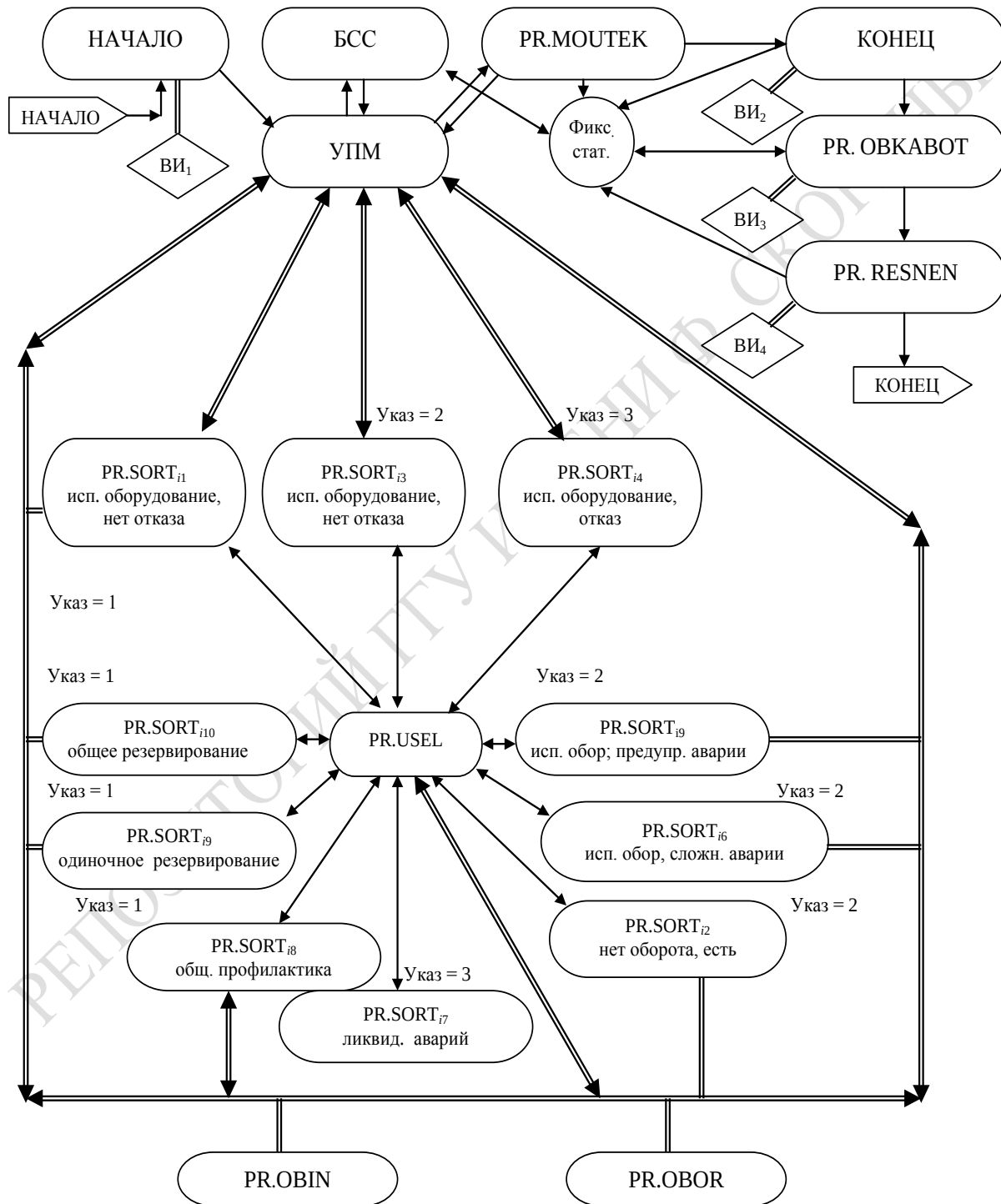


Рисунок 5.14 – Блок-схема программы имитационной модели последовательного ВТПП

Так, например, взаимодействия между $PRO\ REMONT_m$ и $PRO\ PM_k$ синхронизируются с помощью очередей управляющих транзактов $\{UTR_{ij}\}$. Взаимодействие $PRO\ REMONT_{ij}$ организовано с помощью единой очереди запросов $PRO\ RAZBOR_s$ и $PRO\ SBORKA_l$ к программе, которая обслуживается согласно дисциплине FIFO. По окончании выполнения запроса $PRO\ REMONT_m$ программа $PRO\ PM_k$ через УПМ с помощью оператора $CONTIN_{ij}$ активизирует процесс $PR.REMONT_{ij}$, который был остановлен в момент формирования VTR_{ij} и посылки его в очередь к $PRO\ PM_k$, которые также обслуживаются согласно дисциплине FIFO. Отметим, что все операторы сбора статистики также осуществляются централизованно путем периодического просмотра очередей исследуемого ТП.СРО.

5.5 Преобразование формального описания в имитационную модель вероятностного технологического процесса производства

Технологию построения имитационной модели последовательного ВТПП₁ продемонстрируем ниже.

Построение ИМ последовательного ВТПП₁. Используем формальное описание упомянутых процессов-имитаторов операций.

Построение имитационной модели *последовательного* ВТПП требует следующей последовательности действий исследователя.

Во-первых, уточняется блок-схема связи компонентов модифицированной полумарковской модели последовательного ВТПП₁, которая была разработана на предыдущем этапе разработки ИМ в разделе 4.5. Устраняются ошибки формализации последовательного ВТПП₁. Устанавливаем, что процесс $PR.SOST_{ij}$ используется 10 раз. Для каждого состояния разработан свой алгоритм реализации процесса. Процесс $PR.UZEL$ имеет универсальную структуру и присутствует в ИМ ВТПП в одном экземпляре.

Во-вторых, преобразуем $PR.UZEL$ следующим образом. Делим алгоритм $PR.UZEL$ на активности с помощью операторов синхронизации процессов (см. рисунок 5.4):

- $WAITWHILE(UZ)$ – ожидание, пока вычисляется условие $UZ = 1$, означающее активизацию процесса $PR.UZEL$ процессом $PR.SOST_{ij}$, завершившем имитацию состояния номера ij последовательного ВТПП;

- $PUSK\ PR.SOST_{ij}$ – активизация очередного состояния процесса с помощью процедуры нестандартного запуска;

– ASTOP PR.UZEL – автоостанова PR.UZEL после активизации одного из процессов PK.SOST_{ij}.

Кроме того, в «тело» активностей PR.UZEL включаются операторы обращения к процедурам: реализации процедуры Монте-Карло (PR.NONTEK), фиксации статистики времени жизни процесса, формирования интегральной статистики имитации последовательного ВТПП. Собственно стандартный выбор номера активизируемого процесса PR.SOST_{ij} по матрице вероятностей перехода из состояния в состояние остается таким же как в формальном описании. При выполнении условий окончания *l*-й реализации с помощью процедуры PR.MONTEK имитируется запуск процедуры конец имитации, а сам процесс переходит в ожидание выполнения условия *UZ*.

В-третьих, преобразуются все 10 процессов PR.SOST_{ij}. На рисунке 5.15 приведено преобразование блок-схемы алгоритма PR.SOST_{ij} для случая нормального выполнения (согласно полумарковской модели). Алгоритм PR.SOST_{ij} делится на активности с помощью операторов синхронизации процессов:

- WAIT WHILE (PUSK) – ожидание пока выполнится условие PUSK = 1, означающее активизации процесса PR. SOST_{ij};
- WAIT(τ) – ожидания процесса *ij* длительностью τ ;
- PUSK PR.UZEL – активизацию PR.UZEL;
- ASTOP (SOST) – остановка при синхронизации с работой оборудования.

Кроме того, используются: процедура запуска оборудования индивидуального и общего пользования и процедуры сбора статистики жизни процесса и записи ее в файл статистик. На рисунке 5.16 приведена блок-схема алгоритма PR.SOST_{ij} для стандартной ситуации, а на рисунке 5.17 приведена преобразованная блок-схема алгоритма PR.SOST_{i7} для случая ликвидации последствий аварии (нестандартная ситуация). Алгоритм PR.SOST_{i7} делится на активности с помощью операторов синхронизации процессов:

- PUSK PK.UZEL – активизации процесса PK.UZEL;
- WAIT WHIL (PUSK) – ожидания активизации процессов PK.UZEL;
- WAIT(τ) – ожидания процесса *ij* длительностью δ имитации реализации процесса;
- ASTOP (SOST) – автоостанова при синхронизации с работой оборудования;
- PUSK PROC_k – запуска процессов-имитаторов выполнения процедур ликвидации аварий.

Алгоритм процесса делится на ряд активностей операторами фиксации статистики ожидания активизации процесса.

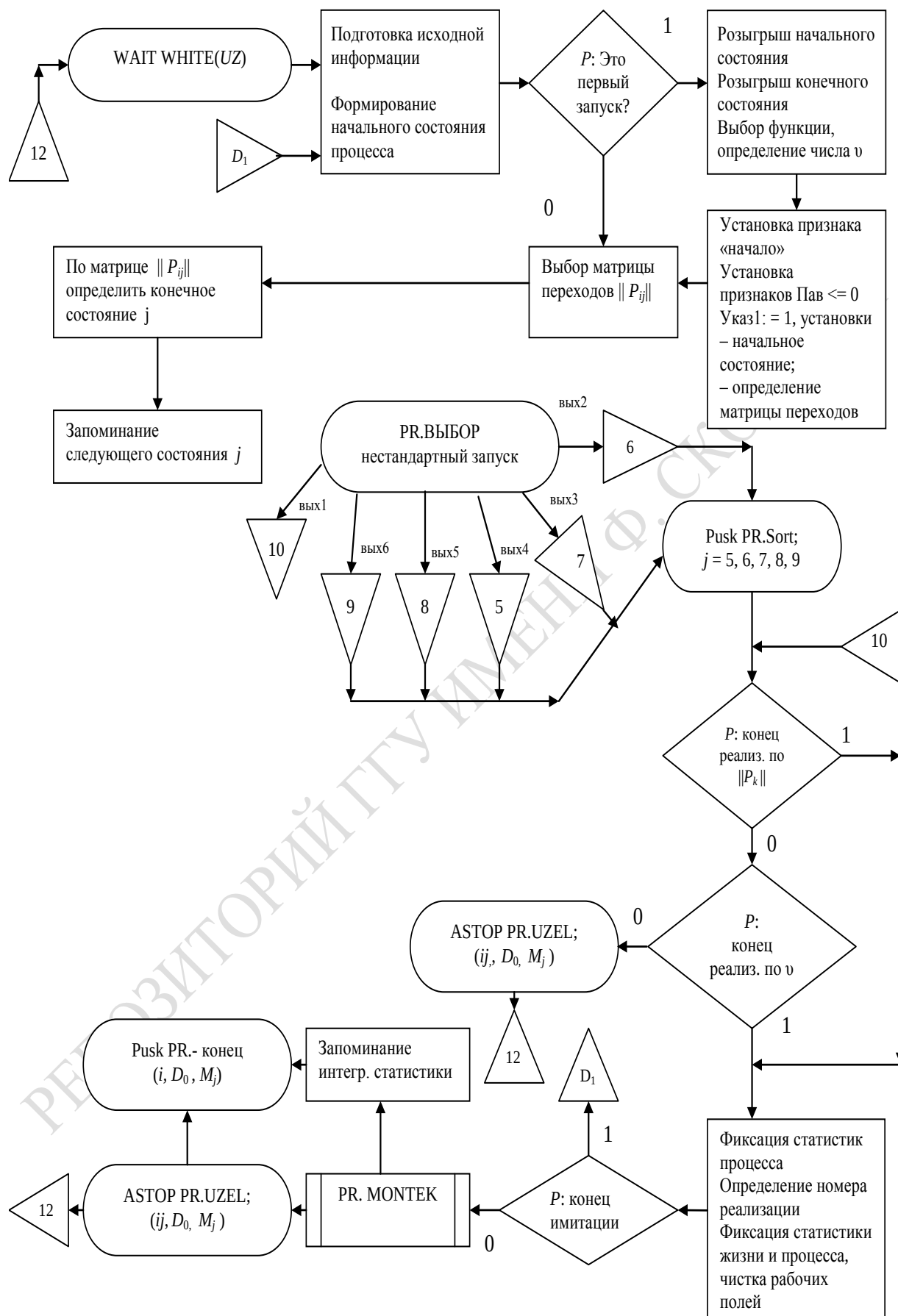


Рисунок 5.15 – Блок-схема алгоритма программы PR.UZEL

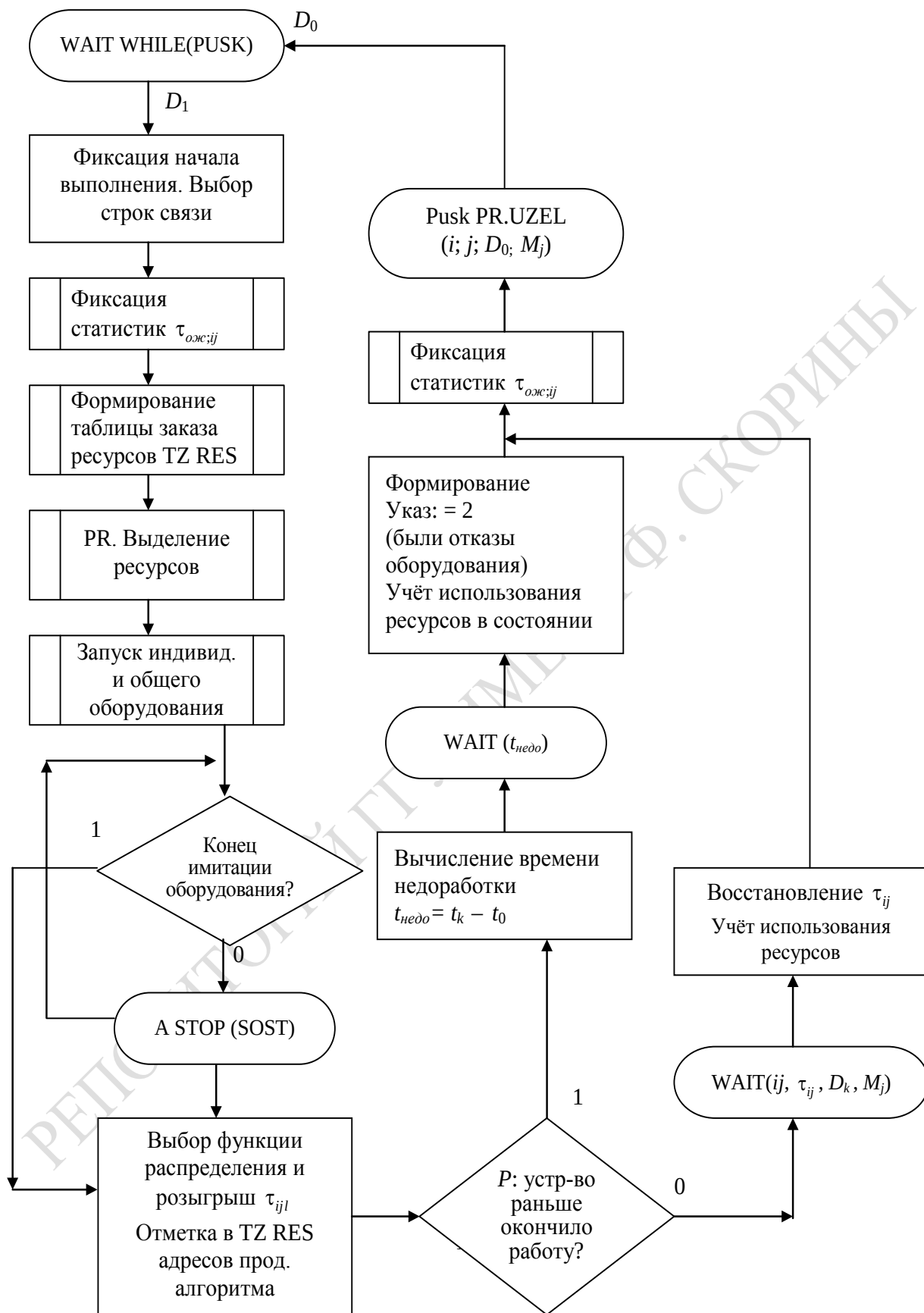


Рисунок 5.16 – Блок-схема алгоритма программы PR.SOST_{ij}
(нормальное выполнение)

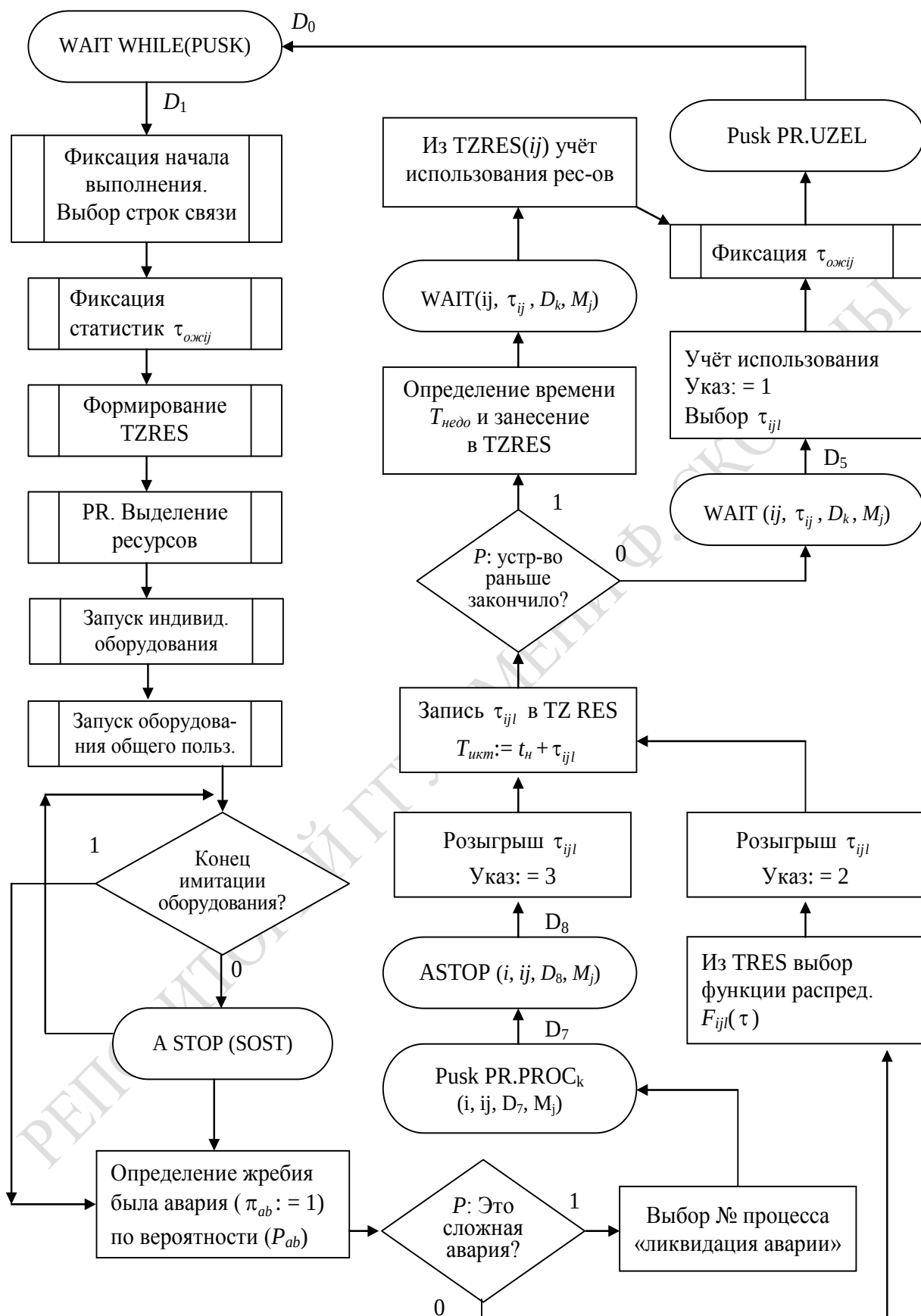


Рисунок 5.17 – Блок-схема алгоритма программы PR.SOST_{ij}
(j = 7, нестандартная ситуация: имитация аварии)

На рисунке 5.14 приведена блок-схема алгоритма имитационной модели последовательного ВТПП. В состав ИМ входят модифицированные программы процессов PR.SOST_{ij} и PR.UZEL, а также процессор-имитаторов оборудования общего и индивидуального использования (PR.OBOR и PR.OBIN). Все эти процессы взаимосвязаны с управляющей программой моделирования (УПМ) с помощью рассмотренных ранее операторов синхронизации процессов. Процессы PR.SOST_{ij} непосредственно взаимодействуют с PR.UZEL с помощью операторов запуска процесса. Друг с другом PR.SOST_{ij} взаимодействуют только через УПМ и затем через PR.UZEL.

Кроме программ перечисленных выше процессов в состав ИМ последовательного ВТПП входят:

- НАЧАЛО – программа ввода исходной информации и установки начального состояния первой реализации ИМ ВТПП;
- БСС – блок сбора статистики, фиксирующей все записи статистики имитации на файле статистики для l -х реализаций ИМ;
- PR.MONTEK – процедура реализации очередных вариантов ИМ при разных значениях базовых генераторов моделирования псевдослучайных чисел, используемых при розыгрыше жребиев 3-го типа;
- PR.OBRABOT и PR.RESHEN организуют вторичную обработку статистики имитации l -х реализаций ИМ, усредняя согласно процедуре Монте-Карло все статистики и отклики имитации.

Эти дополнительные программы связаны с основными процессами ИМ ВСГР с помощью УПМ. Для информационной синхронизации используемой информации применяется файл статистики. Записи на этот файл осуществляют БСС, PR.MONTEK, УПМ и программа КОНЕЦ имитации варианта модели ВТПП₁. После окончания варианта ИМ файл статистики используется программами обработки статистики и принятия решений (PR. OBRABOT и PR.RESHEN). В ходе имитации организовано 4 типа выдачи информации: контроль вводимой информации (ВИ₁), оперативный контроль интегральной статистики и откликов l -й реализации ИМ ВТПП (ВИ₂); результаты вторичной обработки статистики (ВИ₃), результаты анализа вариантов имитации программой принятия решений (ВИ₄).

Перевод вероятностного сетевого графика в имитационную модель ВТПП₂. На этом этапе исследователь должен выполнить следующую последовательность действий.

Во-первых, уточняются состав и структура ВСГР. МТХО_{ij} и SOB_j заменяются соответственно агрегатами АТОР_{ij} и ASOB_j. Добавляются обращения от агрегатов АТОР_{ij} на агрегаты-имитаторы оборудования общего и индивидуального пользования (AOBIN_{r1} и AOBOP_{r2}), а также определяются ситуации использования агрегатов AKAN_{r2} и APROC_k,

отображающих соответственно использование каналов общего оборудования и функционирования процедур ликвидации поставарийной ситуации в ВТПП. Блок-схема ИМ вероятностного сетевого графика тестового примера ВТПП приведена на рисунке 5.18. Как видим, структура ВСГР практически не изменилась, и показаны места использования агрегатами $АТОР_{ij}$ имитаторов оборудования ($АОВОР_{r1}$, $АОВОР_{r2}$, $АКАН_{r2}$). Как видно из рисунка 5.18, в случае использования агрегатов $АОВОР_{r2}$ необходимо указать номер канала этого общего оборудования ($АКАН_{r2}$).

Из рисунка 5.18 видно, что некоторые $АТОР_{ij}$ используют один и тот же агрегат $АОВОР_{r1}$. В итоге ИМ ВСГР компонуется из множеств агрегатов $\{АТОР_{ij}\}$, $\{АОВОР_j\}$, $\{АОВОР_{r1}\}$, $\{АОВОР_{r2}\}$, $\{АКАН_{r2}\}$ и $\{АПРОС_k\}$. Всего реализовано шесть типов агрегатов. Поэтому программы алгоритмов агрегатов необходимо реализовать как реентерабельные программы. Одна и та же реентерабельная программа агрегата использует различные версии информации, относящейся к каждому номеру ij агрегата в ИМ ВСГР. Программы остальных агрегатов также имеют множество версий в информационной базе, соответствуют номерам множеств агрегатов $\{АОВОР_j\}$, $\{АОВОР_{r1}\}$, $\{АОВОР_{r2}\}$, $\{АКАН_{r2}\}$, $\{АПРОС_k\}$. Использование реентерабельных программ агрегатов существенно сократит размеры программы ИМ ВСГР. Блок-схема программы ИМ ВСГР приведена на рисунке 5.19.

Во-вторых, необходимо преобразовать алгоритмы агрегатов путем деления алгоритмов на активности с помощью операторов синхронизации агрегатов, обеспечивающих после завершения соответствующей активности передачу управления на УПМ.

На рисунке 5.20 приведена блок-схема программы агрегата-имитатора $АОВОР_j$. При этом используются следующие операторы синхронизации агрегатов:

- WAIT WHILE (SGPR) – ожидание выполнения условия SOB_j (ожидание прихода сигнала от агрегата $АТОР_{ij}$ на один из входов $АОВОР_{ij}$, т. е. когда переменная $SGPR = 1$);
- USGPR(jn) – формирование сигнала от выхода $АОВОР_j$ на вход агрегата $АТОР_{jn}$ в режиме прямой имитации;
- USGIN(mj) – формирование сигнала от входа $АОВОР_j$ на выход агрегата $АТОР_{mj}$ в режиме инверсной имитации;
- WAIT WHILE (SGIN) – ожидание выполнения условия $SGIN = 1$, т. е. прихода сигнала в режиме инверсной имитации.

Кроме того, в алгоритм агрегата необходимо вставить операторы фиксации статистики ожидания агрегата до момента его активизации соответственно в режимах прямой и инверсной имитации.

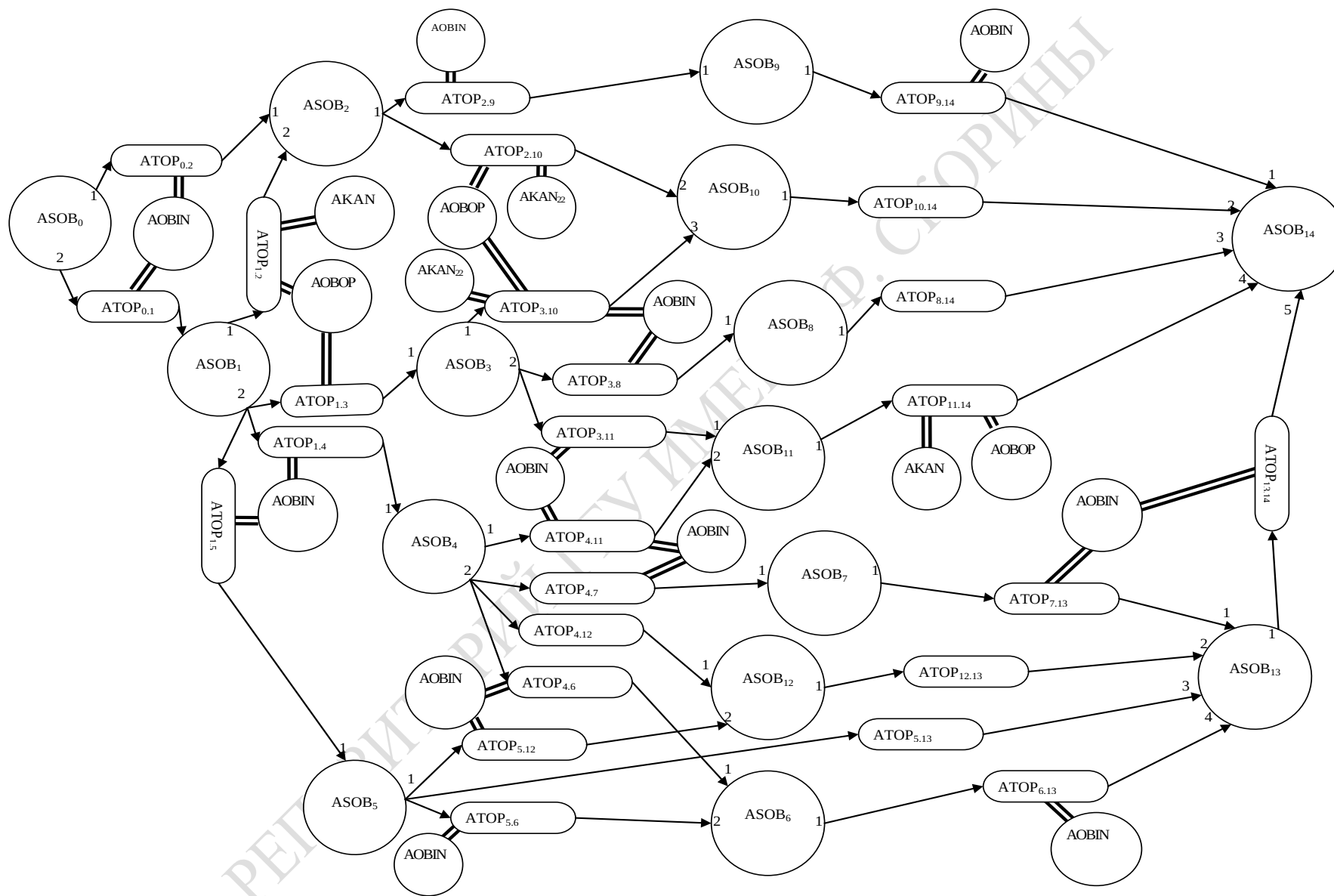


Рисунок 5.18 – Блок-схема ИМ ВСГР тестового примера

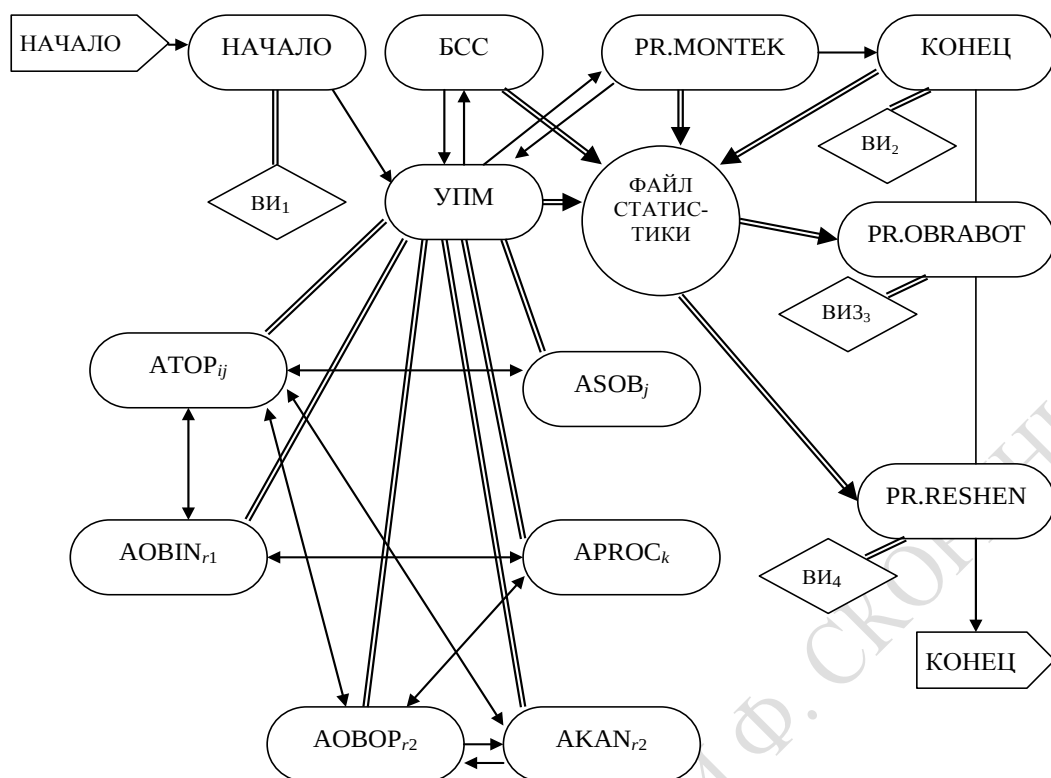


Рисунок 5.19 – Блок-схема имитационной модели параллельно-последовательного ВТПП

На рисунке 5.21 приведена блок-схема алгоритма программы агрегата-имитатора $АТОР_{ij}$. Текст формального описания алгоритма $MTXO_{ij}$ делится на ряд активностей следующими операторами сбора статистики имитации и синхронизации агрегатов:

- WAIT WHILE (SGPS) – ожидания, пока не выполнится условие SGPT, означающее приход сигнала от ASOB в режиме прямой имитации;
- WAIT WHIE (B) – ожидания, пока не выполнится условие, означающее приход от ASOB_j в режиме инверсной имитации;
- SIGEMT(j) – перевода фиктивного сигнала, минуя выполнение алгоритма агрегата непосредственно на один из входов ASOB_j;
- ASTOP (MTXO) – автостанова алгоритма агрегата после активизации агрегатов AOBIN_{r1} и AKAN_{r2} или же в случае заказа ресурса, который занят другим агрегатом;
- PUSK AOBIN_r – активизации алгоритма агрегата имитатора оборудования индивидуального использования;
- PUSK AKAN_{r2} – активизация алгоритма агрегата имитатора использования канала оборудования общего пользования;
- UST SIGPR(j) – послыки действительности сигнала в режиме прямой имитации на ASOB_j;
- UST SIGIN(i) – установка сигнала в режиме инверсной имитации на ASOB_i.

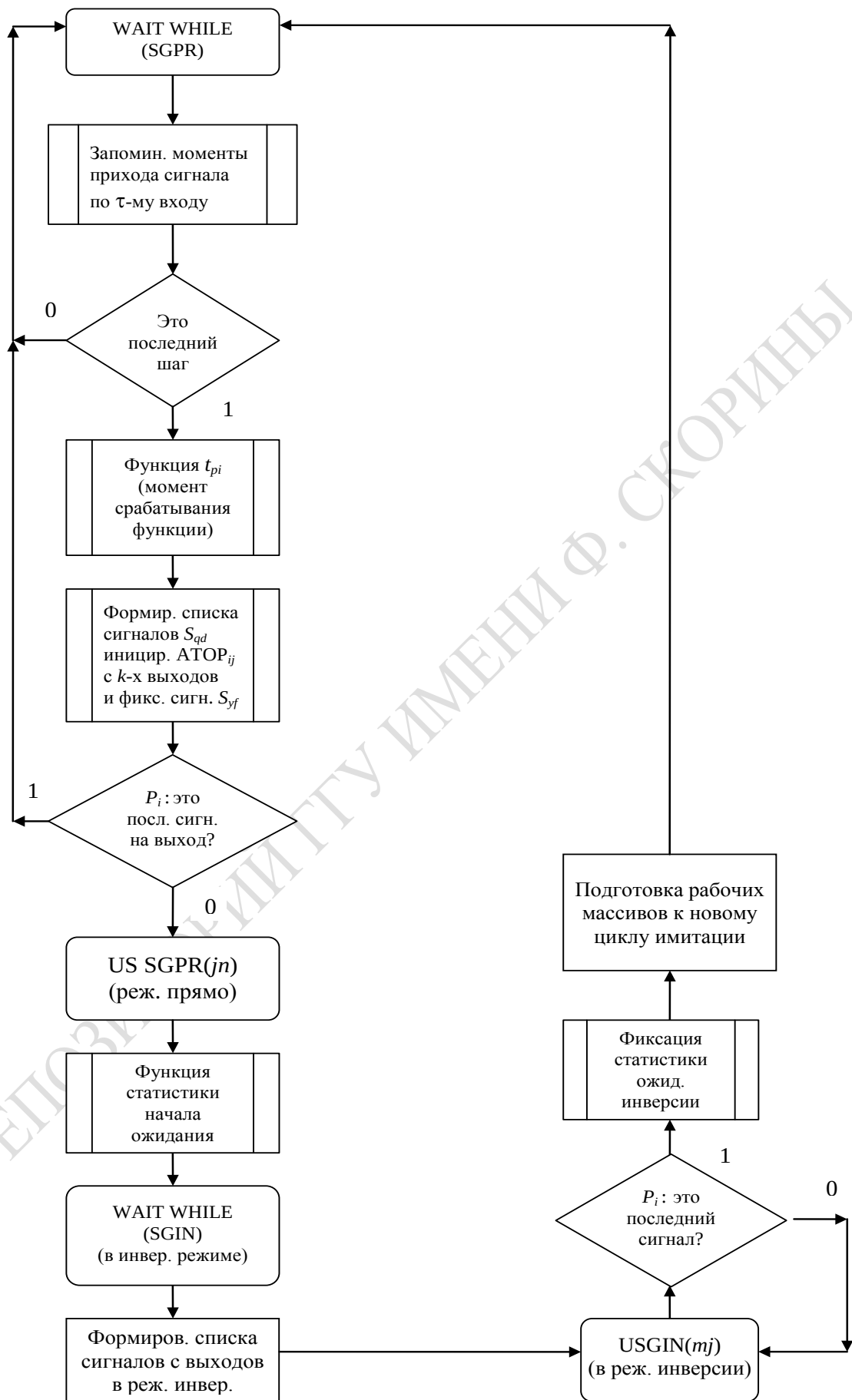


Рисунок 5.20 – Блок-схема алгоритма программы агрегата-имитации АТОВ_j

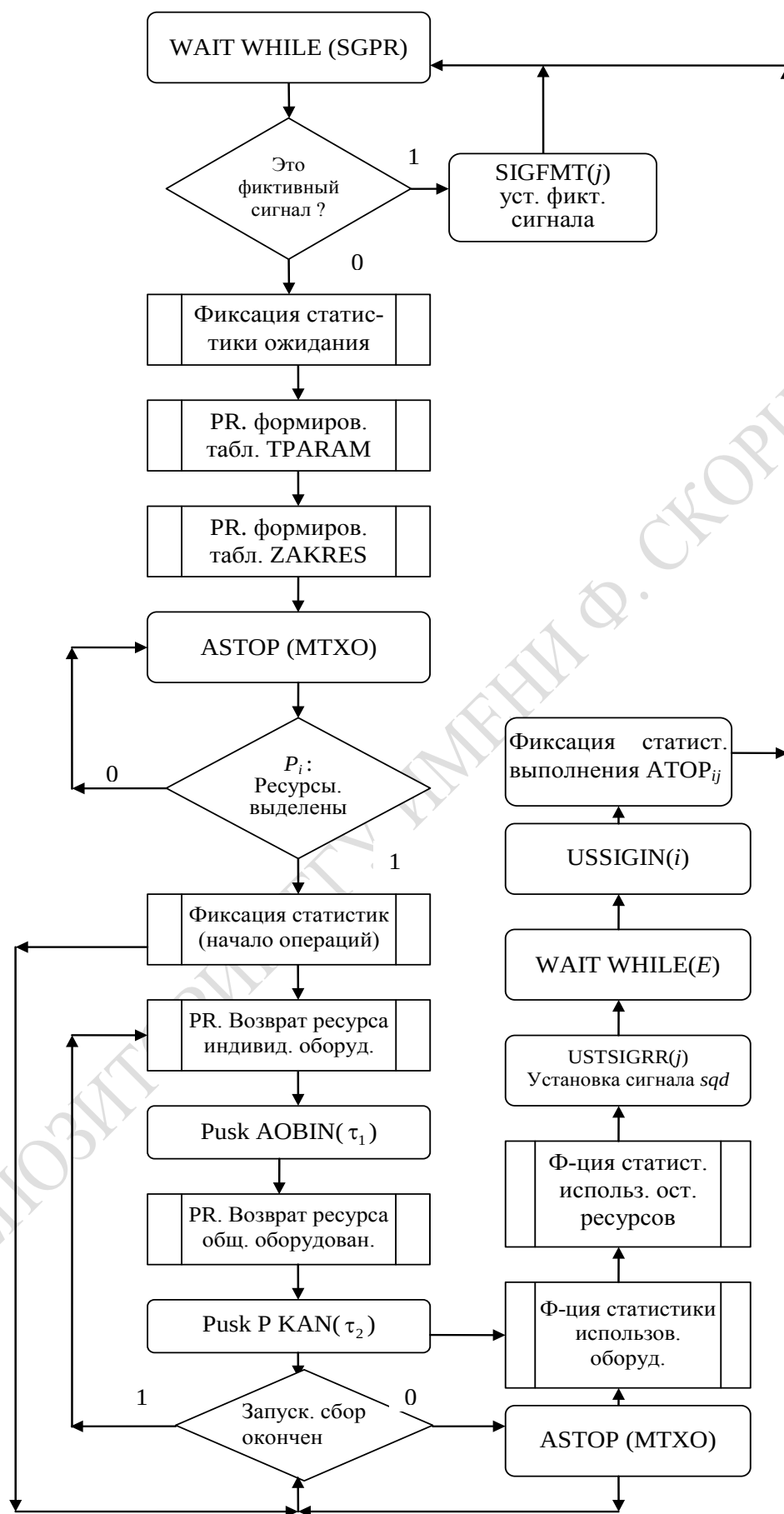


Рисунок 5.21 – Блок-схема алгоритма программы агрегата-имитации АТОР_{ij}

В алгоритм агрегата $АТОР_{ij}$ вставляются также операторы обращения к блоку сбора статистики ожидания и выполнения агрегата в различных его состояниях.

На рисунке 5.22 приведена блок-схема алгоритма взаимодействия программы агрегата-имитатора SOB_i с агрегатом-имитатором оборудования индивидуального использования $AOBIN_{r1}$. Здесь также имеет место деление алгоритма на активности операторами сбора статистики и операторами синхронизации взаимодействия агрегатов с помощью УПМ:

- WAIT WHILE (POBIN) – ожидания, пока не выполнится условие POBIN, означающее запуск агрегата AOBIN агрегатом $АТОР_{ij}$;
- WAIT($\tau_{ожс}$) – ожидание длительностью $\tau_{ожс}$, имитирующее выполнение агрегата $AOBIN_{r1}$;
- CONTIN(ij) – продолжение выполнения алгоритма агрегата $АТОР_{ij}$ по окончании имитации функционирования $AOBIN_{r1}$;
- ASTOP OBIN $_{r1}$ – автостанов агрегата $AOBIN_r$ до окончания выполнения последовательности агрегатов-имитаторов ликвидации аварий (A.PROC $_k$);
- PUSK A.PROC $_k$ – активизация выполнения агрегата-имитатора ликвидации аварий APROC $_k$.

Кроме того, в алгоритме агрегата AOBIN предусмотрена фиксация статистики использования агрегата $AOBIN_{r1}$. На рисунке 5.23 приведена блок-схема алгоритма программы агрегата-имитатора оборудования общего пользования $AOOP_{r2}$. Здесь отображена динамика формирования отказов функционирования агрегата, восстановления функций агрегата и взаимодействия с агрегатом имитатором канала общего ресурса $AKAN_{r2}$ при возникновении отказов функций $AOBOR_{r2}$. До начала имитации имеет место ожидание запуска процедурой Монте-Карло очередной реализации ИМ ВСГР с помощью оператора WAIT WHILE (PUSK.OBOR). По окончании имитации варианта ВТПП процедура PR.MONTEK с помощью оператора автостанова ASTOP.OBOR(r_2) первоначально переводит программу агрегата в режим ожидания.

Алгоритм программы-имитатора оборудования общего пользования делится на ряд активностей следующими операторами организации квазипараллелизма агрегатно-процессным способом имитации:

- WAIT(τ_{BO}) – ожидание длительности τ_{BO} , имитирующее интервал безотказного функционирования $AOBOR_{r2}$, после которого необходимо восстановление функций агрегата и возможна ликвидация аварии последовательностью агрегатов-имитаторов процедур {APROC $_k$ };

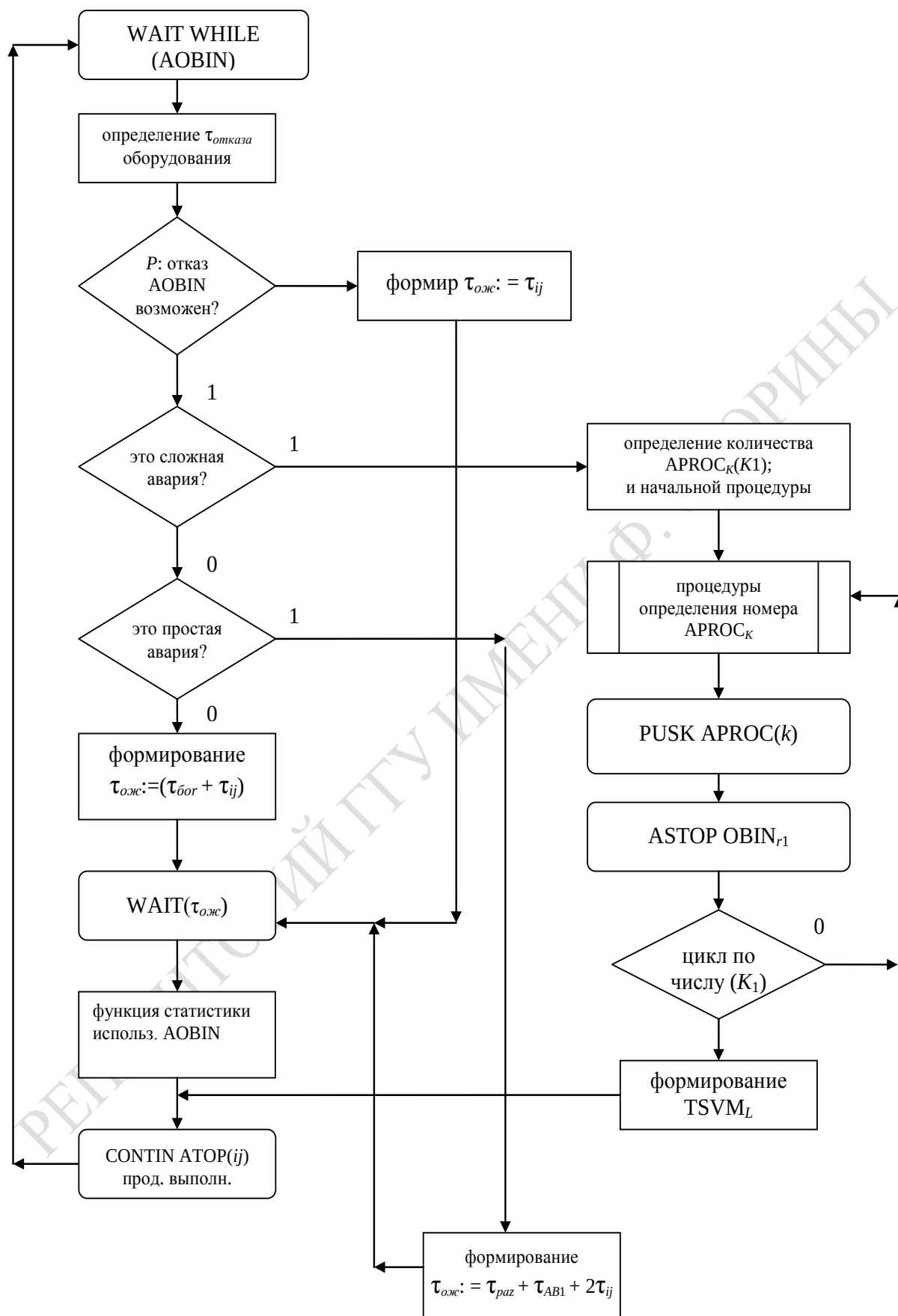


Рисунок 5.22 – Блок-схема алгоритма программы
оценки агрегата-имитатора АОВИН_{r1}

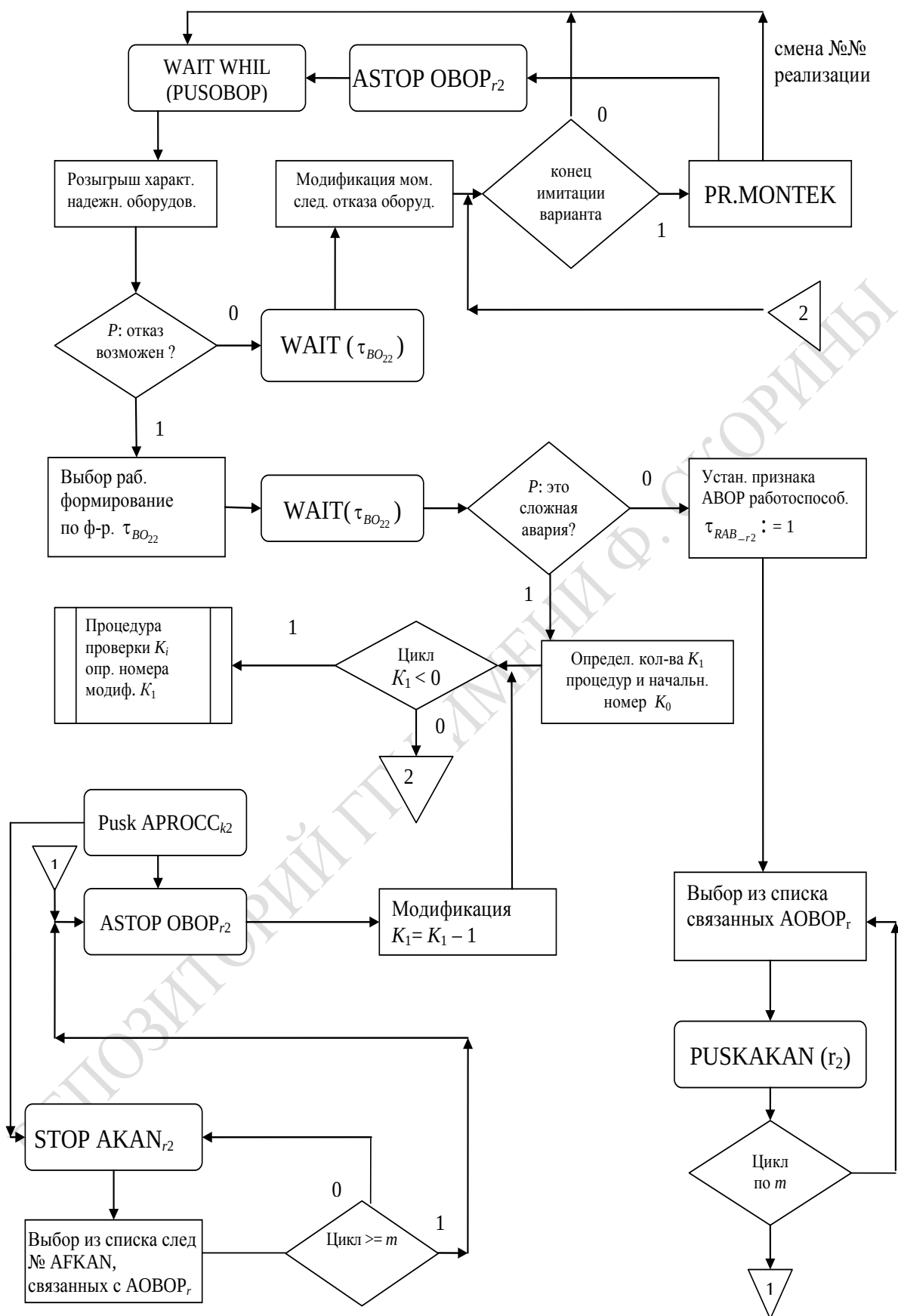


Рисунок 5.23 – Блок-схема алгоритма программы агрегата-имитации АОВОР_{r2}

- WAIT(τ_{VO}) – ожидание длительностью τ_{VO} , имитирующее операцию восстановления функций оборудования агрегата АОВОР_{r2};
- PUSK APROC_{r2} – активизация агрегата-имитатора процедуры номера r₂ ликвидации аварий;
- STOP AKAN_{r2} – останов агрегата-имитатора AKAN_{r2} при появлении аварии на АОВОР_{r2};
- PUSK AKAN_{r2} – активизация агрегата-имитатора AKAN_{r2} после ликвидации аварии на АОВОР_{r2}.

На рисунке 5.23 приведена блок-схема алгоритма программы агрегата-имитатора и использование канала общего оборудования (AKAN_{r2}). Агрегат-имитатор функционирования канала на АОВОР_{r2} все время контролирует состояния АОВОР_{r2}. Если АОВОР_{r2} находится в работоспособном состоянии, то заказ АТОР_{ij} на выполнение агрегата используется. В противном случае происходит ожидание завершения заказа АТОР_{ij} до окончания ликвидации аварии на АОВОР_{r2}. Начинается алгоритм программы ожиданием свершения события (PUSKAN) с помощью оператора WAIT WHILE (PUSKAN). Алгоритм агрегата делится на ряд активностей операторами синхронизации агрегатов:

- STOP АТОР(ij) – останов функционирования агрегата АТОР_{ij}, заказавшего выполнение агрегата AKAN_{r2};
- WAIT(τ_{ij}) – ожидание длительностью τ_{ij} , имитирующее собственно само использование агрегата AKAN_{r2};
- Contin АТОР(ij) – продолжение имитации функционирования агрегата АТОР_{ij} после завершения имитации AKAN_{r2};
- WAIT WHIL (VOST) – ожидание, пока не выполнится условие VOST = 1, означающее конец восстановления аварии на агрегате АОВОР_{k2}.

На рисунке 5.24 приведена блок-схема алгоритма программы агрегата-имитатора процедуры ликвидации аварии APROC_k. Вначале выполнения алгоритма агрегат находится в состоянии ожидания условия PUSPR = 1, означающее запуск агрегата либо АОВИН_{r1}, либо АОВОР_{r2}. Алгоритм выполнения агрегата делится на несколько активностей операторами синхронизации агрегатов:

- Contin АОВИН_{r1} – продолжение выполнения алгоритма агрегата АОВИН номера r₁;
- Contin KAN_{r2} – продолжение выполнения алгоритма агрегата AKAN_{r2}.

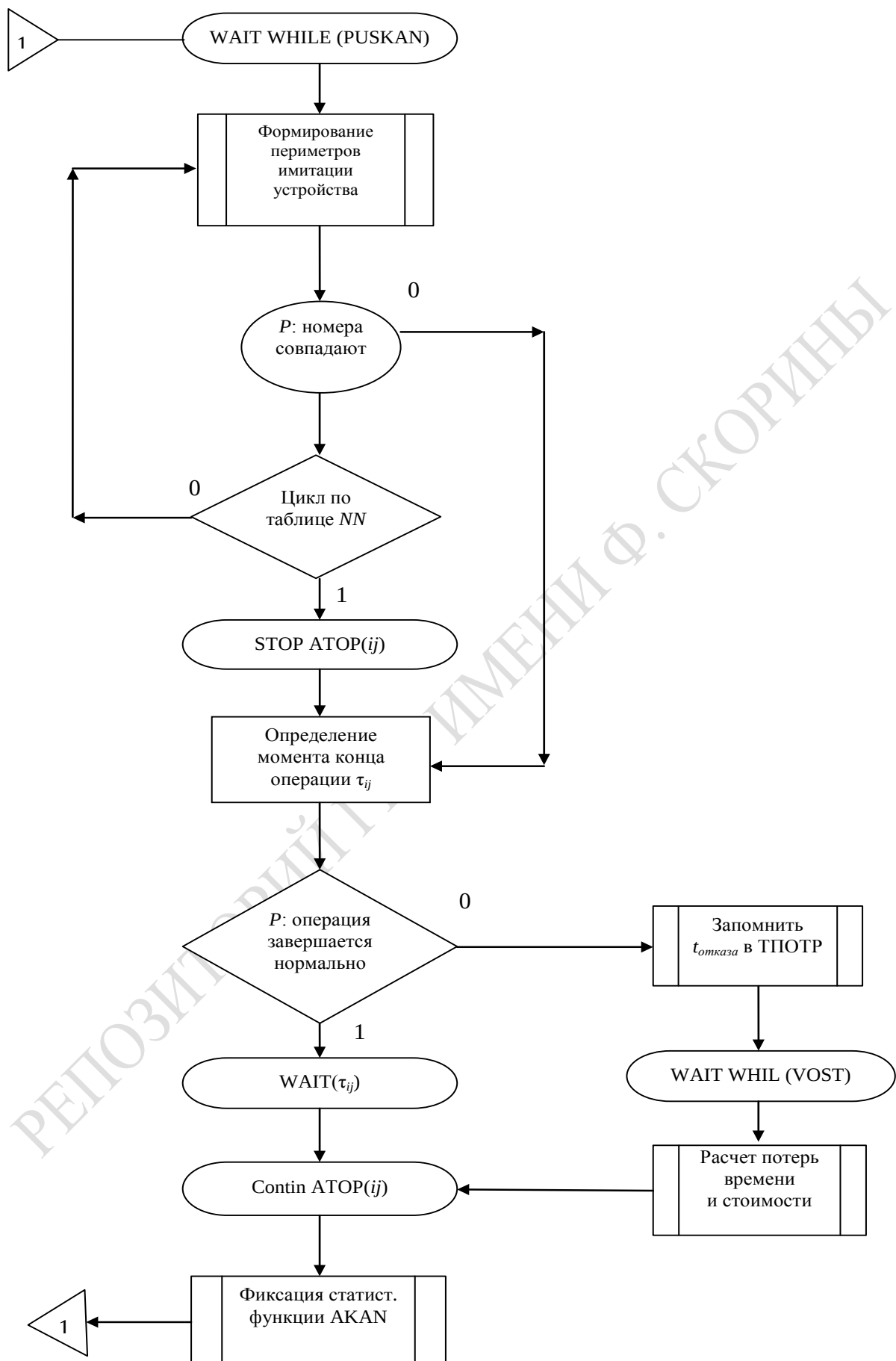


Рисунок 5.24 – Блок-схема алгоритма программы агрегата-имитатора $AKAN_{r2}$

В алгоритме программы $A.PROC_k$ предусмотрен учет использования материалов при имитации процедуры ликвидации аварий оборудования.

На рисунке 5.25 приведена блок-схема программы имитационной модели параллельно-последовательного ВТПП₂. В состав ИМ входят модифицированные программы реентерабельных программ шести типов агрегатов-имитаторов ($АТОР_{ij}$, $ASOB_j$, $AOBIN_{r1}$, $AOBOP_{r2}$, $AKAN_{r2}$, $APROC_k$), а также стандартный набор программ организации имитационного эксперимента. В состав этих программ входят:

- НАЧАЛО – программа ввода исходной информации для начала имитации очередного варианта ВТПП₂;

- БСС – блок сбора статистики, фиксирующий все записи статистики имитации на файле статистики для l -х реализаций ИМ;

- PR.MONTEK – процедура реализаций очередных вариантов ИМ при разных значениях базовых генераторов моделирования псевдослучайных чисел, используемых при розыгрыше жребиев 3-го типа [17];

- PR.OBRABOT и PR.RESHEV, которые организуют вторичную обработку статистики имитации l -х реализаций ИМ, усредняя согласно процедуре Монте-Карло все статистики и отклики имитации.

Все эти дополнительные программы связаны между собой с помощью УПМ. На рисунке 5.25 взаимодействие УПМ с шестью типами реентерабельных программ с помощью операторов синхронизации показано двойными линиями. Кроме того, $АТОР_{ij}$ связан с $ASOB_j$, $AOBOP_{r2}$, $AKAN_{r2}$. Агрегаты-имитаторы $APROC_k$ связаны только с $APBIN_{r1}$ и $AOBOP_{r2}$.

Для информационной синхронизации используемой информации применяется файл статистики. Записи на этот файл осуществляют БСС, PR.MONTEK, УПМ и программа КОНЕЦ имитации варианта моделирования ВТПП₂. По окончании имитации варианта ВТПП₂ файл статистики используется программой обработки статистики (PR.OBKABOT) и программой анализа статистики и принятия решений (PR. RESEN). Во время имитации ВТПП осуществляется ряд выдач информации: программой НАЧАЛО ($БИ_1$) для контроля вводимой информации; программой КОНЕЦ ($БИ_2$) для контроля интегральной статистики имитации; программой PR.OBRABOT ($БИ_3$) для анализа усредненных откликов при имитации процедурой Монте-Карло; программой PR.RESHEN ($БИ_4$) для принятия решений по результатам имитации.

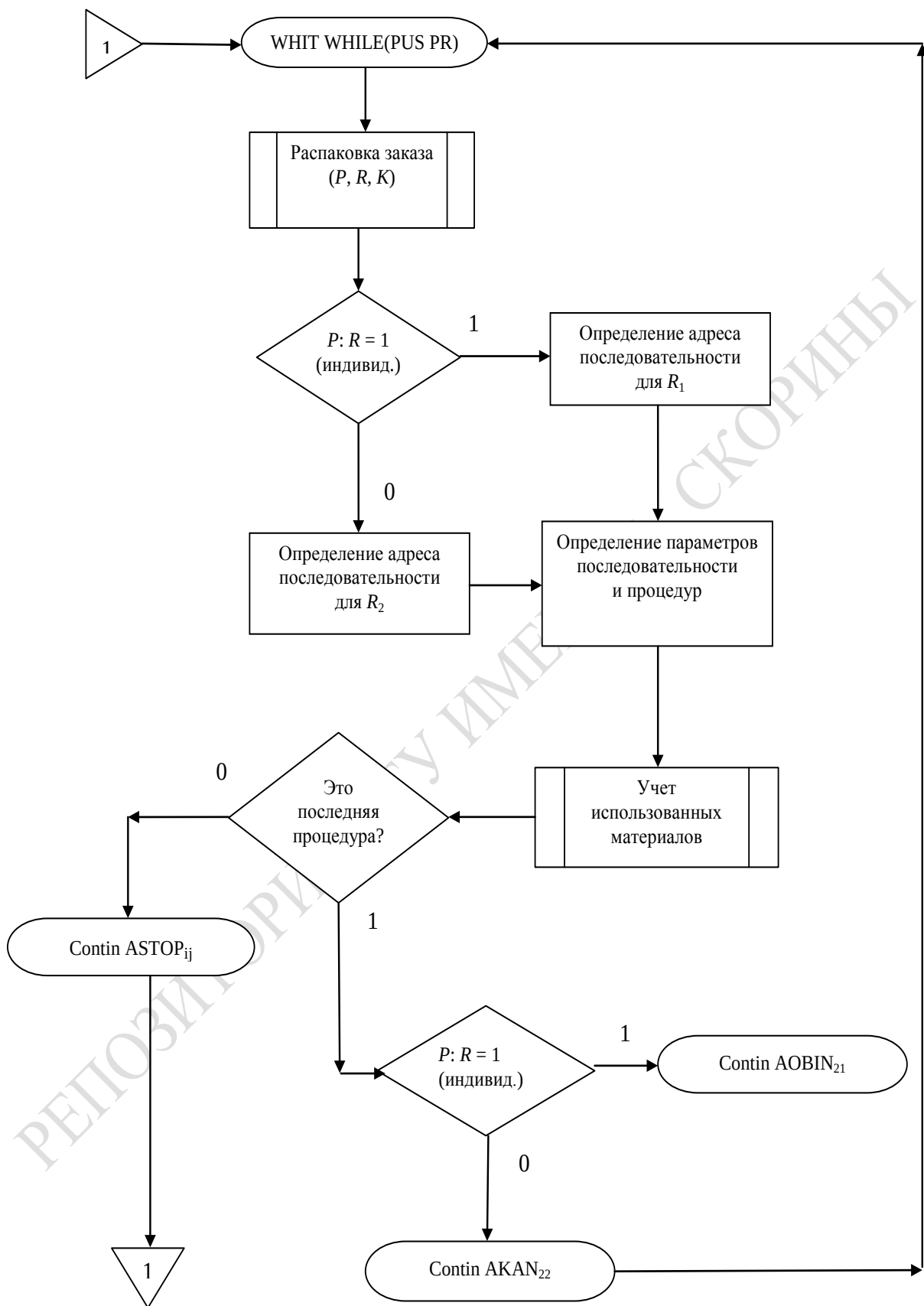


Рисунок 5.25 – Блок-схема алгоритма программы агрегата-имитатора $APROC_k$

5.6 Итоги

Подведем итоги. В 5.1 излагается общая методика преобразования формального описания в имитационную модель. Она включает реализацию следующих действий исследователя: внутреннюю синхронизацию компонент ИМ для всех способов формализации; внешнюю синхронизацию компонент ИМ; синхронизацию моментов появления информации для компоновки ИМ; отображение конфликтных ситуаций в ИМ; уточнение исходной информации для моделирования; организацию окончания имитации; контроль за ходом имитации; организацию сбора статистики; документацию этапа.

В 5.2 общая методика демонстрируется на примере построения ИМ ВП ЛВС (ТОПОЛОГИЯ ЛВС). В 5.1 демонстрируется построение полумарковских моделей РН и ВП на основе использования процедуры Монте-Карло. Излагается технология усреднения результатов имитации l -х реализаций ИМ. ВП и РН и необходимость использования; программ вторичной обработки результатов имитации.

В 5.3 демонстрируется общая методика при построении ИМ УЗЛА ЛВС, использующей информационную подкраску запросов ресурсов для решения задач адаптации РН к составу ресурсов узла ЛВС.

В 5.4 иллюстрируется общая методика на случай построения ИМ поточного производства. Более сложный пример построения ИМ на основе формального описания демонстрируется в 5.4 на случай функционирования сборочно-разборочного производства ремонта сложных изделий.

В 5.5 демонстрируется методика перевода формального описания в ИМ вероятностных технологических процессов производства, показана методика получения ИМ последовательных ВТПП₁, а в 5.6 иллюстрируется перевод вероятностного сетевого графика в ИМ ВТПП₂.

6 ПРОГРАММИРОВАНИЕ И ОТЛАДКА ИМИТАЦИОННОЙ МОДЕЛИ СЛОЖНОЙ СИСТЕМЫ

6.1 Методика программирования и отладки имитационной модели сложной системы

После того, как построено описание имитационной модели сложной системы, можно приступить к его программной реализации. В процессе этой работы необходимо: выбрать вычислительные средства автоматизации моделирования, проверить достоверность схемы программы модели, осуществить кодировку программы модели, выполнить автономную отладку компонент программы модели, провести комплексную отладку программы модели, составить окончательный вариант документации на модель.

Выбор вычислительных средств. В основу выбора можно положить требуемые затраты ресурсов ЭВМ на построение программы модели и организацию модельного эксперимента на ЭВМ. Следует учитывать при этом как размеры оперативной памяти ЭВМ, требующейся для реализации модели, так и машинное время, необходимое для трансляции и моделирования очередного варианта сложной системы. Немаловажную роль играют возможность доступа исследователя к данной ЭВМ и стоимость проведения на ней экспериментов. Однако определяющими факторами при выборе ЭВМ для реализации программы модели являются наличие на ней средств автоматизации постановки модельного эксперимента и их качество. Учет возможностей средств автоматизации моделирования весьма существен, поскольку в общем объеме работ по созданию программы модели сложной системы ее отладка составляет значительную долю.

Выбор средств автоматизации моделирования. В принципе, модель можно реализовать на любом универсальном алгоритмическом языке. Однако для облегчения программирования и отладки ИМ больших систем используют специальные языки моделирования. Языки моделирования различаются методами организации моделирования, наименованиями и структурой блоков модели, способами проверки операций и условий взаимодействия элементов моделирования, видами статистических испытаний, которым можно подвергнуть имеющиеся данные, легкостью изменения структуры модели. Однако их объединяет общее свойство – близость к языку специалистов-аналитиков.

Использование языков моделирования облегчает написание программы модели, обеспечивает более строгое следование выбранной концепции построения модели, помогает четко классифицировать элементы системы, обеспечивает гибкость программы (возможности внесения в нее изменений), позволяет различать элементы одного класса по их характеристикам и свойствам, описывает связи между элементами системы и внешней средой, позволяет корректировать число элементов модели в соответствии с изменениями внутренних условий модели.

Эти возможности обуславливаются специфическими свойствами языков моделирования, отличающими их от универсальных алгоритмических языков программирования на ЭВМ. Среди таких свойств необходимо отметить следующие способности: генерировать случайные числа, продвигать модельное время, накапливать выходные данные моделирования, проводить статистический анализ накапливаемых данных, распределять выходные данные по заранее заданным форматам, выявлять и регистрировать логические несоответствия и другие ситуации, связанные с ошибками в модели.

Кроме того, для организации квазипараллельного выполнения процессов при моделировании языки моделирования позволяют производить: определение типа события после извлечения его из списка событий; вызов стандартных программ для приведения переменных состояний в соответствие с определенным событием; идентификацию конкретных состояний; хранение и извлечение данных из списков, таблиц или массивов, включая список событий и списки состояний процессов модели. На языке моделирования исследователь может четко и ясно описывать модельные понятия.

Существенным достоинством некоторых языков моделирования является наличие у них возможностей использования стандартных блоков организации моделирования. В ряде средств автоматизации моделирования [1, 34] имеются библиотеки специальных процессов, с помощью которых можно формировать блоки задания начальных условий. Они представляют собой программные заготовки на базовом языке моделирования, которые настройкой параметров можно преобразовать к требованиям модели сложной системы. Использование таких языков моделирования значительно облегчает работу исследователя по реализации в составе модели стандартных блоков моделирования.

Проверка достоверности схемы модели. При имитации *активностями* необходимо проверить, правильно ли схема модели отражает взаимодействие активностей друг с другом. Весьма существенно проверить также места модификации глобальных переменных модели,

влияющих на совершение условий инициализации соответствующих активностей.

Для *событийного способа* имитации исследователь должен проверить, не потерял ли физический смысл объекта имитации от раздробления алгоритмов, имитирующих функциональные действия, на множество подпрограмм обслуживания событий.

Для *транзактного способа* проверка достоверности схемы особенно существенна. Может оказаться, что языки моделирования, реализующие транзактный способ имитации, существенно искажают динамику имитации и схема программы модели отличается от схемы концептуальной модели объекта моделирования.

При *процессном способе* имитации исследователь имеет дело с множеством схем соответственно числу типов процессов. Возможны два вида реализации программ процессов: с циклической и древовидной структурами. В первом случае алгоритм процесса замкнут с конца на начало и представляет собой чередование алгоритмических операторов с операторами синхронизации. Во втором случае по окончании выполнения всей цепочки операторов синхронизации процесс переходит в пассивное состояние.

В зависимости от структуры процессов проверка достоверности схемы программы модели также существенно различается. Для процессов с циклическим алгоритмом выполнения положительный исход проверки достоверности схемы каждого отдельного процесса не гарантирует достоверности всей модели. Задача усложняется, если процессы взаимодействуют друг с другом с помощью операторов запуска и останова процессов. Во втором случае, просмотрев процессы поочередно, исследователь может быть уверен в том, что программа модели соответствует замыслу. В обоих случаях при проверке схемы программы модели зачастую используются диаграммы развития процессов в модельном времени.

Агрегатный способ имитации обычно не требует специальной проверки достоверности схемы имитации в силу простоты программы модели и ее практического совпадения с формальным описанием объекта исследования. При ошибках синтаксиса описания программы системы моделирования, реализующие этот способ имитации (агрегатами), выдают исследователю весь список ошибок. Поэтому основная задача исследователя на этом этапе разработки ИМ агрегатного способа имитации сводится к исключению синтаксических ошибок описания программы модели.

При *транзактно-процессном способе* имитации, например СМ MICIS [14], в распоряжение исследователя представляются специальные

средства визуализации структуры закодированной программы. Сравнивая структуру ИМ, полученную на предыдущем этапе разработки модели со структурой, сформированной системой моделирования на основе запрограммированного текста модели, исследователь может визуально определить различие между запланированной и полученной структурами программы ИМ сложной системы.

Для случая создания программы ИМ с *агрегатно-процессным способом* имитации структура программы ИМ практически не отличается от структуры вероятностного сетевого графика. Однако сами средства реализации программы ИМ представляют листинг, практически не отличаются от ВСГР. Дополнительные программы организации имитационных экспериментов специально выделяются в тексте листинга, что существенно облегчает сравнение варианта ИМ с оригиналом текста ИМ «на бумаге», полученным на предыдущем этапе разработки ИМ сложной системы.

Отметим, что проверке по схеме не решаются вопросы информационной стыковки модели. И совсем уж плохо обстоит дело с «прокруткой» программы модели по схеме для случая, когда алгоритм поведения зависит от процессов, с которыми он находится в информационном взаимодействии. В этом случае решение проблем переносится на этап комплексной отладки имитационной модели.

Кодировка программы модели. Следующим шагом этапа программирования имитационных моделей является собственно кодировка программы модели на выбранном языке моделирования. Наиболее четко влияние кодировки на создание модели проявляется при агрегатном и транзактном способах имитации. В обоих случаях дело сводится к использованию унифицированных операторов при реализации текстов формального описания объекта имитации. Для остальных способов имитации кодировка алгоритмов осуществляется средствами базового языка моделирования, а взаимодействия и синхронизации компонент имитационной модели объекта моделирования кодируются соответствующими операторами выбранного языка моделирования.

Автономная отладка программы имитационной модели. В принципе, автономная отладка программы имитационной модели сложной системы не отличается от отладки отдельных подпрограмм. Отладка каждой компоненты модели может вестись по методикам отладки отдельных процедур. Однако необходимо отметить ряд трудностей, возникающих при автономной отладке компонент программы имитационных моделей сложных систем. Необходимо моделирование входов для каждой компоненты во времени в случае, когда управляющая информация поступает распределенно по времени. В тех случаях,

когда в компонентах модели используются стохастические элементы, исследователю при отладке приходится затрачивать дополнительные усилия на запись во внешнюю память последовательности начальных значений генераторов псевдослучайных чисел, с тем чтобы обеспечить повторяемость стохастических характеристик при отладке различных ветвей алгоритма компоненты модели.

При автономной отладке подпрограмм обслуживания активностей или событий целесообразно составлять цепочки взаимодействующих друг с другом активностей или событий, что позволит согласовать их функционирование в модельном времени. Вторым не менее важным аспектом автономной отладки подпрограмм обслуживания активностей или событий является согласование информационного обмена между ними через глобальные переменные или массивы.

Во время автономной отладки *процессов* возникают трудности, обусловленные привлечением к отладке большого числа программистов. Они, в целях ускорения работы, отладку «своих» процессов производят в параллель, используя мультипрограммный режим работы ЭВМ. При отладке ветвей программы возникает необходимость в изменении «своих» процессов (их рабочих массивов). Это зачастую приводит к дополнительной работе по приведению в соответствие отдельных «кусков» модели сложной системы. Эта трудность часто усугубляется неопытностью отдельных программистов.

Для ускорения отладки процессов, которые должны взаимодействовать с другими процессами (а последние к данному моменту времени также еще не отлажены), создаются простейшие макрокоманды на языке моделирования, моделирующие это взаимодействие.

Следует также заметить, что некоторые ошибки в процессах модели проявляются непостоянно по месту и времени, что затрудняет их поиск и устранение.

При автономной отладке модели, реализованной *транзактным способом* имитации, организуются пробные запуски всей модели и начинается отладка синтаксиса программы модели. Затем приступают к проверке правильности алгоритмов обслуживания транзактов по ветвям общей схемы модели.

Для *агрегатного способа имитации* основная трудность автономной отладки состоит в проверке правильности алгоритмов перевода агрегата из состояния в состояние и взаимодействия с УПМ. Обычно при отладке используется принцип наращивания выполняемых агрегатом функций.

При автономной отладке программ ИМ, *реализованных транзактнопроцессным способом* имитации, основная трудность состоит

в изучении полученной структуры «тела» транзакта сложной структуры, которая создана на предыдущем этапе разработки программы ИМ. Зачастую исследователю приходится составлять временные диаграммы получения и использования информации, сосредоточенной в «теле» транзакта. Именно в реализации этой операции сравнения получаемой на отдельных этапах структуры «тела» сложного транзакта с тем, что было предусмотрено исследователем на этапе разработки программы ИМ сложной системы, и состоит основная работа исследователя.

Качество автономной отладки программы ИМ, реализованной на основе *агрегатно-процессного способа* имитации, целиком определяется возможностью визуализации результатов имитации и стандартной статистики. Это обстоятельство определяет наличие в системах автоматизации моделирования с агрегатно-процессным способом имитации специальных средств визуализации: результатов *l*-х реализаций по методу Монте-Карло, формирования структуры временных диаграмм функционирования агрегатов, графиков изменения во времени остатка ресурсов и графиков роста во времени затрат на реализацию множества микротехнологических операций и суммарного расхода во времени материалов и комплектующих деталей по ходу свершения событий в ВСГР.

Важным моментом в организации автономной отладки компонент программ имитационной модели является возможность использования дополнительных средств отладки. Для отладки подпрограмм обслуживания активностей или событий зачастую применяется набор служебных подпрограмм обслуживания активностей или событий, предназначенных для имитации внешнего окружения отлаживаемого алгоритма активности или подпрограммы обслуживания событий. При *агрегатном* и *транзактном* способах имитации дополнительных средств отладки не требуется из-за высокой степени стандартизации взаимодействия блоков или агрегатов с УПМ.

Для организации *отладки процессов* обычно служит стандартный набор средств отладки, представляемый выбранной ЭВМ. В него входят: различные загрузчики рабочих программ, библиотечные и поддерживающие программы, генераторы данных, программы управления отладкой, средства автономного редактирования данных и программ. Различают два вида загрузчиков: в оперативную память и во внешнюю память. Основное их назначение состоит в помещении тела модели и рабочих массивов соответственно либо в оперативную, либо во внешнюю память ЭВМ. Библиотеки и поддерживающие программы имеют средства пополнения и исключения любого процесса, а также

средства поддержания массивов в заданном состоянии после окончания имитации очередной ветви процесса.

При *транзактно-процессном* способе имитации специальные программно-технологические моделирующие комплексы имитации (ПТКИ) имеют специальные средства отладки ИМ. Использование таких средств отладки является основной возможностью отладки реентерабельных программ имитаторов устройств. Например, в различных версиях ПТКИ, построенных на основе базовой системы моделирования MICIS [16], имеются специализированные средства отладки процессов и подсистем программы ИМ дискретных технических систем.

Весьма существенно для разработчика программы модели иметь в используемом языке моделирования средства обнаружения ошибок. Сюда входят: программы распечатки состояний процессов, операторы трассировки алгоритмов процессов, средства мгновенной фотографии развития процесса во внешней памяти ЭВМ.

Программы управления отладкой обеспечивают исследователю три вида услуг: введение вставок в алгоритм процесса; управление вводом и выводом информации для данного процесса; включение отладочных средств, разработанных разработчиком модели.

Средства автономного редактирования данных и программ позволяют пользователю: выводить на печать информацию с внешней памяти, приводить к необходимому формату и сортировать данные; распечатывать последовательности макрокоманд; осуществлять автономную сортировку исходной информации; редактировать массивы с сортировкой входных и выходных данных модели.

Зачастую эффективным средством при отладке программы имитационной модели с древовидной структурой является трассировка алгоритма компонент модели. Возможны три вида организации трассировки компонент модели: каждой команды, фиксации переходов в алгоритмах компонент, регистрации выполнения только макрокоманд.

Вообще автономная отладка компонент модели требует двойного объема работ программистов по сравнению с отладкой процедур обычных рабочих программ. Увеличение объема работ обуславливается необходимостью обеспечить воздействие «внешней среды» компонент модели на их поведение. В состав «внешней среды» компоненты модели входят имитаторы поведения других компонент и, возможно, блоки, имитирующие внешнее окружение моделируемой сложной системы.

Комплексная отладка программы модели. Отладив каждую компоненту модели в отдельности, исследователь объединяет компоненты в единую модель. При этом возникают ошибки из-за несоответствия

в динамике функционирования отдельных компонент и наличия неточностей в информации взаимодействия между компонентами.

При имитации *процессами, активностями и событиями* исследователь вначале упрощает взаимосвязи между двумя процессами (активностями, программами обслуживания событий) по информации (ставит соответствующие «заглушки» в алгоритмы процессов) и лишь после этого переходит к отладке их взаимодействия. Начинается эта отладка по операторам пуска и останова компонент модели. Затем отлаживается информационное взаимодействие между двумя процессами (активностями, программами обслуживания событий). К двум отлаженным компонентам добавляется третья с упрощенными связями по информации (заглушками в алгоритме третьей компоненты). Отлаживают сначала взаимодействие между тремя компонентами по операторам пуска и останова этих компонент, а затем переходят к отладке информационных взаимодействий между тремя компонентами (устраняются заглушки в их алгоритмах). При этом широко используются временные диаграммы функционирования компонент модели. В компоненты модели обычно вставляются операторы выдачи содержимого таблиц процессов (активностей, подпрограмм обслуживания событий) и результатов отладки, предоставляемые системой моделирования.

По окончании проверки соответствия информации алгоритму поведения компонент модели эти операторы отладки исключаются из тела компонент модели. Существенно помогает при этом наличие библиотеки процедур у системы моделирования и средств оперативной компоновки программы модели из библиотеки процедур. Развитые средства автоматизации моделирования, как правило, предоставляют в распоряжение разработчика программы модели сложной системы широкий спектр диагностических сообщений об ошибках динамики имитации. Это значительно облегчает исследователю локализацию динамических ошибок в компонентах модели. Таким способом модель наращивается по методу «этап за этапом». Начиная с отладки взаимодействия двух компонент, исследователь затем наращивает программу модели до полного состава. При каждом добавлении новой компоненты модели необходимо проверить взаимодействие ее с внешней средой. Таким образом, разработчики фактически создают последовательность вложенных программных моделей, и опыт отладки одной модели используется для улучшения последующей модели.

При транзактном способе имитации комплексная отладка модели сводится к отладке семантики обслуживания транзактов по схеме имитационной модели. Например, зачастую для успешной отладки

имитационной модели достаточно отладить синтаксис операторов языка, реализующего транзактный способ. Поэтому-то и язык моделирования GPSS [33], реализующий транзактный способ имитации, получил широкое распространение. Простота использования и высокие технологические возможности (прежде всего в программировании и отладке) для многих исследователей делают GPSS (а с ним способ имитации) предпочтительным перед другими языками моделирования, несмотря на его ограниченные описательные возможности и слабые средства обработки статистики и планирования экспериментов.

В случае агрегатного способа имитации комплексная отладка модели ограничивается согласованием взаимодействия агрегатов по результатам пробного моделирования.

Вообще отладка программы модели имеет много общего с отладкой больших программ реального времени. Поэтому при комплексной отладке программ ИМ можно использовать методики отладки программ реального времени.

Документация программы модели. Завершается создание программы модели сложной системы составлением программной документации на модель. Традиционные методы составления программной документации не подходят для случая проектирования моделей сложных систем. Необходимы более точные и подробные спецификации и описания модели. Обычно при создании рабочих программ моделей сложных систем документация создается после написания и отладки программ компонент модели. При разработке программ моделей больших систем документация не должна рассматриваться как нечто дополнительное, что можно сделать после завершения основной работы по составлению программ. В данном случае документация представляет собой инструмент, без которого разработка комплекса взаимосвязанных программ станет чрезвычайно затруднительной.

Одной из трудностей при разработке компонент модели является частое изменение самого логического построения процедуры, реализующей реальный процесс. При отладке компонент модели исследователь обнаруживает непредвиденные трудности. Он постоянно изменяет процедуры. Однако, учитывая ограничения, налагаемые из-за того, что его работа должна сопрягаться с работой разработчиков других компонент модели, он должен подходить к внесению изменений очень осторожно. Согласно Р. Шеннону [30] разработчик не должен изменять структуру информации, используемой многими другими разработчиками, без консультации с ними. Поэтому все вносимые изменения должны тщательно документироваться. В противном случае

разработка общей системы выйдет из-под контроля, и одни разработчики не будут знать, что делают другие.

В условиях постоянных изменений надо установить жесткий контроль за состоянием и ведением документации, иначе нет никакой надежды на то, что в момент объединения компонент в модель они будут соответствовать друг другу. Именно поэтому необходимо так проработать документацию, чтобы были обеспечены возможности введения существенных изменений и обязательное ознакомление с ними всех программистов, которые должны о них знать. Как следует из [30], при создании программы модели большой системы поток документации может стать таким большим, что возникнет необходимость в принятии специальных усилий к сокращению изменений до минимума после начала отладки компонент модели.

При комплексной отладке осуществляется объединение множества компонент модели. При этом, естественно, могут возникнуть ошибки, которые часто трудно проследить и исправить. Наличие упорядоченной документации может существенно помочь в этом. Уровень детализации документации зависит от сложности разрабатываемой программы модели. Для модели сложной системы принято составлять подробные и тщательно проработанные спецификации компонент модели. Согласно [30] каждой спецификации компоненты должен присваиваться кодовый номер, который точно отражает тип спецификации. Этот номер, в частности, содержит номера всех других спецификаций, связанных с данной компонентой, чтобы можно было проследить внесение изменений в других компонентах. Важно в составе разработчиков программы модели иметь «библиотекаря», который будет контролировать состояние документации у остальных разработчиков и регистрировать все вносимые изменения, появляющиеся по мере развития модели системы. Для регистрации этих постоянных изменений необходимо предусмотреть средства ведения соответствующих архивов. Напомним, что нужно стремиться к тому, чтобы поток информации об изменениях в модели был минимальным, так как иначе разработчики моделей отдельных компонент системы могут «потонуть» в бумагах и некогда будет отлаживать компоненты модели.

Не будем приводить описание программ модели наших примеров. Для специалиста, владеющего языком моделирования, реализующего соответствующий способ имитации, программирование и отладка модели не представляют трудностей. Заметим лишь, что в этой работе многое зависит от степени работоспособности версии системы автоматизации моделирования, имеющейся в распоряжении исследователя. В данном случае (как, впрочем, и в любом другом) можно

рекомендовать использовать для программирования наиболее проверенные и технологичные средства моделирования. Иначе исследователю придется тратить больше времени на отладку самой системы моделирования, чем на отладку программы модели.

6.2 Использование программно-технологических комплексов имитации при отладке ИМ сложных систем

Из работы [11] следует, что основное технологическое достоинство программно-технологических комплексов имитации (ПТКИ) заключается в расширении возможностей базовой системы моделирования MICIS [14] путем введения «информационной подкраски» транзактов. Суть этой подкраски состоит в том, что транзакт i -го номера (типа) TR_i включает в себя целое множество характеристик, которые в совокупности составляют «тело транзакта», расположенное в базе данных модели по адресу (α_i), а сам TR_i обладает приоритетом (π_i) обслуживания его в очередях ИМ. Таким образом, по очередям ИМ движется триада (i, π_i, α_i), а основная информация о транзакте находится в «теле» транзакта. Внутрь «тела» TR_i помещается важная информация для каждой предметной области. В том числе туда помещаются следующие характеристики:

- запросы, влияющие на траекторию движения TR_i в ИМ объекта исследования (ПТКИ УПС) [14];
- технологическая карта движения модели изделия сложной структуры составления части и затем порядок последующей сборки компонент в единое изделие (ПТКИ СПО) [14];
- запросы пользователей на ресурсы ЛВС (ПТКИ ЛВС) [14].

Кроме расширения состава моделей запросов, хранящихся в «подкрашенных» TR_i , используется транзактно-процессный способ имитации, реализованный в базовой СМ MICIS [18].

Для написания программ ИМ с помощью ПТКИ любой версии разработчик ИМ должен владеть классическим языком программирования C^{++} , а также изучить программный интерфейс СМ MICIS 4 [14]. Язык MICIS 4 состоит из трех групп реализации модулей:

- функциональных модулей (ФМОД), используемых для определения и реализации экспериментов с ИМ (определение глобальных данных, выбор типа эксперимента, входных параметров модели);
- модулей запуска ИМ на имитацию, обработки и вывода результатов;

– информационных модулей (ИМОД), используемых для определения информационного взаимодействия элементов модели (установка иерархической структуры ИМ, описание интерфейсов компонентов модели, определение типов экспериментов, программирование активностей, определение необходимого для данной модели множества функций и методов);

– системных модулей (СИМОД), обеспечивающих интерфейсы для программиста ИМ (реализация базовых классов для компонентов модели, статистик, откликов, трасс, постановки эксперимента и стохастических потоков внешних событий, разработка множества системных функций);

– организация квазипараллелизма и управления имитацией, предоставление механизмов работы с датчиками псевдослучайных чисел, настройка типовых переменных для объектов модели, программирование функций и свойств элементов в соответствии со схемой формализации.

Необходимо отметить следующие возможности СМ MICIS, которые использованы при создании версий ПТКИ:

1 Вложенность языка моделирования в стандартный язык программирования C⁺⁺, что обеспечивает кодирование и отладку ИМ внутри любой интегрированной среды разработки C⁺⁺ приложений на произвольной платформе.

2 Открытость, когда MICIS обеспечивает исследователю возможность решения собственных задач.

3 Нет необходимости обучения разработчика ИМ метода программирования.

4 Модульный подход при программировании ИМ и предоставление интерфейса для внедрения модулей MICIS4 в программное обеспечение пользователя.

5 Компактность объектных модулей программы ИМ и реализации множества классов для генерации стандартных и пользовательских отчетов.

6 Предоставление средств надстройки к СМ для обработки и отображения результатов моделирования.

Для облегчения отладки ИМ в среде СМ MICIS имеется возможность графического отображения механизмов обслуживания путем составления графа переходов между активностями. Вершинам соответствуют активности ИМ, а маркированные направления являются дугами (переходы между активностями и внешним управляющим сигналом). Механизм обслуживания содержит две активности: начальную и конечную. Они изображаются графически следующим

образом: у устройства они соединены одинарной сплошной дугой, а у очереди – отображаются пунктирной. Двойные дуги являются висячими и предназначены для указания управляющих сигналов в направлении других элементов ИМ.

В работе [27] приведена базовая схема формализации (БСФ) структурообразующих сервисов, представленных в языке моделирования MICIS 4. Использование БСФ существенно облегчает отладку ИМ в среде ПТКИ. Для этой цели разработчикам СМ MICIS 4 предлагается использовать схемы с минимальным числом сервисов. При построении формальной ИМ объект моделирования нужно рассматривать как совокупность элементов: источников, ресурсов, потребителей и операторов сервиса. Разработчику ИМ предлагается сосредоточить свои усилия только на внутреннем взаимодействии потребителя сервиса и ресурса сервиса. Вторым средством, обеспечивающим отладку ИМ в среде СМ MICIS 4, является механизм использования *контейнеров*, представляющих собой совокупности реализованных элементов и средств (интерфейсов) для работы с ними. В качестве *контейнера* ИМ могут выступать:

- консольное предложение, запускаемое оболочкой операционной системы;
- специализированное или офисное Windows-приложение, которое настроено на задание исходной информации для ИМ, запуск ИЭ на выполнение и каталогизацию результатов моделирования;
- автоматизированные системы управления пользователя с входными и выходными данными моделирования (прежде всего автоматизированным вводом исходной информации и встроенной средой обработки результатов имитации);
- Web-приложения, когда ИМ выполняется на сервере, а результаты ИЭ документируются в одном из Web-форматов, а затем по сети передачи данных они отправляются на компьютер клиента.

В результате *контейнер* ИМ облегчает отладку модели, облегчая работу с функциональным и информационным модулем ИМ удобным способом на более высоком уровне. Он может обеспечивать возможность задания параметров ИМ для постановки работ ИЭ, организации итеративных вычислений. Применение *контейнеров* ИМ существенно облегчает отладку алгоритмов корректировки исходных данных и запуска отдельных ИЭ, а затем в другой среде обработки может осуществить испытание, исследование свойств ИМ и анализ результатов ИЭ.

В итоге решение различных задач с помощью ИМ происходит в специализированной среде обработки. (например, в популярных пакетах статистического анализа данных). Результаты прогонов ИЭ

представляют собой динамические ряды, а результаты опытов и ИЭ в целом представляют собой обычные выборки данных. Статические пакеты G-Quss, Maple, Mathematika, Matlab, MS Excel, Statistika [2, 29, 32] обладают широким набором возможностей обработки данных, освобождая пользователя от программирования алгоритмов анализа результатов моделирования.

Например, в ПТКИ ЛВС имеется возможность использования дополнительных средств отладки в виде набора служебных подпрограмм, входящих в состав стандартного набора СМ MICIC4. В него входят: загрузчики рабочих программ; библиотечные и поддерживающие программы, генераторы данных, программы управления отладкой, средства редактирования данных и программных модулей. Эффективным средством комплексной отладки ИМ в среде ПТКИ ЛВС [14] является трассировка компонентов ИМ. Комплекс допускает три вида организации трассировки: покомандная фиксация переходов в алгоритмах компонента; регистрация выполнения только макрокоманд имитации. По окончании проверки соответствия информации алгоритму поведения компонента ИМ операторы отладки исключаются из текста ИМ. Существенно помогает при этом наличие библиотеки процедур у СМ и средств оперативной компоновки программы ИМ по библиотеке процедур.

Предлагается наращивать ИМ по методу «этап за этапом», начиная с отладки двух компонентов ИМ и кончая отладкой взаимодействий всех остальных компонент модели. При каждом добавлении нового компонента ИМ необходимо проверить взаимодействие его с внешней средой; таким образом, разработчик ИМ фактически создает последовательность вложенных программных моделей и опыт отладки одной модели используется для улучшения последующей модели. Комплексная отладка обычно проводится либо по ретроспективному способу, либо по способу анализа контрольных точек. Обычно для ИМ задаются такие значения параметров и начальных значений, которые они должны пройти через определенное количество шагов изменения модельного времени t_0 к состоянию, известному заранее исследователю. Затем необходимо повторять расчет итоговых откликов ИМ заданное число раз. Целью повторений является имитация различных вариантов конфигураций и характеристик исследуемой системы. За счет использования графических и диалоговых возможностей контроля за постановкой ИЭ в среде MICIC 4 существенно облегчается отладка ИМ. Кроме того, имеются возможности: выбора новой модели; определения новых параметров модели; активизации модели, изменения состояния модели, мониторинга процесса имитации.

При определении новых параметров ИМ обеспечиваются ввод, просмотр и корректировка их значений в ходе имитации. На экране монитора возникает окно с таблицей, которую необходимо заполнить, указав в ней три этапа и изменив тип и размерность параметра. Специальные клавиши ПЭВМ используются для ввода, просмотра и корректировки параметров ИМ. Таким образом, отделив описание параметров ИМ от их определения, исследователь получает в лице ПТКИ программные средства динамического контроля за параметрами ИМ. Просмотр откликов ИМ сопровождается их отображением на экране компьютера в ходе ИЭ. При этом выводится такая же таблица, что и при определении параметров ИМ. До запуска на имитацию прообраза ИМ значения откликов модели равны нулю. При каждой остановке ИЭ отклики автоматически пересчитываются, что позволяет оценить их. Активизация ИМ приводит к имитации ее до очередного останова в режиме однопрогонного ИЭ. ПТКИ ЛВС предлагает исследователю меню, где перечислены способы окончания очередного интервала моделирования по истечении промежутка времени T ; свершении некоторого условия; по оператору языка разработчика ИМ Finish; специальному сигналу с клавиатуры; выходу контролируемых параметров за указанные заранее границы; по смене модельного времени t_0 ; выполнению некоторой последовательности активностей.

На начальных этапах комплексной отладки ИМ рекомендуется применять событийный механизм останова имитации, согласно которому при смене модельного времени t_0 ПТКИ ЛВС спрашивает пользователя, остановить или продолжить имитацию. Если же локализовать ошибку моделирования не удастся событийным способом, то рекомендуется следующая последовательность действий. Активируется ИМ с остановом по специальному сигналу. Поскольку в ходе имитации значение модельного времени t_0 отображается на экране компьютера, то можно визуально определить момент, когда t_0 достаточно долго не меняется, и нажать клавишу F11. Если исследователь поторопился дать сигнал останова имитации, он может продолжить имитацию с прерванного места. При установлении факта «зацикливания» имитации необходимо запомнить время этого зацикливания (t^*), перезагрузить ИМ, запустить прообраз ИМ, восстановить структуру модели и активизировать модель на интервал времени меньший чем момент зацикливания ($t < t^*$). Затем, используя средства мониторинга, необходимо установить режим «просмотра» всех параметров ИМ, значения которых, по мнению исследователя, могли породить ошибку имитации. Далее продолжается отладка ИМ с шагом изменения модельного времени по активностям, при этом сравниваются контролируемые

значения откликов и статистик имитации и редакция СМ до момента «зацикливания» ИМ. Для этого можно использовать диалоговые средства просмотра откликов и статистики и изменения структуры ИМ в среде ПТКИ ЛВС. Если же ИМ вновь «зациклилась», то после «перезагрузки» ИМ рекомендуется выйти на режим редактирования текста «циклящегося» участка компоненты ИМ, проанализировать ошибочный фрагмент и затем откорректировать его. На более поздних шагах комплексной отладки следует применять интервальный способ останова имитации.

Дополнительным средством отладки ИМ является использование процедур изменения состояния ИМ. С помощью нажатия соответствующих функциональных клавиш возможно сохранение текущего состояния ИМ, восстановление одного из предыдущих ее состояний, удаление множества старых состояний. Для удобства принятия решения о выборе некоторой точки останова все сохраняемые состояния образуют дерево, у которого вершины являются именами соответствующих файлов, а дуги означают направления исследований на ИМ. В режиме восстановления исследователь перебирает вершины дерева, применяя быстрый поиск по введенному шаблону, и отмечает то состояние, в которое он снова хотел бы вернуться. В результате этого текущее состояние ИМ объявляется «старым» и к дереву вариантов поиска состояний добавляется новая ветка. Далее исследователь может продолжить отладку ИМ в другом направлении – меняя либо структуру и параметры ИМ, либо способ их активизации.

Для динамического отображения процесса имитации предлагаются возможности системы мониторинга ПТКИ ЛВС (диалог и отображение результатов имитации). Диалог позволяет оперативным образом установить или модифицировать состав тех числовых и графических объектов имитации, значение которых исследователь хочет контролировать в ходе ИЭ отображение результатов. Обеспечивается вывод на экран и запись в файл статистики значений объектов наблюдения, которые установлены в ходе диалога. Для динамического изменения структуры ИМ исследователь в среде ПТКИ ЛВС может осуществить следующие действия: просмотр и редактирование параметров устройства; изменение состояния устройств ИМ; редактирование состава очередей к устройству; просматривание «тела» транзактов; перемещение копии транзактов; просматривание всех устройств, на которых обслуживались транзакты выбранного типа.

Более подробно с технологиями отладки ИМ в среде ПТКИ можно ознакомиться в работе [14], а также в технической документации каждой версии ПТКИ [21, 27].

6.3 Технологические возможности отладки ИМ агрегатного типа в среде системы моделирования САИМ

Для отладки ИМ ВТПП предлагается использовать соответствующую версию системы автоматизации имитационного моделирования (САИМ) состав, структура и технологические возможности которой изложены в работе [26]. Например, отладку ИМ последовательных ВТПП можно проводить с помощью первой версии САИМ-1, реализующей процессный способ имитации. Она позволяет автоматизировать все этапы создания и использования программы ИМ последовательных ВТПП. Для отладки ИМ систем управления (СУ) оборудованием ВТПП предлагается использовать вторую версию САИМ-2, реализующую агрегатный способ имитации. Третья версия САИМ-3, реализующая агрегатно-процессный способ имитации, автоматизирует этапы отладки, испытания и эксплуатации ИМ параллельно-последовательных ВТПП графовой структурой взаимодействия $\{MTXO_{ij}\}$ [26]. Причем с помощью САИМ-3 исследуются ВТПП при наличии отказов оборудования ВТПП.

Все три версии САИМ используют ряд универсальных подсистем для обеспечения этапов отладки и имитация ИМ, а также для обработки статистики и визуализации результатов имитации. Каждая программа этих универсальных подсистем реализована в среде системы программирования DELPHI [32]. Она готова к использованию, и поэтому исследователю нет необходимости в ее модификации. Исходя из этого соображения, рассмотрим процесс отладки новых ИМ ВТПП для тех случаев, когда структура ИМ ВТПП не устраивает исследователя. Дело в том, что ИМ ВТПП в составе САИМ настолько специализированы по предметным областям, что можно использовать уже готовые ИМ, находящиеся библиотек версий САИМ [26].

Отладка новых ИМ ВТПП в среде версий САИМ сводится не к программированию алгоритмов процессов или агрегатов, а представляет собой компоновку структуры ИМ из различного числа копий реентерабельных программ процессов и агрегатов. Поэтому отладка вариантов ИМ ВТПП сводится к реализации процесса «запитки» исходной информацией для использования универсальных реентерабельных программ процессов или агрегатов. Например, этап отладки ИМ ВТПП с помощью САИМ-3 реализуется следующей последовательностью шагов:

- 1 Формирование структуры ИМ ВТПП в среде соответствующей версии САИМ.

2 Подготовка исходной информации о моделируемом ВТПП.

3 Задание параметров компонент ИМ ВТПП и запись их в базу данных модели.

4 Составление таблиц коммутации агрегатов в структуре ИМ ВТПП.

5 Инициация выполнения базового варианта моделирования и настройка универсальной структуры ИМ ВТПП для конкретной структуры варианта ВТПП.

Рассмотрим последовательность действий исследователя при реализации перечисленных шагов технологии получения работоспособного варианта ВТПП с помощью версии САИМ-3 [26].

На шаге 1 структура ИМ ВСГР, отображающая параллельно-последовательный ВТПП, задается следующим образом:

- формируются таблицы структуры множества агрегатов $\{АТОР_{ij}\}$;
- задается информация о структуре множества агрегатов $\{АSOB_j\}$ в виде таблицы;
- формируется таблица коммутации агрегатов $АТОР_{ij}$ с агрегатами $АSOB_i$ и $АSOB_j$ с помощью типовых сигналов (действительных Sgd и фиктивных Sgf).

Фактически в таблице коммутации агрегатов $АSOB_i$ и $АТОР_{ij}$ указывается коммутация агрегатов при прямой имитации ВСГР ($АSOB_i \rightarrow АТОР_{ij} \rightarrow АSOB_j$) и при инверсной имитации ВСГР ($АSOB_i \leftarrow АТОР_{ij} \leftarrow АSOB_j$). Сигналы Sgd и Sgf формируются в режиме диалога в процессе указания в таблице структуры коммутации агрегатов. Сигналы указываются сразу для каждого агрегата $АSOB_j$ (в порядке возрастания номеров j). В таблице коммутации выходов $АSOB_i$ указываются выходные сигналы в виде строк, в которых задана адресация («откуда» – «куда»). Предлагается информацию о выходных сигналах располагать в порядке возрастания номеров событий. При этом каждому событию соответствует своя группа строк, содержащих информацию о выходных сигналах.

На шаге 2 организуется либо натурный эксперимент на реальном ВТПП, либо используются экспертные значения параметров реализации $АТОР_{ij}$. Основная трудность в подготовке исходной информации состоит в определении вероятностных характеристик параметров $АТОР_{ij}$. В тех случаях, когда трудно найти аналитический вид для аппроксимирующих функций распределения, их можно сформировать в виде табличных функций распределения. Табличная запись функций распределения стандартизирована для всех функций распределения. При каждой записи значений параметров $АТОР_{ij}$ осуществляется преобразование их во внутреннее представление.

На шаге 3 осуществляется контроль корректности описания $АТОР_{ij}$ путем устранения ошибок коммутации агрегатов. Технология взаимодействия исследователя с САИМ-3 реализуется на основе «меню» в режиме «вопрос – ответ». По окончании этого шага синтаксические ошибки в описании ВСГР будут исправлены.

На шаге 4 таблицы коммутации агрегатов $ASOB_i$, $ASOB_j$ и $АТОР_{ij}$ проверяются на соответствие входов и выходов четырех типов ($АТОР_{ij}$) с соответствующим входом и выходом агрегатов ($ASOB_i$ и $ASOB_j$). Любое дублирование входов в $АТОР_{ij}$ или $ASOB_j$ немедленно фиксируется, и формируется соответствующее сообщение исследователю на экран монитора. При этом осуществляется автоматическая проверка входов в $ASOB_i$, как при прямой имитации (слева направо по ВСГР), так и при инверсной имитации (справа налево по ВСГР). Все выявленные несоответствия сообщаются исследователю, и далее в режиме диалога все ошибки коммутации агрегатов исправляются. По завершении этого шага исследователю выдается окончательная структура всех таблиц коммутации, в которых исправлены ошибки.

На шаге 5 осуществляется начальный запуск базового варианта ИМ ВСГР, и САИМ-3 предлагает исследователю в режиме «пошагового выполнения» просмотреть переходы всех агрегатов из состояния в состояние с автоматической документацией этого процесса. Таким образом, достигается автоматизация процедуры отладки ИМ ВСГР во всех режимах имитации.

Как видим, сама отладка ИМ ВСГР не требует от исследователя высокой квалификации по программированию. Поэтому ее осуществляет сам технолог предприятия, не являясь при этом специалистом по программированию и имитации. Другое дело, когда структура ВСГР не является стандартной и необходимо переработать алгоритмы реентерабельных программ агрегатов имитаторов, функционирующих на основе агрегатно-процессного способа имитации, тогда, используя обычные средства программирования, например систему программирования DELPHI [32], исследователю необходимо разработать новые программы агрегатов-имитаторов $МТХО_{ij}$. Такая ситуация может иметь место в том случае, когда состав ресурсов ВТПП исследователь хочет существенно расширить. В этом случае изменяются алгоритмы агрегатов и характер использования новых типов оборудования ВТПП и ресурсов предприятия.

С нашей точки зрения, такую модификацию ИМ ВТПП должны выполнять разработчики версий САИМ, ибо изменяются состав операторов организации квазипараллелизма и алгоритм УПМ, реализующий выполнение новых операторов организации имитации.

Для основной части пользователей САИМ-3 [26] приведенных выше средств автоматизации и технологии имитации ВТПП достаточно. Поэтому на этом мы завершим рассмотрение типовой структуры ВСГР.

6.4 Итоги

Подведем итоги. В 6.1 излагается общая методика программирования и отладка ИМ СС, которая предлагает реализацию следующих действий исследователя: выбор вычислительных средств; выбор средств автоматизации моделирования; проверку достоверности схемы модели по типам имитации; организацию автономной отладки программ ИМ; проведение комплексной отладки программы ИМ, документация программы модели. Для реализации этой методики демонстрируется два технологических средства автоматизации имитации СС. В 6.2 демонстрируется методика использования ПТКИ при отладке СС. В распоряжение исследователя представляются три версии комплекса: ПТКИ ЛВС, ПТКИ СРО, ПТКИ УПС [14]. В 6.3 демонстрируется применение для отладки программ ИМ версий системы моделирования САИМ реализующих: процессный способ имитации (САИМ-1); агрегатный способ имитации (САИМ-2); агрегатно-процессный способ имитации (САИМ-3) [26].

7 ИСПЫТАНИЕ ИМИТАЦИОННЫХ МОДЕЛЕЙ СЛОЖНЫХ СИСТЕМ

7.1 Методика испытания имитационных моделей СС

После того как ИМ реализована на ЭВМ, исследователь должен выполнить следующую последовательность технологических этапов: испытание ИМ, исследование свойств модели, составление плана имитационных экспериментов на ЭВМ, эксплуатацию модели.

Как уже отмечалось, в ИМ можно различать такие составляющие, как компоненты, переменные, параметры, функциональные зависимости, ограничения, целевые функции. Под компонентами понимают составные части, которые при соответствующем объединении образуют систему. На реальную систему воздействуют переменные G^* , которые можно измерять, но нельзя ими управлять, и параметры X^* , которые исследователь может изменять в ходе натурных экспериментов. На выходе системы возможно измерение выходных характеристик Y^* . При этом существует некоторая неизвестная исследователю функциональная зависимость между выходными характеристиками, переменными и параметрами системы: $Y^* = \varphi^*(X^*, G^*)$. Модель системы, в свою очередь, определяется как совокупность компонент, объединенных для выполнения заданной функции $Y = \varphi(X, G)$. Здесь Y, X, G – векторы соответственно результата действия модели системы, параметров моделирования, переменных модели. Параметры модели X исследователь в ходе имитационного эксперимента может выбирать произвольно. Переменные G модели должны принимать только те значения, которые характерны для данного объекта моделирования или условий его функционирования. Поэтому в ходе имитации реальных процессов вектор G формируется на основе реальной информации. Для этой цели обычно замеряются фактические значения переменных G^* (внешней нагрузки) и результатов поведения системы Y^*_Q при заданных значениях параметров системы Q^* (Q^* является выборкой измеренных значений вектора X^*).

Испытания ИМ начинаются с задания исходной информации моделирования. Перед исследователем чаще всего возникают следующие проблемы: где взять исходную информацию, имеется ли возможность использования прототипов объекта моделирования для измерения переменных G^* , применимы ли статистические гипотезы при

стохастическом характере переменных G^* , возможна ли аппроксимация сложных процессов при определении G .

Затем необходима верификация модели, которая состоит в проверке соответствия ИМ тому замыслу, который был заложен при ее разработке. Исследователь должен получить гарантию того, что создание ИМ во всех ситуациях будет правильно отражать поведение реальной системы. С помощью верификации устанавливается верность логической структуры модели. Обычно верификация выполняется в ходе комплексной отладки модели на реальном потоке данных (при заданных значениях G и X).

Будем различать модели существующих и проектируемых систем. При моделировании существующих систем исследователя прежде всего интересует, насколько хорошо модель представляет моделируемую систему. Модель, поведение которой слишком отличается от поведения моделируемой системы, практически бесполезна. Поэтому для моделей существующих систем исследователь должен выполнить проверку адекватности ИМ объекту моделирования. Необходимо проверить соответствие между поведением реальной системы (Y_Q^*) и поведением ИМ (Y). Для построения модели проектируемой системы производится упрощение намеченной проектировщиком структуры до такой степени, когда становится возможной имитация проектируемых процессов системы на ЭВМ. Возможности и способы упрощения структуры проектируемых систем для целей построения моделей рассматриваются в [11, 13]. Для моделей проектируемых систем проверку адекватности выполнить очень трудно, поскольку нет реального объекта для сравнения с ИМ. Поэтому в данном случае этап проверки адекватности ИМ объекту моделирования зачастую отсутствует.

Для обеспечения адекватности ИМ реальному объекту исследования осуществляется калибровка модели. Цель калибровки состоит в модификации вида функции $\varphi(X, G)$ модели таким образом, чтобы модель стала адекватно отображать поведение реальной системы.

Исследование свойств ИМ предполагает последовательность выполнения следующих действий: оценка погрешностей имитации из-за неидеальности генераторов псевдослучайных чисел, используемых в ИМ; определение времени моделирования, необходимого для достижения заданной точности моделирования; оценка устойчивости результатов моделирования Y при различных изменениях параметров X и заданных значениях переменных G ; оценка чувствительности результатов моделирования к изменению параметров модели X и исходных данных моделирования G .

Процедура задания исходной информации для моделирования зависит от типа модели. Если мы имеем дело с функционирующей системой, то зачастую организуется измерение характеристик поведения системы Y_Q^* , рабочей нагрузки системы G^* и параметров функционирования системы Q^* . В ходе этих измерений определяются точностные характеристики каждой компоненты векторов G^* , Q^* , Y_Q^* . Важную роль при этом играют качество измерительных средств и их точностные характеристики. Одним из основных требований, выдвигаемых к измерителям параметров рабочей нагрузки и поведения системы, является их малое влияние на алгоритм функционирования реальной системы и соответствие их возможностей диапазону измеряемых значений характеристик.

При моделировании проектируемых систем, а также при моделировании функционирующих систем (когда нельзя организовать измерения) для формирования исходной информации проводят измерения на прототипах систем. Если таких прототипов не существует, то исследователю при построении модели приходится использовать экспертные оценки внешней нагрузки G^* и характеристик поведения системы Y_Q^* .

Итак, в ходе натурного эксперимента по результатам измерений параметров рабочей нагрузки G^* определяется вектором точностей представления ее компонент δG и уровнем детализации алгоритма системы, т. е. точностью аппроксимации зависимости $\varphi^*(X^*, G^*)$ функцией $\varphi(X, G)$. Весьма существенно соблюдать при этом правило баланса точностей [22]. Согласно этому правилу необходимо соизмерять отклонения модели от описания системы с погрешностью в задании параметров описания модели. Должно быть достигнуто соответствие точностей представления отдельных элементов модели и модели входной нагрузки на систему.

Когда нет возможности определить значения компонент вектора G^* на основании измерений, приходится полагаться на субъективные (экспертные) оценки. Чаще всего при этом используется метод Дельфи [30]. Это итерационная процедура, которая позволяет подвергать мнение каждого эксперта критике со стороны всех остальных специалистов, не заставляя их фактически сталкиваться лицом к лицу. Координатор опроса регулирует процедуру анализа мнений и сохраняет их анонимность. Полученная групповая оценка усредняется и доводится до каждого члена группы экспертов. Затем экспертов просят пересмотреть свою оценку или кратко обосновать свое мнение, если оно выходит за рамки общепринятого. Членам группы сообщаются результаты второго тура и письменные объяснения предельных значений.

При этом сохраняется их анонимность. После нескольких итераций координатор подводит итог и завершает процедуру экспертных оценок векторов G^* , Q^* , Y_Q^* .

При построении модели рабочей нагрузки G по значениям измеренных или установленных экспертным путем компонент вектора G^* широко применяются статистические методы проверки гипотез о близости [80]. В качестве задаваемых характеристик качества модели рабочей нагрузки на систему G используются уровень доверия β и вектор точностей моделирования переменных δG . В связи с этим перед исследователем возникает множество проблем, связанных с конкретным применением измерительных средств на реальных системах. Одна из проблем – это необходимость удовлетворения требованиям, чтобы измерители значений компонент вектора G не искажали их и по возможности не влияли на те физические явления и процессы, которые являются предметом моделирования. Дать общие рекомендации по этому вопросу вряд ли возможно. В каждом конкретном случае приходится вновь решать задачу измерения параметров в ходе натурных экспериментов на объекте исследования.

Верификация имитационной модели состоит в доказательстве утверждений соответствия алгоритма ее функционирования замыслу моделирования путем формальных и неформальных исследований реализованной программы модели.

Неформальные исследования программы модели имеют целью убедить исследователя и заказчика ИМ в соответствии построенной модели замыслу, заложенному конструктором при ее разработке. Они представляют ряд процедур проверки правильности алгоритма функционирования модели объекта исследования. Проведение этих процедур входит в состав комплексной отладки ИМ. Вся последовательность действий комплексной отладки направлена на достижение у разработчика уверенности в том, что поведение реализованной программы модели соответствует замыслу и своему назначению. Важно при этом тщательно отладить и проверить поведение программы той части модели, которая прогнозирует работу системы, т. е. имитирует режимы работы, в которых исследователь не мог наблюдать поведение реальной системы.

Формальные методы верификации программы модели включают в себя: использование специальных препроцессоров и постпроцессоров в качестве «читателей» программ; замену некоторых стохастических элементов модели детерминированными и проверку на «ожидаемость» результатов моделирования; использование теста на «непрерывность» моделирования. Препроцессоры и постпроцессоры анализируют текст

программы модели путем доказательства логических утверждений и выдают исследователю сообщения о возможности наличия логических ошибок в тексте модели. Если при замене стохастических элементов модели детерминированными исследователь получает очень далекие от ожидаемых ответы программы, то это означает, что программная реализация модели неудачна. В ходе тестирования на «непрерывность» моделирования проверяется соответствие выходных характеристик воздействиям на входе модели по всему диапазону значений параметров модели X .

В любой вероятностной ИМ сложной системы применяются генераторы псевдослучайных чисел. Как правило, система автоматизации моделирования предоставляет в распоряжение исследователя несколько различных генераторов псевдослучайных чисел. Каждый из них использует базовый генератор, числа с выхода которого затем преобразуются для получения псевдослучайных величин с заданными законами распределения. Оказывается, что такой генератор является источником погрешности имитации. В силу особенностей алгоритма работы генераторов псевдослучайных чисел результаты нескольких прогонов ИМ при одних и тех же значениях параметров X и переменных G будут отличаться друг от друга.

Некоторые исследователи по неопытности применяют единый генератор, обращаясь к нему из различных мест ИМ в ходе имитации реальных функциональных действий. В этом случае процесс моделирования может быстро вырождаться из-за выхода псевдослучайной последовательности за пределы отрезка апериодичности [22]. Зачастую исследователи не проверяют базовые генераторы не только на «длину отрезка апериодичности», но даже на «истинную равномерность».

Для проверки качества генераторов псевдослучайных чисел имеется множество тестов. При начальном знакомстве с методами тестирования можно рекомендовать работу [31]. Существует проблема компромисса между точностью воспроизведения псевдослучайных величин согласно выбранному закону их распределения и сложностью реализации таких генераторов. Еще А. А. Колмогоров выявил тесную связь между «степенью случайности» последовательности и сложностью ее представления. Поэтому широко распространенные датчики, использующие простые алгоритмы имитации псевдослучайных чисел, в принципе не могут быть хорошего качества. Но применение датчиков со сложными алгоритмами имитации зачастую не представляется возможным из-за ограничений ресурса времени ЭВМ на моделирование вариантов сложной системы. Что же делать исследователю в подобных ситуациях? В ряде случаев после определения

погрешности, связанной с использованием генераторов псевдослучайных чисел, может оказаться, что точность имитации приемлема и проблема отсутствует. Чем и как оценить эту погрешность? В средней точке области изменения параметров модели X организуется несколько (автор рекомендует $N = 30$) прогонов модели с одними и теми же значениями X и G , но с разными начальными значениями для базовых генераторов псевдослучайных чисел ξ_{ok} [18]. Для каждого k -го прогона модели вычисляются значения n -й компоненты вектора отклика Y_{nk} . В результате получают выборки значений отклика $\{Y_{nk}\}$, $k = \overline{1, 30}$. По выборкам определяют оценки математического ожидания и дисперсии.

После того как погрешность оценена и она оказалась достаточно высокой, исследователь должен *пересмотреть состав генераторов псевдослучайных чисел* и процедуру использования самих генераторов в ходе имитации. В таких случаях можно рекомендовать выбрать большее число базовых генераторов псевдослучайных чисел с обязательным хранением своего очередного начального значения ξ_{0ij} для каждой ij -й активности модели. Для увеличения периода аппериодичности генераторов псевдослучайных чисел рекомендуются комбинированные алгоритмы [18].

Одно из основных отличий имитационных экспериментов от натуральных испытаний объекта моделирования заключается в простоте запуска, прерывания и возобновления эксперимента. Это позволяет экспериментатору полностью контролировать модельный эксперимент. Однако необходимо заботиться о том, как задать начальные условия и когда приступить к сбору данных. Обычно имитационные модели применяются для изучения системы в типичных для нее и повторяющихся изо дня в день условиях. К сожалению, в большинстве стохастических моделей требуется некоторое время T_0 для достижения моделью необходимого установившегося состояния. Поэтому исследователь должен позаботиться об уменьшении влияния начального периода моделирования или его исключении из результатов моделирования. Существуют *три способа уменьшения влияния начального периода на динамику моделирования* сложной системы [15]: использование длинных прогонов модели; исключение из рассмотрения начального периода прогона; выбор такого начального условия, которое ближе всего к типичному, и тем самым существенное уменьшение длительности переходного режима. Первый подход можно применять только в случае, если прогон модели не требует много машинного времени. При втором подходе часть машинного времени тратится бесполезно и из-за сокращения объема выборки увеличивается

дисперсия отклика модели Y . При этом весьма трудно установить, когда закончится переходный режим. Не существует полностью надежных методов для решения вопроса о том, достигнуто ли установившееся состояние или нет. Установившееся состояние вовсе не означает, что переменная отклика Y достигла некоторого постоянного уровня. Переменная отклика Y продолжает флуктуировать присущим ей образом. Под статистическим равновесием или установившимся состоянием мы понимаем такое состояние, в котором противодействующие влияния сбалансированы и компенсируют друг друга. Предполагается, что для каждой стохастической модели существует распределение вероятностей значений отклика, являющееся характеристикой системы. Модель находится в равновесии, когда отклик не выходит за предельные значения.

Для *отделения переходного режима* от стационарного у исследователя должна быть возможность наблюдения за моментом входа контролируемого параметра моделирования в стационарный режим. Это можно осуществить, например, путем наблюдения за длиной очереди заказов к наиболее загруженной компоненте модели. Такой контроль может быть выполнен на любой компоненте модели, если исследователь предполагает, что она позже всех входит в стационарный режим. Необходимо задать временной интервал, по окончании которого происходит вычисление оценки средней длины очереди. Проверку можно выполнить по критерию Вилкоксона [2], который требует задания уровня значимости.

Необходимо обратить внимание на трудность выбора контролируемого параметра, поскольку эта процедура имеет неформальный характер. В ряде случаев исследователи строят графики изменения значений контролируемого параметра в модельном времени t_0 и по виду этого графика устанавливают, как отделить длину переходного режима. Сбор статистики моделирования начинается именно с этого значения модельного времени. Некоторые средства автоматизации моделирования [11] позволяют для определения момента входа модели в стационарный режим использовать специальные процессы-контролеры. Эти процессы функционируют по различным алгоритмам, описываемым самим пользователем и реализуемым в виде специальных процедур.

Не всегда целью моделирования является исследование поведения системы в стационарном режиме. Бывают случаи, когда предметом имитации является переходный режим, имеющий место в реальной системе. Но и в этих случаях важно уметь уловить момент окончания переходного режима в модель, чтобы завершить имитацию,

не затратив лишних ресурсов ЭВМ. Исследователь может оказаться и в такой ситуации, когда за максимально возможное в реальных условиях время моделирования варианта системы не удастся достичь стационарного режима. Вместе с тем он знает, что все результаты моделирования могут иметь смысл только при достижении стационарного режима в системе. Несмотря на то, что в такой ситуации исследователь мало доверяет результатам имитации, он все же вынужден ими пользоваться, поскольку зачастую другого средства для изучения явления у него нет. Чтобы как-то повысить доверие к такой модели, ему приходится строить последовательность вложенных моделей, отличающихся друг от друга уровнем детализации, но позволяющих достигать стационарного состояния. Более подробно об использовании для подобных целей последовательностей вложенных имитационных моделей можно найти в [15].

Под *устойчивостью* результатов имитации будем понимать степень нечувствительности ее к изменению входных условий. Универсальной процедуры подобной проверки не существует. Поэтому создатели моделей вынуждены прибегать к методам «для данного случая», частичным тестам и к здравому смыслу. Часто бывает полезна апостериорная проверка. Она состоит в сравнении предсказаний модели и результатов измерений после того, как будет произведено какое-либо изменение в установке, предусмотренное в исследовании. Если модель окажется приемлемо точной, уверенность пользователей в ее устойчивости оправданно возрастет.

В общем случае можно следовать принципу: чем ближе структура модели к структуре системы и чем выше степень ее детальности, тем обширнее область пригодности модели. Однако иногда в процессе структурной калибровки модели достигается точка, за которой устойчивость результатов моделирования может ухудшаться из-за чрезмерной подстройки модели к частным входным условиям, использованным для ее калибровки. К сожалению, крайне трудно определить момент, когда эта точка достигнута. Практика моделирования вычислительных систем показала, что устойчивость результатов моделирования можно оценивать дисперсией значений отклика (по выбранной компоненте). Если эта дисперсия при увеличении времени моделирования T_n не увеличивается, значит, результаты моделирования устойчивы.

Методика оценки дисперсии может быть следующей. В модельном времени t_0 задаются шаг Δt контролирования показателя качества Y и число шагов n для контроля, а также экспертное значение изменения контролируемого параметра ΔY_0 . По достижении стационарного

состояния оценивается амплитуда изменений парастационарного состояния параметра Y и вычисляется модуль амплитуды как функции от n . Каждый раз при этом проверяется выполнимость неравенства $\Delta Y \leq \Delta Y_0$. Если на всем интервале исследования Y окажется в заданных пределах, то считается, что модель находится в устойчивом состоянии. Рост разброса контролируемого параметра от начального значения при изменении n указывает на неустойчивый характер имитации исследуемого процесса.

Исследователь должен установить диапазон изменения отклика модели Y при изменении каждой компоненты вектора параметров X . В зависимости от диапазона изменения откликов Y определяется стратегия планирования экспериментов на ИМ. Если при значительной амплитуде изменений некоторой компоненты вектора параметров модели X отклик Y 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 модели Y окажется высокочувствительным к изменению некоторой компоненты вектора X , то это служит прямым указанием на необходимость представления ее в модели с максимально возможной точностью.

Вторым, не менее важным моментом *исследования чувствительности* модели является проверка зависимости отклика модели Y от изменений параметров внешней среды G . Меняя характеристики G в обе стороны на некоторую величину ΔG , исследователь оценивает диапазон изменений вектора отклика модели δY . Если δY незначителен, то требования к точности задания модели внешней среды могут быть несущественными. В противном случае измерение характеристик G^* , по которым строится модель внешней среды G , и сами способы стабилизации значения G в модели должны быть высокоточными. Анализ поведения приращения отклика модели δY при колебаниях X и G иногда позволяет скорректировать алгоритмы ИМ в сторону их упрощения.

Обычно пространство значений параметра X задано и определяется целями моделирования и степенью осведомленности исследователя о компонентах объекта моделирования. Определение чувствительности ИМ параметров легче всего проводить в центральной точке пространства значений параметра. Выбор центральной точки осуществляется, как правило, на основании априорных суждений [1, 30] и носит неформальный характер. Для вычисления отклика применяются процедуры, описываемые самим пользователем на языке реализации модели, результаты работы которых затем уже используются стандартной процедурой оценки чувствительности. Исходя из тех же

априорных сведений пользователь должен задать интервал изменения каждой компоненты вектора параметров X модели системы.

Итак, мы подошли к оценке пригодности ИМ для нашего случая, которая называется проверкой адекватности ИМ. Оценка адекватности модели объекту исследования проводится чаще всего для случая, когда можно определить значение отклика системы в ходе натурных испытаний. Пусть известен отклик реальной системы Y_Q^* при нагрузке G^* , параметрах Q^* и неизвестной из-за сложности и малой изученности протекающих в объекте процессов функции $Y_Q^* = \varphi^*(Q^*, G^*)$. Модель представляет собой аппроксимирующую зависимость $Y_k = \varphi(Q_k, G)$, $k = \overline{1, N}$ (k и N – соответственно номер и число опытов на модели и реальной системе), найденную в ходе эксплуатации объекта по результатам наблюдений входных воздействий Q_k^* , $k = \overline{1, N}$, и выхода Y_{Qk}^* при заданных значениях G^* .

Проверку адекватности модели реальному объекту исследования можно проводить различными способами. Мы остановимся на трех наиболее употребляемых способах: по средним значениям откликов модели от среднего значения откликов системы; по максимальному значению абсолютных отклонений откликов модели от откликов системы.

При первом способе проверяется гипотеза о близости средних значений каждой n -й компоненты откликов модели $\bar{Y}_n$ известным средним значениям n -й компоненты откликов реальной системы $\bar{Y}_{Qn}^*$. Проводят N_1 опытов на реальной системе и измеряют по каждой n -й компоненте откликов системы выборки значений $\{Y_{Qnk}^*\}$, $k = \overline{1, N_1}$. Выполняют N_2 опытов на модели системы и получают по тем же n -м компонентам откликов модели выборки значений $\{Y_{nk}\}$, $k = \overline{1, N_2}$. Обычно стараются, чтобы объемы выборок были одинаковы ($N_1 = N_2$), но в ряде случаев натурные эксперименты весьма дороги и поэтому $N_2 > N_1$. По выборкам вычисляются оценки математического ожидания и дисперсии откликов модели и системы [2].

При втором способе для каждой n -й компоненты откликов проверяется гипотеза о значимости различий – оценка двух дисперсий D_n^* и D_{on} ; F -критерию ($F = D_{on}/D_n^*$). Отметим, что поскольку F всегда должна быть больше единицы, то в числителе берется большая из оценок дисперсий. Чаще всего выполняется неравенство $D_{on} \geq D_n^*$. Обычно задаются уровнем значимости $\alpha = 0,05$ и при конкретных значениях степеней свободы $\gamma_1 = \gamma_2 = N_2$ (при данном способе проверки

адекватности стремятся к $N_2 = N_1$) по таблицам для F-распределения Фишера [2] находят критическое значение F-статистики ($F_{кр}$). Выполнение неравенства $F > F_{кр}$ указывает на то, что гипотеза о значимости различий двух оценок дисперсий принимается и отсутствует адекватность между n -ми компонентами откликов модели и реальной системы. Если отсутствует адекватность модели хотя бы по одной из компонент, то считают, что модель неадекватно отображает реальную систему.

При третьем способе проверяется соответствие отклонений откликов модели и реальной системы по каждой компоненте. Отклонения этих откликов должны быть не более заданной величины. Этот способ применяется при проверке адекватности моделей систем управления. Аналогично первому способу для каждой n -й компоненты откликов формируют две выборки: $\{Y_{nh}\}$ и $\{Y_{Qnk}^*\}$, $k = \overline{1, N_1 = N_2}$. По выборкам определяют вектор отклонений откликов модели от откликов реальной системы в процентах. Компонентами этого вектора являются

$$\delta Y_n = \max_k \frac{|Y_{nk} - Y_{Qnk}^*|}{\bar{Y}_{Qn}^*} 100 \%, \quad (7.1)$$

где $\bar{Y}_{Qn}^*$ – среднее значение n -й компоненты отклика системы.

Таким образом, δY_n представляет собой максимальное значение отклонений n -й компоненты откликов модели в N_2 модельных экспериментах от соответствующих откликов реальной системы, нормированное средним значением n -й компоненты отклика реальной системы, процентным значением допустимого отклонения каждой компоненты откликов модели от откликов системы δY_d , проверяют выполнимость неравенства $\delta Y_n \leq \delta Y_d$. Невыполнение неравенства (7.1) хотя бы по одной компоненте вектора отклика модели ставит под сомнение адекватность модели реальной системе.

После того как завершена верификация ИМ, проведена оценка технологических характеристик ИМ и достигнута адекватность ИМ реальному объему или если для проектируемой системы исследователь получает некоторую уверенность в том, что модель соответствует своему назначению. Только в этом случае этап испытания ИМ можно считать завершенным. Наиболее существенными, с нашей точки зрения, являются процедуры: верификация ИМ; оценка технологических характеристик ИМ (точности имитации, длины переходного периода, устойчивости имитации); оценка чувствительности откликов ИМ в вариациях параметров модели. Перейдем к более подробному изложению этих процедур.

7.2 Верификация имитационной модели

Одним из первых способов верификации является использование «заглушек» в алгоритме ИМ СС. Последовательно отдельными «заглушками» меняют активности процессов или агрегатов, а затем постепенно их заменяют, убеждаясь в том, что алгоритмы процессов или других компонентов функционируют должным образом. Этот известный прием в программировании эффективен на начальных этапах разработки ИМ сложных систем. Способ не требует большого количества операторов. Когда разрабатываются крупные ИМ, используется способ *структурного разбора* ИМ несколькими специалистами по программированию. Этот способ широко распространен при программировании больших и сложных программ. Третьим способом, применяемым для отладки дискретно-событийных ИМ, является *трассировка*, когда все переменные состояния ИМ выдаются на дисплей компьютера. При этом обращают основное внимание на способность программы ИМ обрабатывать «предельные условия». В современных СМ для этой цели используются интерактивные отладчики, которые позволяют исследователю остановить имитацию в выбранные моменты времени и изучить состояния ИМ.

Для многих ИМ СС существует строгая зависимость между активностями, требующая согласования активностей и по информации, и по порядку их следования друг за другом. В работе [27] предложен способ *верификации ИМ с помощью контроля за выполнением графа активностей*. Результаты l -й реализации ИМ записываются в файле статистики. В результате формируется стандартная трасса ИЭ. В каждой записи этой трассы статистики содержится: значение модельного времени t_0 на момент свершения событий, номер активности и сопутствующая информация. Далее используется табличный процессор Excel в режиме «Автофильтра» для каждого номера активностей. Анализируя эту информацию, исследователь убеждается в том, что все активности графа задействованы в соответствующих ситуациях; а отслеживаемые потребителем ресурсы используются в правильном порядке их следования и в планируемые моменты модельного времени t_0 . Общая последовательность действий при верификации ИМ способом контроля графа активностей имеет следующий вид:

- задание графа переходов между активностями для ИМ СС;
- добавление в программу ИМ СС специализированных функций трассировки интересующих событий или использование стандартной схемы трасс событий;

- проведение серии ИЭ с целью выхода ИМ на различные режимы (в том числе для выявления редких событий) и проверка правильности соответствующих переходов для этих событий;

- использование утилиты контроля графа активности и анализ результатов этого контроля для выявления ошибок реализации в программе ИМ и их устранение;

- доказательство правильности работы алгоритма имитации, дополнительный анализ полученных трасс и при необходимости возврат на предыдущий этап построения ИМ СС.

Одним из распространенных способов верификации ИМ СС является контроль изменения откликов в модельном времени t_0 . Идея способа заключается в том, что по всем типам запросов ресурсов в ИМ должны отображаться на изменении откликов ИМ. Обычно проводят три ИЭ, в которых запросы в систему поступают с малой интенсивностью ($\lambda_{\min}$), сбалансированной интенсивностью (λ_0) и высокой активностью ($\lambda_{\max}$). Сравниваются диаграммы использования ресурсов ИМ. По этим диаграммам, а также путем анализа интегральных откликов ИМ определяются такие свойства ИМ, как: периодичность, неубывание, невозрастание, разрывность, анализ ярусности, анализ обнулений и спадов откликов, разрежение и скопление точек графиков откликов. В программу при необходимости добавляются специальные функции фиксации интересующего отклика в момент изменения его величины. Вспомогательные переменные, относящиеся к событию, и изменения значения отклика фиксируются в трассе имитации. Если в ИМ СС используются ресурсы системы по расписанию, то необходимо убедиться, что данное расписание находит отражение в графиках изменения откликов в модельном времени t_0 . Например, в диаграмме изменения времени обслуживания потребителей ресурса по расписанию, обязательно будут видны «разрывы» расписания. Или, например, при наличии в системе группового обслуживания элементами модели на диаграммах, отображающих ожидания в очередях, должны быть отражены факты резкого увеличения количества элементов в очередях к устройствам.

В подобных случаях рекомендуется следующая последовательность действий исследователя при верификации модели:

- в программу ИМ добавляются операторы для создания трассы исследователя (реализуется опыт, включающий до 30 имитационных экспериментов с данной моделью);

- результаты опыта после усреднения согласно процедуре Монте-Карло находятся в файле с усредненной трассой;

- строятся графики с помощью доступных исследователю статистических пакетов обработки данных, и уже по результатам анализа

графиков и диаграмм необходимо убедиться в том, что ожидаемая картина находит свое отображение.

При выявлении несоответствий ведется поиск ошибок реализации подобных ситуаций в программные ИМ СС.

Еще одним способом верификации является проверка эвристических гипотез. Алгоритм верификации ИМ СС этим способом реализуется в 3 шага:

- выдвигается гипотеза о взаимосвязи параметров и откликов ИМ;
- проводится ИЭ, позволяющий получить статистику имитации, необходимый для проверки гипотезы;
- если гипотеза отвергается, то анализируются причины несоответствия и при необходимости продолжается отладка ИМ или осуществляется возврат на предыдущий этап создания ИМ;
- возможно выдвижение новой эвристической гипотезы и выполнение предыдущих шагов анализа программы ИМ СС.

Рассмотренные выше способы верификации программ ИМ СС для детерминированных ситуаций в СС назовем способами *статической верификации* ИМ. Они позволяют убедиться в том, что программа ИМ отображает замысел исследователя в тех случаях, когда основная часть параметров ИМ является детерминированными величинами. Но как только ИМ становится вероятностной, ее необходимо использовать только в сочетании с методом статистических испытаний. Далее уже не ясно, насколько точно отображает программа ИМ замысел исследователя, поскольку усреднение статистик и откликов может ввести в заблуждение исследователя. Поэтому ниже предлагается способ *динамической верификации* программы вероятностных ИМ сложных систем. При верификации вероятностных ИМ СС необходимо реализовать две фазы проверки программами ИМ СС:

- правильности алгоритмов имитации при конкретных значениях запросов процессами или агрегатами ресурсов системы (проектирование «единичной нити»);
- правильности алгоритмов имитации компонентами ИМ при общей нагрузке (проектирование вероятностной нагрузки).

Реализацию способа рассмотрим на примере верификации ИМ сложной системы, построенной на агрегатно-процессном способе имитации (пример № 6). Предположим на время испытания, что все распределения моделируемых вероятностных величин подчиняются нормальному закону с параметрами $N(M, S)$. Реализацию первой фазы верификации осуществляем согласно следующей методике.

1. Выберем типовой пример ВСГР, в котором отражены все особенности взаимодействия $АТОР_{ij}$ с $ASOB_i$ и $ASOB_j$.

2. Определим 5 вариантов конкретных значений, полученных с помощью нормальных распределений $N(\bar{X}, S)$, где $\bar{X}$ – среднее значение, S – среднее квадратичное отклонение значения запроса ресурса. Пусть в варианте 1 с вероятностью $P_1 = 0,05$ все значения вероятностных параметров ИМ равны $x = \bar{x} - s/2$; в варианте 2 все значения запросов ресурсов с вероятностью $P_2 = 0,2$ равны $x = \bar{x} - s/4$; в варианте 3 с вероятностью $P_3 = 0,5$ значения всех запросов ресурсов равны $x = \bar{x}$; в вариантах 4 и 5 соответственно с вероятностями $P_4 = 0,2$ и $P_5 = 0,05$ установим $x = \bar{x} + s/4$ и $x = \bar{x} + s/2$. Таким образом перекрывается основной диапазон возможных значений запросов ресурсов ВТПП при выполнении множества $\{MTXO_{ij}\}$.

3. Зададим в тестовом примере 6 конкретные значения запросов АТОР_{ij} ресурсов системы, как указано в пункте 2 методики: $F_{ijh}(\tau)$ – времена выполнения АТОР_{ij} в h -м варианте верификации ($h = \overline{1,5}$ – это уже детерминированные величины), а также векторы значений:

$\{C_{ijh}\}$ – стоимость выполнения АТОР_i в h -ом варианте верификации;

$\{KO_{r1ijh}\}$ – запрос АТОР_{ij} количества комплектующих деталей r_1 -го типа;

$\{Mt_{r2ijh}\}$ – запросы АТОР_{ij} количества материалов r_2 -го типа.

Остальные запросы ресурсов в тестовом примере ВСГР задаются одинаковыми для каждого варианта имитации (в виде векторов количества):

$\{n_{6ij}\}$ – бригад исполнителей АТОР_{ij};

$\{n_{5ij}\}$ – индивидуальных исполнителей АТОР_{ij};

$\{n_{4ij}\}$ – ресурсов общего пользования;

$\{n_{3ij}\}$ – ресурсов индивидуального использования;

$\{n_{2ij}\}$ – устройств оборудования общего пользования;

$\{n_{1ij}\}$ – устройств оборудования индивидуального использования.

Запросы места на ресурсах общего пользования и оборудовании общего пользования сделаем равными их средним значениям $\{\bar{V}_{1ij}\}$ и $\{\bar{V}_{2ij}\}$ (хотя в ИМ они могут иметь другие значения, формируемые по соответствующим функциям распределения).

Далее используем известный алгоритм [12] расчета ранних и поздних сроков свершения событий (t_{pj} и t_{nj}), а также резерва их свершения ($R_j = t_{nj} - t_{Rj}$). В итоге для каждого варианта ($h = \overline{1,5}$) формируется полигональная форма $POLGR_h$ вероятностного сетевого графика реализации ВСГР.

4. Определяем интегральные отклики h -го варианта ИМ ВСГР:
 - критический путь KRP_h для h -го варианта ВСГР (состоящий из последовательности агрегатов $АТОР_{ij}$, переменяющихся агрегатами $АТОР_{ij}$);
 - критическое время реализации $ВСГР_h (T_{kph})$;
 - суммарное количество использования ресурсов детерминированными запросами $АТОР_{ij}$:

$(n_{60h}, n_{50h}, n_{40h}, n_{30h}, n_{20h}, n_{10h})$

5. Формируем графики изменения во времени расхода:

- финансов на реализацию $ВСГР_h C_{oh} (\bar{t}_{pj})$;
- комплектующих изделий r_1 -го типа $КО_{4oh} (\bar{t}_{pj})$;
- материалов r_2 -го типа $mt_{r2oh} (\bar{t}_{pj})$.

6. Формируются диаграммы изменения остатка места на общем ресурсе и общем оборудовании системы:

$DIAGR (V_{OST4} (\bar{t}_{pjh}) \text{ и } V_{OST2} (\bar{t}_{pjh}))$.

7. Формируются диаграммы использования $\{АТОР_{ij}\}$ за интервал времени $(0, T_{kph})$ в моменты ранних сроков свершения событий (V_{pjh}) детерминированных ресурсов: бригад ($DIAGBR_r$) исполнителей ($DIAGRISP_r$), ресурсов R -го типа ($DIAGRRES_R$), устройств общего и индивидуального использования ($DIAGR AOBIN_{r1}$ и $DIAGR AOBOP_{n2}$).

8. В ходе отладки алгоритмов ИМ ВТПП h -го варианта необходимо добиться совпадения данных имитации с рассчитанными «вручную» значениями для всех перечисленных статистик имитации. Только после совпадения (с точностью до ошибок имитации) расчетных данных и результатов ИЭ можно говорить, что фаза 1 процедуры *динамической верификации* оказалась успешной и можно переходить ко второй фазе.

На второй фазе верификации ИМ ВСГР будем придерживаться следующей методики проверки результатов ИЭ.

1. В тексте программы агрегатов $АТОР_{ij}$, $АОВОР_{r1}$, $АОВОР_{r2}$ и $АКАН_{r2}$ включаются все генераторы формирования жребиев 3-го типа [17] по функциям распределения $F_{ij}(\tau)$. Причем в тестовом варианте ИМ ВСГР все функции распределения величины запросов ресурсов являются нормально распределенными величинами с параметрами $N(\bar{X}, S)$. В ходе 2-ой фазы верификации (проектирование вероятностной нагрузки) проводится серия ИЭ согласно процедуре Монте-Карло.

2. Предполагаем, что каждый h -й вариант $(h = \overline{1,5})$ выполнен соответственно (5, 20, 50, 20, 5) раз. Исходя из этого соображения,

рассчитываются ожидаемые значения откликов ИМ ВСГР (среднее значение и дисперсия) для $R = \overline{1,5}$:

$$\{T_{kph}^*, C_{oh}^*, KO_{oh}^*, mt_{oh}^*\} \quad (7.2)$$

$$\begin{aligned} T_{kp}^* &= T_{kp1}^* \cdot 0,05 + T_{kp2}^* \cdot 0,2 + T_{kp3}^* \cdot 0,5 + T_{kp4}^* \cdot 0,2 + T_{kp5}^* \cdot 0,05; \\ C_0^* &= C_{01}^* \cdot 0,05 + C_{02}^* \cdot 0,2 + C_{03}^* \cdot 0,5 + C_{04}^* \cdot 0,2 + C_{05}^* \cdot 0,05; \\ KO_0^* &= KO_{01}^* \cdot 0,05 + KO_{02}^* \cdot 0,2 + KO_{03}^* \cdot 0,5 + KO_{04}^* \cdot 0,2 + KO_{05}^* \cdot 0,0; \\ mt_0^* &= mt_{01}^* \cdot 0,05 + mt_{02}^* \cdot 0,2 + mt_{03}^* \cdot 0,5 + mt_{04}^* \cdot 0,2 + mt_{05}^* \cdot 0,05. \end{aligned} \quad (7.3)$$

Значения количества использований остальных ресурсов устанавливаются одинаковыми для всех вариантов: $\{n_{60}^*, n_{50}^*, n_{40}^*, n_{30}^*, n_{20}^*, n_{10}^*\}$.

3. Проводится 100 раз ИЭ с программой тестируемой ИМ ВСГР и находим оценки откликов: (средние значения и дисперсии):

$$\{(T_{kp}, D_{Tkp}), (C_0, D_{C_0}), (KO_0, D_{KO_0}), (mt_0, D_{mt_0})\}, \quad (7.4)$$

$$(n_{60}, n_{50}, n_{40}, n_{30}, n_{20}, n_{10}). \quad (7.5)$$

4. Сравниваем интегральные отклики, полученные в результате усреднения откликов реализаций ИМ ВСГР, с рассчитанными откликами по формулам (7.3). Определяем t -статистики различий откликов, вычисленных заранее по данным исследований единичной нити, с результатами, полученными усреднением результатов имитации по методу статистических испытаний, представленными в виде зависимостей:

$$\begin{aligned} t_{01} &= |T_{kp} - T_{kp}^*| \sqrt{\frac{50}{D_{Tkp}}}; \\ t_{02} &= |C_0 - C_0^*| \sqrt{\frac{50}{D_{C_0}}}; \\ t_{03} &= |KO_0 - KO_0^*| \sqrt{\frac{50}{D_{KO_0}}}; \\ t_{04} &= |mt_0 - mt_0^*| \sqrt{\frac{50}{D_{mt_0}}}. \end{aligned} \quad (7.6)$$

По таблице распределений Стьюдента для уровня значимости $\alpha = 0,05$ и чисел степеней свободы $\nu = 100 - 2 = 98$ для каждого типа зависимостей (7.6) находим допустимое значение различия интегрального отклика: $t_{kp1}; t_{kp2}; t_{kp3}; t_{kp4}$.

5. Сравниваем покомпонентно различие векторов интегральных откликов расчетных и полученных с помощью программы ИМ ВСГР. Устанавливаем выполнимость неравенств:

$$t_{kps} \geq t_{os}; \quad s = \overline{1, 4}. \quad (7.7)$$

Только в случае, когда все четыре неравенства (7.7) выполняются, можно считать, что вторая фаза верификации ИМ ВСГР прошла успешно. Невыполнение какого-либо из неравенств (7.7) дает указание, что при имитации соответствующего ресурса в ИМ ВСГР имеются проблемы и исследователь обязан изменить ИМ ВСГР таким образом, чтобы устранить это несоответствие. Для остальных откликов (7.5) необходимо полное соответствие расчетных данных и результатов имитации варианта ИМ ВСГР. Несоответствие какого-либо интегрального отклика вида (7.5) дает указание на то, что в программе имеются ошибки отображения расхода соответствующего типа ресурсов ВСГР.

7.3 Определение технологических характеристик имитационных моделей СС

При определении технологических характеристик ИМ СС необходимо определить: точность имитации (ε_n), длину переходного периода имитации ($T_{пп}$), устойчивость имитации (UST). С этих процедур начинается любое испытание программы ИМ СС. Рассмотрим технологию реализации этих трех процедур.

Оценка точности имитации. В любой вероятностной ИМ СС применяются генераторы моделирования псевдослучайных чисел, равномерно распределенных на интервале $[0, 1]$, называемых *базовыми* генераторами. Система моделирования предоставляет пользователю несколько различных базовых генераторов. Эти генераторы и сам вероятностный характер ИМ являются источником погрешности имитации. Для проверки качества этих генераторов в документации СМ приведены описания тестов базовых генераторов. Иногда даже трудно выбрать, какой из базовых генераторов подходит для конкретного

использования. Поэтому нами опробована на различных ИМ следующая технология оценки погрешности имитации. В серединной точке области изменения управляющих параметров $\{X\}$ организуется серия прогонов ИМ ($k \leq 30$) с одними и теми же значениями параметров и переменных (X и G), но с различными начальными значениями базовых генераторов (ξ_{okm} , k – номер прогона, m – номера базового генератора).

Определяются значения n -й компоненты вектора откликов Y_{nk} в каждом прогоне. В итоге формируются выборки откликов $\{Y_{nk}\}$. По этим выборкам определяем оценки их математического ожидания и дисперсии $(\bar{Y}_n, D_n)$.

Находим **доверительный интервал** нахождения истинного значения математического ожидания n -й компоненты (Y_{hn}). При этом допускаем наличие нормальности распределения отклонения $\bar{Y}_{hk}$ от $\bar{Y}_{hn}^*$. Поскольку объемы выборок малы ($N \leq 30$), то для нахождения доверительных интервалов используем t -статистику:

$$t = \left(\bar{Y}_h - \bar{Y}_{hn}^* \right) \sqrt{\frac{N-1}{D_n}}, \quad (7.8)$$

которая имеет распределение Стьюдента [2]. Задавшем уравнением значимости (например $\alpha = 0,05$) с вероятностью 0,95, можно утверждать, что Y_{hn}^* лежит в пределах:

$$\bar{Y}_n - t_{0,05} \sqrt{\frac{D_h}{N-1}} \leq Y_{hn} < \bar{Y}_n + t_{0,05} \sqrt{\frac{D_h}{N-2}}, \quad (7.9)$$

где $t_{0,05}$ – значение t -статистики при $(N-1)$ степенях свободы и уровне значимости $\alpha = 0,05$.

Таким образом, для каждой компоненты можно определить погрешность:

$$\alpha_n = t_{0,05} \sqrt{\frac{D_n}{N-2}}. \quad (7.10)$$

Отметим, что минимальное число ИЭ должно быть равно 30. В этом случае с достоверностью $\beta = 0,95$ погрешность $\varepsilon = 0,753 S_n$ (где S_n – среднее квадратичное значение выборки объема $N = 30$), которая будет верхней границей погрешности, связанной с вероятностным

характером ИМ и использованием базовых генераторов псевдослучайных величин. Если эта погрешность окажется достаточно высокой, то исследователь обязан пересмотреть состав генераторов и процедуру их использования в ИМ СС. В этих случаях следует использовать большее число базовых генераторов с гарантированным качеством моделирования ξ_{okm} .

Определение длительности переходного режима имитации. Одно из основных отличий ИЭ от натурных испытаний объекта моделирования заключается в простоте повторения и воспроизведении условий эксперимента, а также в простоте запуска, прерывания и возобновления эксперимента на ИМ. Это позволяет исследователю контролировать модельный эксперимент. Однако необходимо заботиться о том, как задать начальные условия и когда приступить к сбору статистики имитации. В большинстве вероятностных ИМ требуется некоторое время ($T_{пп}$) для достижения установившегося состояния. Поэтому исследователь должен позаботиться об уменьшении влияния начального периода имитации и исключения статистики этого периода из всех результатов моделирования.

Для отделения переходного режима от стационарного у исследователя должна быть возможность наблюдения за моментом перехода контролируемого параметра моделирования в стационарный режим его изменения. Обычно стараются наблюдать за длиной очереди заказов к наиболее загруженной компоненте ИМ СС. Необходимо задать временной интервал, по окончании которого вычитается средняя длина очереди. Для этой цели используются графики изменения значений контролируемого параметра в модельном времени t_0 , и по виду этого графика устанавливается момент окончания переходного режима и начала сбора статистики имитации. По собранной статистике длин очередей строится ряд диаграмм по каждому из откликов для определения максимального из моментов наступления стационарного режима заполнения и освобождения очередей. Одновременно с этим определяются коэффициенты загрузки компонентов (η_j) ИМ, у которых имеются контролируемые очереди запросов ресурса ИМ. По графику изменения η_j во времени t_0 уточняется факт перехода компоненты в установившееся состояние обслуживания запросов ресурсов ИМ СС.

Отметим, что переходный период моделирования характерен для ИМ транзактно-процессного способа имитации. В ИМ, построенных на основе агрегатного или агрегатно-процессного способа имитации, переходный режим практически отсутствует, поскольку сигналы в очереди не собираются. В некоторых случаях в ИМ, реализующих процессный или событийный способ имитации, также существует

период «разогрева» ИМ, однако для таких ИМ трудно найти контролируемый параметр имитации.

Оценка устойчивости имитации. Под устойчивостью имитации обычно понимают степень чувствительности ИМ к изменению входных условий, или же при возникновении редких событий, приводящую модель объекта моделирования к непредусмотренным ИМ режимам имитации. Для этой цели часто используется апостериорная проверка функционирования ИМ. Она состоит в сравнении предсказаний ИМ при дополнительном увеличении периода моделирования. Чем ближе структура ИМ к описанию реальной системы, тем выше детальность ее представления. В этих случаях обширнее область пригодности ИМ. Однако из-за чрезмерной подстройки ИМ к частным ситуациям снижается устойчивость результатов имитации. Обычно устойчивость имитации оценивают дисперсией j -го отклика ИМ (D_j). Возможно, при увеличении времени имитации ($T_{\text{ИМ}}$) на порядок начинают влиять на отклики редкие события, и дисперсии откликов резко увеличиваются. В таких ситуациях считают, что ИМ становится неустойчивой в моделировании реальных объектов. В случае срабатывания жребиев, имеющих малую вероятность выполнения алгоритма ИМ, существующие аппроксимации процессов неверны или функции распределения не обеспечивают требуемой достоверности воспроизведения вероятностной компонентой ИМ СС.

Когда увеличение времени имитации ($T_{\text{ИМ}}$) на порядок не дает уверенности в «устойчивости» режима имитации из-за возможных ошибок аппроксимации или из-за редких ситуаций, предлагается способ вызова возможной неустойчивости имитации. Например, существенно увеличивается вероятность появления «редких событий» и проверяется поведение ИМ. Если в этих случаях не возникает роста амплитуды откликов с шагом имитации $\Delta t_{\text{и}}$, то считают, что ИМ СС «устойчива» к появлению редких ситуаций. Если подозрение вызывает наличие ошибок аппроксимации, то уменьшают интервал времени выполнения активностей на порядок и измеряют выполнимость устойчивой имитации с меньшим интервалом изменения модельного времени t_0 откликов ИМ СС. Таким способом стремятся «спровоцировать» влияние на отклики модели способов аппроксимации в ИМ реальных процессов. Вообще при оценке «устойчивости имитации» проявляется утверждение Р. Шеннона [30], что имитационное моделирование является больше искусством, чем собственно наукой. Общих правил оценки устойчивости имитации, также как способов верификации ИМ, не существует. В каждом конкретном случае исследователю следует надеяться на свой опыт и здравый смысл.

Правило автоматической остановки имитации. До сих пор мы считали, что ресурсов на моделирование достаточно и вычислялось определенное число переходов компонентов ИМ из состояния в состояние. Особенно этот подход использовался при усреднении результатов имитации на основе метода статистических испытаний. Исходя из достоверности откликов имитации, определялось число реализаций процедуры Монте-Карло (N_m). При этом в каждой l -й реализации ИМ согласно процедуре Монте-Карло число смен состояния (m), контролируемого n -го отклика заранее устанавливалось до имитации. Однако более часто исследователю приходится решать обратную задачу, когда необходимо определить стационарное среднее отклика с точностью не меньшей, чем заданная. Из-за выборочных свойств этих функций никогда нельзя быть уверенным, что данная цель достигнута [27, 31]. Поэтому определяют требуемый объем выборки n через вероятность $(1 - \alpha)$ того, что истинное значение отклика $\bar{Y}_j$ отличается от рассчитанной оценки r не более, чем на C единиц $P(|r - \bar{Y}_j| \leq C) = 1 - \alpha$. Из выражения для абсолютной погрешности независимых нормально распределенных величин $\{Y_j\} \quad j = \overline{1, n}$ получают оценку значения n [27]. В ходе имитации СС отклик является случайной величиной, так как зависит от оценки дисперсии. Для правильного определения момента, когда можно закончить имитацию варианта ИМ, достигнув желаемой точности, рекомендуется использовать последовательный, двухвыборочный и многошаговые методы, изложенные в работе [27].

7.4 Оценка чувствительности откликов имитационной модели СС

Определение чувствительности результатов имитационного моделирования состоит в определении диапазона изменения откликов ИМ $\{Y_j\}$ в зависимости от изменения множества параметров ИМ $\{X_k\}$ на всем диапазоне их изменения. В зависимости от диапазона изменения откликов $\{Y_j\}$ определяется будущая стратегия эксплуатации ИМ. Если при незначительной амплитуде компоненты вектора параметров (иногда даже на всем диапазоне изменений от $X_{\min}$ до $X_{\max}$) отклик Y_j меняется незначительно, то это означает, что точность представления Y_j в ИМ не играет существенной роли. Даже при планировании ИЭ этот отклик не будет использоваться как основной или вообще его

можно исключить из рассмотрения при постановке серий ИЭ на ИМ. Если же отклик Y_j окажется высокочувствительным к изменению параметра X_k , то это служит прямым указанием на необходимость представления его в ИМ с максимально возможной точностью.

Процедура определения чувствительности результатов моделирования заключается в следующем. Пусть каждая q -я компонента вектора параметров $\{X_q\}$ отклоняется от своего значения в центральной точке в обе стороны на длину выбранного интервала его изменения ($\min X_q, \max X_q$). И пусть остальные компоненты вектора $\{X_q\}$ остаются неизменными и соответствуют центральной точке в пространстве параметров. При указанных значениях вектора параметров $\{X_q\}$ проводим пару модельных экспериментов и вычисляем значения j -го отклика, обозначив их Y_j^- (для $X_{q\min}$) и Y_j^+ (для $X_{q\max}$). Вычисляем коэффициент чувствительности отклика Y_j к изменениям параметра X_q по формуле:

$$\delta Y_{kj} = \frac{|Y_j^+ - Y_j^-| \cdot 2 \cdot 100 \% }{(Y_j^+ + Y_j^-)}, \quad (7.11)$$

где Y_j^+ и Y_j^- – соответственно изменение отклика Y_j при изменении параметра X_q от максимального, до минимального его изменения на всем диапазоне его изменения. Далее строим матрицу чувствительности откликов $\|\delta Y_{kj}\|$, в которой q -я строка соответствует изменениям компоненты вектора параметра X_q в динамике изменения его значений, а j -м столбцам соответствуют номера компонент вектора откликов ИМ. Расчет δY_{kj} осуществляется по формуле (7.11).

Далее содержимое матрицы $\|\delta Y_{kj}\|$ анализируется следующим образом. Для каждой компоненты этой матрицы проверяется выполнимость неравенства:

$$\delta Y_{kj} \leq \xi_u \%, \quad (7.12)$$

где $\xi_u \%$ – максимальная ошибка имитации вариантов в ИМ СС. Формируется булева матрица $\|\gamma_{kj}\|$, в которой

$$\gamma_{kj} = \begin{cases} 0, & \text{при выполнении неравенства} \\ 1, & \text{в противном случае} \end{cases} \quad (7.13)$$

Если k -я строка этой матрицы оказалась нулевой, то это означает, что в данном варианте ИМ СС эта компонента вектора параметра практически ИС влияет на все компоненты вектора откликов $\{Y_j\}$ и этот параметр в данном исследовании можно исключить. В результате минимизируется число компонент-вектора параметров ИМ СС. Если же j -й столбец матрицы $\|\gamma_{kj}\|$ равен 0, то это означает, что в данном исследовании можно не рассматривать j -ю компоненту вектора откликов ИМ. Таким способом минимизируется число откликов ИМ СС при заданном составе остальных переменных имитации $\{G\}$.

Полезно также проранжировать все отклики таким образом, что слева окажутся отклики, имеющие максимальные значения δY_{kj} . Все отклики, для которых не выполняется неравенство (7.12), назовем основными, а остальные считаем незначительными откликами (для конкретного значения компоненты вектора параметров). Аналогичным образом для Y_j -ю компоненты вектора откликов проранжируем параметры. Все параметры, для которых выполняется неравенство (7.12), аналогично назовем существенными, а остальные считаем несущественными. Очевидно, что только основные параметры должны участвовать при планировании экспериментов на последующих этапах эксплуатации ИМ СС (исследователь должен уделить внимание взаимовлиянию основных откликов с существенными параметрами).

7.5 Проверка адекватности ИМ прототипу сложной системы

В работе [27] предлагается универсальная процедура, позволяющая судить об адекватности ИМ и объекта имитации. На рисунке 7.1 приведена блок-схема процедуры с расшифровкой блоков процедуры в таблице 7.2. Согласно приведенному алгоритму необходимым условием адекватности ИМ и объекта имитации является подтверждение гипотезы о равенстве средних. Непараметрический критерий Монте-Карло используется в случае нормальности отклика. Если не подтверждается гипотеза о равенстве дисперсий, но принимается гипотеза о равенстве средних. Отметим, что окончательный вердикт выносится с помощью этого критерия при уменьшенном в 3 раза уровне точности. Основной недостаток этого критерия состоит в том, что требуется хранить обе выборки. Для реализации этого способа проверки адекватности ИМ можно использовать доступные пакеты статистической обработки данных, например пакет MS Excel. Отметим,

что в библиотеке ПТКИ реализована эта процедура и ее можно использовать при проверке адекватности модели [14].

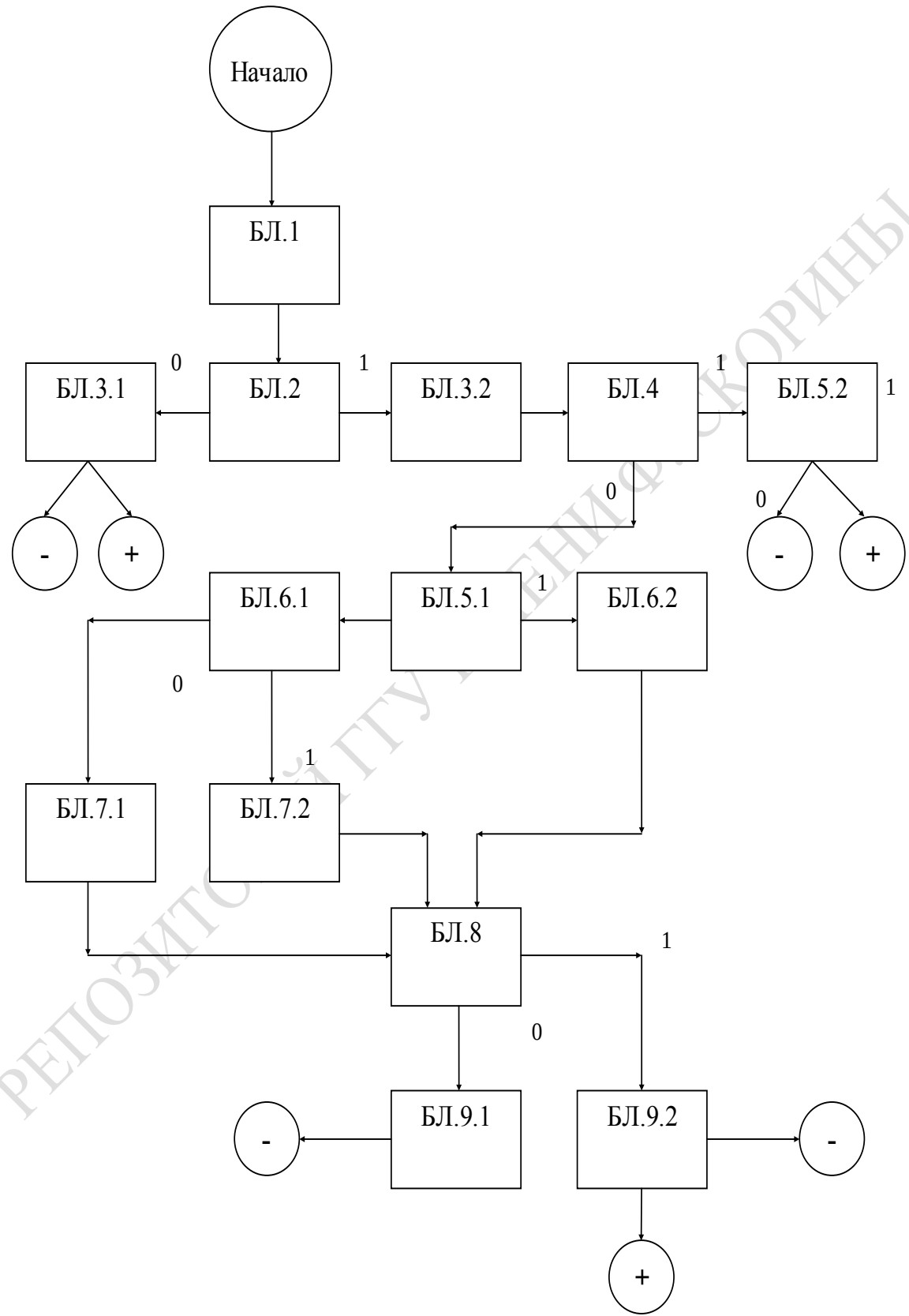


Рисунок 7.1 – Блок-схема алгоритма процедуры испытания ИМ СС

Таблица 7.1 – Содержание блоков алгоритма процедуры испытания ИМ СС

№ бло-ков	Операции, выполняемые блоками
1	Реализовать n_1 экспериментов с первым объектом n_2 экспериментом вторым, вычисляя оценки средних и дисперсий r_1, r_2 и S_1^2 и S_2^2 одного и того же отклика Y
2	По критерию Хи-квадрат или Колмогорова-Смирнова проверка гипотезы о нормальности двух совокупностей. Если гипотеза отвергается, то Бл. 3.1 иначе Бл. 3.2
3.1	Принятие решения об адекватности объектов по u -критерию о равенстве средних с уровнем значимости $\alpha/2$
3.2	Проверка гипотезы о равенстве дисперсий по F-критерию
5.1	Если $n_1 \leq 30$ и $n_2 \leq 30$, то на Бл. 6.1, иначе на Бл. 6.2
5.2	При $n_1 + n_2 < 30$ принять решение об адекватности объектов по результатам t -критерия с $(n_1 + n_2 - 2)$ степенями сводок: $t = (r_1 - r_2) \left(\sqrt{\frac{n_1 S_1^2 + n_2 S_2^2}{n_1 + n_2 - 2}} \times \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \right)$
6.1	Если $n_1 - n_2$, то Бл. 7.1 иначе Бл. 7.2
6.2	Выполнить z -критерий: $Z = (r_1 - r_2) / \left(\sqrt{\frac{S_1^2}{n_1 - 1}} + \sqrt{\frac{S_2^2}{n_2 - 1}} \right)$
7.1	Реализовать t -критерий с $n_1 - 1$ степенями свободы $t = (r_1 - r_2) / \left(\sqrt{\frac{n_1 (S_1 + S_2)^2}{2(n_1 - 1)}} \times \sqrt{\frac{1}{2n_1}} \right)$
7.2	Вычислить t -критерий из выражения $t = (r_1 - r_2) / \left(\sqrt{\frac{S_1^2}{n_1 - 1}} + \sqrt{\frac{S_2^2}{n_2 - 1}} \right);$ и сравнить его со *** значением $t_2 = \frac{w_1 t_1 + w_2 t_2}{w_1 + w_2};$ где $w_1 = \frac{S_1^3}{n_1}; w_2 = \frac{S_2^2}{n_2};$ t_1 и t_2 – критические точки t -критерия при $n_1 - 1$ и $n_2 - 1$ степенях свободы для заданного уровня значимости α
8	Если гипотеза о равенстве средних не подтвердится же, то Бл. 9.1, иначе Бл. 9.2
9.1	Гипотеза об адекватности агрегатов
9.2	Принятие решения об адекватности объектов U -критерию с уровнем значимости, равным 3α

Калибровка имитационной модели. Когда модель оказывается неадекватной реальной системе, исследователь переходит к калибровке ИМ. Цель калибровки состоит в уменьшении неточностей формулировки ИМ, обусловленных ошибочной или недостаточно подробной моделью $\varphi(X, G)$. Исследователь может корректировать либо модель внешней нагрузки G , либо алгоритм функционирования модели $\varphi(X, G)$. В ходе калибровки модели сложной системы можно проводить изменения трех типов: глобальные структурные изменения (например, добавление программ процессов, изменения событий в процессах); локальные изменения в модели (например, замена компонент модели внешне эквивалентными, но более точными); изменение некоторых параметров, используемых в качестве калибровочных.

Можно предположить, что структурные изменения как более трудно осуществимые дорогие, и должны рассматриваться в последнюю очередь, когда все попытки откалибровать модель путем изменения параметров и локальных модификаций окажутся безуспешными. На самом деле, при проведении калибровки модели существует опасность появления структурных несоответствий или недостаточной степени детальности описания реальных явлений системы в модели, которые могут вызвать сокращение области пригодности модели. Поэтому исследователи стараются «подправить» само описание модели и таким способом откалибровать модель.

При глобальной модификации модели исследователю можно рекомендовать методику, изложенную в [15]. Калибровка модели согласно этой методике состоит из трех шагов: сравнения между собой распределений откликов Y_Q^* и Y ; балансировки модели; оптимизации модели. Каждый из этих шагов является по своей природе итеративным. Он включает ряд изменений модели, за каждым из которых следует тест, который определяет, завершен этот шаг или его следует продолжить.

Шаг 1. Сравнение распределений. Определяется погрешность моделирования, которая должна быть минимизирована в процессе калибровки. В качестве такой погрешности может выступать ранее определенная величина E_{1n} (разность между средними значениями компонент векторов Y_Q^* и Y). В ходе калибровки пытаются изменить алгоритмы модели таким образом, чтобы минимизировать погрешность E_{1n} . Однако более удовлетворительное решение вопроса имеет место при использовании в качестве сравниваемых показателей функций распределения вероятностей значений компонент $F(Y_{Qn}^*)$ и $F(Y_n)$. При этом сравнение производится локально (поточечно). Для сравнения выбираются абсолютные значения разностей между

функциями распределений $|F(Y_{Qn}^*) - F(Y_n)|$ такими, чтобы избежать частичной компенсации погрешностей из-за различия знаков. Мету расхождения функций распределения вероятностей значений компонент векторов Y_{Qn}^* и Y обозначим E_{2n} . Для одних и тех же выборок Y_{Qn}^* и Y она зависит от статистического критерия сравнения выборок. Если используется критерий Пирсона, то в качестве E_{2n} выступает статистика хи-квадрат [2], если применяется критерий Колмогорова, то в качестве E_{2n} применяется D-статистика Колмогорова [14].

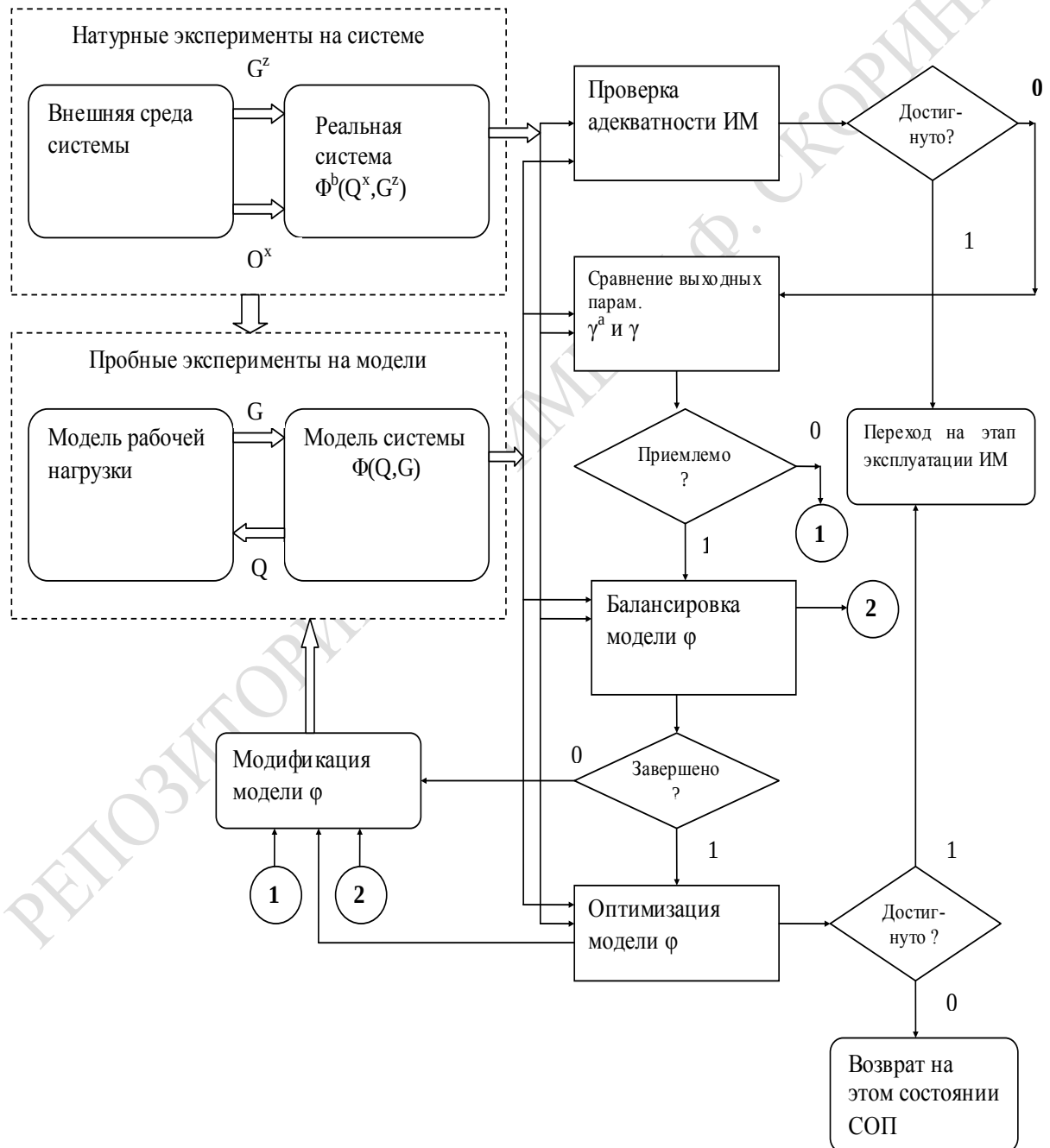


Рисунок 7.2 – Блок-схема итеративной процедуры проверки адекватности и калибровки ИМ

При использовании критерия Пирсона исследователь задается уровнем значимости α при числе степеней свободы, определяемом по числу интервалов группирования L функций распределения $F(Y_{Qn}^*)$, находится критическое значение величины $\chi^2(\alpha, L - 1)$. Если выполняется неравенство $E_{2n} \leq \chi^2(\alpha, L - 1)$, считают, что различие между функциями распределения незначимо и достигнута близость между n -ми компонентами откликов модели и системы. Проверяя значимость различия между функциями распределения по всем компонентам, исследователь определяет, улучшает ли проведенная модификация компонент модель или нет. Отметим, что данный шаг предполагает макроскопические изменения модели, например, модификацию структуры или замену главных составляющих модели.

Шаг 2. Балансировка модели. В ходе натурального эксперимента была зафиксирована выборка значений вектора Y_Q^* при некотором изменении вектора параметров системы Q^* . Поэтому перед этапом балансировки модели исследователь располагает выборками объема N значений каждой n -й компоненты отклика реальной системы $\{Y_{Qnk}^*\}$, $k = \overline{1, N}$, при последовательном изменении вектора параметров системы Q аналогично натурному эксперименту и фиксирует выборки объема N откликов модели системы $\{Y_{nk}\}$, $k = \overline{1, N}$.

Определяются средние значения E_{3n} квадратов разностей компонент откликов Y_{Qnk}^* и Y_{nk} соответственно в натурном и модельном экспериментах и по возможности минимизируется

$$E_{3n} = \frac{1}{N} \sum_{k=1}^N (Y_{Qnk}^* - Y_{nk})^2 \rightarrow \min, \quad (7.14)$$

где k – номер эксперимента;

Y_{Qnk}^* и Y_{nk} – значения откликов n -й компоненты соответственно в k -м натурном и модельном экспериментах.

Составляется уравнение регрессии:

$$E_{3n} = \sum_{q=0}^h \beta_q Q_q. \quad (7.15)$$

Оцениваются коэффициенты β_q зависимости (7.15), где h – размерность вектора параметров. Затем проверяется значимость коэффициентов уравнения регрессии β_q при выбранном уровне значимости. Если коэффициенты уравнения окажутся значимыми, то на следующей

итерации меняются те параметры модели, которые позволяют уменьшить эту значимость. Целью этой процедуры, которая называется методом хорошего баланса, является получение незначимой регрессии $E_{3п}$.

Шаг 3. Оптимизация модели. Некоторые параметры системы могут оказаться наиболее существенными для изменения откликов реальной системы Y_Q^* . Поэтому не обязательно стремиться к тому, чтобы по всем параметрам регрессия $E_{3п}$ была незначима. Достаточно достичь незначимости регрессии $E_{3п}$ только для наиболее существенных параметров. Такие параметры называют калибровочными. На третьем шаге калибровки ИМ рассматриваются только изменения модели, связанные с калибровочными параметрами. Пусть H – вектор калибровочных параметров, а H_n – наилучшее значение H среди тех, которые использовались при различных прогонах модели системы $\varphi(X, G)$, так как оно дает минимальное значение погрешности $E_{3п}$.

Строится линейная регрессия между калибровочными параметрами и $E_{3п}$, основанная только на данных, собранных во время прогонов модели при соответствующих значениях H . Задавшись уровнем значимости α , проверяют значимость этой линейной регрессии. Если регрессия значима, уравнения регрессии, содержащие только те члены, коэффициенты которых значимы, могут рассматриваться как локальная аппроксимация функции $E_{3п}(H)$ после исключения шумовой компоненты. Направление вектора параметров в этой регрессии близко к направлению вектора наискорейшего спуска этой функции. Таким образом, следующим «хорошим» значением H окажется $H_{nm} = H_n - B$ ($B > 0$ – константа). Следующий прогон модели выполняется уже с новым значением H_{nm} . Определяются новые значения погрешностей $E_{3п}$, которые сравнивают с ранее полученными их значениями при H_n . Так как погрешности $E_{3п}$ являются случайными величинами, то сравнения любых двух значений еще недостаточно для определения, является ли модифицированная модель значимо лучшей. Поэтому сравниваются последовательности пар наблюдений $E_{3п}$, представляющие собой квадраты разностей компонент Y_Q^* и Y . Для каждой пары наблюдений проверяется, является ли модифицированная модель системы лучшей, чем предыдущая. Если улучшение характеристик модели достигнуто, то составляется новая регрессионная зависимость уже вблизи H_{nm} и вся процедура проверок повторяется. В противном случае этот шаг и все процедуры калибровки модели завершаются.

Все эти сравнения поведения модели и системы не могут быть исчерпывающими, так как модель нельзя откалибровать для всех экспериментальных условий, в которых она должна работать. Это связано

с большим количеством условий, возникающих зачастую непредсказуемо. В свою очередь, если бы была выполнена исчерпывающая калибровка с данными измерений, модель оказалась бы бесполезной, поскольку не требовалось бы предсказывать поведение системы. Модель обычно используется в условиях, в которых ее точность не всегда может быть проверена, поэтому калибровка обычно ограничена несколькими комбинациями параметров системы.

Итак, при определенных требованиях к точности в терминах допустимых погрешностей мы говорим, что модель пригодна для некоторого входного условия, если она удовлетворяет этим требованиям, когда ее переменные соответствуют этому условию. Область пригодности модели есть множество входных условий, для которых пригодна модель. При калибровке модели мы добиваемся ее пригодности для входных условий моделирования.

Схема изложенной выше 3-х-шаговой итеративной процедуры калибровки модели реальной системы представлена на рисунке 7.2. Если в результате процедуры оптимизации модели адекватность ее объекту исследования не достигнута, то необходимо вернуться на этап составления содержательного описания системы, добавить недостающую информацию о системе и провести весь процесс построения модели заново. Для более углубленного изучения данной проблемы можно рекомендовать работу [27].

7.6 Итоги

Подведем итоги. В 7.1 излагается общая методика испытания ИМ СС, которая предполагает выполнение следующих действий исследователя: оценку погрешности имитации; пересмотр состава генераторов псевдослучайных чисел; уменьшение влияния начального периода имитации на динамику моделирования СС; определение длительности переходного режима имитации; оценку устойчивости результатов имитации; исследование чувствительности ИМ; проверку адекватности ИМ реальному объекту исследования. В 7.2 детализируется методика верификации программы ИМ. Рассматриваются три способа верификации: структурного разбора ИМ; контроля за выполнением графа активностей; динамической верификации путем испытания «единичной нити» и затем с помощью проектирования вероятностной нагрузки. В 7.3 обсуждаются следующие шаги технологии определения технологических характеристик ИМ СС: оценка точности программы ИМ СС; нахождение доверительного интервала математического ожидания

n -й компоненты отклика; определение длины переходного периода имитации, оценки устойчивости имитации; реализация правил автоматической остановки имитации. В 7.4 излагаются шаги технологии оценки чувствительности ИМ СС: вычисление коэффициентов чувствительности откликов к вариациям параметров; составление матрицы коэффициентов чувствительности; отбраковка состава откликов и параметров с помощью коэффициентов чувствительности ИМ. В 7.5 излагается технология проверки адекватности ИМ прототипу СС, которая предполагает реализацию следующих шагов: проверка близости средних у ИМ и реальной системы; уменьшение точности в 3 раза в качестве критерия адекватности ИМ; калибровка ИМ, включающая три операции (глобальные структурные изменения в ИМ, локальные изменения путем замены компонентов реальной системы эквивалентами, использование калибровочных параметров). В завершение 7.5 излагается методика калибровки ИМ, состоящая из трех шагов: сравнение распределений откликов ИМ с реальными СС, балансировка ИМ на основе уравнения разностей между откликами ИМ откликами реальной СС; оптимизация ИМ по наиболее существенным откликам ИМ и реальной СС.

8 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И СРЕДСТВА ОПТИМИЗАЦИИ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

8.1 Метод пошаговой реструктуризации имитационных моделей

В данном разделе рассматривается новый подход к построению имитационных моделей вероятностных технологических процессов [28]. Излагается метод пошаговой реструктуризации имитационных моделей для решения задачи исследования вероятностных процессов с последовательной организацией технологического цикла. На основе метода пошаговой реструктуризации предлагается способ для построения оптимальной, в рамках выбранного критерия качества, структуры технологических процессов с последовательной организацией.

Вероятностные технологические процессы как объект имитации обладают рядом особенностей, отличающих их от других объектов, обычно исследуемых с помощью имитационного моделирования: вероятностный характер взаимодействия компонентов технологического процесса с системой управления технологическим циклом; наличие ограничений на использование ресурсов технологического процесса и их качество, влияющее на надежность выполнения технологических операций; надежность характеристики задействованного оборудования и необходимость оперативной синхронизации функционирования элементов системы управления при появлении случайных возмущений (сбоев и отказов оборудования). Основные аспекты проблемы исследования вероятностных технологических процессов с последовательной организацией определяются следующими факторами:

а) многообразием сложных технологических систем, в ходе реализации которых могут изменяться параметры их функционирования и структура технологического цикла;

б) важностью решения задач оптимизации структуры технологического цикла и управления технологическими системами с учетом текущих значений используемых ресурсов и состояния системы в режиме реального времени;

в) сложностью практических задач при оценке уровня надежности и безопасности потенциально опасных промышленных объектов;

г) необходимостью учета человеческого фактора при выполнении работ на потенциально опасных объектах.

Как показал анализ современного состояния разработок в этой области, проблема моделирования вероятностных технологических процессов состоит в недостаточной результативности существующих методов их исследования при увеличении количества учитываемых параметров, в особенности для тех случаев, когда структура технологического цикла изменяется в процессе функционирования объекта исследования.

Метод пошаговой реструктуризации. В основу предлагаемого метода пошаговой реструктуризации имитационной модели вероятностных процессов положен способ восстановления текущего состояния технологической системы в режиме реального времени (инвариантное погружение) через случайные временные интервалы с использованием принципа организации квазипараллелизма «до следующего события» при применении процедур Монте-Карло. Формальное представление технологического цикла при этом осуществляется с помощью вероятностного сетевого графика переменной структуры или полумарковской модели с переходами в нестандартные состояния.

Описание процесса реализации технологического цикла основано на использовании в структуре соответствующей имитационной модели агрегатов-имитаторов со стандартными элементами и сигналами, которые формируют управляющие воздействия на структуру модели в режиме реального времени путем их логической комбинации в зависимости от особенностей реализации вероятностного технологического процесса. Типы элементов различаются между собой степенью сложности алгоритма их выполнения в зависимости от состава используемого оборудования, ресурсов и надежности технологических операций.

Реализация имитационных моделей вероятностных технологических процессов при пошаговой реструктуризации. Построение и использование имитационной модели вероятностного технологического процесса при пошаговой реструктуризации реализуется следующей последовательностью этапов.

На *этапе 1* задаётся начальная структура исходной имитационной модели вероятностного процесса с помощью операторов подсистемы *PS.FORMSG*.

На *этапе 2* реализуется натурный эксперимент с целью получения исходной информации для последующей проверки адекватности имитационной модели реальному объекту имитации. В случаях, когда не удаётся найти аналитический вид аппроксимирующих функций распределения, используется табличная форма их представления.

На *этапе 3* осуществляется верификация базового варианта имитационной модели. Задаётся начальный состав ресурсов, материалов,

исполнителей и оборудования. Далее формируются управляющие таблицы-справочники и таблицы базы данных для хранения статистики имитации, используемые в дальнейшем при организации имитационного эксперимента.

На *этапе 4* реализуется функция испытания и анализа свойств имитационной модели, представляющая собой многошаговую процедуру использования стандартных методик испытания сложных систем, в процессе которой реализуются типовые этапы испытания имитационных моделей.

Начиная с *этапа 5*, осуществляется контроль реализации имитационной модели в пошаговом режиме через случайные интервалы времени (до следующего события) с одновременной проверкой текущего состояния вероятностного технологического процесса на соответствие базовому варианту имитационной модели. В случае наличия полного соответствия объекту имитации часы модельного времени переводятся на следующее событие, и функционирование имитационной модели продолжается. При изменении текущей структуры вероятностного процесса производится запись реализовавшейся части вероятностного процесса в базу данных модели. Последнее свершившееся событие фиксируется в качестве начального для дальнейшего продолжения имитации, после чего выполняется возврат на *этап 1*.

По окончании имитации в базе данных имитационной модели завершается процесс формирования рациональной структуры технологического цикла, построенной в режиме реального (модельного) времени и соответствующей заданным целям имитации с учетом реализовавшейся оперативной обстановки.

Обсуждение результатов. Для решения задачи исследования и управления вероятностными процессами разработан метод пошаговой реструктуризации имитационной модели вероятностного технологического процесса в реальном масштабе времени. На основе метода пошаговой реструктуризации и новой версии системы автоматизации имитационного моделирования агрегатного типа предложен способ имитации вероятностных процессов для построения рациональной структуры вероятностного технологического процесса.

Предложенный способ имитации вероятностных процессов на основе метода пошаговой реструктуризации ориентирован на случаи, когда динамику функционирования исследуемого объекта можно описать с помощью вероятностного сетевого графика переменной структуры или полумарковской модели на уровне элементов управления со сложной логикой при использовании технических средств сопряжения имитационной модели с вероятностным технологическим процессом.

Данный способ применим для технологических объектов с последовательной организацией технологического цикла, в том числе при отсутствии технических средств сопряжения имитационной модели с вероятностным технологическим процессом.

Новый способ имитации вероятностных процессов дает возможность его использования при проектном моделировании производственных технологических систем, выполнении научных исследований на объектах повышенной опасности и проектировании высоконадежных технологических комплексов для различных областей производственной деятельности.

В основу структурной реконфигурации технологического процесса производства положена технология имитационного моделирования, применяемая ко всему спектру задач синтеза оптимальной структуры вероятностных технологических систем с использованием агрегатной системы автоматизации моделирования, которая опирается на построение динамических имитационных моделей объекта исследования.

8.2 Динамическая имитация вероятностных технологических процессов производства

В данном разделе рассматривается новый подход к унификации методов системных исследований в области проектного моделирования сложных систем с вероятностными параметрами их функционирования на основе конечного множества математических моделей. В качестве инструмента реализации предлагаемого подхода используется динамическое имитационное моделирование, базирующееся на разработке динамических имитационных моделей вероятностных производственных систем.

В настоящее время накоплен большой опыт построения математических моделей объектов исследования из различных отраслей науки, техники, промышленности и экономики. Однако проявляющаяся в последнее время тенденция к использованию общесистемных принципов и методов исследований в различных областях знаний наталкивается на определенные трудности. Это связано, как известно, с целым комплексом различных причин: наличием больших объемов разнородной информации в различных областях знаний, использованием различных понятийных аппаратов, профессиональной разобщенностью исследователей и других.

Многочисленные попытки унификации системного подхода при решении конкретных задач науки и практики привели к понятию

сложной системы как многопараметрического объекта, представляемого конечным множеством математических моделей, каждая из которых отражает конкретную группу свойств исходной системы. При этом появляется возможность выделить классы сложных систем со специфическими свойствами, на основании которых разрабатываются методологические принципы построения математических моделей, которые характеризуются единой математической терминологией и могут быть доступны специалистам различных предметных областей.

Такой подход дает основания рассчитывать в дальнейшем на создание необходимой базы, позволяющей исследователю работать с системами любой степени сложности, вне зависимости от ее физической сущности или ограниченности рамками определенной формализации. К классам сложных систем относятся производственные и экономические системы, вычислительные сети, системы управления, системы энергообеспечения, а также другие системы, призванные обеспечивать безопасность жизнедеятельности и производства.

Системный анализ сложных систем положен в основу исследований производственных систем с вероятностными характеристиками их функционирования (вероятностных производственных систем) как класса сложных систем. При этом под вероятностными характеристиками функционирования могут пониматься характеристики надежности функционирования оборудования, задействованного в ходе реализации технологического цикла производства, характеристики выполнения технологических операций, качественные характеристики используемых в процессе производства материалов и комплектующих изделий и другие.

Краткий анализ состояния вопроса. Как показал анализ современного состояния разработок в области исследования и проектирования технологических объектов с вероятностными характеристиками их функционирования, проблема моделирования вероятностных производственных систем состоит в недостаточной результативности методов их исследования при увеличении количества учитываемых параметров, в особенности для тех случаев, когда структура технологического цикла изменяется в процессе функционирования объекта исследования.

Это связано, в первую очередь, с многообразием сложных технологических систем, в ходе реализации которых могут изменяться параметры их функционирования и структура технологического цикла; сложностью практических задач при оценке уровня надежности и безопасности потенциально опасных промышленных объектов;

необходимостью учета человеческого фактора при выполнении работ на потенциально опасных объектах.

В соответствии с вышесказанным является актуальной задача обеспечения надежности и безопасности функционирования производственных систем, а также повышение эффективности их использования. Одним из подходов на пути решения данной проблемы является проектное моделирование рациональной структуры вероятностных производственных систем как на стадии проектирования новых, так и при эксплуатации существующих технологических линий.

Поэтому разработка новых методов, программных средств анализа и синтеза оптимальной структуры наиболее сложных вероятностных технологических процессов, использующихся при организации работ на ответственных объектах при наличии оборудования, которое может отказывать и служить источником возникновения аварий техногенного характера, являются весьма актуальными с научной и практической точек зрения.

Методика построения компонентов динамической имитационной модели вероятностной производственной системы. Для построения компонентов динамической имитационной модели вероятностных производственных систем предлагается использовать агрегатную систему автоматизации моделирования вероятностных технологических процессов производства, реализующую агрегатный способ имитации сложных систем, а также способ формализации вероятностных производственных систем на основе вероятностных сетевых графиков (ВСГР). При этом используется шесть типов агрегатов-имитаторов:

$АТОР_{ij}$ – агрегат-имитатор выполнения технологической операции ($МТХО_{ij}$);

$АSOB_i$ – агрегат-имитатор свершения i -го события в процессе имитации функционирования вероятностной производственной системы (ВПС);

$АOBI\bar{N}_r$ – агрегат-имитатор функционирования оборудования индивидуального пользования с номером r ;

$АKAN_r$ и $АОВОР_r$ – агрегат-имитатор совместного использования оборудования общего пользования с помощью выделенного канала;

$АPROC_k$ – процедура-имитатор с номером k ликвидации аварийной ситуации при выполнении имитационной модели (ИМ) ВПС.

Каждый из этих агрегатов представляет собой реентерабельную программу из конечного множества математических моделей, составляющих математическую модель исходной вероятностной

производственной системы. Упомянутые модели агрегатов-имитаторов в базе данных модели имеют свой набор переменных и статистик моделирования в соответствии с версией модели. Количество версий равно числу агрегатов данного типа в ИМ ВПС. Динамика реализации алгоритмов этих агрегатов одинакова для каждого типа агрегатов, но каждая из версий может находиться в различных состояниях, согласно временной диаграмме реализации агрегатов в модельном времени. Рассмотрим кратко динамику реализации алгоритма одной из версий этих агрегатов.

$AMTXO_{ij}$ представляет собой четырехполюсный агрегат, имитирующий выполнение технологической операции $MTXO_{ij}$ в ИМ ВПС. В режиме прямой имитации приходит сигнал от $ASOB_i$, который по соответствующим функциям распределения формирует значения параметров агрегата (τ_{ijl} ; c_{ijl} ; $\{V_{rij}\}$; $\{mt_{ijl}\}$; $\{ko_{ijl}\}$) в l -й реализации ИМ. Затем каждый $ATOR_{ij}$ определяет индивидуальные запросы на ресурсы и оборудование в виде списков запросов ($SPINRS_{ijl}$; $SPOBR_{ijl}$; $SPISP_{ijl}$). Далее происходит обращение к системе распределения ресурсов, которая выделяет требуемые ресурсы на время имитации $MTXO_{ij}$. Когда при имитации $ATOR_{ij}$ на выделенном ему оборудовании возникают опасные отказы, то интервал времени выполнения $ATOR_{ij}$ будет увеличиваться до тех пор, пока не будут завершены восстановительные работы на соответствующем агрегате-имитаторе оборудования. При этом проводится анализ возможных последствий случившегося опасного отказа оборудования путем розыгрыша с вероятностью P_{ar} жребия первого типа «Произошла авария», который позволяет идентифицировать наличие аварии при работе агрегата на оборудовании с номером r . Если возникла сложная авария, то для ее ликвидации используется «штатная» последовательность процедур ликвидации аварий (агрегатов-имитаторов) $\{APROC_k\}$. Каждый из них обладает своим оборудованием, составом исполнителей и действует по утвержденному алгоритму ликвидации последствий аварийной ситуации. Длительность τ_{prkl} реализации $APROC_k$ также является случайной величиной и формируется по известной до начала имитации функции распределения $\Phi_{4k}(\tau_{pr})$. По окончании реализации этой последовательности агрегатов выполнение операции на отказавшем оборудовании продолжается с прерванного места, формируется признак «была авария» ($\pi_{ak} = 1$), а время выполнения операции на

оборудовании увеличивается на величину, равную суммарному времени выполнения всех процедур $APROC_k$. По окончании имитации последовательности $\{APROC_k\}$ инициируется возврат на $ATOP_{ij}$, где и формируется сигнал, поступающий на r -й вход агрегата $ASOB_j$. При этом передается признак $\pi_{ak} = 1$ («была авария» на оборудовании номера k), а агрегат $ATOP_{ij}$ переходит в режим ожидания инверсного сигнала от $ASOB_j$.

Агрегаты $ASOB_i$ являются многополюсными с различным числом входов и выходов. Выходы у $ASOB_i$ могут быть одиночными и «кустовыми». Из «кустовых» выходов агрегата формируются сигналы двух типов: действительный Sg_d , разыгрываемый по вектору вероятностей $\{P_{ijkl}\}$ и $(k-1)$ фиктивных сигналов Sg_f . Выходы $ASOB_i$ нумеруются, поэтому при адресации сигнала указывается номер события i и номер входа r в агрегат $ASOB_j$. Только действительные сигналы Sg_d , поступающие в режиме прямой имитации на вход $ATOP_{ij}$, инициируют его работу по изложенному алгоритму. Фиктивные сигналы Sg_f обходят алгоритм выполнения $ATOP_{ij}$. При этом у агрегатов $ASOB_j$ используется еще один тип выходных «кустовых» сигналов, называемых резервными выходами $ASOB_j$. С их помощью реализуется «технологическое резервирование» в ИМ ВПС. При появлении отказов оборудования во время выполнения $ATOP_{ij}$, которые посылают сигналы на входы $ASOB_j$, внутри сигнала устанавливается признак аварии (π_{av}). Если $\pi_{av} = 1$ (это означает, что $ASOB_j$ инициируется в ситуации, когда на предыдущем $ATOP_{ij}$ была авария), то в этих случаях инициируются «резервные» выходы. С этих выходов может формироваться любая комбинация сигналов Sg_d и Sg_f . Если $\pi_{av} = 0$ на всех входах $ASOB_j$, то это означает, что аварий до активизации агрегата не было, поэтому активизируются «штатные» $ATOP_{js}$.

Если хотя бы на один из входов $ASOB_j$ приходят сигналы от $ATOP_{ij}$, во время выполнения которых была авария, то это означает активизацию резервного кустового выхода. Механизм формирования действительных сигналов у «кустовых» выходов третьего типа основан на использовании булевой матрицы $\|\gamma_{nr}\|$, которую составляет технолог-эксперт до начала имитации. Наличие единицы на пересечении

n -й строки с r -м столбцом в этой матрице означает необходимость включения резервных $АТОР_{jr}$ в том случае, когда во время выполнения $АТОР_{nj}$ произошла авария ($\pi_{авн} = 1$). С помощью этого механизма кустовые выходы третьего типа становятся динамическими регуляторами подключения резервных $АТОР_{jr}$, если на входы $АSOB_j$ поступают сигналы от $АТОР_{ij}$, во время выполнения которых на оборудовании, используемом ими, происходили аварии.

Имитация функций оборудования инициируется запусками $АТОР_{ij}$, который может использовать любое число имитаторов оборудования. За каждым $АТОР_{ij}$ закрепляются два типа оборудования: индивидуального пользования с номером r , имитируемого агрегатами $АОВIN_r$; общего пользования, имитируемого агрегатами $АОВОР_r$. На агрегатах оборудования общего пользования $АТОР_{ij}$ использует некоторый объем общего ресурса V_{r2ij} на время его имитации. По окончании имитации занятый объем ресурса возвращается агрегату оборудования общего пользования $АОВОР_r$. Поскольку общим оборудованием могут пользоваться несколько $АТОР_{ij}$, то синхронизация их доступа реализуется следующим образом: работа оборудования общего пользования имитируется одним агрегатом $АОВОР_r$ и множеством агрегатов-имитаторов каналов общего пользования $АKAN_r$, число которых равно числу агрегатов $АТОР_{ij}$, использующих совместно ресурсы $АОВОР_r$. Агрегат $АОВОР_r$ функционирует непрерывно (последовательно переходит из состояния S_1 в состояние S_2 , а затем снова в состояние S_1) в течение l -й реализации ИМ ВПС. Длительности нахождения оборудования в этих состояниях определяются при их смене соответственно функциям распределения: k -й интервал безотказного функционирования $\tau_{БОkl}$ находится по функции распределения $\Phi_{1r}(\tau_{BO})$, определяя длительность нахождения общего оборудования в состоянии S_1 ; k -й интервал восстановления работоспособности оборудования $\tau_{БОkl}$ также находится по функции распределения $\Phi_{2r}(\tau_{BO})$, определяя таким образом длительность нахождения общего оборудования в состоянии S_2 . Это чередование состояний агрегата продолжается до окончания l -й реализации ИМ. Далее весь процесс имитации $АОВОР_r$ продолжается аналогичным образом, но уже в $(l+1)$ -й реализации имитационной модели.

Поскольку $АТОР_{ij}$ и $АОВОР_r$ функционируют независимо, каждый согласно своим алгоритмам, то для синхронизации совместного использования несколькими $АТОР_{ij}$ одного и того же $АОВОР_r$ (когда он находится именно в состоянии S_1) введено использование агрегата-канала $АKAN_r$, имитирующего использование общего ресурса V_{rjl} . Количество таких $АKAN_r$ определяется числом использований $АОВОР_r$ всеми агрегатами $АТОР_{ij}$.

Таким образом, с помощью комбинаций различных типов кустовых выходов и задания числа разветвлений каждого кустового выхода, эксперту-технологу на стадии проектирования предоставляется возможность динамического регулирования выполнения множества $\{АТОР_{ij}\}$ в зависимости от сложившейся операционной обстановки.

Синтез динамической структуры технологического цикла на имитационной модели вероятностной производственной системы.

Как видно из методики построения компонентов динамической имитационной модели, исходная вероятностная производственная система представляется в виде конечного набора взаимосвязанных математических моделей. Связь между компонентами математической модели исходной ВПС осуществляется посредством синхронизации взаимодействия агрегатов-имитаторов, входящих в состав моделей-компонентов.

На основании значений множества $\{\pi_{avij}\}$ агрегат события $АSOB_j$ осуществляет технологическое резервирование по изложенной выше методике. Кроме технологического резервирования, в ИМ ВПС предусмотрено автоматическое индивидуальное резервирование оборудования, которое включается при достижении фактической наработки оборудованием его критического значения. До начала имитации эксперт-технолог устанавливает для каждого устройства порог Q_{0r} этой наработки. Фактическая наработка $Q_{\phi r}$ накапливается на индикаторе Ind_r путем добавления к значению этого индикатора при каждом использовании устройства некоторого приращения $\Delta Q_{\phi r}$. С интервалом $\Delta \tau_{I3}$ значения этих индикаторов посылаются системе принятия решений *SPRESH* для обеспечения оперативного управления процессом имитации.

Окончательный вариант организации рациональной структуры технологического цикла, который соответствует текущему состоянию оборудования, принимается на основе анализа статистической информации, собранной в процессе функционирования имитационной модели.

Важной статистикой реализации ВСГР, лежащего в основе формализации вероятностной производственной системы, является граф критических путей (GR_KRP_h) в N реализациях в h -го варианта ВПС, который получен следующим образом. Каждый элемент l -й реализации ВСГР критического пути KRP_{lh} представляет собой последовательность чередования статистик выполнения агрегатов ($ASOB_i$, $ATOP_{ij}$, $ASOB_j$). При этом статистиками выполнения $ASOB_i$ являются четверки значений ($i, t_{pil}, t_{nil}, R_{il} = 0$), а статистиками $ATOP_{ij}$ служат тройки значений (ij, τ_{vijl}, c_{ijl}). Используя исходную структуру ВСГР, эти последовательности накладываются, а в итоге формируется вероятностный граф критических путей GR_KRP_h . На выходе $ASOB_i$ подсчитывается количество запусков $ATOP_{ij}$ (v_{ij}) во всех N реализациях исходного ВСГР. Затем происходит формирование вероятностей P_{ij} запусков $ATOP_{ij}$ с выходов $ASOB_i$ графа GR_KRP_h . Таким образом, кроме четверки статистик $ASOB_i (i, \bar{t}_{pi}, \bar{t}_{ni}, \bar{R}_i)$, на каждом выходе этого агрегата формируется вектор $\{P_{ks}\}$ вероятностей запусков $ATOP_{ij}$, где s – номера компонентов этого вектора вероятностей.

Другой группой статистик $\{ST_{hl}\}$ являются графики расхода ресурсов в модельном времени при l -й реализации ИМ ВСГР: ресурсов (GR_{1rl}) с номером r , материалов и комплектующих изделий (GR_{2rl}) с номером r , суммарного расхода финансовых средств (GR_{3l}). Фиксируются также временные диаграммы использования: оборудования ($DIAG_{1l}$), исполнителей ($DIAG_{2l}$), бригад исполнителей ($DIAG_{3l}$).

Таким образом, в конце l -й реализации имитационной модели ВПС множество статистик $\{ST_{hl}\}$ включает в себя следующие графики и диаграммы: $\{GR_{1rl}\}$, $\{GR_{2rl}\}$, $\{GR_{3l}\}$, $\{DIAG_{1l}\}$, $\{DIAG_{2l}\}$, $\{DIAG_{3l}\}$. Эти статистики запоминаются в виде файлов графиков и диаграмм в базе данных имитационной модели, и по решению эксперта-технолога могут направляться в систему принятия решений ($SPRESH$) для выполнения анализа в цикле каждой l -й реализации имитационной модели. По завершении реализаций ИМ графики и диаграммы (в модельном времени t_0) накладываются друг на друга системой $SPRESH$ и в виде набора параметров (θ_6) передаются эксперту для выбора окончательного варианта структуры ВПС.

Обсуждение результатов. Новизна класса формализуемых систем производственных систем состоит в том, что в процессе функционирования технологического цикла может произойти изменение его

структуры в результате управляющих воздействий аппаратуры управления или оператора технологического цикла вследствие сбоев или отказов функционирования задействованного оборудования. При этом предложенный подход позволяет решать следующие задачи:

а) проектного моделирования надежных и безопасных производственных систем с учетом операций резервирования и ликвидации последствий сбоев и аварий оборудования;

б) синтеза оптимальной структуры технологического цикла с учетом надежности характеристик используемого оборудования на стадии проектирования;

в) обеспечения динамической имитации класса производственных систем с вероятностными параметрами их функционирования на стадии проектного моделирования с использованием конечного множества математических моделей,

Это в совокупности дает возможность ускорения разработки новых проектов технологических комплексов и снижения себестоимости проектирования их оптимальной структуры при наличии элементов потенциальной опасности.

Разработанный подход и программное обеспечение проектного моделирования производственных систем с вероятностными характеристиками позволяют: повысить надежность и безопасность производства существующих технологических систем за счет резервирования цепочек технологических операций при возникновении аварийных ситуаций; снизить вероятность возникновения аварийных ситуаций на производстве, на основе построения рациональной структуры технологического цикла на стадии проектирования; обеспечить резервирование потенциально опасных технологических операций при проведении пуско-наладочных работ и монтаж технологических линий, имеющих определенный срок предварительной эксплуатации.

8.3 Система оперативного контроля имитации управляемых технологических процессов

Возникновение понятия сложной системы как многопараметрического образа объекта исследования, представляемого конечным множеством математических моделей, каждая из которых отражает конкретную группу свойств исходной системы, связано с многочисленными попытками унификации системного подхода в процессе решения конкретных задач науки и практики. При этом появилась возможность выделить классы сложных систем со специфическими

свойствами, что обеспечило возможность разработки методологических принципов построения математических моделей, которые характеризуются единой математической терминологией и могут быть доступны специалистам различных предметных областей.

В данном разделе системный анализ сложных систем положен в основу исследований управляемых технологических процессов с вероятностными характеристиками их функционирования (управляемых вероятностных технологических процессов) как класса сложных систем. Под вероятностными характеристиками функционирования могут пониматься характеристики надежности функционирования оборудования, задействованного в ходе реализации технологического цикла производства, характеристики выполнения технологических операций, качественные характеристики используемых в процессе производства материалов и комплектующих изделий и другие.

Предлагается новый подход к унификации методов системных исследований в области проектного моделирования сложных систем с вероятностными параметрами их функционирования на основе конечного множества математических моделей. В качестве инструмента реализации предлагаемого подхода используется динамическое имитационное моделирование, базирующееся на разработке динамических имитационных моделей управляемых вероятностных технологических процессов.

Для построения компонентов динамических имитационных моделей объектов исследований используется агрегатная система автоматизации моделирования вероятностных технологических процессов, реализующая агрегатный способ имитации сложных систем, а также способ формализации вероятностных производственных систем на основе вероятностных сетевых графиков. Система контроля имитации при этом обеспечивает возможность принятия управляющих воздействий на основе текущей операционной обстановки в процессе функционирования технологического цикла.

Структура системы контроля имитации и управления. Под управлением понимается упреждение конфликтных ситуаций в функционировании вероятностного технологического процесса (ВТП), которое достигается с помощью аппаратно-программного комплекса, состоящего из средств аппаратного сопряжения ВТП с параллельно действующей ЭВМ.

Система оперативного контроля имитации управляемых вероятностных технологических процессов (ВТП) состоит из следующих компонентов: имитационной модели (ИМ) управляемого ВТП;

подсистемы планирования имитационных экспериментов (*PS.PLANEX*); подсистемы оперативного контроля имитации (*PS.CONTROL*); подсистемы анализа результатов (*PS.ANALIS*); подсистемы-эксперта (*PS.EXPERT*), контролирующей ход имитации и управляющей изменением параметров модели.

Функционирование компонентов системы оперативного контроля осуществляется с помощью управляющих воздействий:

- взаимодействия подсистем с «экспертом» θ_k ($k = \overline{1,5}$): чтения информации о состоянии технологического цикла ($\theta_1, \theta_2, \theta_4$); оперативного воздействия «эксперта» (θ_3) по текущему варианту реализации структуры ВТП в имитационной модели; установки начальных значений состава ресурсов в h -м варианте организации структуры ВТП (θ_5);

- взаимодействия подсистем с ИМ (выдача оперативной информации о ходе имитации ВТП с помощью множества индикаторов состояния оборудования $\{ind_r\}$; формирование множества статистик $\{ST_{kl}\}$ и откликов $\{Y_{0jl}\}$ в l -й реализации ИМ ВТП по методу Монте-Карло);

- множества постоянных характеристик модели $\{G_h\}$ и параметров $\{X_{0jh}\}$ моделирования ВТП, задаваемых для «запитки» модели при запуске на имитацию h -го варианта состава ресурсов и структуры ВТП.

Основным компонентом системы является агрегатная имитационная модель, реализующая имитацию объекта исследования с помощью вероятного сетевого графика (ВСГР) и построенная на основе системы автоматизации имитационного моделирования.

В процессе реализации прогона имитационной модели осуществляются следующие виды внутреннего управления динамикой имитации:

- автоматическое одиночное резервирование оборудования в тех случаях, когда наработка на отказ r -го устройства оборудования (Q_r) достигает критической величины;

- ликвидация последствий аварий при опасных отказах функционирования r -го устройства последовательностью процедур $\{PROC_k\}$;

- технологическое резервирование, означающее изменение состава и последовательности выполнения технологических операций $\{MTXO_{ij}\}$ после возникновения аварии.

В имитационной модели предусмотрено также множество управляющих воздействий ($\{\alpha_k\}$, $k = \overline{1,4}$) подсистемы *PS.CONTROL*,

включающих в себя: переключение на режим профилактики оборудования (α_1), изменение шага наблюдения и формирования информации для построения графиков расхода ресурсов ВТП (α_2), выдачу диаграмм перехода компонентов имитационной модели из состояния в состояние (α_3); досрочную остановку имитации использования h -го варианта состава ресурсов ВТП (α_4).

Функциональное назначение компонентов ИМ ВТП. Для постановки серий имитационных экспериментов (ИЭ), отображающих взаимодействие имитационной модели с компонентами системы контроля, используется агрегатная система автоматизации моделирования, которая реализует агрегатный способ имитации ВТП и состоит из следующих подсистем, библиотек и компонентов:

- библиотеки типовых агрегатов *LB.AGREG*;
- подсистемы формирования имитационных моделей из множества агрегатов-компонентов ВСП (*PS.FORMSG*);
- управляющей программы моделирования (*UPMA*);
- подсистем обработки статистики имитации ВСП (*PS.OBRAB*), визуализации результатов ИЭ (*PS.VISIAL*), первичного анализа результатов экспериментов (*PS.ANALIZ*).

Рассмотрим структуру и функциональное назначение перечисленных компонентов. Библиотека *LB.AGREG* содержит несколько типов реентерабельных программ, используемых для имитации функций агрегатов $AMTXO_{ij}$, $ASOB_i$, $AOBIN_k$, $APROC_k$. За время постановки ИЭ для каждой l -й реализации имитационной модели программы $AMTXO_{ij}$ и $ASOB_i$ циклически переходят в различные состояния под управлением *UPMA*.

При построении ИМ ВТП компоненты этой библиотеки используются в качестве «заготовок» для построения подпрограмм агрегатов, которые совместно с *UPMA* образуют ее основную часть. Подсистема *PS.FOMSG* организует ввод исходной информации, который иницируется *PS.PLANEX*, проверяет правильность написания состава и структуры ИМ, сообщает эксперту о наличии ошибок коммутации, организует верификацию функционирования скомпонованных агрегатов в виде единой имитационной модели.

Подсистема *PS.MONTEK* содержит библиотеку процедур формирования случайных величин, программу реализации статистических испытаний и вычисления оценок математических ожиданий и дисперсий откликов ИЭ. Подсистема *PS.OBRAB* автоматизирует операции обработки первичной статистики моделирования, являясь при этом адаптацией известного пакета *STATISTIKA*.

Подсистема *PS.VIZUAL* формирует временные диаграммы использования устройств и ресурсов ($DIAG_{rhl}$), а также графики $Z_{1rhl}(t_0)$ и $Z_{2rhl}(t_0)$ для l -й реализации ВСГР. Управляющая программа моделирования *UPMA* организует переход агрегатов из состояния в состояние и обеспечивает сочетание прямого и инверсного способа изменения модельного времени t_0 с реализацией процедур Монте-Карло, а также контролирует моменты вычисления t_{pil} и t_{pjl} агрегатов $ASOB_i$.

Таким образом, структура имитационной модели формируется с помощью библиотек агрегатной системы имитации с помощью таблиц коммутации агрегатов $AMTXO_{ij}$ и $ASOB_i$. Эту работу выполняет *PS.PLANEX*, воздействуя через управление α_5 на *UPMA*. На следующем этапе исследования проводится серия экспериментов на реальном ВТП. Если же проектируется новая структура ВТП, то используются экспертные данные о параметрах и надёжности функционирования $AMTXO_{ij}$. Результатом этапа является формирование базы данных модели, содержащей информацию о параметрах $MTXO_{ij}$. На следующих этапах исследования организуется серия ИЭ для испытания ИМ ВТП согласно известным методикам.

Контроль хода имитации управляемого ВТП. Возможность возникновения аварии оборудования ВТП при выполнении $AMTXO_{ij}$ требует предусмотреть выход из ситуации, возникшей после аварии, хотя она уже ликвидирована последовательностью $\{APROC_k\}$. На этот случай в ИМ ВТП предусмотрены «резервные» цепочки $AMTXO_{ij}$, которые активизируются только при появлении аварий. Переключение на «резервную» ветвь $AMTXO_{ij}$ реализуется за счёт использования булевой матрицы коммутации $\|\gamma_{ns}\|$, формируемой экспертом-технологом до начала имитации. Строками матрицы – номера $AMTXO_{ij}$ на входе $ASOB_i$, а столбцами являются номера резервных $AMTXO_{ij}$ на выходе $ASOB_j$, которые необходимо инициировать после аварии. Таким образом, подобное «технологическое» резервирование является динамическим регулятором поставарийной ситуации в ВТП.

Ещё одним способом недопущения аварий оборудования ВТП является автоматический переход на резервное устройство, когда хотя бы для одного из устройств оборудования фактическая наработка на отказ превышает пороговые значения в ind_r . В этом случае множество $\{ind_r\}$ поступает в *PS.OPEREX*, которая проверяет близость к пороговому значению наработки всех устройств оборудования. При

достижении близости к пороговым значениям у нескольких устройств формируются следующие воздействия на ИМ ВТП: групповой переход на резервирование тех устройств, у которых наработка близка к критической (α_1); переход на общую профилактику оборудования из-за неэффективности группового резервирования или нехватки резервных устройств (α_2); допускается возможность аварии в тех случаях, когда останавливать ВТП нельзя, и состояния индикаторов игнорируются (α_3); если оборудование изношено, а общая профилактика неэффективна, принимается решение о досрочном завершении имитации h -го варианта ИМ, поскольку появление аварии для данного ВТП недопустимо (α_4).

PS.ANALEX использует статистику имитации $\{ST_k\}$ и множество откликов модели $\{Y_{0j}\}$. Откликами Y_{0j} являются усреднённые по числу реализаций их интегральные значения для h -го варианта ВТП: критическое время выполнения технологического цикла (T_{KPh}), стоимость реализации ВТП (C_{0h}), интегральный расход материалов и комплектующих изделий (mt_{0h}), количество использований ресурсов r -го номера ($v_{\sum rh}$), суммарная стоимость ликвидации аварий ($C_{\sum ABh}$), общие потери на профилактику (T_{OPPh}). Эти интегральные отклики модели составляют многомерный вектор откликов VO_h варианта ВТП, у которого все компоненты требуют минимизации, но имеют различную размерность и свои диапазоны изменения. Поэтому необходима нормировка компонентов этого вектора их максимальными значениями для всех возможных вариантов структуры ВТП.

Важной статикой реализации ИМ ВТП является граф критических путей ($GRKRP_k$), который получен после наложения всех критических путей. С помощью сообщения θ_4 *PS.ANALEX* выдаёт эксперту графики расхода ресурсов r -го типа $Z_{1rh}(t_0)$, финансовых средств $Z_{2rh}(t_0)$, а также временные диаграммы использования оборудования и исполнителей ($DIAG_{rh}$). Оперативная статистика реализации ВТП в виде сообщения θ_4 предоставляется эксперту-технологу для принятия решений на основе классической теории принятия решений. В результате эксперт формирует набор управляющих воздействий ($\theta_1, \theta_2, \theta_3$ и θ_5), которые затем через подсистему *SPRESH* обеспечивают возможность досрочной остановки l -й реализации ИМ, перевод всех устройств оборудования на профилактику, оперативное изменение характеристик надёжности функционирования оборудования.

Для сравнения вариантов структуры ВТП осуществляется «свёртка» этого вектора к скалярному показателю W_h способом весовых коэффициентов важности ($\sum_j \delta_j = 1; 0 \leq \delta_j \leq 1$) откликов с номером j . Вариантам организации ВТП соответствуют значения вектора $\{X_{oih}\}$ параметров ВТП и постоянных параметров имитации $\{G_h\}$. Каждая из составляющих векторов параметров может изменяться на различных уровнях. Поэтому количество вариантов K_0 ($h = \overline{1, K_0}$) определяется стратегией изменения каждого уровня параметров. Выбор оптимальной стратегии осуществляется на основе классических методов планирования экспериментов. Эксперт с помощью воздействия θ_5 инициирует *PS.ANALEX* и последующий запуск h -го варианта ИМ. Минимальное значение W_h по всему множеству вариантов с номером h_0 и будет решением задачи определения рационального варианта организации структуры ВТП.

Обсуждение результатов. Разработанный подход имитационного моделирования для нового класса объектов (управляемых производственных систем с вероятностными параметрами их функционирования) обеспечивает возможность корректировки последствий отказов оборудования и возврата переменных управления в заданные диапазоны значений с использованием зарезервированных технологических операций.

Программное обеспечение, которое реализует данный способ имитационного моделирования, обеспечивает решение задачи проектного моделирования рациональной структуры технологического цикла при наличии сбоя и отказов оборудования. Как следует из вышесказанного, предложенный подход и программное обеспечение для исследования управляемых вероятностных технологических процессов как класса сложных систем позволяют:

- а) повысить надежность и безопасность производства существующих технологических систем за счет резервирования цепочек технологических операций при возникновении аварийных ситуаций;
- б) снизить вероятность возникновения аварийных ситуаций на производстве, на основе построения рациональной структуры технологического цикла на стадии проектирования;
- в) обеспечить резервирование потенциально опасных технологических операций при проведении пуско-наладочных работ и монтаж технологических линий, имеющих определенный срок предварительной эксплуатации;
- г) повысить производительность технологических линий;

- д) снизить себестоимость проектного моделирования высоконадежных производственных систем;
- е) обеспечить конкурентоспособность выпускаемой продукции, как на внешнем, так и на внутреннем рынках Республики Беларусь.

8.4 Методика анализа надежности функционирования вероятностных производственных систем

В данном пункте рассматривается методика анализа надёжности функционирования вероятностных производственных систем, позволяющая определить параметры выполнения технологических операций, обеспечивающие заданный уровень надёжности функционирования технологического цикла.

Практическая значимость предложенного подхода состоит в реализации процедуры оценки вероятностных значений параметров надёжности функционирования компонентов производственной системы, обеспечивающей оптимальные варианты организации структуры технологического цикла в рамках заданного критерия качества его выполнения.

Анализ функционирования технологических процессов при наличии элементов потенциальной опасности обладает рядом особенностей и связан с необходимостью учета вероятности сбоев, отказов и аварий оборудования. Наиболее сложными объектами для исследования являются технологические системы, в ходе реализации которых могут изменяться вероятностные параметры их функционирования. При этом под вероятностными параметрами могут пониматься параметры выполнения технологических операций, обеспечивающих заданный уровень надёжности функционирования технологического цикла производства, а также надёжные характеристики оборудования, используемого при реализации технологических операций процесса производства.

Новизна класса формализуемых объектов вероятностных технологических процессов при наличии элементов потенциальной опасности состоит в том, что в процессе функционирования технологического цикла могут произойти изменения в его структуре в результате управляющих воздействий аппаратуры управления или оператора, управляющего технологическим циклом, а также при необходимости упреждения выхода бракованной продукции.

В данной работе предлагается новый подход, связанный с погружением моделей технологических объектов, имеющих графовую

структуру, в пространство возможных состояний объекта исследования при наличии функциональных зависимостей между операциями для реализации сквозной технологии объектно-ориентированного программирования контура управления технологическим циклом производства.

Следует отметить, что единственной технологией, обладающей возможностью структурной реконфигурации технологического процесса, остается технология имитационного моделирования, применяемая ко всему спектру задач синтеза оптимальной структуры вероятностных технологических систем в рамках предложенного нами подхода.

Методика анализа интегральной статистики и принятия проектных решений. Отличительной особенностью векторной (многокритериальной) оптимизации является наличие множества недоминирующих решений, каждое из которых может быть выбрано исследователем в качестве оптимального. Как известно, в общем виде задачи векторной оптимизации могут быть сформулированы следующим образом:

$$f_i(x) \rightarrow \max, i = \overline{1, k}; f_j(x) \rightarrow \min, j = \overline{k+1, m}; x \in G \quad (1)$$

Понятие оптимального решения для таких задач заменяется понятием эффективного решения. В агрегатной системе имитации процедура *PR.USTUPK* подсистемы *PS.RESHEN* реализует наиболее часто применяемые методы многокритериальной оптимизации, при этом выделяется главный критерий ($f_1(x) \rightarrow \max$). В этом случае задача векторной оптимизации имеет вид:

$$f_i(x) \rightarrow \max; f_i(x) \geq f_i^*, i = \overline{1, k}; f_j(x) \leq f_j^*, j = \overline{k+1, m}, \quad (2)$$

где f_i^* и f_j^* – множество экспертных оценок для неглавных критериев оптимизации. Подобная процедура реализует метод последовательных уступок, согласно которому все критерии эффективности располагаются в порядке уменьшения степени их важности ($f_{i1}(x), f_{i2}(x), \dots, f_{in}(x)$).

Процедура *PR.SOSTKP* подсистемы *PS.RESHEN* реализует метод составного критерия. Исследователь определяет важность γ_i , где $\gamma_i \geq 0$, i -го критерия (например, дисперсия $f_i(x)$ может быть «весом» критерия). Составной критерий имеет вид:

$$u(x) = \sum_{i=1}^n \gamma_i f_i(x) \rightarrow \max \quad (3)$$

В более общем случае используется процедура *PR.BEZRA* подсистемы *PS.RESHEN*, реализующая общий случай нормативных методов векторной оптимизации. Все нормативные методы предполагают возможность предварительного получения нормативов λ_{f_i} , $i = \overline{1, m}$ на основе приближенного решения многоцелевой задачи и приближения к этим нормативам по заданной метрике $\rho(f(x), \lambda_{f_i}) \rightarrow \min$, которая может вычисляться одним из следующих способов:

$$\rho_1(f(x), \lambda_{f_i}) = \sum_{i=1}^m [f_i(x) - \lambda_{f_i}]^2; \quad (4)$$

$$\rho_2(f(x), \lambda_{f_i}) = \sum_{i=1}^m |f_i(x) - \lambda_{f_i}|; \quad (5)$$

$$\rho_3(f(x), \lambda_{f_i}) = \max_i |f_i(x) - \lambda_{f_i}|. \quad (6)$$

В ряде случаев в качестве критериев выступают булевы переменные и функции их использующие. Для таких случаев в *PS.RESHEN* имеется процедура *PR.GLKP*, которая реализует метод логического объединения критерия и используется, когда $f_i(x)$, при $i = \overline{1, m}$, могут принимать значения только 1 (i -я цель достигнута) или 0 (в противном случае).

Для выбора рационального варианта используется процедура *PR.CRIT* подсистемы *PS.RESHEN*, которая выбирает рациональный вариант, согласно одной из стратегий выбора:

1) усредненные значения критериев

$$u_1 = \max_s (f_{is} = \frac{1}{2}(\max_i f_{is} + \min_i f_{is})), i = \overline{1, s}; \quad (7)$$

2) оптимистическое значение оценки принятия решений

$$u_2 = \max_s (f_{ir} = \max_i f_{is}), i = \overline{1, s}; \quad (8)$$

3) пессимистическая оценка

$$u_3 = \max_s (f_{ir} = \min_i f_{is}), i = \overline{1, s}; \quad (9)$$

4) нейтралитета

$$u_4 = \max_s (\frac{1}{s} \sum_{i=1}^s f_{ir}), i = \overline{1, s}; \quad (10)$$

5) Севиджа

$$u_5 = \min_s (f_{ir} = \max_i (\max_s f_{is} - l_{is})), i = \overline{1, s}. \quad (11)$$

Отметим, что все отклики f_{ir} и f_{is} являются скалярами. Набор классических критериев (8.7)–(8.11) позволяет принимать эффективные решения в условиях неопределенности и риска. Отметим, что эти критерии вычисляются при условии постоянства значений множеств $\{G\}$ и $\{Z\}$ в ходе выбора оптимального варианта решения.

Если же $\{G\}$ и $\{Z\}$ таковыми не являются, то составляется матрица принятия решений $\|f_{ir}\|$, в которой строками r являются варианты структуры технологического цикла, а столбцами – вероятности нахождения множеств $\{G\}$ и $\{Z\}$ в различных состояниях. В этом случае используется методика выбора варианта рациональной организации структуры по матрице принятия решения.

Таблица 8.1

Вектор вероятностей начального состояния v_n	i_n	1	2	3	4	5	6	7
	P_n	0,60	0,10	0,05	0,05	0,05	0,05	0,10
Функция распределения цикла смен состояний $F_0(v)$	P_k	0,10	0,20	0,30	0,40	0,50	0,60	1,00
	v_k	20	30	40	50	60	70	80
вектор вероятностей конечного состояния v_k	i_k	1	2	3	4	5	6	7
	P_k	0,10	0,05	0,05	0,10	0,10	0,10	0,50
	j	1	2	3	4	5	6	7
Матрица вероятностей перехода из состояния i в состояние j $\ P_{ij}\ $	i							
	1	0,05	0,30	0,20	0,10	0,05	0,20	0,10
	2	0,30	0,05	0,10	0,05	0,05	0,05	0,40
	3	0,20	0,10	0,05	0,10	0,15	0,20	0,20
	4	0,05	0,30	0,30	0,05	0,10	0,10	0,10
	5	0,05	0,05	0,05	0,05	0,05	0,70	0,05
	6	0,10	0,10	0,10	0,20	0,25	0,05	0,20
	7	0,20	0,10	0,05	0,05	0,50	0,05	0,05

В таблице 8.1 приведены значения вероятностей перехода технологического цикла контрольного примера из состояния в состояние при определении рационального варианта организации его структуры при наличии сбоев и отказов оборудования.

Оценка ошибок имитации при оптимизации структуры технологического цикла. Для автоматизации постановки серий имитационных экспериментов на модели использовалась агрегатная система имитации, в библиотеке которой (*LIB.PROC*) реализованы соответствующие реентерабельные программы. С помощью подсистемы формирования имитационной модели (*PS.FORMPR*) на основании

исходной информации происходит компоновка структуры программы модели.

В результате создается вариант программы имитационной модели, для которого параметры состава ресурсов и надежность характеристики функционирования оборудования технологического цикла находятся в «серединной» точке пространства параметров состояний модели. Вариант имитационной модели, для которого параметры находятся в серединной точке пространства, в дальнейшем будем называть «базовой моделью». Таким образом, универсальная имитационная модель из библиотеки *LIB.PROC* системы автоматизации моделирования превращается в модель, соответствующую конкретной структуре тестового варианта объекта исследования.

На данном этапе требуется определение достаточного числа реализаций процедуры Монте-Карло при постановке серий имитационных экспериментов, оценка погрешности имитации, анализ «чувствительности» откликов модели к вариациям параметров моделирования в диапазоне их изменения.

На *первом шаге* этапа с помощью универсальной процедуры *PR.TOCHN* определялась *точность* имитации. Число «прогонов» модели в имитационном эксперименте было установлено равным тридцати ($N_1 = 30$). Каждый «прогон» предполагает запись интегральной статистики имитации в базу данных имитационной модели. Процедура *PR.TOCHN* по выборке объема $N_1 = 30$ для каждого l -го «прогона» ($l = \overline{1, N_1}$) определяет оценку математического ожидания и дисперсии n -го отклика ($\bar{Y}_n$ и D_n). Точность оценивается по формуле:

$$\varepsilon_n \% = \frac{0,75L}{\bar{Y}_n} \sqrt{\frac{D_n}{N_1 - 1}} \cdot 100 \% . \quad (8.12)$$

Оценка общей ошибки имитации при этом равна $\varepsilon = \max_n \varepsilon_n \%$. При каждом «прогоне» начальные значения базовых генераторов равномерно распределенных величин ξ_0 определялись путем совмещения значений двух конгруэнтных генераторов псевдослучайных величин ξ_0 . После проведения тридцати имитационных экспериментов для имитационной модели с помощью процедуры *PR.TOCHN* была определена точность имитации ($\varepsilon_n = 10 \%$). Для иллюстрации реализации предложенной методики исследования точность в 10 %, с нашей точки зрения, достаточна. В том случае, когда подобная точность имитации мала, необходимо увеличить число реализаций

процедуры Монте-Карло до величины N_2 , большей N_1 но для этого необходимы дополнительные расходы ресурсов машинного времени.

На *втором шаге* этапа с помощью процедуры *PR.CHUVS* подсистемы *PS.ISPITAN* оценивается «чувствительность» откликов имитационной модели к изменению параметров имитации при числе реализаций $N_1 = 30$ (использовалась известная методика [3]). Процедура *PR.CHUVS* позволила вычислить *коэффициент чувствительности* откликов к вариациям параметров по формуле:

$$\delta Y_n = \frac{|Y_h^+ - Y_h^-| \cdot 2}{(Y_h^+ + Y_h^-)} \cdot 100 \%, \quad (8.13)$$

где Y_h^+ и Y_h^- – значения откликов при значениях параметров имитационной модели соответственно из X_q^+ и X_q^- (здесь диапазон изменения q -го параметра равен (X_q^+, X_q^-)). Результаты расчетов коэффициентов с помощью процедуры *PR.CHUVS* показывают, что для всех откликов *коэффициент чувствительности* δY_n изменяется от 20 % до 30 %. Поскольку $\delta Y_n > \varepsilon_n$ для всех параметров, то в дальнейших имитационных экспериментах нельзя уменьшать состав параметров и откликов имитационной модели.

На *третьем шаге* этапа определялось число реализаций N процедуры Монте-Карло в имитационной модели с помощью расчетной формулы:

$$N > \frac{S_n}{d^2(1-\alpha)}; \quad (8.14)$$

где $\alpha = 0,05$ – коэффициент значимости,

$d = S_n / 4$ – точность имитации.

Расчеты по формуле (8.14) определили необходимость задания числа реализаций $N_1 \geq 100$ экспериментов.

Апробация методики анализа надежности функционирования вероятностных производственных систем. Анализ надежности функционирования вероятностных производственных систем осуществлялся на имитационной модели в рабочей точке пространства параметров надежности $\{GH\}$ на основании матрицы надежностных характеристик оборудования технологического цикла. В ходе перехода из состояния i в состояние j процесс *PR.UZEL* определял времена и стоимости нахождения технологического цикла в состояниях ij .

Каждый j -й процесс $PR.SOST_{ij}$ заказывал ресурсы в состояниях ij l -й реализации ($l \leq 100$). Процедура Монте-Карло активизировала $PR.UZEL$ с начального состояния $i_n = 1$.

При выполнении алгоритмов $PR.SOST_{ij}$ для тех случаев, когда использовалось оборудование, проверялась ситуация возникновения отказов. В l -й реализации для каждого устройства оборудования с номером k определялись конкретные значения характеристик надежности оборудования (τ_{BOkl} , τ_{VOkl} , C_{VOkl} , τ_{AV1kl} , C_{AV1kl} , τ_{AV2kl} , C_{AV2kl}). Эти характеристики оставались неизменными до возникновения очередного отказа оборудования номера k . При очередной активизации процессу $PR.SOST_{ij}$ процессом $PR.UZEL$ передаются номера предыдущего i и последующего состояний j . В момент активизации $PR.SOST_{ij}$ по матрице условных распределений определяет время нахождения технологического цикла в состоянии ij τ_{ijkl} и стоимость реализации в состоянии ij C_{ijkl} .

При использовании оборудования номера k проверяется условие возникновения отказа из-за превышения граничного времени наработки устройства $Q_{fkl} \leq Q_{nark}$, где Q_{fkl} – накопленное значение времени наработки k -го устройства в l -й реализации; Q_{nark} – граничное значение времени наработки k -го устройства в l -й реализации имитационной модели.

После накопления статистик имитации процесс $PR.SOST_{ij}$ активизирует процесс $PR.UZEL$ вместе с указанием признаков ($UKAZ$ и π_{av}), на основе которых процесс выбирает номер активизируемого процесса $PR.SOST_{ij}$.

Процесс $PR.SOST_{ij}$ в j -м состоянии формирует очередную точку временной диаграммы использования ресурсов при переходе ВТП1 из состояния i в состояние j , состоящих из последовательностей статистик имитации.

В результате завершения имитационного эксперимента в файле статистики находится множество временных диаграмм реализации процесса в виде записи ($\{ST1_{jhl}\}; j = \overline{1,10}, h = \overline{1,100}, l = \overline{1,100}$) и множество интегральных статистик ($\{STIN_{jl}\}; j = \overline{1,10}, l = \overline{1,100}$). По окончании $N_1 = 100$ реализаций исследования варианта структуры в срединной точке параметров надежности $\{GH\}$ в файле статистик имитации имеется N_1 записей первого и второго типа. Этот файл

статистики был обработан *PS.OBRABOT*. Обработка статистики имитации состояла в усреднении статистик имитации $\{STIN_{jl}\}$ и нахождения средних значений по N_1 реализациям интегральных откликов нахождения ВТП1 в j -х состояниях и определении векторов откликов $\{S\tau_j\}$, $\{SC_j\}$, $\{Sko_j\}$, $\{Smt_j\}$; $j = \overline{1,10}$ – среднее значение статистик соответственно времени нахождения, стоимости реализации, расхода комплектующих и материалов $PR.SOST_{ij}$ в j -х состояниях. При этом вычислялись дисперсии статистик расхода ресурсов $\{D\tau_j\}$, $\{DC_j\}$, $\{Dko_j\}$, $\{Dmt_j\}$; $j = \overline{1,10}$.

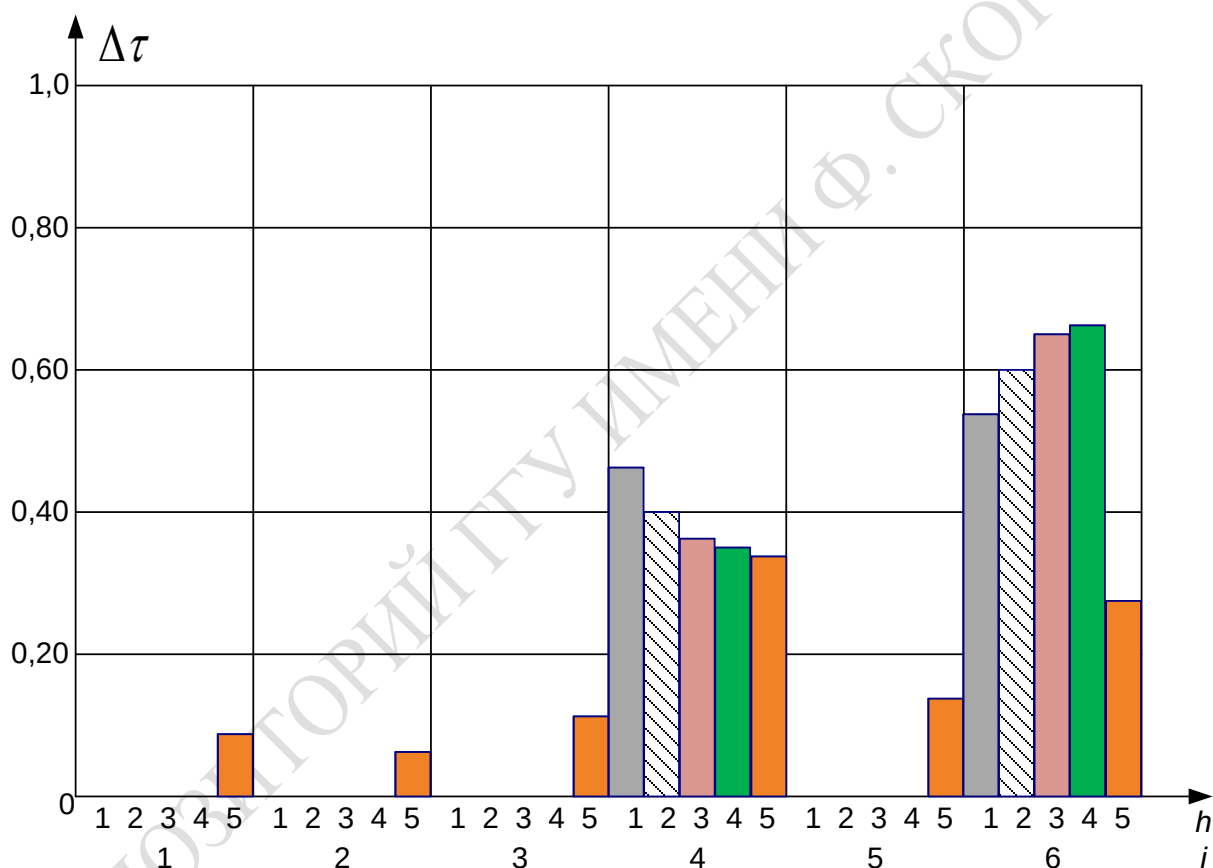


Рисунок 8.1 – Диаграмма превышения времени нахождения технологического цикла в i -х состояниях из-за отказов оборудования

Анализ нормированных значений откликов, представленных на рисунках 8.1 и 8.2, позволяет разделить состояния технологического процесса на 3 группы по суммарно используемым ресурсам (SC_j , Sko_j , Smt_j) и суммарному времени нахождения в состояниях ($S\tau_i$).

Анализ динамики реализации запросов ресурсов при наличии отказов оборудования в ходе серий экспериментов на модели позволил установить значение следующих статистик имитации:

- коэффициент превышения фактического времени τ_{ij} выполнения $PR.SOST_{ij}$ на значениях контрольного примера составляет $k\tau_z = 0,69$, а растяжение времени выполнения $PR.SOST_{ij}$ из-за отказов оборудования составляет $\gamma_R = 1,44$;
- вероятности использования ресурсов и оборудования составляют соответственно $P_{ispR} = 0,76$ и $P_{ispO} = 0,77$.

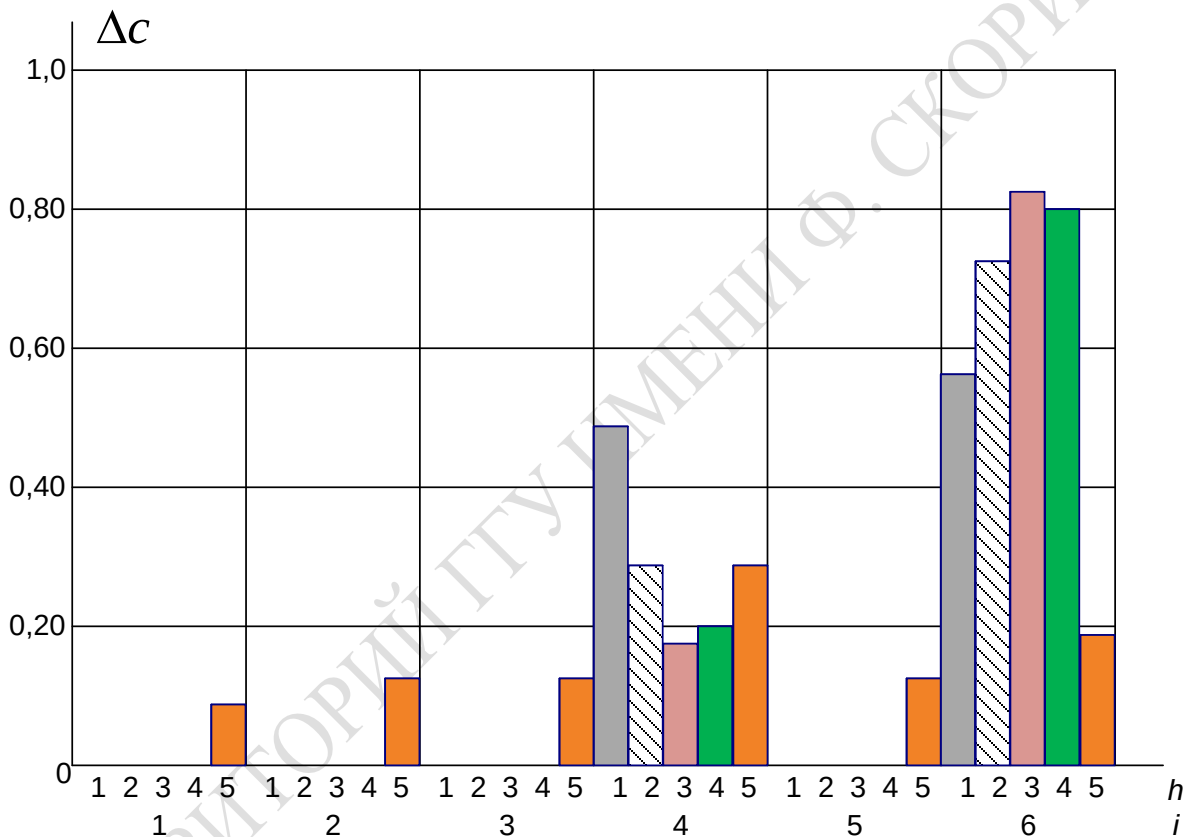


Рисунок 8.2 – Диаграмма превышения стоимости технологических операций в i -х состояниях из-за отказов оборудования

Как видно из рисунка 8.2, более 2/3 состояний технологического цикла используют ресурсы и оборудование. Из-за отказов оборудования и появления аварий время нахождения в состояниях увеличивается в среднем в 1,4 раза. При 100 сменах состояний оборудование отказывало в 7 случаях. Поэтому можно считать, что интегральная вероятность отказа оборудования в среднем составляет $P_{от} = 7/100 = 0,07$.

На основании полученных результатов можно сделать заключение, что основная доля времени и стоимости выполнения технологического

цикла приходится на состояние третьей группы ($i = 4$ и $i = 6$) и равна соответственно 0,535 и 0,477. При этом удельный вес нестандартных состояний по времени и стоимости нахождения в этих состояниях равен соответственно 0,257 и 0,22.

Таким образом, решающую роль в структурной реконфигурации технологического цикла играют надежность характеристики функционирования оборудования, а предложенный подход позволяет оценить параметры выполнения технологических операций, обеспечивающие заданный уровень надежности функционирования технологического цикла.

Предлагается новый подход, связанный с погружением модели объекта исследования в пространство возможных состояний при наличии функциональных зависимостей между операциями для реализации сквозной технологии объектно-ориентированного программирования контура управления технологическим циклом производства.

Разработана методика анализа надежности функционирования вероятностных производственных систем, позволяющая определить параметры выполнения технологических операций, обеспечивающие заданный уровень надежности функционирования технологического цикла. Методика реализует процедуру комплексной оценки вероятностных характеристик выполнения технологических операций с использованием аппарата имитационного моделирования.

Новизна предложенной методики определяется введением пространства состояний параметров надежности технологического цикла, основанного на определении вероятностей надежности технологических операций, и введением пространства функций, позволяющих отобразить отношения и связи между технологическими операциями.

Практическая значимость данного подхода состоит в реализации процедуры оценки вероятностных значений параметров надежности функционирования компонентов производственной системы, обеспечивающей оптимальные варианты реализации структуры технологического цикла в рамках многофункционального критерия качества его выполнения.

8.5 Синтез оптимальной структуры технологического цикла управляемых производственных систем

В последнем пункте настоящей главы рассматривается способ синтеза структуры технологического цикла для управляемых производственных систем на основе конечного множества математических

моделей. В качестве инструмента реализации предлагаемого подхода используются динамические имитационные модели. Приводится теоретическое обоснование и технология применения динамической имитации как способа системного анализа сложных систем для класса управляемых производственных систем с вероятностными характеристиками их функционирования.

Многообразие сложных технологических систем, в ходе реализации которых могут изменяться параметры их функционирования и структура технологического цикла, сложность возникающих практических задач при оценке уровня надежности и безопасности потенциально опасных промышленных объектов в настоящее время требуют разработки новых подходов и технических решений при оптимизации структуры производственных систем на стадии их проектного моделирования. Эффективным средством решения актуальных проблем в подобных случаях является системный анализ объектов исследования, в качестве которых в данной работе рассматриваются управляемые производственные системы (УПС) с вероятностными параметрами их функционирования. При этом под вероятностными параметрами функционирования могут пониматься характеристики надежности функционирования оборудования, задействованного в ходе реализации технологического цикла производства, характеристики выполнения технологических операций, качественные характеристики используемых в процессе производства материалов и комплектующих изделий.

Для управляемых производственных систем как класса сложных систем предлагается гибкий математический аппарат реструктуризации имитационных моделей, позволяющий представить объект исследования в виде конечного множества математических моделей. Подобное представление обеспечивает получение результирующей структуры технологического цикла, которая содержит оптимизированные (в рамках заданного критерия качества функционирования объекта исследования) схемы резервирования технологических операций в процессе имитации.

В качестве инструмента реализации предлагаемого подхода используется динамическое имитационное моделирование, базирующееся на разработке динамических имитационных моделей управляемых производственных систем. Управление производственной системой в контексте данной статьи понимается в смысле упреждения конфликтных ситуаций в процессе функционирования технологического цикла производственной системы.

Новизна предложенного подхода состоит в обеспечении возможности построения интегральной графовой структуры технологического

цикла, что позволяет получить результирующую структуру, содержащую конкретные схемы резервирования технологических операций в зависимости от вероятностных параметров функционирования управляемой производственной системы.

Построенные при этом динамические имитационные модели могут быть использованы в качестве составных компонентов в системах анализа функционирования при автоматизации технологических процессов и производств, а также при разработке и автоматизации систем проектирования новых технологических объектов.

Описание математического аппарата, используемого при реструктуризации имитационных моделей. Возможность отказов оборудования управляемой производственной системы при выполнении агрегатов-имитаторов технологических операций $AMTXO_{ij}$ ставит эксперта-технолога перед необходимостью на стадии проектирования предусмотреть выход из состояний, возникших после аварии. На этот случай в имитационной модели (ИМ) предусматриваются «резервные» цепочки $AMTXO_{ij}$, которые активизируются только при появлении аварий. Переключение на «резервную» ветвь $AMTXO_{ij}$ реализуется за счёт использования булевой матрицы коммутации $\|\gamma_{ns}\|$, формируемой экспертом-технологом до начала имитации. Строками этой матрицы (n) являются номера агрегатов $AMTXO_{ij}$ на входе агрегата-события $ASOB_i$, а столбцами (s) являются номера резервных $AMTXO_{ij}$ на выходе $ASOB_j$, которые необходимо инициировать в поставарийной ситуации. Подобное «технологическое резервирование» является динамическим регулятором поставарийной ситуации в УПС.

Другим способом недопущения аварий оборудования является автоматический переход на резервные устройства, когда хотя бы для одного из устройств оборудования фактическая «наработка» превышает пороговые значения из множества $\{ind_r\}$. Элементы этого множества $\{ind_r\}$ поступают в подсистему $PS.OPEREX$, которая проверяет близость к пороговому значению наработки всех устройств оборудования. При достижении близости к пороговым значениям у нескольких устройств формируются следующие воздействия на ИМ: групповой переход на резервирование тех устройств, у которых наработка близка к критической (α_1); переход на общую профилактику оборудования из-за неэффективности группового резервирования или нехватки резервных устройств (α_2); допускается возможность аварии в тех случаях, когда процесс производства останавливать нельзя, и состояния индикаторов игнорируются (α_3); если оборудование изношено, и общая профилактика будет неэффективна, принимается решение о досрочном завершении

имитации h -го варианта ИМ УПС, поскольку появление аварии для данного технологического процесса недопустимо (α_4).

PS.ANALEX использует статистику имитаций $\{ST_k\}$ и множество откликов модели $\{Y_{oj}\}$. Откликами Y_{oj} являются усреднённые по числу реализаций N их интегральные значения для h -го варианта УПС: критическое время выполнения (T_{KPh}) технологического цикла производства, стоимость его реализации (C_{oh}), интегральный расход материалов и комплектующих изделий (mt_{oh}), количество использованных ресурсов r -го номера ($v_{\Sigma rh}$), суммарная стоимость ликвидации аварий (C_{ABh}), общие потери времени на профилактику (T_{Oph}). Интегральные отклики модели составляют многомерный вектор откликов V_{OH} варианта структуры УПС, у которого все компоненты требуют минимизации, но имеют различную размерность и диапазоны изменения. Поэтому необходима нормировка компонентов этого вектора максимальными их значениями из всех вариантов организации структуры технологического цикла. Для сравнения вариантов структуры осуществляется «свёртка» этого вектора к скалярному показателю W_h способом весовых коэффициентов важности ($\sum_j \delta_j = 1; 0 \leq \delta_j \leq 1$) откликов с номером j . Вариантам организации структуры УПС соответствуют значения вектора параметров $\{X_{ojh}\}$ и постоянных параметров имитации множества $\{G_h\}$. Каждая из составляющих векторов параметров может меняться на различных уровнях. Поэтому общее количество вариантов N_h ($h = \overline{1, K_0}$) определяется стратегией изменения каждого уровня параметров. Выбор оптимальной стратегии осуществляется на основе классических методов планирования экспериментов. Эксперт с помощью воздействия θ_5 иницирует подсистему *PS.ANALEX* и последующий запуск h -го варианта ИМ УПС. Минимальное значение W_h по всему множеству вариантов с номером h и будет решением задачи построения оптимального варианта организации структуры УПС.

Важной статикой реализации имитационной модели является граф критических путей ($GRKRP_h$), который получается после наложения всех реализовавшихся критических путей. С помощью сообщений θ_4 подсистема *PS.ANALEX* выдаёт эксперту графики расхода (в модельном времени t_0) ресурсов r -го типа $Z_{1rh}(t_0)$, финансовых средств $Z_{2rh}(t_0)$, а также временные диаграммы использования оборудования и исполнителей ($DIAGR_{rh}$). Оперативная статистика реализации ИМ в виде сообщения θ_4 предоставляется эксперту-технологу для принятия решений на основе классической теории принятия решений. В результате эксперт формирует набор управляющих воздействий ($\theta_1, \theta_2, \theta_3$ и θ_5), которые затем через подсистему принятия

решений *SPRESH* обеспечивают возможность досрочной остановки l -й реализации имитационной модели, перевод всех устройств оборудования на профилактику, оперативное изменение характеристик надёжности функционирования оборудования.

Оптимизация схем резервирования технологических операций. При оптимизации схем резервирования технологических операций на стадии проектного моделирования эксперт-технолог может использовать следующие стратегии: задание начальных значений множества констант-характеристик процесса $\{H\}$, определяющих состав ресурсов предприятия; выбор алгоритмов модификации структуры технологического цикла при возникновении аварий в ходе реализации технологических операций $AMTXO_{ij}$; выбор оборудования, обладающего необходимой надёжностью функционирования и удовлетворительными временами восстановления работоспособности; изменение ценовой политики, определяющей стоимостные показатели реализации всех технологических операций. При этом варианты организации УПС будут отличаться составом множества $\{X\}$ параметров $AMTXO_{ij}$. Общее число (m) вариантов организации УПС определяется количеством комбинаций уровней элементов множества $\{X\}$.

Откликами имитационной модели УПС являются:

$\{Y_1\}$ – вектор усредненных значений коэффициентов растяжения времени выполнения технологических операций $\{MTXO_{ij}\}$ из-за отказов и ликвидации аварий $\{\bar{p}_{ij}\}$, где $\bar{p}_{ij} = \tau_{vij}/\tau_{ij}$;

$\{Y_2\}$ – вектор стоимостных показателей реализации УПС, состоящий из трех компонентов (реализации структуры при безотказной работе $\{Y_{21}\}$, стоимости восстановительных работ $\{Y_{22}\}$ и стоимости ликвидации аварий $\{Y_{23}\}$);

$\{Y_3\}$ – вектор суточного расхода материалов и комплектующих изделий;

$\{Y_4\}$ – вектор коэффициентов использования ресурсов технологического цикла;

$\{Y_5\}$ – вектор коэффициентов использования места на общих ресурсах;

$\{Y_6\}$ – вектор использования индивидуального оборудования с учетом наличия отказов и восстановлений его работоспособности;

$\{Y_7\}$ – вектор значений времени реализации технологического цикла при безотказной работе оборудования (Y_{71}) и при наличии отказов оборудования (Y_{72}).

Технология выбора управляющего воздействия на производственную систему при возникновении отказов и аварий оборудования

предполагает их предварительную имитацию. Для принятия обоснованного решения, необходимо выполнить следующую последовательность действий: ввод в базу данных модели исходной информации о структуре технологического процесса; задание начальных значений ресурсов согласно плану имитационных экспериментов; составление матрицы переключения резервных агрегатов $AMTXO_{ij}$ для каждого кустового выхода агрегата $ASOB_j$; выбор весовых коэффициентов δ_i откликов моделирования на основе априорной информации и стратегиях управления; составление плана и проведение серий имитационных экспериментов, в ходе которых определяется статистика и вычисляются значения откликов моделирования $\{Y_l\}$; приведение откликов к одному масштабу и типу с последующим расчетом интегрального показателя L_{ks} и формирование матрицы решений; использование одного из классических критериев принятия решений для выбора рационального состава ресурсов технологического цикла.

Процедура реализации динамических имитационных моделей. Построение и использование динамических имитационных моделей управляемых производственных систем реализуется следующей последовательностью этапов.

На *этапе 1* задаётся начальная структура исходной имитационной модели вероятностного процесса с помощью операторов подсистемы формирования модели $PS.FORMSG$.

На *этапе 2* реализуется натурный эксперимент с целью получения исходной информации для последующей проверки адекватности имитационной модели реальному объекту имитации. В случаях, когда не удаётся найти аналитический вид функций распределения, используется табличная форма их представления.

На *этапе 3* осуществляется верификация базового варианта имитационной модели. Задаётся начальный состав ресурсов, материалов, исполнителей и оборудования. Далее формируются управляющие таблицы-справочники и таблицы базы данных для хранения статистики имитации, используемые в дальнейшем при организации имитационного эксперимента.

На *этапе 4* реализуются функции испытания и анализа свойств имитационной модели, представляющую собой многошаговую процедуру использования стандартных методик испытания сложных систем, в процессе которой реализуются типовые этапы испытания имитационных моделей.

На *этапе 5* осуществляется контроль реализации имитационной модели в пошаговом режиме через случайные интервалы времени (до следующего события) с одновременной проверкой текущего состояния

управляемой производственной системы на соответствие базовому варианту модели.

При изменении текущей структуры производственной системы производится запись реализовавшейся части технологического цикла в базу данных модели. Последнее свершившееся событие фиксируется в качестве начального для дальнейшего продолжения имитации, после чего выполняется возврат на *этап 1*.

По окончании имитации в базе данных модели завершается процесс формирования оптимизированной структуры технологического цикла, построенной в режиме модельного времени в соответствии с выбранным критерием качества функционирования объекта имитации.

Предлагается способ синтеза структуры технологического цикла управляемых производственных систем на основе конечного множества математических моделей на стадии их проектного моделирования. При реализации данного подхода предложено использовать динамические имитационные модели. Приведены теоретическое обоснование и технология применения динамической имитации как способа системного анализа сложных систем для класса управляемых производственных систем с вероятностными характеристиками их функционирования.

Следует отметить, что новизна предложенного подхода состоит в предоставлении способа построения интегральной графовой структуры технологического цикла, который позволяет получить результирующую структуру, содержащую оптимизированные схемы резервирования технологических операций в рамках выбранного критерия функционирования объекта исследования.

Изложенный способ построения динамических имитационных моделей опирается на применение агрегатной системы автоматизации имитационного моделирования и ориентирован на случаи, когда динамику функционирования имитационных моделей управляемых производственных систем можно описать на уровне элементов управления со сложной логикой с использованием процедур метода Монте-Карло.

В основу формализации управляемых производственных систем положены вероятностные графовые структуры. Полученные результаты [28] могут быть использованы в качестве составных компонентов в системах анализа функционирования при автоматизации технологических процессов и производств, а также при разработке и автоматизации систем проектирования новых технологических объектов.

ЗАКЛЮЧЕНИЕ

В настоящей монографии «Разработка имитационных моделей сложных технических систем» изложен новый подход к унификации методов системных исследований в области проектного моделирования сложных технических систем с вероятностными параметрами их функционирования. В качестве инструмента реализации применяется динамическое имитационное моделирование, базирующееся на разработке динамических имитационных моделей сложных технических систем, в основу которых положен системный анализ.

Для построения компонентов динамических имитационных моделей объектов исследований используется система автоматизации имитационного моделирования, реализующая агрегатный способ имитации сложных технических систем, а также способ их формализации на основе построения графовых моделей, отражающих структуру объекта исследования. Система контроля имитации обеспечивает возможность принятия управляющих воздействий на основе текущей операционной обстановки в процессе функционирования объекта исследования.

При исследовании технологических объектов как класса сложных технических систем предложен гибкий математический аппарат реструктуризации имитационных моделей, позволяющий представить объект исследования в виде конечного множества математических моделей. Подобное представление позволяет осуществлять оптимизацию структуры цикла управления в процессе функционирования системы.

В данной работе изложена также суть метода пошаговой реструктуризации имитационных моделей, который составляет основу построения оптимальной структуры технологических процессов с последовательной и параллельно-последовательной организацией технологического цикла в рамках выбранного критерия качества. Новизна представленного подхода состоит в обеспечении возможности построения интегральной графовой структуры технологического цикла, что позволяет получить результирующую структуру, содержащую конкретные схемы резервирования технологических операций в зависимости от вероятностных параметров функционирования управляемой производственной системы.

Построенные при этом динамические имитационные модели могут быть использованы в качестве составных компонентов в системах анализа функционирования при автоматизации технологических процессов и производств, а также при разработке и автоматизации систем проектирования новых технологических объектов.

ЛИТЕРАТУРА

1. Асатурян, В. И. Теория планирования эксперимента / В. И. Асатурян. – М.: Радио и связь, 1983. – 123 с.
2. Боровиков, В. П. STATISTIKA. Искусство анализа данных на компьютере: для проектировщиков / В. Д. Боровиков, 2-е изд. СПб.: Питер, 2003. – 688 с.
3. Бусленко, Н. П. Лекции по теории сложных систем / Н. П. Бусленко, В. В. Калашников, И. Н. Коваленко. – М.: Сов. Радио, 1973. – 439 с.
4. Быченко, О. В. Метод, средства и технология адаптации рабочей нагрузки к составу ресурсов узла ЛВС / О. В. Быченко // Дисс. к.т.н., спец. 05.13.13. УО «Гомельский государственный университет им. Ф. Скорины», 2006. – Гомель. – 232 с.
5. Воруев, А. В. Мониторинг и адаптация вычислительного процесса в узлах локальной сети с использованием рабочей нагрузки / А. В. Воруев // Дисс. к.т.н., спец. 05.13.13. УО «Гомельский государственный университет имени Ф. Скорины», 2002. – Гомель. – 250 с.
6. Гмурман, В. Б. Теория вероятностей и математическая статистика: учеб. пособие для вузов / В. Б. Гмурман. – М.: Высш. шк. 1998. – 479 с.
7. Демиденко, О. М. Методы и средства исследования и адаптации вычислительного процесса под рабочую нагрузку на локальную вычислительную сеть / О. М. Демиденко // Дисс. к.т.н., спец. 05.13.13. УО «Гомельский государственный университет имени Ф. Скорины», 2003. – Гомель. – 534 с.
8. Демиденко, О. М. Технология мониторинга и адаптации вычислительного процесса под рабочую нагрузку на локальную вычислительную сеть / О. М. Демиденко. – Мн.: Белорусская наука, 2002. – 193 с.
9. Демиденко О. М. Проектное моделирование вычислительного процесса в локальных вычислительных сетях / О. М. Демиденко, И. В. Максимей. – Мн.: Белорусская наука, 2001. – 252 с.
10. Демиденко, О. М. Методика и средства выбора рациональных вариантов организации вычислительного процесса для заданной рабочей нагрузки на локальную вычислительную сеть / О. М. Демиденко [и др.] // Веснік Гродненскага ўніверсітэта імя Я. Купалы. – Гродно, 2002. – № 2. – С. 113–120.
11. Имитационное моделирование систем / под. ред. А. А. Вавилова. – М.: Машиностроение; Берлин. Техника, 1983. – 416 с.
12. Жогаль, С.И. Задачи и модели исследования операций. Аналитические модели исследования операций: учеб. пособие. Ч. 1 /

С. И. Жогаль, И. В. Максимей. – Гомель: БелГУТ, 1994. – 109 с.

13. Зайченко, Ю. П. Анализ и синтез структуры глобальных вычислительных сетей / Ю. П. Зайченко. – К.: ЗАО «Спецмонтаж проект», 1998. – 108 с.

14. Левчук В. Д. Программно-технологические комплексы имитации сложных дискретных систем : монография / В. Д. Левчук, И. В. Максимей. – Гомель, 2006. – 263 с.

15. Максимей, И. В. Имитационное моделирование на ЭВМ» / И. В. Максимей. – М.: Радио и связь, 1988. – 232 с.

16. Левчук, В. Д. Базовая схема формализации системы моделирования MICIS 4 / В. Д. Левчук // Проблемы программирования. – Киев. – 2005, № 1. – С. 85–96.

17. Максимей, И. В. Задачи и модель исследования операций. Методы нелинейного и стохастического программирования : учеб. пособие. Ч. 2. / И. В. Максимей, В. С. Серегина. – Гомель, БелГУТ, 1999. – 103 с.

18. Максимей, И. В. Задачи и модели исследования операций. Технология имитации на ЭВМ и принятие решений : учеб. пособие. Ч. 3 / И. В. Максимей [и др.]. – Гомель: БелГУТ, 1999. – 150 с.

19. Максимей, И. В. Языковые и технологические возможности МК АСИМ для построения имитационных моделей больших систем / И. В. Максимей, Ю. А. Семишин // Моделирование дискретных управляющих и вычислительных систем. Тр. IV Всесоюзный семинар. – Свердловск, 1984. – С. 61–63.

20. Смородин, В. С. Метод динамической имитации вероятностных производственных систем / В. С. Смородин // Математичні машини і системи (Mathematical Machines and Systems). – 2012. – № 2. – С. 96–101.

21. Никишаев, В. А. Рациональный выбор организации узла ЛВС с помощью средств мониторинга и имитационного моделирования вычислительного процесса и рабочей нагрузки / В. А. Никишаев. Дисс. к.т.н. сек. 05.13.13. УО «Гомельский государственный университет имени Ф. Скорины». 2004. – Гомель. – 285 с.

22. Поляк, Ю. Г. Вероятностное моделирование на электронных вычислительных машинах / Ю. Г. Поляк. – М.: Сов. Радио, 1971. – 399 с.

23. Попова, Е. О. Имитационная модель технологического процесса дискретного производства / Е. О. Попова // Известия ГГУ им. Ф. Скорины. – 2003, № 3. – С. 59–63.

24. Потрашкова, М. В. Анализ эффективности функционирования локальной вычислительной сети при заданной рабочей нагрузки на

основе имитационной модели вычислительного процесса / М. В. По-трашкова // Дисс. к.т.н. спец. 05.13.13. УО «Гомельский государственный университет имени Ф. Скорины». 2004. – Гомель. – 250 с.

25. Принявичюс, Г. Модели и методы исследования вычислительных систем / Г. Принявичюс. – Вильнюс: Моклас, 1982. – 227 с.

26. Смородин, В. С. Методы и средства имитационного моделирования технологических процессов производства : монография / В. С. Смородин, И. В. Максимей. – Гомель, 2007. – 360 с.

27. Смородин, В. С. Проектное моделирование управляемых производственных систем с резервированием схем управления / В. С. Смородин, А. В. Клименко // Известия Гомельского гос. ун-та им. Ф. Скорины. – 2014. – № 3(84). – С. 150–156.

28. Смородин, В. С. Имитационное моделирование и средства оптимизации сложных технических систем / В. С. Смородин, А. В. Клименко // Математичні машини і системи (Mathematical Machines and Systems). – 2014. – № 4. – С. 98–109.

29. Хорошевский, В. Г. Инженерный анализ вычислительных машин и систем / В. Г. Хорошевский. – М.: Радио и связь, 1987. – 258 с.

30. Шеннон, Р. Имитационное моделирование систем / Р. Шеннон. – Искусство и наука. – М.: Мир, 1971. – 417 с.

31. Шевченко, Д. Н. Теория вероятностей и математической статистики : учеб. пособие / Д. Н. Шевченко. – Гомель: УО «БелГУТа», 2006. – 318 с.

32. Шумаков, Г. В. Delphi 7. Руководство разработчика БД / Г. В. Шумаков, В. В. Фаронов. – М. 2001. – 350 с.

33. Шрайбер, Т. Дж. Моделирование на GPSS / Т. Дж. Шрайбер. – М.: Машиностроение, 1980. – 592 с.

34. Якимов, А. Н. Имитационное моделирование в ERP-системах управления / А. Н. Якимов, С. И. Альховик. – Мн.: Белорусская наука, 2005. – С. 97.

ОГЛАВЛЕНИЕ

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ.....	3
ВВЕДЕНИЕ	4
1. ЭТАПЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ.....	6
1.1. Использование имитационных моделей при проектировании СС	6
1.2. Этапы разработки программы имитации функций СС.....	9
1.3. Этапы испытания имитационных моделей СС.....	15
1.4. Этапы эксплуатации программ ИМ СС.....	18
1.5. Формулировка рекомендаций по результатам имитации СС.....	19
1.6. Итоги.	20
2. СОСТАВЛЕНИЕ СОДЕРЖАТЕЛЬНОГО ОПИСАНИЯ ОБЪЕКТА ИМИТАЦИИ.....	22
2.1. Методика составления содержательного описания СС.....	22
2.2. Описание вычислительного процесса в локальных вычислительных сетях.....	24
2.3. Содержательное описание поточного производства изготовления и ремонта изделий.....	36
2.4. Содержательное описание вероятностных технологических процессов производства.....	53
2.5. Итоги.....	70
3. РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ОБЪЕКТА МОДЕЛИРОВАНИЯ.....	72
3.1. Методика составления концептуальной модели СС.....	72
3.2. Примеры составления концептуальных моделей СС.....	83
3.3. Итоги.....	93
4. ФОРМАЛИЗАЦИЯ СЛОЖНОЙ СИСТЕМЫ.....	95
4.1. Методика формализации объекта моделирования.....	95
4.2. Формализация вычислительного процесса в ЛВС оператором полумарковских процессов.....	102
4.3. Формализация ВП в узле ЛВС с помощью информационной подкраски запросов ресурсов пользователей.....	116
4.4. Формализация технологии поточного производства.....	123
4.5. Формализация вероятностных технологических процессов производства.....	136
4.6. Итоги.....	148
5. ПРЕОБРАЗОВАНИЕ ФОРМАЛЬНОГО ОПИСАНИЯ СЛОЖНОЙ СИСТЕМЫ В ИМИТАЦИОННУЮ МОДЕЛЬ.....	150
5.1. Методика преобразования формального описания СС в имитационную модель.....	150
5.2. Представление имитационной модели вычислительного процесса в ЛВС аппаратом полумарковских процессов.....	160

5.3. Построение имитационной модели вычислительного процесса в узле ЛВС на основе информационной подкраски запросов ресурсов системы пользователями.....	164
5.4. Построение ИМ технологии поточного производства.....	170
5.5. Преобразование формального описания в имитационную модель вероятностного технологического процесса производства.....	187
5.6. Итоги.....	205
6. ПРОГРАММИРОВАНИЕ И ОТЛАДКА ИМИТАЦИОННОЙ МОДЕЛИ СЛОЖНОЙ СИСТЕМЫ.....	206
6.1. Методика программирования и отладки имитационной модели сложной системы.....	206
6.2. Использование программно-технологических комплексов имитации при отладке ИМ сложных систем.....	216
6.3. Технологические возможности отладки ИМ агрегатного типа в среде системы моделирования САИМ.....	222
6.4. Итоги.....	225
7. ИСПЫТАНИЕ ИМИТАЦИОННЫХ МОДЕЛЕЙ СЛОЖНЫХ СИСТЕМ.....	226
7.1. Методика испытания имитационных моделей СС.....	226
7.2. Верификация имитационной модели.....	237
7.3. Определение технологических характеристик имитационных моделей СС.....	243
7.4. Оценка чувствительности откликов имитационной модели СС	247
7.5. Проверка адекватности ИМ прототипу сложной системы.....	249
7.6. Итоги.....	256
8. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И СРЕДСТВА ОПТИМИЗАЦИИ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ.....	258
8.1. Метод пошаговой реструктуризации имитационных моделей...	258
8.2. Динамическая имитация вероятностных технологических процессов производства.....	261
8.3. Система оперативного контроля имитации управляемых технологических процессов.....	269
8.4. Методика анализа надежности функционирования вероятностных производственных систем.....	276
8.5. Синтез оптимальной структуры технологического цикла управляемых производственных систем.....	285
ЗАКЛЮЧЕНИЕ.....	292
ЛИТЕРАТУРА.....	293

Научное издание

Максимей Иван Васильевич ,

Сморозин Виктор Сергеевич,

Демиденко Олег Михайлович

РАЗРАБОТКА ИМИТАЦИОННЫХ МОДЕЛЕЙ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

Подписано в печать 29.12.2014. Формат 60x84 1/16.

Бумага офсетная. Ризография. Усл. печ. л. 17,4.

Уч.-изд. л. 19,1. Тираж 100 экз. Заказ 719.

Издатель и полиграфическое исполнение:
учреждение образования

«Гомельский государственный университет
имени Франциска Скорины».

Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/87 от 18.11.2013.

Специальное разрешение (лицензия) № 02330 / 450 от 18.12.2013.

Ул. Советская, 104, 246019, Гомель.

