Содержание

Введение

Актуальность темы исследования. В рамках Четвертой промышленной революции или по-другому - Индустрии 4.0 в мире происходит постепенное внедрение киберфизических систем в жизнь человека. Данная тенденция ведет к растущей потребности в данных системах и их состовляющих: физической (датчики) и программной (вычислительные системы). Это обуславливает создание интеллектуальных вычислительных систем, которые смогут обеспечить большую точность и скорость планирования деятельности производств, что позволит сократить потери производственных мощностей и ресурсов, а также ускорит актуализацию планов по результатам деятельности предприятия. Из всего вышесказанного следует необходимость в разработке компонент для данной системы, которые будут решать поставленные задачи, а также, учитывая модульную архитектуру системы, могут быть в будущем использованы в последующих проектах.

сбросить нумерацию содержания?

Цель работы. Целью данной работы является разработка и тестирование компонент системы планирования производства, для чего были поставлены следующие задачи:

— разработка модели ресурса сборочной линии;

— разработка модуля отображения логического времени на физическое;

— организация консистентного хранилища данных;

— тестирование полученных модулей и верификация структуры базы данных.

Теоретическая и практическая значимость. Результаты данного исследования могут быть применены:

— при разработке программных комплексов управления предприятиями;

— при разработке имитационных моделей производства;

— при разработке структуры консистентного хранилища данных.

Методы исследования. Для достижения поставленных задач используются следующие методы:

— анализ сборочной линии, синтез модели сборочной линии и моделирование в подсистеме имитационного моделирования;

— анализ логического и физического времён, их различий и синтез соответствующего алгоритма отображения логического времени на физическое;

— анализ данных, которые необходимо хранить в базе данных, синтез структуры базы данных, формализация и обоснование предложенной структуры.

Апробация результатов исследования. Результаты исследования используются в основной составляющей интегрируемого программного комплекса интеллектуальной системы управления предприятием (ИС-системе планирования и прогнозирования (СПП). Результаты исследований по отображению логического времени на физическое были представлены в статье для Конгресса молодых ученых (2019) “Задача отображения временных промежутков на рабочий календарь”

или производства?

fix subsections

1 Обзорная часть

rename?

Написать переход

1.1 Индустрия 4.0

Индустрия 4.0 - прогнозируемое массовое внедрение киберфизических систем в промышленность и повседневный быт человека. Под киберфизической системой (CPS - cyber-physical system) подразумевается взаимодействие цифровых, аналоговых, физических и человеческих компонентов, разработанных для функционирования посредством интегрированной физики и логики или другими словами: киберфизические системы - это интеллектуальные системы, которые состоят из тесно взаимосвязанных сетей физических и вычислительных компонентов [1].

Появилось понятие индустрии 4.0 во время ганноверской выставки 2011 года как обозначение стратегического плана развития и поддержания конкурентоспособности немецкой экономики, предусматривающий совершение прорыва в области информационных технологий для промышленности. Также считается, что данное направление знаменует собой четвертую промышленную революцию [2].

Если рассматривать производство, на что и нацелена индустрия 4.0, то, используя определение данное выше, процесс внедрения киберфизической системы разделяется на внедрение физической части (например датчики, собирающие и передающие данные посредством различных сетей) и вычислительную - систему планирования производства.

Система планирования производства (СПП) обеспечивает расчет объемно-календарного и оперативного планов, автоматизированный подбор поставщиков, автоматизированный перерасчет планов по фактическим результатам деятельности, направленный на минимизацию временных и финансовых потерь.

Объемно-календарный план - задание для каждой производственной площадки на заданный интервал времени, представляющее собой план-график, на котором каждому интервалу соответствует но-

менклатура и объем подлежащих к производству изделий [3].

Оперативный план - план, согласно которому выполняется привязка каждой операции для каждой единицы продукции к временным интервалам, конкретному работнику и конкретным производственным средствам [3].

С некоторыми оговорками можно сказать, что оперативный план является более подробной версией объемно-календарного плана, которая, в силу ограничений со стороны вычислительной сложности расчета, составляется на срок в порядок меньший, чем для объемно-календарного плана.

1.2 Обзор существующих решений

Сегодня, чтобы сохранять конкурентоспособность, предприятиям необходимо развивать и внедрять системы управления и планирования производственных процессов. Актуальным направлением, ориентированным на решение этой задачи, является разработка СПП, которая должна обеспечивать решение следующих задач: исполнения планов производства и целевых показателей, оптимизации операционной деятельности, снижения затрат и повышения эффективности производства.

Производственное планирование — это систематический, структурированный, направленный на достижение поставленной задачи процесс планирования промышленного предприятия, состоящий из отдельных, иерархически распределенных этапов, осуществляемый с помощью специальных методов и инструментов, начиная с момента появления замысла и заканчивая запуском производства [4]. Производственное планирование может также включать в себя корректирующие мероприятия в процессе эксплуатации. Планирование промышленного предприятия может основываться на различных целях и задачах, охватывать самые разные проектные ситуации.

В процессе планирования используется ряд инструментов и программных компонент, которые поддерживают реализацию метода планирования [5]. Говоря в общем, это программные средства (прикладные

программы), которые применяются пользователями на персональных компьютерах в различных фазах планирования жизненного цикла промышленного предприятия. К основным инструментам планирования относятся: APS/SCM; CAP; АСУП. Ниже даны подробные описания данных типов инструментов [6].

APS/SCM (системы синхронного планирования, системы управления логической цепочкой) поддерживают расчеты, эксплуатацию и оптимизацию логистических цепочек. Особые свойства логистической цепочки получаются благодаря взаимодействию участников. Важная роль отводится структуре логистических цепочек, соответствующих рынку, а также координации и интеграции всех индивидуальных действий.

CAP (система автоматизированного регулирования) характеризует область компьютеризованного планирования работы. При этом используется система электронной обработки данных для создания рабочего графика, выбора эксплуатационных средств, создания указаний по изготовлению и монтажу, а также программирования для станков с ЧПУ.

АСУП (CAM — автоматическая система управления производством) включает в себя компьютерное техническое управление и контроль над производственными линиями и эксплуатационными средствами при проведении производства, т.е. прямое управление обрабатывающими и перерабатывающими машинами, устройствами манипуляции, транспортировки, перегрузки и хранения, а кратко — техническое управление всеми устройствами потоковых систем [7].

Наиболее распространенными программными продуктами, предназначенными для планирования производственных процессов являются «1С:Предприятие» и «SAP R/3»: первый является наиболее распространенным на российском рынке, второй, в свою очередь, широко используется за рубежом. На примере этих, зарекомендовавших себя с положительной стороны продуктов, проведем анализ системных компонент и сравним их место в архитектуре систем.

1.2.1 «1С: Предприятие 8.0»

Комплекс программ «1С: Предприятие 8.0» состоит из технологической платформы и прикладных компонентов, которые создаются на её основе и предназначены для автоматизации деятельности предприятий. Технологическая платформа не является готовым программным продуктом, предназначенным для внедрения на предприятие, вместо нее обычно применяют несколько компонентов, разработанных на её базе. Данное решение делает возможным автоматизировать различные виды деятельности, применяя единую основную технологическую платформу.

Система «1С: Предприятие 8.0» использует следующие основные компоненты:

— «Управление торговлей»;

— «Управление персоналом»;

— «Управление производственным предприятием»;

— «Управление складом»;

— «Управленческий учет и расчет себестоимости».

Наиболее интересными для рассмотрения являются компоненты «Управление персоналом» и «Управление производственным предприятием», поскольку их реализация является ключевой с точки зрения планирования деятельности предприятия и не имеет сегодня строгой математической формализации.

Компонент «1С: Предприятие 8.0. Управление персоналом» позволяет эффективно управлять кадровыми процессами в следующих областях: планирование потребностей в персонале; обеспечение организации новыми кадрами; эффективное планирование занятости персонала; кадровый учет и анализ персонала; управление персоналом.

Компонент «Управление производственным предприятием» предназначен для автоматизации процессов управления и учета на производственном предприятии. Он позволяет создать единую информационную систему для управления различными сторонами деятельности предпри-

ятия.

В платформе «1С: Предприятие 8.0» заложен ряд подходов, которые формируют основную концепцию разработки типовых компонентов. Эти подходы предназначены для максимального сближения технологических возможностей с бизнес-процессами разработки и интеграции прикладных решений. Важными моментами, которые следует отметить, являются: изоляция разработчика от технологических деталей, алгоритмическое программирование конкретной бизнес-логики приложения, использование собственной модели базы данных и гибкость прикладных решений без их доработки.

Механизм обмена данными, используемый в технологической платформе «1С: Предприятие 8.0», позволяет создавать территориально распределенные информационные системы на основе баз данных «1С: Предприятия 8.0», и использовать другие информационных систем, не относящиеся к «1С: Предприятии 8.0». Например, можно организовать работу главного офиса, филиалов и складов предприятия в одной базе данных, или обеспечить взаимодействие базы данных «1С: Предприятия 8.0» с существующей базой данных «Oracle».

Технологическая платформа «1С: Предприятие 8.0» предоставляет средства разработки, с помощью которых создаются новые или модифицируют существующие прикладные решения. Этот инструмент разработки называется «конфигуратор». Благодаря тому, что он поставляется со стандартным пакетом «1С: Предприятия 8.0», то пользователь может свободно разработать или модифицировать прикладное решение (адаптировать его под себя), возможно, с привлечением сторонних специалистов.

Среди преимуществ данной системы можно выделить:

— открытость системы;

— регулярные программные обновления;

— широкие функциональные возможности системы.

Однако необходимо отметить, что подходы «1С: Предприятия» ориентированы на решение проблем автоматизации бухгалтерского

и организационного управления предприятием. Использование проблемно-ориентированных объектов позволяет разработчику решать задачи складского, бухгалтерского, управленческого учета, расчетам заработной платы, анализа данных и управлению бизнес-процессами. Однако области экономического и бухгалтерское учета характеризуются высокой степенью математического формализации и их реализация происходит с относительно малыми трудозатратами, тогда как компоненты планирования и производственного расписания сегодня являются актуальными направлением для исследований и прикладной разработки.

1.2.2 «SAP R/3»

Система «SAP R/3» предоставляет собой набор разноплановых инструментов, направленных на повышение эффективности производственного процесса, увеличение экономической стабильности, автоматизацию процессов планирования. Она дает возможность интегрировать инновационные подходы централизованного планирования и управления, и повысить качество управления на разных организационных уровнях предприятий.

«SAP R/3» использует модульную архитектуру, где каждый отдельный модуль предназначен, для решения специализированной задачи процесса предприятия, взаимодействие между ними происходит в режиме реального времени. Наибольший интерес для рассмотрения представляют компоненты, не имеющие прямого отношение к бухгалтерской и экономической деятельности предприятия - модуль PP; модуль HR; модуль BC.

Модуль PP (планирование производства) дает возможность организовать управление и планирование производства предприятия. Он реализует следующие функции: формирование спецификаций, создание технологических карт, управление производственными площадками, планирование сбыта, планирование потребности в материалах, управление производственными заказами, планирование затрат на изготовление изделие, учет затрат производственных процессов, планирование

производственной деятельности, управление серийным производством, планирование автоматизированного производства.

Модуль HR (управление персоналом) решает задачи планирования и управления работой персонала. Ключевые элементы: администрирование персонала, расчет данных для вычисления заработной платы, сбор и анализ данных о рабочем времени, учет командировочных расходов, создание информационной модели внутренней структуры компании.

Модуль BC (базовый модуль) предназначен для интеграции в систему «SAP R/3» всех отдельных прикладных модулей и обеспечивает независимость от аппаратной платформы. Модуль BC позволяет организовать работу с многоуровневой распределенной архитектуре клиент-сервер. Система «SAP R/3» работает на серверах UNIX, AS/400, Windows NT, S/390 и с различными СУБД (Informix, Oracle, Microsoft SQL Server, DB2). [8].

На данный момент система «SAP R/3» является наиболее распространенной системой управления предприятием. Благодаря тому, что она является модульной системой, ее можно настроить в соответствии с конкретными потребностями отдельного предприятия. Степень технического уровня системы определяется возможностью ее перенастройки без необходимости переписывать программный код. Эта опция «SAP R/3» также позволяет занимать ведущее место в мире в системе управления.

С помощью инструментов управления, включенных в систему «SAP R/3», можно реализовывать задачи мониторинга и анализа, без дополнительного программирования, для чего система предлагает следующие способы:

— мониторинг БД;

— мониторинг операционной системы сервера;

— мониторинг коммуникаций;

— мониторинг и управление сервером приложений:

— формирование и запуск новой конфигурация ядра R/3;

— снятие и редактирование текущей конфигурации ядра R/3;

— формирование временного графика в зависимости от нагрузки (например, в ночное время можно увеличивать количество процессов, отвечающих за фоновые задания);

— управление системой архивирования;

— управление текущими пользователями, процессами.

Многоуровневая клиент-серверная архитектура позволяет разделять задачи управления данными, ориентированные на нужды пользователя. В версии 3.0 системы «SAP R/3» SAP AG были расширены возможности решения для организации взаимодействия с другими приложениями и распределения операций «SAP R/3» в масштабируемой компьютерной структуре. Технология внедрения «SAP R/3» основана на многоуровневой архитектуре с использованием программного обеспечения среднего уровня. Между тем, промежуточное ПО отделяет пользовательские приложения от аппаратного и программного обеспечения, с другой стороны, решает проблему взаимодействия программных приложений и аппаратного обеспечения. В «SAP R/3» SAP Basis функционирует как промежуточное программное обеспечение [8].

Основные данные — это информация, которая хранится в базе данных, достаточно долгий промежуток времени. К ним относятся такие данные как: информация о кредиторах, поставщиках, материалах и счетах. Основные данные создаются централизованно и доступны для всех приложений. Например, они включают данные клиента, которые используются в заявках, поставках, для выставления счетов и платежей и так далее. Основные данные клиента могут быть присвоены следующим организационным единицам: балансовая единица, сбытовая организация, канал сбыта, сектор.

Основная запись материала является центральным объектом данных системы «SAP R/3». Она включает в себя: сырье; оборудование; расходные материалы; полуфабрикаты; продукты; вспомогательное

производственное оборудование и инструменты. Она является главным источником данных предприятия и используется всеми компонентами логистической системы SAP. Благодаря объединению всех материалов в единый объект базы данных устраняются проблемы избыточности данных. Сохранённые данные могут использоваться во всех областях, такими как закупки, контроль запасов, планирование потребностей в материалах, проверка счетов.

Данные, хранящиеся в основной записи материала, необходимы логистическому модулю системы для решения следующих задач:

— обработки запасов на поставку;

— обновления движения материалов и инвентаризационной обработки;

— проводки счетов;

— обработки клиентских заказов;

— планирования потребностей и календарного планирования.

Структурная логика поставщика и клиента также применяется к основной записи материала. При оформлении заказа для клиентов необходимо учитывать: согласование о перевозке, условия доставки, оплаты и т.д. Данные, необходимые для таких операций, дублируются из основной записи делового партнера, чтобы исключить необходимость повторного ввода информации о каждой транзакции. В основной записи материала могут одновременно храниться данные, обработанные во время ввода заказа, например, цена за единицу цены товара, запасы на другом складе и т.д. Этот принцип полезен для обработки данных в каждой основной записи, связанной с выполнением операции.

Для каждой транзакции необходимо присваивать соответствующую организационную единицу. Присвоение структуры предприятия генерируется в дополнение к данным, доступным по данным клиента и по данным материала. Поэтому документ, созданный при помощи транзакции, содержит все основные данные из организационных еди-

ниц [9].

Среди достоинств данной системы можно выделить:

— прозрачность деятельность предприятия;

— повышение оборотов товарно-материальных запасов;

— сокращение персонала управления;

— единые стандарты управления.

К недостаткам относятся:

— требуется высокий уровень подготовки персонала;

— сложность интеграции;

— большие финансовые вложения.

2 Система планирования производства

2.1 Архитектура системы планирования производства

Система планирования производства представляет из себя набор программных модулей, взаимодействующих согласно схеме (рисунок 2.1).

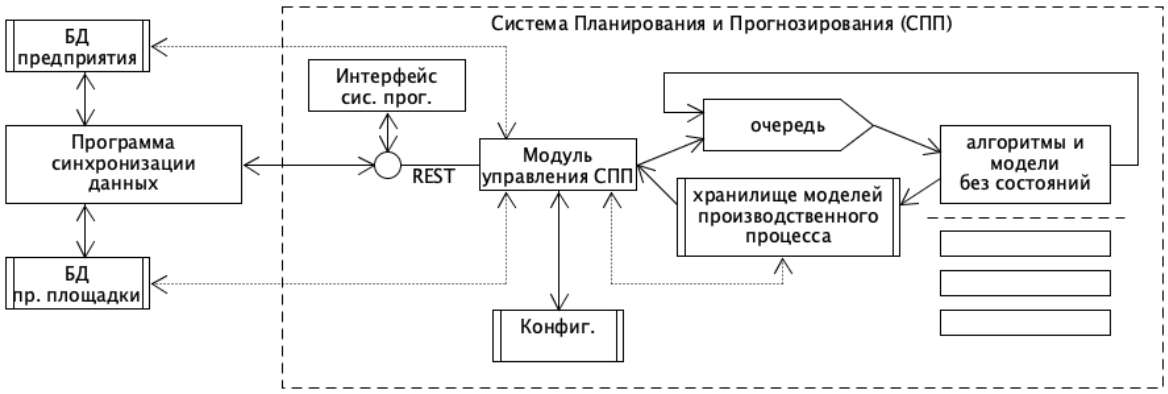

Рисунок 2.1 — Схема системы планирования производства [10]

Так как для выбора оптимального плана сотруднику требуется несколько планов, непосредственно из которых нужно будет оптимальный, система производит запуск множества параллельных расчетов, каждый из которых отличается конфигурацией смен либо количества доступных ресурсов. За запуск и синхронизацию отвечает “Модуль управления СПП” (рисунок 2.1), который создает очередь расчетов на запуск. Затем пул потоков извлекает их в порядке очереди и начинает работу ядра имитационного моделирования (“Алгоритмы и модели без состояний”, рисунок 2.1). После окончания работы полученный результат передается обратно в модуль управления для возврата пользователю посредством RESTful API (REST - архитектурный стиль взаимодействия компонентов распределённого приложения в сети). Пользователь, зная с какими параметрами запускался полученный расчет, может поменять конфигурацию и отправить его на повторное вычисление, что будет сохранено в базе данных предприятия и приведет к запуску всего цикла с начала.

2.2 Подсистема имитационного моделирования

Подсистема имитационного моделирования (ядро имитационного моделирования) - модуль, производящий моделирование производственных процессов для получения приблизительной оценки времени выполнения набора операций (например карты технологического процесса).

Карта технологического процесса - документ, предназначенный для операционного описания технологического процесса изготовления или ремонта изделия (составных частей изделия) в технологической последовательности по всем операциям одного вида формообразования, обработки, сборки или ремонта с указанием переходов, технологических режимов и данных о средствах технологического оснащения, материальных и трудовых затратах.

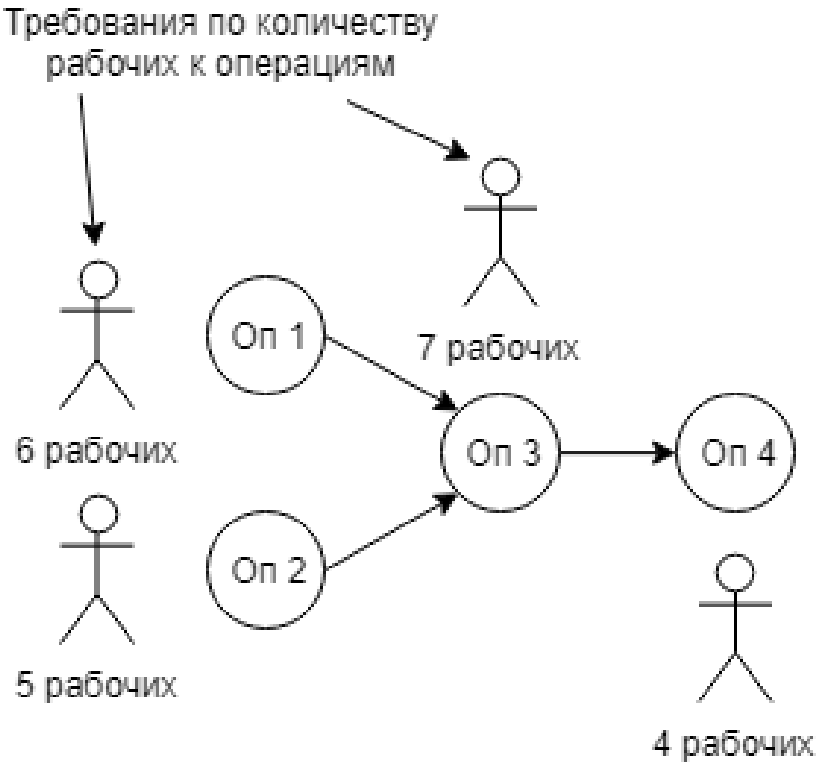

Рисунок 2.2 — Визуальное представление технологической карты с ограничением по количеству рабочих на операцию

Для определения оценки времени модуль рассматривает карту технологического процесса как систему уравнений, в которой неизвестными являются времена начала и окончания выполнения операций:

$$\begin{cases} op1_2 = op1_1 + dur1 \\ op2_2 = op2_1 + dur2 \\ op3_2 = op3_1 + dur3 \\ op4_2 = op4_1 + dur4 \\ op1_2 \leqslant op3_1 \\ op2_2 \leqslant op3_1 \\ op3_2 \leqslant op4_1 \end{cases} \tag{2.1}$$

Здесь операция обозначается двумя числами: отметкой начала и конца, которые обозначаются индексами 1 и 2 соответственно. Первые четыре уравнения задают расчет отметки окончания операции, другие три - накладывают ограничения на последовательность следования операций друг за другом.

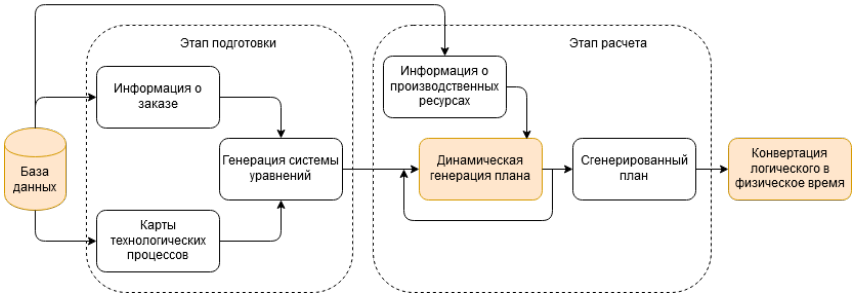

Рисунок 2.3 — Схема ядра имитационного моделирования

В начале работы система выберет одну из независимых переменных, например $op1_1$ или $op2_1$ из системы 2.1 - список переменных может меняться в зависимости от накладываемых моделью ресурса ограничений (о них речь в следующем разделе). Система может быть рассчитана тогда, когда она может быть приравнена к логическому нулю и произведет подсчет времени окончания выполнения данной операции, то есть по соответствующему уравнению найдет значение окончания операции $op1_2$ или $op2_2$. Затем по тому же принципу будет выбрано следующее уравнение и так далее пока есть неразрешенные переменные. Когда они закончатся, подсистема завершит свое выполнение, передав результирующее значение и необходимые данные другому

модулю, который произведет отображение полученного подсистемой имитационного моделирования числа в физическое время, о чем речь пойдет далее.

2.3 Модель ресурса сборочной линии

В процессе создания оперативного плана, для получения корректной оценки времени выполнения операции или набора операций, необходимо ввести систему ограничений для СПП, которая будет отражать влияние ресурса, участвующего в операции, на её время выполнения. Это привело к созданию модели ресурсов, накладывающей ограничения на выбор операции для расчета ядром имитационного моделирования. Под ресурсом подразумевается любое устройство, деталь, инструмент или средство, за исключением сырьевого материала и промежуточного продукта, находящееся в распоряжении предприятия для производства товаров и услуг. В соответствии с данным определением к ресурсам относятся в том числе и человеческие ресурсы, которые в данной системе не рассматриваются с точки зрения поведения или других аспектов человеческой жизни, а лишь с точки зрения возможности выполнить конкретную задачу. Также необходимо обозначить, что в данном разделе под моделью ресурса будет пониматься упрощенная модель реального ресурса, отражающая его основные (в рамках выполняемых операций) характеристики.

Каждая модель ресурса представляет из себя структуру данных, которая должна реализовывать три метода:

— метод привязки операции к модели ресурса;

— метод, осуществляющий проверку возможности выполнения данной операции моделью ресурса;

— метод, реализующий логику работы, в котором происходит изменение состояния данной модели.

Под привязкой операции к модели подразумевается добавление операции в очередь на выполнение и, если это первая привязанная

для данного продукта операция, добавление продукта в очередь на распределение. Привязка осуществляется в начале работы системы, что позволяет ресурсам манипулировать ядром имитационного моделирования разрешая или запрещая выбирать привязанные к ним операции для расчета, что может повлечь за собой изменение последовательности выполнения операций и, соответственно, расчетного времени выполнения карты технологического процесса.

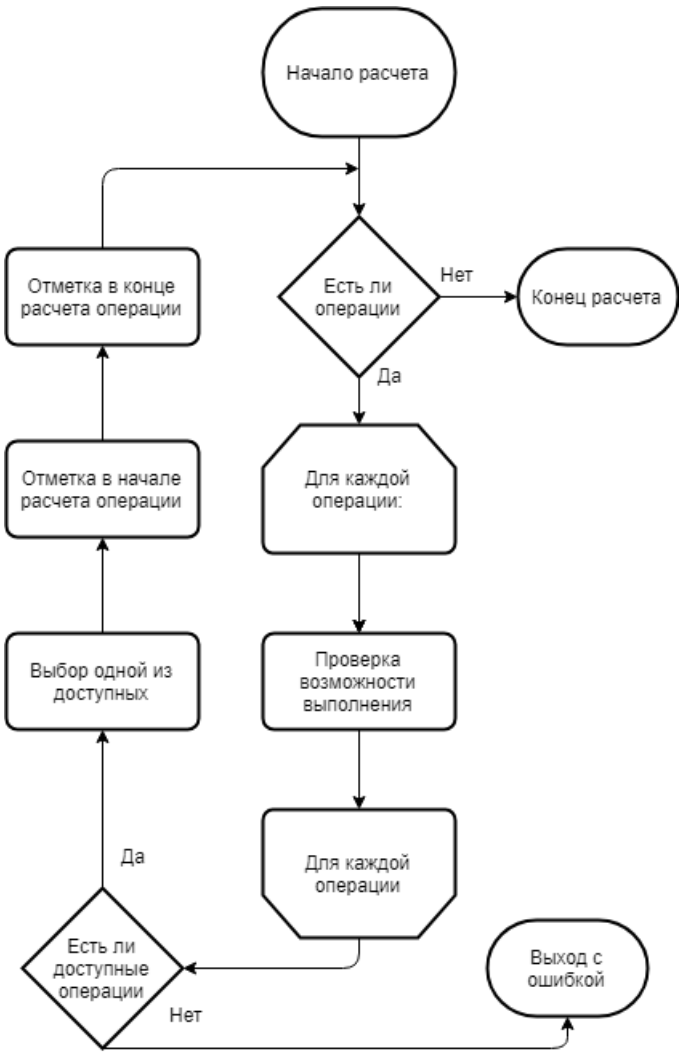

Рисунок 2.4 — Схема работы ядра моделирования с ресурсами в процессе расчета оперативного плана

Проверка производится во время работы системы и именно в этот момент происходит отбор операций в соответствии с внутренним состоянием модели.

Метод, реализующий логику работы вызывается при выборке операции ядром и для каждой вызывается два раза: чтобы отметить

состояние модели в начале и в конце расчета операции (см. рисунок 2.4).

СПП имеет несколько видов ресурсов, одним из которых является модель ресурса сборочной линии. Она описывает несколько однотипных, то есть с одинаковым числом рабочих постов, физических сборочных линий. Сборочная линия - это способ перемещения заготовки от одного рабочего поста к другому; на каждом посту выполняются закрепленные за ним операции. Под постами понимаются заготовкоместа, оснащенные соответствующим технологическим оборудованием и предназначенными для технического воздействия на заготовку для осуществления фиксированного перечня операций. Объединение нескольких сборочных линий в одну обуславливается упрощением как взаимодействия с ядром имитационного моделирования, так и управления моделью, потому как в любом случае (даже когда все линии будут различны по количеству постов) количество линий в модели будет всегда меньше либо равно количеству физических сборочных линии. Благодаря такому объединению представляется возможным инкапсулировать реализацию распределения заготовок и связанных с ними операций по сборочным линиям внутри модели.

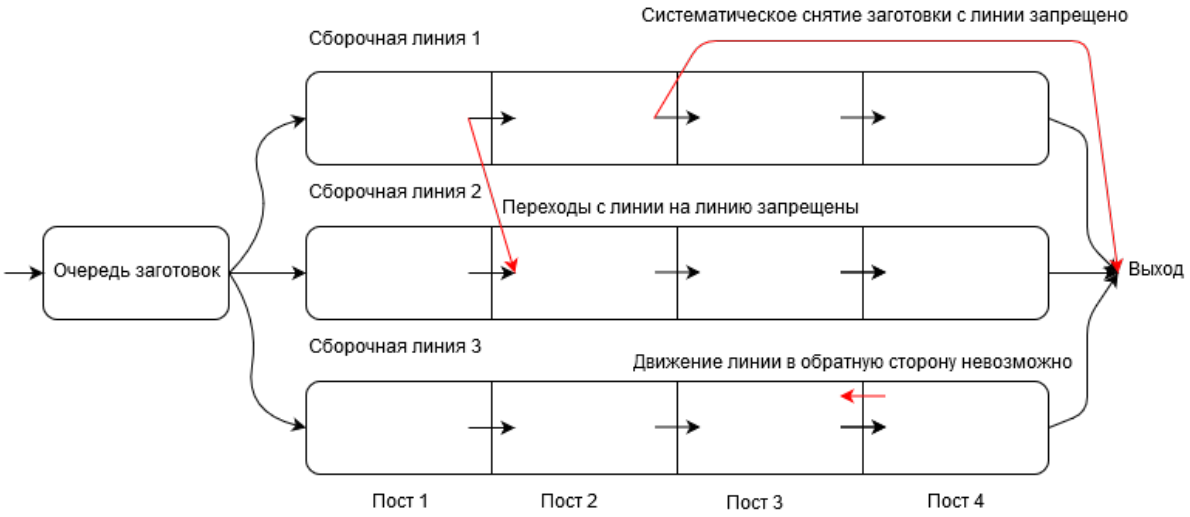

Рисунок 2.5 — Схема модели ресурса с вариантами перемещения заготовки внутри

Главным предназначением данной модели является ограничение перемещения продукции внутри ресурса (см. рисунок 2.5), и моделирование работы физического сборочного конвейера. С одной стороны ограничивается перемещение между сборочными линиями: к какой продукт был привязан, на той он и останется до окончания выполнения всех операций, относящихся к данному продукту и привязанных к постам данной сборочной линии. С другой - перемещение продукции между рабочими постами: продукт должен двигаться последовательно с поста на пост (см. рисунок 2.5).

Одной из ключевых особенностей практически любой сборочной линии является синхронизация передвижения продукции между постами. Это означает что, перемещение продукции на сборочной линии будет осуществляться с периодом, равным максимальной длительности выполнения всех операций на постах - эта длительность называется тактом сборочной линии. Каждая заготовка сможет сменить пост только после того, как все остальные заготовки будут готовы к смене своих постов.

Для реализации необходимого функционала, были введены структуры, описывающие линии, посты, очередь продукции, и проекцию привязки операций к постам линий. Каждая линия, моделируемая компонентом, характеризуется временем начала производственного цикла, структурой данный, описывающей набор рабочих постов (в свою очередь описываемые состоянием: "выполняются работы "простаивает "отсутствует продукция на посту технической картой и серийным номером заготовки),

Проекция привязки операций к постам линий - структура данных необходимая для динамической привязки операций. Так как во время привязки операции система не может оценить длительность изготовления продукции, а распределение заготовок происходит до начала работы, может сложиться ситуация, когда одна из линий будет работать намного дольше или наоборот по сравнению с другими линиями, что приведет к простою производства. Следовательно необходимо, не привязывая операцию к определенной линии, обозначить к какому

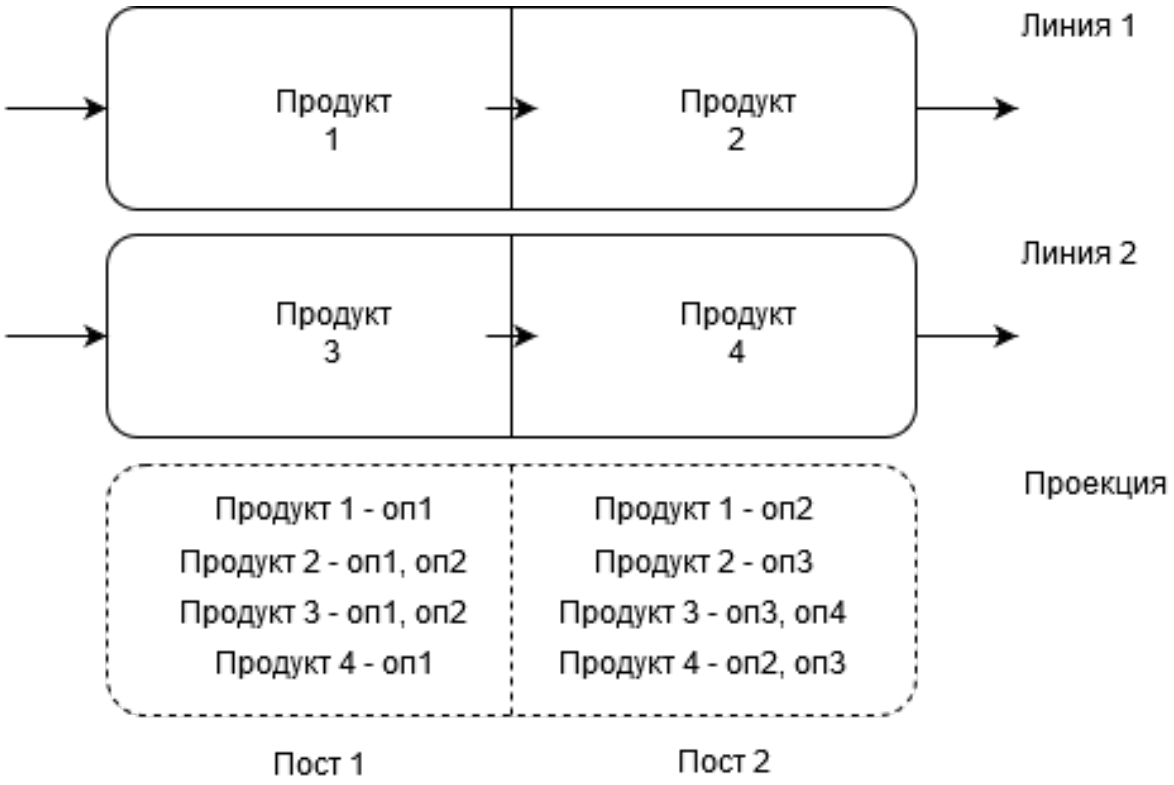

Рисунок 2.6 — Схема сборочной линии с проекцией привязки операций к постам

посту она относится (см. рисунок 2.6). Это является основной задачей разработанной структуры - она содержит то же количество постов, что и остальные линии, но не описывает какую либо физическую сборочную линию, а является проекцией всех линий в целом. Внутри каждого поста данной проекции находится хэш-таблица (структура данных, позволяющая хранить пары (ключ, значение) и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу), в которой ключом является конкретная единица продукции (характеризующаяся типом продукции и серийным номером), а значением - список операций, который необходимо выполнить над данной заготовкой на посту.

Во время выполнения система, для определения возможности выполнения операции опрашивает все линии с целью определения возможности выполнения данной операции и местонахождения продукта:

— при отсутствии продукта на всех линиях:

1) инициируется проверка всех линий на возможность загрузки первого поста (то есть пуст ли он);

2) из получившегося списка линий (если количество больше нуля, иначе переход к 4 пункту) выбирается линия, временная отметка которой является наименьшей среди всех доступных линий;

3) возвращается разрешение на выполнение данной операции и временная метка, выбранная на предыдущем шаге;

4) иначе (количество линий равно нулю) возвращается запрет на выполнение данной операции на текущей итерации.

— если продукт находится на какой-либо линии, то производится опрос проекции:

а) при нахождении нужной операции на посту, на котором находится в данный момент продукт, возвращается разрешение на выполнение и временная метка, с которой может производиться данная операция;

б) иначе возвращается запрет выполнения.

Ядро имитационного моделирования последовательно перебер ет все доступные на данной итерации и выберет одну из тех, что получили разрешение от всех моделей на выполнение. Если таковых не будет, то система известит о невозможности дальнейшей работы вследствие логической ошибки при задании входных данных. В ином случае произойдет вызов метода для того, чтобы отметить состояние модели в начале выполнения операции. При этом, если продукт не был загружен на линию, то произойдет загрузка продукта на линию с минимальной временной отметкой, иначе произойдет проверка на возможность сдвига линии.

Во второй раз метод будет вызван для того, чтобы отметить окончание операции - это означает что данная операция будет удалена из проекции, временная метка поста будет изменена с учетом дли-

тельности операции, и, если операций на данном посту для данного продукта не осталось, то состояние поста изменится на “готов к сдвигу линии” и будет совершена проверка возможности сдвига линии.

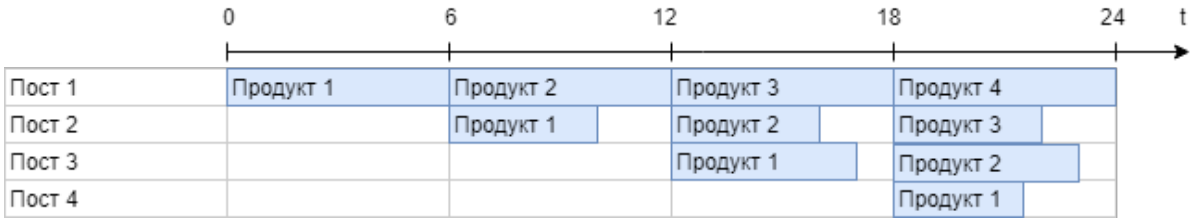

Рисунок 2.7 — Диаграмма, изображающая сдвиг линии

Сдвиг линии - процесс, при котором все посты на сборочной линии передают свою заготовку на следующий пост и принимают заготовку с предыдущего. Данный процесс происходит одновременно для всех постов, не может быть разделен или проигнорирован каким-либо постом и выполняется после получения от всех постов сигнала о готовности к сдвигу. При этом происходит синхронизация постов, то есть все посты, и, соответственно вся линия, получают одну временную отметку - сумма предыдущей отметки начала рабочего такта и длительности текущего рабочего такта линии. С этой отметки начинается отсчет следующего такта (см. 2.7).

Для наглядного представления процессов происходящих в данной модели, на каждой итерации работы создается несколько записей в лог для отслеживания состояния модели в данный момент - по ним можно отследить некорректное поведение, либо просто узнать причины, по которым ядро имитационного моделирования отработало именно в такой последовательности (естественно при условии работы с моделью ресурса сборочной линии). На рисунке 2.8 видно, что модель ресурса охватывает две линии (0 и 1) физических конвейеров и работает с тремя продуктами типа “M” (0, 1 и 2). Также в данном логе видны все передвижения продуктов между постами (station), все изменения временных меток и результирующие метки отгрузки с последних постов.

По завершении основной разработки, было произведено как ручное тестирование (просмотром логов в поисках ошибок работы, рисунок 2.8), так и автоматизированное с написанием модульных (для проверки

```
2019/05/12 19:25:08 Line (0) product (M, 0) loaded on station 0
2019/05/12 19:25:08 Line (0) product (M, 0) changing timestamp (4)
2019/05/12 19:25:08 Line (0) product (M, 0) changing station (0 → 1)
2019/05/12 19:25:08 Line (0) product (M, 0) changing timestamp (7)
2019/05/12 19:25:08 Line (1) product (M, 1) loaded on station 0
2019/05/12 19:25:08 Line (1) product (M, 1) changing timestamp (4)
2019/05/12 19:25:08 Line (1) product (M, 1) changing station (0 → 1)
2019/05/12 19:25:08 Line (1) product (M, 1) changing timestamp (7)
2019/05/12 19:25:08 Line (0) product (M, 2) loaded on station 0
2019/05/12 19:25:08 Line (0) product (M, 2) changing timestamp (8)
2019/05/12 19:25:08 Line (0) product (M, 0) changing station (1 → 2)
2019/05/12 19:25:08 Line (0) product (M, 2) changing station (0 → 1)
2019/05/12 19:25:08 Line (0) product (M, 0) changing timestamp (14)
2019/05/12 19:25:08 Line (1) product (M, 1) changing station (1 → 2)
2019/05/12 19:25:08 Line (1) product (M, 1) changing timestamp (13)
2019/05/12 19:25:08 Line (1) product (M, 1) unloaded from line (timestamp: 13)
2019/05/12 19:25:08 Line (0) product (M, 2) changing timestamp (11)
2019/05/12 19:25:08 Line (0) product (M, 0) unloaded from line (timestamp: 14)
2019/05/12 19:25:08 Line (0) product (M, 2) changing station (1 → 2)
2019/05/12 19:25:08 Line (0) product (M, 2) changing timestamp (20)
2019/05/12 19:25:08 Line (0) product (M, 2) unloaded from line (timestamp: 20)
```

Рисунок 2.8 — Лог с результатами работы модели ресурса сборочной линии

самой модели) и интеграционных тестов (для проверки работы с ядром имитационного моделирования и другими моделями ресурсов) - которые показали правильную работу во всех рассмотренных случаях.

2.4 Модуль отображения логического времени на физическое

Так как расчет выполнения операции (как и всей карты технологического процесса) ядром имитационного моделирования производится в логическом времени,то есть во времени отсчитываемом от нуля, существует необходимость в отображении (соответствие между элементами двух множеств) логического времени на физическое, которое используется в повседневной жизни. Одной из главных сложностей, возникающих при этом, является неоднородность рабочего времени, которая проявляется в рабочем графике (чередование интервалов рабочего и нерабочего времени), наличии выходных, перенесенных дней. Другой сложностью является наличие в системе “обратного расчета”, при котором планирование ведется от даты “дедлайна” (дата или

время, к которому должна быть выполнена задача), что накладывает некоторые ограничения на реализацию данной компоненты.

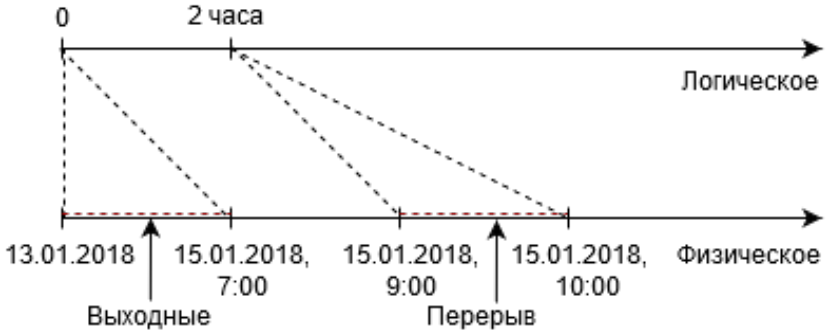

Рисунок 2.9 — Пример отображения оси логического на ось физического времени

Для отображения логического времени на физическое был предложен итеративный процесс, который осуществляет “переход” к необходимому времени путем последовательного перебора дат.

Как было сказано ранее, из-за того, что рабочее время является дискретным, мы не имеем возможности просуммировать начальную дату и значение поданного логического времени. Это ведет к тому, что необходимо синхронизировать логическое и физическое время - это достигается путем последовательного периодического отображения конкретного логического времени на физическое (см. рисунок 2.9). Это подразумевает под собой наличие двух массивов чисел или “осей”:

— оси логического времени, которая начинается с нуля и единица которой соответствует одной секунде (необходимости в более точном отображении нет);

— оси физического времени, на которой может быть отложено любая дата физического времени, отсчет которой начинается 1 января 1970 года 00:00:00 (эпоха Unix).

Особенностью оси физического времени является наличие на ней “выколотых” промежутков времени, в которые работа не ведется и операции не выполняются, следовательно, об этих промежутках

системе необходимо знать, что подразумевает данную информацию как входные данные.

Входными данными для модуля являются:

— дата, с которой необходимо начинать отсчет;

— логическое время, которого необходимо достигнуть;

— конфигурация модуля.

Дата является точкой на физической оси, куда будет отображаться нуль логической. Она представляет собой количество секунд, прошедшее с начала эпохи Unix.

Логическое время - количество секунд, которое должно быть отложено на логической оси. В силу непрерывности физической оси, каждой логической точке сопоставляется отрезок на физической оси, сопоставляется пара чисел - границ данного отрезка.

Конфигурация модуля - вспомогательные данные используемые для определения модулем какие промежутки необходимо пропускать в процессе работы. Состоит из данных о рабочем графике занятого персонала (интервалы рабочего времени), шаблонном расписании на неделю (например, суббота, воскресенье - выходные, пятница - “короткий” день, остальные - стандартные рабочие дни), набора информации о датах, которые являются днями-исключениями и соответствующей информацией о графике работы в данные дни.

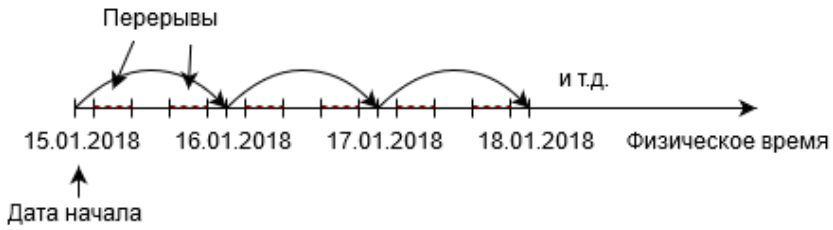

Рисунок 2.10 — Схема прямого расчета

После запуска модуль получает параметры и совершает проверку последних на корректность и непротиворечивость (например, если два дня имеют пересечения временных промежутков, то они противоре-

Рисунок 2.11 — Схема обратного расчета

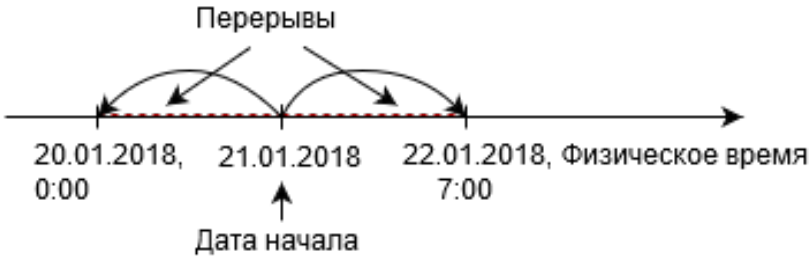

Рисунок 2.12 — Схема проверки времени

чивы, ведь ресурс не может работать одновременно в двух сменах) как в рамках смен одного, так соседних дней. Далее производится определение режима работы: прямой, обратный расчет или проверка времени:

— прямой расчет - задается дата начала отсчета, логическое время и расчет ведется до нахождения даты окончания работ (см 2.10);

— обратный расчет - задается дата дедлайна, логическое время и расчет ведется до нахождения времени начала работ(см 2.11);

— проверка времени - задается дата и логическое время равное нулю, что запускает оба предыдущих расчета пока не будет найдено первое ненулевое время в обоих направлениях от даты начала расчета(см 2.12).

После выбора режима работы сбрасывается счетчик текущего логического времени до нуля и счетчик текущего физического времени до стартовой даты. Затем итеративно, пока текущее логическое время не превысит необходимое производится поиск следующей даты. Алгоритмически поиск даты работает следующим образом (рисунок 2.13):

1) определяются интервалы рабочих смен относящихся к текущему дню:

а) при отсутствии таковых, к текущей дате прибавляется один день и затем возврат к п.1.

2) отсортированные в порядке возрастания интервалы последовательно перебираются и их длительности прибавляются к текущему логическому и физическому времёнам:

а) при превышении текущим логическим временем необходимого, переход к п.3;

б) если все интервалы были просуммированы, но необходимое логическое время не превышено - переход к п.1;

3) разность текущего и необходимого логического времён вычитается из физического времени, при этом сохраняя данное значение как левую (правую при обратном расчете) границу, после чего продолжается расчет для выявления правой (левой) границы промежутка.

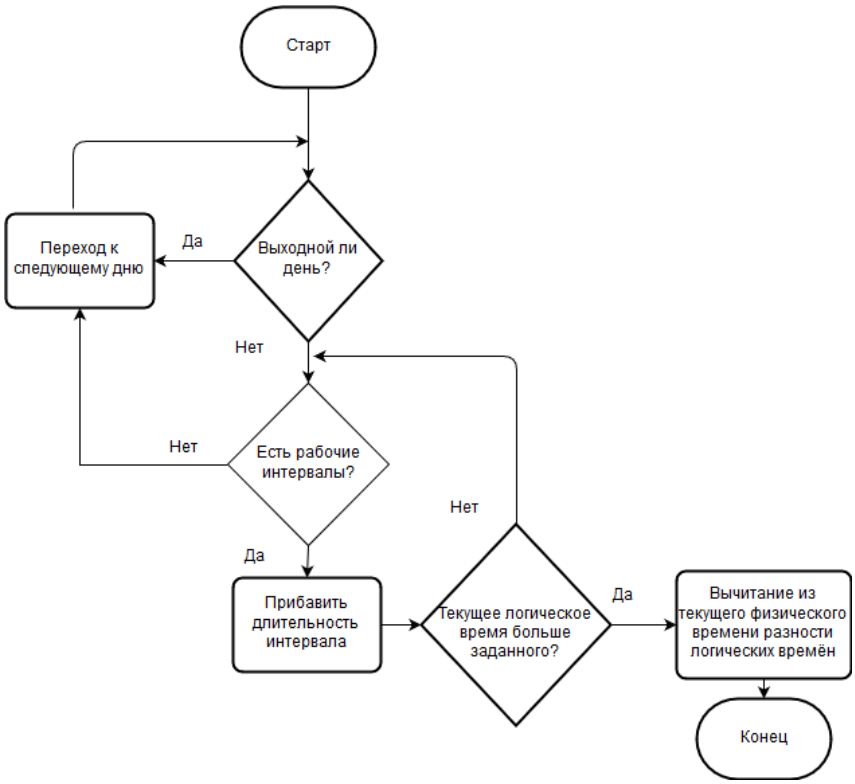

Рисунок 2.13 — Блок-схема модуля

Определение интервалов рабочего времени происходит взятием даты из текущего физического времени - затем начинается определение принадлежности данной даты к перенесенным датам после чего есть два варианта развития ситуации:

— дата является перенесенным днем и модуль получает информацию о расписании которое нужно применить;

— дата не является перенесенным днем и получение информации происходит исходя из того, каким днем недели является данная дата.

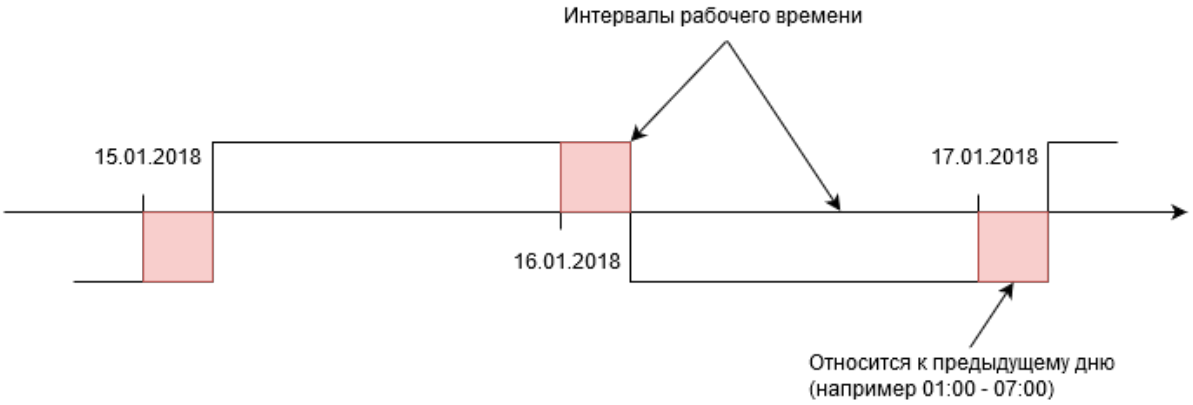

Рисунок 2.14 — Смещение интервалов рабочего времени относительно рабочего дня

Рисунок 2.15 — Граница двух дней

Так как на реальных производственных площадках нередко случается так, что рабочие интервалы, принадлежащие к рабочему дню и сам рабочий день смещены относительно друг друга (см. рисунок 2.14). При этом возникают трудности с определением интервалов рабочего времени в связи с тем, что для определения используется количество секунд с начала эпохи Unix и до нуля часов нуля минут и нуля секунд нужной даты. По количеству, используя инструментарий языка, определяется каким днем недели является нужная дата и, соответственно,

задать для нее шаблон рабочего времени. Эти трудности практически не влияют на прямой расчет, но с обратным все немного сложнее. Так как в одном дне 86400 секунд и, если рассматривать границу двух дней (см. рисунок 2.15), то 86400 секунда предыдущего дня будет равна нулевой секунде нового дня. Можно сказать, что 86400 - левый предел, а 0 - правый предел в данной точке (особенно, если изображать в виде окружности). Но, в связи с тем, что данная недетерминированность присутствует лишь при рассмотрении ситуации человеком, а система может распознавать диапазон от 0 до 86399 секунд, тогда для определения интервалов рабочего графика (при обратном расчете) используется именно величина в 86399 секунд, при условии, что данное допущение не влияет на расчет, как последняя секунда текущего дня.

```
2019/05/10 16:19:10 New config №1: {
    ScheduleTemplate: {
        Sunday    : Day Off,
        Monday    : Workday,
        Tuesday   : Workday,
        Wednesday: Workday,
        Thursday  : Workday,
        Friday    : Shortday,
        Saturday  : Day Off,
    },
    ExceptionDates:{
        1514764800 (01/01/2018 00:00:00): Day Off,
        1514851200 (02/01/2018 00:00:00): Day Off,
        1514937600 (03/01/2018 00:00:00): Day Off,
        1515024000 (04/01/2018 00:00:00): Day Off,
        1515110400 (05/01/2018 00:00:00): Day Off,
        1515196800 (06/01/2018 00:00:00): Day Off,
        1515283200 (07/01/2018 00:00:00): Day Off,
        1515369600 (08/01/2018 00:00:00): Day Off,
        1515456000 (09/01/2018 00:00:00): Day Off,
        1515542400 (10/01/2018 00:00:00): Day Off,
        1515628800 (11/01/2018 00:00:00): Day Off,
        1515715200 (12/01/2018 00:00:00): Day Off,
        1515801600 (13/01/2018 00:00:00): Day Off,
        1515888000 (14/01/2018 00:00:00): Day Off,
    },
    ScheduleForDay:{
        Day Off : {
        }
        Shortday : {
            { ShiftID: 2, Starts: 28800  (01 day 08:00:00), Ends: 46800  (01 day 13:00:00) }
            { ShiftID: 2, Starts: 50400  (01 day 14:00:00), Ends: 64800  (01 day 18:00:00) }
        }
        Workday : {
            { ShiftID: 1, Starts: 28800  (01 day 08:00:00), Ends: 46800  (01 day 13:00:00) }
            { ShiftID: 1, Starts: 50400  (01 day 14:00:00), Ends: 64800  (01 day 18:00:00) }
            { ShiftID: 1, Starts: 68400  (01 day 19:00:00), Ends: 86400  (02 day 00:00:00) }
            { ShiftID: 1, Starts: 90000  (02 day 01:00:00), Ends: 111600 (02 day 07:00:00) }
        }
    }
}
```

Рисунок 2.16 — Пример конфигурации модуля

Возвращаясь к пункту 3 алгоритма при нахождении нужного времени работа модуля не прекращается, а ведется до момента

нахождения второй границы промежутка, на который отображается необходимое логической время.

Как было сказано ранее - физическое время непрерывно, а следовательно когда производится отображение на него логического времени, в результате получается промежуток (см. рисунок 2.9), который и характеризуют две границы. Эта пара чисел, характеризующая начало и конец отрезка, которые отображаются на логическую ось в точке, где значение равно входному логическому времени и являются выходными данными данного модуля.

```
2019/05/10 16:54:09 Start time is: 1514678400 (31/12/2017 00:00:00)
2019/05/10 16:54:09 Given logical time is: 518400 (144 in hours)
2019/05/10 16:54:09  Logical time | Physical time
2019/05/10 16:54:09 --------------|---------------------------------
2019/05/10 16:54:09  0             | 1515974400 (15/01/2018 00:00:00)
2019/05/10 16:54:09  18000         | 1516021200 (15/01/2018 13:00:00)
2019/05/10 16:54:09  32400         | 1516039200 (15/01/2018 18:00:00)
2019/05/10 16:54:09  50400         | 1516060800 (16/01/2018 00:00:00)
2019/05/10 16:54:09  72000         | 1516086000 (16/01/2018 07:00:00)
2019/05/10 16:54:09  90000         | 1516107600 (16/01/2018 13:00:00)
2019/05/10 16:54:09  104400        | 1516125600 (16/01/2018 18:00:00)
2019/05/10 16:54:09  122400        | 1516147200 (17/01/2018 00:00:00)
2019/05/10 16:54:09  144000        | 1516172400 (17/01/2018 07:00:00)
2019/05/10 16:54:09  162000        | 1516194000 (17/01/2018 13:00:00)
2019/05/10 16:54:09  176400        | 1516212000 (17/01/2018 18:00:00)
2019/05/10 16:54:09  194400        | 1516233600 (18/01/2018 00:00:00)
2019/05/10 16:54:09  216000        | 1516258800 (18/01/2018 07:00:00)
2019/05/10 16:54:09  234000        | 1516280400 (18/01/2018 13:00:00)
2019/05/10 16:54:09  248400        | 1516298400 (18/01/2018 18:00:00)
2019/05/10 16:54:09  266400        | 1516320000 (19/01/2018 00:00:00)
2019/05/10 16:54:09  288000        | 1516345200 (19/01/2018 07:00:00)
2019/05/10 16:54:09  306000        | 1516366800 (19/01/2018 13:00:00)
2019/05/10 16:54:09  320400        | 1516579200 (22/01/2018 00:00:00)
2019/05/10 16:54:09  338400        | 1516626000 (22/01/2018 13:00:00)
2019/05/10 16:54:09  352800        | 1516644000 (22/01/2018 18:00:00)
2019/05/10 16:54:09  370800        | 1516665600 (23/01/2018 00:00:00)
2019/05/10 16:54:09  392400        | 1516690800 (23/01/2018 07:00:00)
2019/05/10 16:54:09  410400        | 1516712400 (23/01/2018 13:00:00)
2019/05/10 16:54:09  424800        | 1516730400 (23/01/2018 18:00:00)
2019/05/10 16:54:09  442800        | 1516752000 (24/01/2018 00:00:00)
2019/05/10 16:54:09  464400        | 1516777200 (24/01/2018 07:00:00)
2019/05/10 16:54:09  482400        | 1516798800 (24/01/2018 13:00:00)
2019/05/10 16:54:09  496800        | 1516816800 (24/01/2018 18:00:00)
2019/05/10 16:54:09  514800        | 1516838400 (25/01/2018 00:00:00)
2019/05/10 16:54:09  518400        | 1516845600 (25/01/2018 02:00:00)
2019/05/10 16:54:09 Result:
2019/05/10 16:54:09 ------------------------------------------------------------
2019/05/10 16:54:09  518400        | [1516845600          - 1516845600         ]
2019/05/10 16:54:09  518400        | [25/01/2018 02:00:00 - 25/01/2018 02:00:00]
2019/05/10 16:54:09 Found config №1
2019/05/10 16:54:09 Start time is: 1514678400 (31/12/2017 00:00:00)
2019/05/10 16:54:09 Given logical time is: 504000 (140 in hours)
2019/05/10 16:54:09  Logical time | Physical time
2019/05/10 16:54:09 --------------|---------------------------------
```

Рисунок 2.17 — Пример прямого расчета модулем

В результате работы над модулем было проведено тестирование, результаты одного из которых можно увидеть на изображениях 2.16 и 2.17. На рисунке 2.16 изображена конфигурация модуля, которая

показывает, как производится совмещение логического и физического дня, обработка перенесенных дней. На рисунке 2.17, перед расчетом нового отображения, можно отметить, что при использовании той же конфигурации не производится ее повторный вывод в лог, что позволяет сократить его длину, а следовательно улучшить читаемость. Рисунок 2.17 показывает итеративный процесс как результат прибавления каждого нового интервала к текущему времени.

```
2019/05/10 19:33:39 Start time is: 1516845600 (25/01/2018 02:00:00)
2019/05/10 19:33:39 Given logical time is: -518400 (-144 in hours)
2019/05/10 19:33:39   Logical time | Physical time
2019/05/10 19:33:39 --------------|---------------------------------
2019/05/10 19:33:39   0            | 1516845600 (25/01/2018 02:00:00)
2019/05/10 19:33:39   3600         | 1516838400 (25/01/2018 00:00:00)
2019/05/10 19:33:39   21600        | 1516820400 (24/01/2018 19:00:00)
2019/05/10 19:33:39   36000        | 1516802400 (24/01/2018 14:00:00)
2019/05/10 19:33:39   54000        | 1516780800 (24/01/2018 08:00:00)
2019/05/10 19:33:39   75600        | 1516752000 (24/01/2018 00:00:00)
2019/05/10 19:33:39   93600        | 1516734000 (23/01/2018 19:00:00)
2019/05/10 19:33:39   108000       | 1516716000 (23/01/2018 14:00:00)
2019/05/10 19:33:39   126000       | 1516694400 (23/01/2018 08:00:00)
2019/05/10 19:33:39   147600       | 1516665600 (23/01/2018 00:00:00)
2019/05/10 19:33:39   165600       | 1516647600 (22/01/2018 19:00:00)
2019/05/10 19:33:39   180000       | 1516629600 (22/01/2018 14:00:00)
2019/05/10 19:33:39   198000       | 1516406400 (20/01/2018 00:00:00)
2019/05/10 19:33:39   212400       | 1516370400 (19/01/2018 14:00:00)
2019/05/10 19:33:39   230400       | 1516348800 (19/01/2018 08:00:00)
2019/05/10 19:33:39   252000       | 1516320000 (19/01/2018 00:00:00)
2019/05/10 19:33:39   270000       | 1516302000 (18/01/2018 19:00:00)
2019/05/10 19:33:39   284400       | 1516284000 (18/01/2018 14:00:00)
2019/05/10 19:33:39   302400       | 1516262400 (18/01/2018 08:00:00)
2019/05/10 19:33:39   324000       | 1516233600 (18/01/2018 00:00:00)
2019/05/10 19:33:39   342000       | 1516215600 (17/01/2018 19:00:00)
2019/05/10 19:33:39   356400       | 1516197600 (17/01/2018 14:00:00)
2019/05/10 19:33:39   374400       | 1516176000 (17/01/2018 08:00:00)
2019/05/10 19:33:39   396000       | 1516147200 (17/01/2018 00:00:00)
2019/05/10 19:33:39   414000       | 1516129200 (16/01/2018 19:00:00)
2019/05/10 19:33:39   428400       | 1516111200 (16/01/2018 14:00:00)
2019/05/10 19:33:39   446400       | 1516089600 (16/01/2018 08:00:00)
2019/05/10 19:33:39   468000       | 1516060800 (16/01/2018 00:00:00)
2019/05/10 19:33:39   486000       | 1516042800 (15/01/2018 19:00:00)
2019/05/10 19:33:39   500400       | 1516024800 (15/01/2018 14:00:00)
2019/05/10 19:33:39   518400       | 1514592000 (30/12/2017 00:00:00)
2019/05/10 19:33:39 Result:
2019/05/10 19:33:39 ------------------------------------------------------------
2019/05/10 19:33:39   518400       | [1516003200          - 1514570400         ]
2019/05/10 19:33:39   518400       | [15/01/2018 08:00:00 - 29/12/2017 18:00:00]
```

Рисунок 2.18 — Пример обратного расчета модулем

Также были проведены тесты обратного расчета и проверки времени, результаты которых можно видеть на изображениях 2.18 и 2.19. Пояснение к анализу заданного физического времени: используется конфигурация, где 2 января является выходным днем, анализ которого и производится, что позволяет продемонстрировать работу данного

смещен ан
отдельную
страницу,
пофиксить

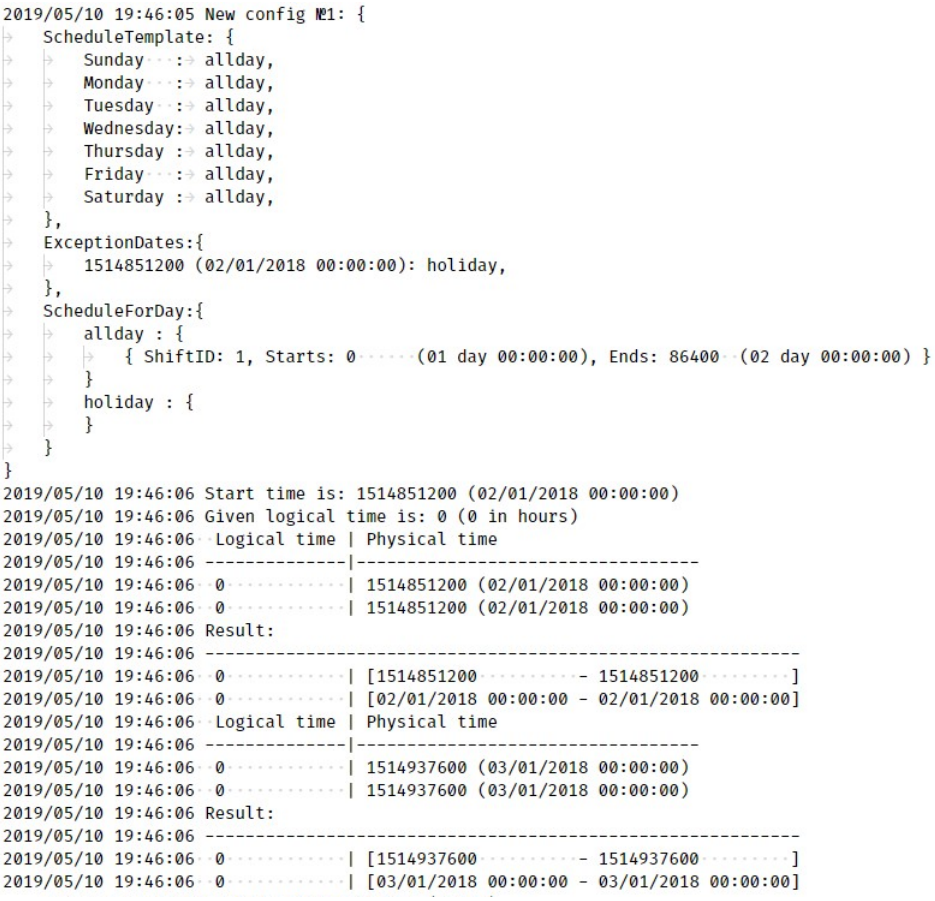

Рисунок 2.19 — Пример анализа заданного физического времени

3 Организация консистентного хранилища данных

3.1 Выбор хранилища данных

Система планирования производства, как и практически любая система, получающая и обрабатывающая данные, нуждается в хранилище данных - базе данных. Хранилище данных может быть разделено на две составляющие:

— хранилище временных данных;

— хранилище постоянных данных.

Под временными данными подразумеваются данные, которые получаются в процессе работы системы (например оперативные и объемно-календарные планы) и, при необходимости, могут быть рассчитаны заново, хоть и с некоторыми затратами (время или вычислительные мощности). В данной работе этот вид данных и хранилище для них не рассматривается.

С другой стороны существуют постоянные данные - информация которая задается пользователем (например предприятием), потеря которой в лучшем случае приведет к необходимости заново добавлять их в систему, а в худшем - приведет к её утрате. В любом случае потеря постоянных данных ведет к критическим нарушениям в работе системы, что обуславливает необходимость организации консистентного хранилища данных.

Консистентность - требование к данным, получаемым из базы данных, которое заключается в том, что последние должны быть целостны и непротиворечивы. Под целостностью данных подразумевается соответствие имеющейся в базе данных информации её внутренней логике, структуре и явно заданным правилам. Любое правило, направленное на ограничение возможного состояния базы данных,называют ограничением целостности. Помимо целостных, данные должны также быть непротиворечивыми - это означает, что в базе данных нет логического противоречия, то есть некоторого утверждения и его отрицания.

В случае системы планирования производства, в качестве по-

стоянных данных требуется хранить информацию о каждом запуске системы для того, чтобы можно было затем выбрать оптимальный план работы предприятия. Это ведет к тому, что появляется несколько версий одних и тех же данных и приводит к необходимости организации хранения и извлечения этих версий.

3.2 Понятия реляционной теории

Для организации и теоретического обоснования описанного ранее хранилища, использовалась теория реляционных баз данных.

Теория реляционных баз данных оперирует понятиями отношения (relation, от которого и пошло само название теории и баз данных основанных на ней), атрибута, домена и кортежа.

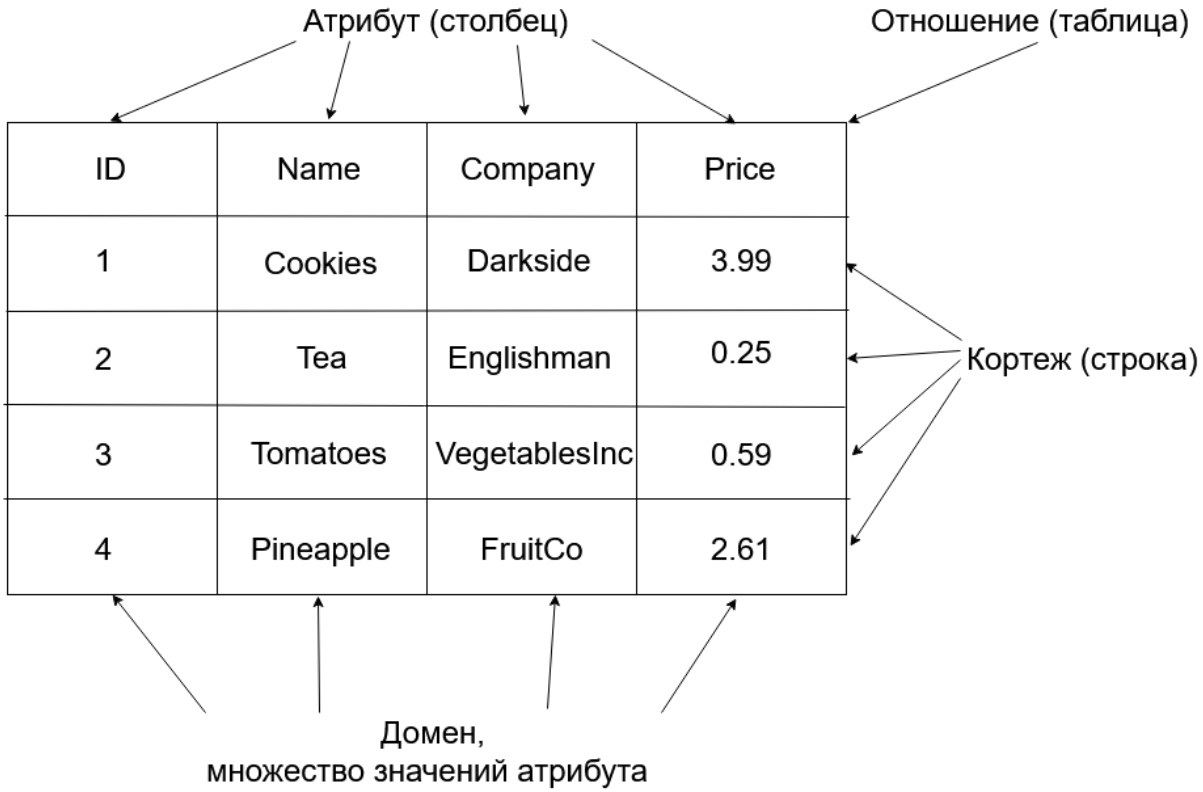

Рисунок 3.1 — Понятия реляционной базы данных

Атрибут - именованный столбец отношения. В пределах одного атрибута все значения должны быть одного типа данных, то есть принадлежать одному домену.

Домен - тип данных, множество всех допустимых значений ат-

рибута.

Кортеж - упорядоченный набор из N элементов, где N - это число атрибутов отношения. Иначе говоря, кортеж - это строка или запись таблицы.

Отношение - множество упорядоченных N-кортежей. Другими словами отношение - это двумерная (плоская) таблица, состоящая из столбцов и строк - атрибутов и кортежей (см. рисунок 3.1).

Также необходимо ввести понятие целостности по ссылкам заключающееся в отсутствии в любом её отношении внешних ключей (атрибут, указывающий на атрибут в другом отношении и совпадающий с ним по значению), ссылающихся на несуществующие кортежи.

Так как любое отношение это множество, то им свойственны все операции определенные для множеств:

— пересечение;

— объединение;

— вычитание;

— декартово произведение.

Помимо этих четырех операций, в теории реляционной алгебры, вводится еще 4 операции свойственные только отношениям:

— выборка ($\sigma_\phi(R)$) - накладывает ограничение ϕ на отношение R;

— проекция ($\pi_\phi(R)$) - отображает только те атрибуты отношения R, которые были представлены в ϕ;

— соединение ($R_1 \bowtie_\phi R_2$) - соединяет кортежи из двух отношений R_1 и R_2 по условию ϕ;

— переименование ($\rho_{a,b}(R)$) - переименовывает атрибут отношения R из a в b .

дать определения операциям над множествами

3.3 Обоснование

Пусть заданы два отношения R_1 и R_2, такие что:

$$\begin{aligned} R_1 &= < name, version, ... > \\ R_2 &= < FK(R_1, name), version, ... > \end{aligned}$$

$FK(R, a)$ - внешний ключ, ссылающийся на атрибут a на отношения R, тогда введем оператор “версионного соединения”, который будет отражать взаимосвязь двух отношений:

$$\forall R_1,\, R_2 \;:\; R_1.name \supset R_2.name,\, R_1.version \leqslant R_2.version \;: \\ R_1,\, R_2 \Rightarrow R_1 \overrightarrow{\bowtie} R_2 \quad (3.1)$$

Формула 3.1 отражает эквивалентность “версионного соединения” и связанных по условию отношений R_1 и R_2. То есть для версионного соединения должны выполняться два условия:

— внешний ключ $R_2.name$ второго отношения существует в домене $R_1.name$;

— версия указанная во втором отношении не меньше версии указанной в первом.

При их соблюдении возможно применять свойство следующее из определения версионного соединения:

$$\forall\; f : f(R_1 \overrightarrow{\bowtie} R_2,\, R_1' \overrightarrow{\bowtie} R_2') \Leftrightarrow f(R_1,\, R_1') \overrightarrow{\bowtie} f(R_2,\, R_2') \quad (3.2)$$

Другое свойство версионного соединения:

$$\forall R, R', R_1', R_2', f : R \subsetneq f(R, R'), R_1 \overrightarrow{\bowtie} R_2, R_2'.name \subset R_1.name : \\ \left.\begin{aligned} f(R_1, R_1') &\Rightarrow \bot \\ f(R_2, R_2') &\Rightarrow R_2'' \\ f(R_1, R_1'), f(R_2, R_2') &\Rightarrow R_1'', R_2'' \end{aligned}\right\} \Rightarrow \begin{aligned} &\bot \\ &R_1 \overrightarrow{\bowtie} R_2'' \\ &R_1'' \overrightarrow{\bowtie} R_2'' \end{aligned} \quad (3.3)$$

как это адекватно записать?

В выражении 3.3 показано свойство, что изменение множества кортежей отношения R_1 функцией $f(R, R')$ недопустимо, при том что

либо только R_2 либо и то и другое отношение одновременно могут быть изменены при соблюдении условий версионного соединения.

Оператор версионного соединения ($\overrightarrow{\bowtie}$) не следует понимать как обычное пересечение ($\bowtie$), потому как он лишь отражает взаимосвязь двух отношений не соединяя их в одно.

Теорема 1. *Любая выборка (σ_ϕ) из отношения полученного путем объединения двух версионных соединений R и R' будет включать в себя выборку по тому же условию из R.*

$$R \leftarrow (R_1 \overrightarrow{\bowtie} R_2),\ R' \leftarrow (R'_1 \overrightarrow{\bowtie} R'_2) \tag{3.4}$$

$$f :: R_1 \overrightarrow{\bowtie} R_2 \rightarrow R_1 \overrightarrow{\bowtie} R_2 \rightarrow R_1 \overrightarrow{\bowtie} R_2 \tag{3.5}$$

$$\forall \quad R,\ R', f : R \subsetneq f(R,\ R') \tag{3.6}$$

$$\forall \quad \phi,\ R,\ R' :\ \sigma_\phi(R) \subset \sigma_\phi(f(R,\ R')) \tag{3.7}$$

Если предположить, что теорема 3.7 неверна, тогда объединение не включают в себя R для любого условия ϕ:

декомпозировать

$$R \not\subset f(R,\ R') \tag{3.8}$$

$$R \not\subset R \cup R' \tag{3.9}$$

$$R \not\subset (R_1 \overrightarrow{\bowtie} R_2) \cup (R'_1 \overrightarrow{\bowtie} R'_2) \tag{3.10}$$

$$R_1 \overrightarrow{\bowtie} R_2 \not\subset (R_1 \cup R_2) \overrightarrow{\bowtie} (R'_1 \cup R'_2) \tag{3.11}$$

$$\tag{3.12}$$

Так как множества могут быть в том числе и пустыми, приравняем R'_1 и R'_2 пустому множеству:

$$\left.\begin{array}{rcl} R'_1 & = & \varnothing \\ R'_2 & = & \varnothing \end{array}\right\} \Rightarrow R_1 \overrightarrow{\bowtie} R_2 \not\subset R_1 \overrightarrow{\bowtie} R_2 \tag{3.13}$$

Из выражения 3.13 видно, что в результате получается, что версионное соединение множеств не входит само в себя, что неверно. Из этого следует что изначальное предположение оказалось неверным, а следовательно теорема 3.7 доказана.

Доказать

монотонность версий

оформить свойства и определение

3.4 Структура базы данных

Из раздела 3.1 очевидно, что в постоянной базе данных необходимо хранить информацию, на основе которой производится расчет объемно-календарного и оперативного планов. Для этого необходимо хранить:

— данные о заказах;

— данные о типах продукции;

— данные о продукции, которая относится к каждому заказу;

— данные о самой последовательности операций и самих операциях для каждого типа продукции;

— данные о каждой модели ресурсов;

— данные о связях между моделью ресурса и операциями.

Помимо этого, нужно учитывать, что кроме простого хранения данных, необходимо соблюдать “версионность”, ввиду того, что карта технологического процесса может меняться во времени. Значит и в базе данных требуется следить за тем, чтобы в любой момент времени было возможно использовать любую из версий данной карты, иначе новый расчет оперативного плана (при условии, что другие данные остались неизменными) приведет к созданию новой версии плана, а, следовательно, предыдущую версию восстановить будет либо очень сложно, либо, в худшем случае, невозможно.

Для достижения данной цели в определенных таблицах было введено дополнительное поле - временная отметка обновления - “upd”. С помощью этой метки можно получить как последние данные из

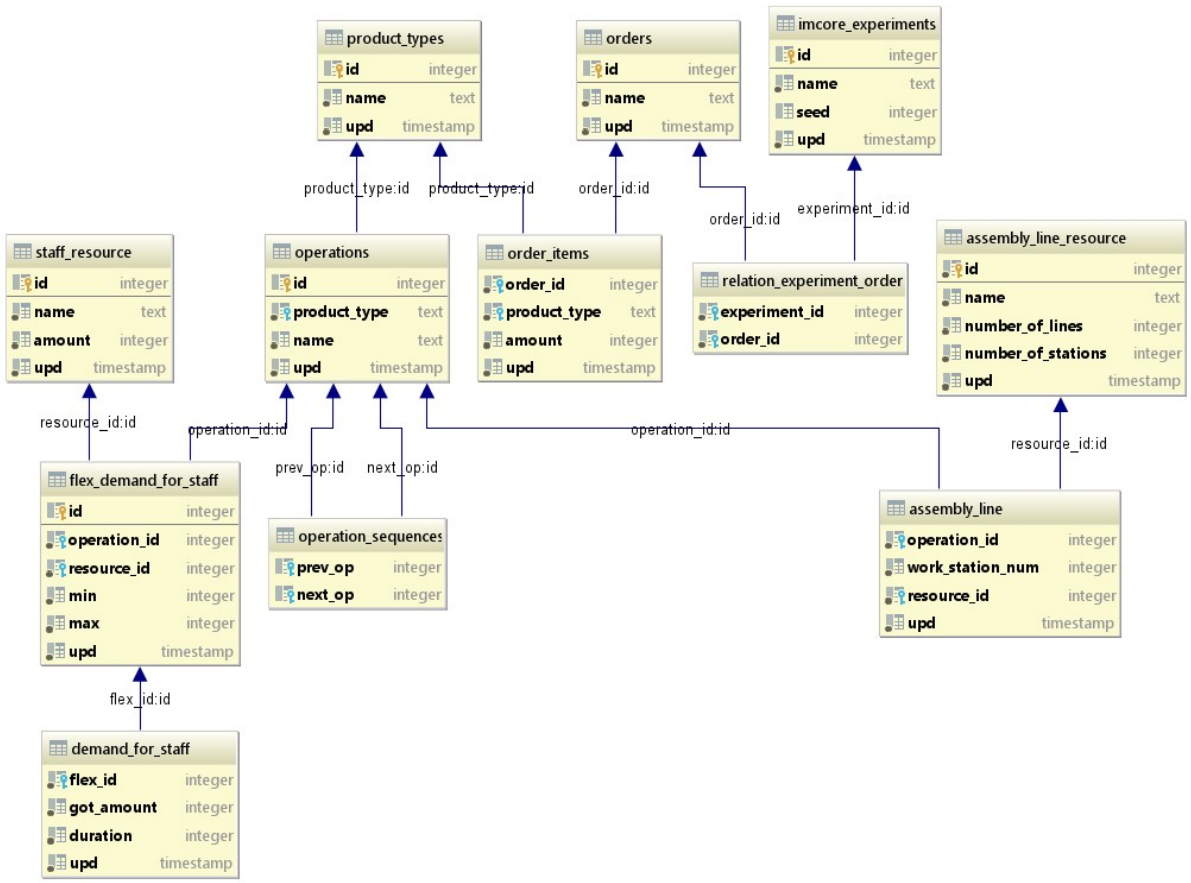

Рисунок 3.2 — Схема базы данных

базы, просто максимизируя метку “upd”, так и данные на определенный момент времени ограничивая эту метку необходимым временем.

Заключение

В рамках данной работы были рассмотрены, разработаны и протестированы компоненты системы планирования производства, а также организованно хранилище данных, работа которого была формально верифицирована.

По итогу выполнения работы были достигнуты следующие результаты:

— произведен анализ логического и физического времени;

— синтезирован, реализован и протестирован алгоритм отображения логического времени на физическое;

— проанализирована сборочная линия, её функции и ограничения, накладываемые на перемещение продукции;

— синтезирована, реализована и протестирована модель сборочной линии для подсистемы имитационного моделирования СПП;

— проанализированы данные предприятия, которые необходимо хранить в базе данных;

— синтезирована и обоснована структура базы данных.

Разработанные модули в настоящий момент интегрированы в СПП с соответствующими интеграционными тестами.

Структура базы данных была реализована и используется для ...ия и извлечения тестовых данных, что используется для тестирования ядра имитационного моделирования. Также были реализованны введенные в процессе обоснования структуры базы данных операторы и выражения, что упростило работу с версионированными отношениями.

обезличенных
ссылка на Диаконт без упоминания!

Написать и исправить

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Roth T. P., Song Y., Burns M. J. et al. CYBER-PHYSICAL SYSTEM DEVELOPMENT ENVIRONMENT FOR ENERGY APPLICATIONS. 2017. URL: https://www.nist.gov/publications/cyber-physical-system-development-environm

2. Олег Новиков. ЧТО ТАКОЕ ИНДУСТРИЯ 4.0? ЦИФРЫ И ФАКТЫ. 2015. URL: http://holzex.ru/chto-takoe-industriya-4-0-tsifryi-i-faktyi.

3. Интеллектуальные технологии цифрового производства: Tech. Rep.: / Кремлев А.С., Маргун А.А., Юрьева Р.А. [и др.]: 2018.

4. Э. Мюллер, М. Шенк, З. Вирт. Планирование и эксплуатация промышленных предприятий: Рабочие методики для адаптивного, сетевого и ресурсосберегающего предприятия. 2017.

5. Дж. Лодон, К. Лодон. Управление информационными системами. 2005.

6. Д. О'Лири. ERP системы. Современное планирование и управление ресурсами предприятия. Выбор, внедрение и эксплуатация. 2004.

7. Гибсон, Дж.Л. Организации: поведение, структура, процессы. 2000.

8. Маззулло Джим, Уитли Питер. SAP R/3 для каждого. Пошаговые инструкции, практические рекомендации, советы и подсказки. 2008.

9. Герхард Келлер, Томас Дикерсбах Йорг. Планирование и управление производством с помощью решений SAP ERP. 2011.

10. Система планирования и прогнозирования ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ТЕХНИЧЕСКОМУ ПРОЕКТУ: Tech. Rep.: / Кремлев А.С., Маргун А.А., А.А. Иващенко [и др.]: 2018.