

УДК 681.3

О. В. Быченко, И. В. Максимей

Учреждение образования

«Гомельский государственный университет им. Ф. Скорины»

ул. Советская, 104, 246019 Гомель, Республика Беларусь

Универсальный способ имитации дискретных сложных систем

Изложена история развития агрегатного, процессного и транзактного способов имитации. Рассмотрены попытки объединения двух способов имитации. Предложена основная идея универсального способа имитации дискретных сложных систем (ДСС). Приведен пример описания ДСС с помощью универсального способа имитации. Обсуждаются состав и структура системы автоматизации моделирования UNIVERSAL, реализующей предложенный способ имитации.

Ключевые слова: сложные системы, способы имитации, агрегаты, транзакты, процессы, динамические, статистические элементы.

Введение

Первым из способов имитации сложных систем (СС), который был предложен Н.П. Бусленко [1], являлся *агрегатный* способ имитации. Для исследования производства было предложено использовать кусочно-линейные агрегаты (КЛА), обладающие простым алгоритмом имитации, и которые взаимодействовали друг с другом с помощью сигналов. Приход сигналов синхронизовался внутри КЛА, и с помощью функций перехода формировались выходные сигналы. Весь производственный процесс (ПП) представлялся в виде графа (GR), узлами, которого были КЛА_{ij}, а связями являлись сигналы (Sig_{ij}) простой структуры (здесь *i* и *j* — номера узлов графа). Для автоматизации построения и эксплуатации имитационных моделей графа технологии (ИМ ГТ) была разработана система автоматизации моделирования (САМ), которая имела достаточно простую управляющую программу моделирования (УПМ). Представление структуры СС возлагалось на пользователя, поскольку КЛА имели универсальную структуру, а УПМ выполняли функции передачи сигналов от одного КЛА к другому. Однако все сложности программирования ИМ ГТ лавинообразно возникали перед пользователем и, кроме того, реализация этой САМ была на ЭВМ первого поколения, что породило множество

© О. В. Быченко, И. В. Максимей

проблем построения ИМ ГТ даже на этапах эксплуатации этих моделей. Это обстоятельство привело к тому, что агрегатный способ не получил достаточного распространения. Известны только несколько реализаций САМ, реализующих классический способ имитации с помощью агрегатов [2, 3].

Поэтому возник *процессный способ* имитации, при котором каждый процесс (ПР) имел сложный алгоритм имитации активных компонентов СС. Все компоненты СС связывались друг с другом через общую базу данных (ОБД) имитационной модели (ИМ) и с помощью команд управления (КОМ_{ki}) типа: пустить, остановить, продолжить выполнение ПР. Каждая k -я команда реализовывалась с помощью УПМ. Причем, было разработано несколько языков имитации, отличающихся набором команд управления с помощью которых УПМ управляли как ходом имитации во времени функций компонентов СС, так и способом взаимодействия ПР друг с другом. На основе приоритетов ПР_{ij}, установленных пользователем САМ, УПМ синхронизовала в модельном времени динамику выполнения ПР_{ij} и контролировала окончание имитации. В языке САМ имелся большой набор операторов сбора статистики имитации и формирования псевдослучайных чисел для моделирования в алгоритмах ПР «жребиев» свершения случайных событий и для имитации реализаций значений случайных величин по функциям распределения вероятностей (ФРВ). По началу *процессный способ* имитации получил довольно широкое распространение, и вершиной его была САМ СИМУЛА [4]. Однако богатство возможностей этой САМ требовало постоянного использования специалистов по программированию. Кроме того, реализации этой САМ на ЭВМ первого поколения рождали еще большее количество проблем построения на этапах их испытания и эксплуатации ИМ.

Это обстоятельство привело к появлению *транзактного способа* имитации, в котором активные компоненты СС представляются в виде систем массового обслуживания (СМО). Структура графа СС (ГСС) представлялась в виде блок-схемы, состоящей из множества СМО, которые связаны друг с другом с помощью системы очередей (на входе и выходе из СМО). Связь между СМО в ИМ обеспечивалась с помощью транзактов (TR_{ij}) простой структуры. Каждому TR_{ij}, кроме идентификатора ij , присваивался приоритет π_{ij} и в ИМ ГСС, состоящей из множества {СМО_i}, имитировалось:

- движение транзактов от данной к другой очереди к СМО_i;
 - выбор TR_{ij} из очереди, которое осуществлялось самим обслуживающим устройством согласно установленной дисциплине обслуживания (DIS_{jk}) (здесь j — номер очереди, а k — номер СМО);
 - собственное обслуживание TR_{ij} путем задержки на время обслуживания τ_{ij} .
- Для автоматизации построения и эксплуатации ИМ ГСС была разработана САМ GPSS [5]. Простота перевода ГСС в ИМ на языке GPSS привела к широкому ее распространению среди специалистов.

Существует большое число версий САМ на основе GPSS [6]. Причем, разработчики этих версий увеличивали технологические возможности имитации, постоянно добавляя новые изобразительные возможности, создающие дополнительное удобство их использования в разных предметных областях. Но слабые описательные возможности языка GPSS остались существенным тормозом для создания многих реальных СС. Возникли большие трудности в создании ИМ на основе

транзактного способа имитации. В любой версии САМ на основе GPSS УПМ организовала перемещение TR_i от одной к другой очереди и имитацию захвата TR_{ij} соответствующей СМО_{*j*}. Простота имитации в GPSS стала тормозом для использования версий САМ в тех случаях, когда необходимо было имитировать взаимодействие компонентов в СС иерархической структуры. Обзор описательных и технологических возможностей САМ, реализующих перечисленные три способа имитации, приведены в монографии [7]. Необходимо отметить, что эти САМ позволяли автоматизировать создание ИМ СС, построенных только на одном способе имитации.

Объединение двух способов имитации дискретных сложных систем

Проблемы имитации СС, рассмотренные в работе [6], обусловили необходимость создания *транзактно-процессного* способа имитации. Компоненты СС представляются процессами двух типов: обслуживающих устройств (ОУ) имеющих сложный алгоритм обслуживания транзактов; процессов (ПР_{*i*}), которые состоят из набора активностей управления имитацией других процессов, и активностей, организующих имитацию самой компоненты по сложному алгоритму на основе информационной связи элементов ИМ через общую базу данных. Связь в ИМ между ОУ в ИМ осуществляется с помощью системы очередей, на которые поступают транзакты сложной структуры (STR_{ij}). В отличие от GPSS, использующего TR_{ij} классической структуры, STR_{ij} состоят из двух составляющих:

1) информационной части транзакта (inf_{ij}) *ij*-го номера и π_{ij} приоритета, в которой содержится три группы информации: программа поведения транзакта в ИМ (ППП_{*i*}); исходная информация (ИСХ_{*ij*}), позволяющая имитировать вероятностные характеристики использования STR_i ресурсов ИМ СС; поле сбора статистики (ПСТ_{*ij*}) обслуживания STR_{ij} компонентами ИМ СС;

2) головки транзакта, состоящего из триады (*ij*, π_{ij} , α_{ij}), в которой α_{ij} означает адрес STB_{ij} в общей базе данных ИМ СС (место нахождения самой inf_{ij}).

Таким образом, в ИМ СС к очередям от одного ОУ_{*i*} к другому ОУ_{*j*} движутся триады транзактов. Собственно перемещение этих триад обеспечивает УПМ. Сами процессы и ОУ_{*i*} имеют сложный алгоритм обслуживания STR_{ij} . Причем, одни процессы могут являться генераторами STR_{ij} , а другие уничтожают их по окончании цикла имитации транзактов в ИМ СС. Кроме того, некоторые процессы взаимодействуют с другими процессами и ОУ_{*j*} через УПМ с помощью целой системы команд (типа «пустить», «остановить», «продолжить работу» процесса). Структурно ОУ_{*j*} и процессы состоят из набора активностей (АКТ_{*kj*}) (где *k* — номер активности в программе процесса, а *j* — номер самого процесса). Каждая АКТ_{*kj*} состоит из двух частей:

1) алгоритма моделирования компоненты СС на основе информации, содержащейся в информационной части STR_{ij} ;

2) операторов обращения к УПМ либо для имитации во времени функций процесса, либо для управления другими процессами, либо для пересылки STR_{ij} в одну из очередей последующего ОУ_{*j*}.

Для автоматизации создания ИМ СС на основе транзактно-процессного спосо-

ба имитации была разработана САМ МІСІС [6], которая явилась основой создания набора программно-технических комплексов имитации (ПТКИ) сложных систем для разных предметных областей. Больше всего прошли апробацию ПТКИ ЛВС, позволяющие автоматизировать создание, испытание и эксплуатацию ИМ ЛВС разного уровня детализации вычислительного процесса (ВП) в ЛВС. Результаты апробации версий ПТКИ ЛВС изложены в работах [8–13]. Собственно состав и структура ПТКИ и его использование для проектного моделирования в других предметных областях изложены в работе [6].

Однако, сочетание транзактного и процессного способа имитации СС возможно при выполнении ряда ограничений на использование ОУ_{*j*} самим подходом к рассмотрению СС в виде множества СМО. В ряде случаев эти ограничения не выполняются, и это привело к появлению нового способа имитации СС на основе сочетания агрегатного и процессного способа имитации [14]. Компоненты СС представляются либо процессами, либо агрегатами сложной структуры. В отличие от классического агрегатного способа имитации [1] агрегаты имеют сложную структуру, являясь по своей сути многофункциональными процессами, предметно ориентированными на проектное моделирование вероятностных технологических процессов (ВТП), и состоят из множества АКТ_{*kj*}. Каждая АКТ_{*kj*} завершается оператором обращения к УПМ либо для связи с другими агрегатами, либо для взаимодействия с другими процессами посредством команд управления работой этих процессов с помощью УПМ.

Связи между агрегатами осуществляются с помощью сигналов сложной структуры (CSig_{*ij*}). Связи между исполнительными (AGISP_{*ij*}) агрегатами осуществляются с помощью соответствующих агрегатов синхронизации (AGSIN_{*j*}). Различают два типа AGSIN_{*j*}:

— функционирующие по логической схеме совпадения «И», которая активизируется приходом на его входы самого позднего в модельном времени SSig_{*ij*} (ALAS_{*j*});

— функционирующие по логической схеме совпадения «ИЛИ», которая активизируется приходом самого раннего в модельном времени SSig_{*ij*} (AFIRST_{*j*}.)

Агрегаты AGISP_{*ij*} представляют собой универсальные двухполюсники, которые после прихода на их входы сигнала от одного из агрегатов AGSIN_{*i*} имитируют расход ресурсов СС по типовому алгоритму по запросам SSig_{*ij*}. Далее по окончании имитации на выходе AGISP_{*ij*} формируются сигналы (здесь *i* — номер AGSIN_{*i*}, активизирующий этот алгоритм, а *j* — номер AGSIN_{*j*} на который поступает сигнал после имитации алгоритма AGISP_{*ij*}). Таким образом, ИМ СС представляется в виде сочетания двух множеств агрегатов {AGISP_{*ij*}} и {AGSIN_{*j*}}. Передачу сигналов SSig_{*ij*} между этапами типами агрегатов осуществляет УПМ. Новым при таком гибридном способе имитации СС является сложная структура сигнала. Помимо адресной части SSig_{*ij*} (AD_{*ij*}) имеется еще и информационная часть сигнала (IN_{*ij*}), в которой содержится предыстория имитации компонентов ИМ СС в виде признаков: «была авария» (π_{ij}) при имитации AGISP_{*ij*} и «был выход» управляющей переменной *U* для СС за допустимые пределы (π_{uij}). В дальнейшем к этим признакам в IN_{*ij*} сложного сигнала агрегаты и процессы могут добавлять другую информацию. Вся информационная часть SSig_{*ij*} используется в качестве оперативной информации для использования ее алгоритмами агрегатов синхрони-

зации или исполнительными агрегатами. $AGISP_{ij}$ имитируют расход ресурсов СС во время имитационного эксперимента. Алгоритмы $AGISP_{ij}$ моделируют расход в СС двух типов ресурсов: детерминированным множеством ресурсов $\{DRES_{kj}\}$, где k — номер детерминированного ресурса; множеством ресурсов вероятностей природы $\{VRES_{hj}\}$. Обычно в состав множества $\{DRES_{kj}\}$, используемых агрегатами j , входят ресурсы номера k : ($k = 1 \div 6$):

$DRES_1$ — количество устройств оборудования индивидуального пользования (n_{1j});

$DRES_2$ — количество устройств оборудования общего пользования (n_{2j});

$DRES_3$ — количество ресурсов индивидуального пользования (n_{3j});

$DRES_4$ — количество ресурсов общего пользования (n_{4j});

$DRES_5$ — количество индивидуальных исполнителей (n_{5j});

$DRES_6$ — список исполнителей, входящих в состав бригады исполнителей (SP_6).

Каждый j -й элемент множества $\{RES_{nj}\}$ использует четыре типа вероятностных ресурсов:

$DRES_7$ — количество материалов k -го типа (mt_{nij});

$DRES_8$ — количество комплектующих изделий k -го типа (KO_{kij});

$DRES_9$ — стоимость выполнения операции, имитируемой j -м агрегатом (ST_{ij});

$DRES_{10}$ — время имитации j -м агрегатами выполнения операции (τ_{ij}).

Поскольку эти четыре типа ресурсов являются вероятностными, то их конкретные значения в l -й реализации процедуры Монте–Карло во время имитации выполнения операции агрегат формирует с помощью соответствующих функций распределения вероятностей:

$$F_{4kij}(mt); F_{3kij}(KO); F_{2ij}(st); F_{1ij}(\tau). \quad (1)$$

Очевидно для описания запросов ресурсов агрегатом номера ij эти функции распределения (1) должны быть заданы до имитации вариантов СС для тех компонентов СС, функции которых имитирует $AGISP_{ij}$ — по функциям распределения во время l -й реализации ИМ СС формируются соответственно конкретные значения запросов ресурсов вероятностной природы: mt_{kijl} , KO_{kijl} , st_{ijl} , τ_{ijl} .

Новым в формализации агрегатно-процессным способом имитации является формирование сигналов двух типов:

— действительных (DS_i), которые всегда формируются на выходах $AGSIN_i$, и активизируют $AGISP_{ij}$;

— вероятностных (VS_i), когда только по одному из выходов $AGSIN_i$ формируются DS_i и группы фиктивных сигналов (FS_{ih}), здесь h — номера разветвлений одного из выходов $AGSIN_i$, которые формируются по вектору вероятностей $\{P_{ih}\}$.

Фиктивные сигналы не активизируют $AGISP_{ij}$, а только обходят алгоритм выполнения агрегата, поступая непосредственно на один из входов $AGSIN_j$. В общем случае используется 4 типа кустовых выходов агрегатов $AGSIN_i$:

тип 1 — формирует на всех h разветвлениях куста только действительные сигналы;

тип 2 — формируют по вектору $\{P_{ih}\}$ только один действительный сигнал, и остальное подмножество фиктивных сигналов (fS_{ih});

тип 3 — по специальному алгоритму формируется комбинация подмножеств сигналов $\{DS_i\}$ и $\{fS_{ih}\}$ только в том случае, когда в информационной части сигналов, пришедших на входы $AGSIN_i$, имеется признак $\pi\alpha_{ij}$ («была авария» во время выполнения $AGISP_{ij}$);

тип 4 — по другому специальному алгоритму формируется другая комбинация подмножеств действительных и фиктивных сигналов только в том случае, когда в информационной части сигналов, пришедших на вход $AGSTN_i$ имеется признак π_{ij} (был выход переменной U за допустимые пределы).

Наличие выходов 3 и 4 типа позволяет зарезервировать некоторые агрегаты $AGISP_{ij}$, которые скорректируют оперативным образом последствия аварий в ходе имитации ВТП или же с помощью других $AGISP_{ij}$ осуществится возврат переменной U в допустимые пределы.

Для автоматизации создания и эксплуатации ИМ СС на основе агрегатно-процессного способа имитации была разработана система моделирования САИМ [14], три версии которой позволяют моделировать ВТПП трех типов: последовательного типа (САИМ-1), сложных логических схем управления ВТП2 (САИМ-2); параллельно-последовательного типа (САИМ-3). Результаты апробации версий САИМ приведены в монографии [14].

Как следует из работ [6, 14], сочетание двух способов имитации существенно расширили возможности проектного моделирования СС. Однако каждый из этих способов имеет свои ограничения. Опыт разработки и эксплуатации этих САИМ определил актуальность разработки нового универсального способа имитации, который позволяет, по нашему мнению, организовать проектное моделирование СС различной структуры и уровня сложности без существенных ограничений для современных высокопроизводительных ЭВМ.

Основная идея универсального способа имитации

Идея универсального способа имитации состоит в совмещении всех способов формализации СС в одном тексте ИМ СС. Например, если СС имеет графовую структуру, то компонентами могут быть *статические элементы* (СТЭЛ_{*j*}), находящиеся на узлах графа, или элементы связи разных вершин графа друг с другом. Связи между *статическими элементами* осуществляются либо информационно через общую базу данных ИМ СС, либо с помощью *динамических элементов* (ДЭЛ_{*in*}), которые перемещаются по ИМ СС от одного статического элемента (СТЭЛ_{*i*}) к множеству других статических элементов. Причем, к одному СТЭЛ_{*n*} может поступать любое число $\{ДЭЛ_{in}\}$, и после имитации функций соответствующих компонентов СС в модельном времени t_0 могут формироваться на выходе СТЭЛ_{*n*} любое число $\{ДЭЛ_{in}\}$, здесь i и n — номера динамических элементов. Как видим, все ДЭЛ_{*ij*} пронумерованы таким образом, что не может быть двух ДЭЛ_{*i*} с одним и тем же номером. Аналогичным образом множество СТЭЛ_{*j*} также пронумеровано, не допуская при этом одинаковых номеров у СТЭЛ_{*j*}.

В ИМ СС могут одновременно присутствовать следующие типы СТЭЛ_{*j*}:

тип 1 — агрегаты классического типа (например, кусочно-линейные агрегаты в терминологии Н.П. Бусленко [1]);

тип 2 — процессы классического типа (в терминологиях языка моделирования

СИМУЛА [4]);

тип 3 — обслуживающие устройства, представляющие собой разные модификации систем массового обслуживания (в терминах классического транзактного способа имитации на языке моделирования GPSS [5]);

тип 4 — процессы со сложным алгоритмом обслуживания SPP_j транзактов сложной структуры STR_{ij} (в терминах системы моделирования MICIC [6]);

тип 5 — агрегаты со сложным алгоритмом (SAG_j) обслуживания сигналов сложной структуры $SSig_{ij}$ (в терминах системы моделирования САИМ [14]);

тип 6 — универсальные статические элементы ($UNЭЛ_j$) со сложным алгоритмом обслуживания поступающих на них множества $\{ДЭЛ_{ij}\}$ и формирующих на выходе $UNЭЛ_j$ после имитации компонента СС множество $\{ДЭЛ_{in}\}$.

Аналогично в ИМ СС могут присутствовать следующие типы динамических элементов:

тип 1 — сигналы простой структуры Sig_{ij} для связи агрегата i с агрегатом j (в терминологии О.Н. Бусленко [2, 3]);

тип 2 — команды управления процессами KOM_{ki} (в терминологии САМ СИМУЛА [4]);

тип 3 — транзакты классического типа TR_{ij} (в терминологии САМ GPSS [5]);

тип 4 — транзакты сложной структуры STR_{ij} (в терминологии САМ MICIC [6]);

тип 5 — сигналы сложной структуры $SSig_{ij}$ для связи агрегата синхронизации i с агрегатом j (в терминологии САМ САИМ [14]);

тип 6 — информационные сообщения (SOB_{ij}) от $UNЭЛ_i$ к $UNЭЛ_j$.

Таким образом, структура ИМ СС формируется только из множества статических элементов $\{СТЭЛ_j\}$ и обеспечения между ними связей с помощью множества динамических элементов $\{ДЭЛ_{in}\}$. Исследователю необходимо выбрать способ формализации СС и заменить j -е компоненты СС соответствующими $СТЭЛ_j$. В зависимости от способа формализации СС необходимо указать номер типа $СТЭЛ_j$ и выбрать соответствующие типы $ДЭЛ_{in}$. В тех случаях, когда СС формализуется несколькими способами, исследователь должен предусмотреть соответствие типов $ДЭЛ_{in}$ и $СТЭЛ_j$. Для имитации расхода ресурсов СС исследователи в качестве исходной информации должны задать в БД для каждого типа $СТЭЛ_j$ состав запросов ресурсов ИМ СС, используемых для обслуживания $ДЭЛ_{in}$. В ряде случаев часть запросов ресурсов, расходуемых $СТЭЛ_j$ при имитации операций, обслуживания $ДЭЛ_{ij}$, может заключаться в информационной части $ДЭЛ_{in}$. Статистика обслуживания $ДЭЛ_{in}$ и расхода ресурсов ИМ СС также накапливаются в информационной части $ДЭЛ_{in}$. Кроме того, исследователь может запрограммировать в информационной части $ДЭЛ_{in}$ порядок выбора $ДЭЛ_{in}$ номера тех $СТЭЛ_n$, на которые $ДЭЛ_{in}$ поступает по окончании имитации его обслуживания алгоритмом $СТЭЛ_n$. Это дополнительная возможность организации $ДЭЛ_{in}$ в ИМ СС, поскольку некоторые типы $СТЭЛ_j$ могут обладать сложным алгоритмом формирования на выходе $ДЭЛ_{in}$ за счет использования кустовых выходов $СТЭЛ_j$ 2÷4 типа. С помощью шестого способа имитации СС исследователь может организовать одновременное выполнение несколько типов ИМ СС (агрегатных, транзактных, процессных). С нашей точки зрения, это основное преимущество универсального способа имитации. Ибо при использовании предыдущих способов имитации чрезвычайно

сложно организовать идентичность операционной обстановки в СС из-за необходимости использования процедуры Монте–Карло при проектном моделировании ВТП.

Пример представления дискретных сложных систем с помощью универсального способа имитации

На рис. 1 приведен пример ВТП, имеющего графовую структуру (GRCC1), для случая, когда его компоненты представляют собой СТЭЛ_{*j*} нескольких типов, которые взаимодействуют друг с другом с помощью ДЭЛ_{*ij*} нескольких типов. Нумерация ДЭЛ_{*ij*} — двойная, где *i* — номер СТЭЛ_{*i*} инициатора создания ДЭЛ_{*ij*}, а *j* — номер СТЭЛ_{*j*} адресата ДЭЛ_{*ij*}, активизирующего имитацию соответствующего компонента ВТП. На входе СТЭЛ_{*j*} типа 4 имеется очередь накопления STR_{*ij*} номера *j*, служащая для организации дисциплины обслуживания STR_{*ij*} статическим элементом 4-го типа номера *j* и синхронизации динамики его взаимодействия СТЭЛ_{*i*} и СТЭЛ_{*j*}. Инициатором цикла имитации согласно процедуре Монте–Карло является агрегат СТЭЛ₁, а завершается *l*-й цикл процедуры Монте–Карло по окончании процесса СТЭЛ₂₃. Если цикл имитации СС не завершен ($1 < N_M$, где N_M — число реализации ВТП), то имеет место повторная инициализация СТЭЛ₁ с помощью ДЭЛ_{23,1}, являющегося сложным сигналом повторной активизации агрегата СТЭЛ₁. На рис. 1 указан номер типа СТЭЛ_{*j*} (в данном графе используются 5 типов СТЭЛ_{*j*} (1, 2, 4, 5 и 6)). В очередях ОЧ₁₂ – ОЧ₁₆ к процессам поступают транзакты сложной структуры STR_{*ij*}. СТЭЛ_{*j*} типа 2, являющиеся процессами, имитируют взаимодействие агрегатов и процессов ВТП с оборудованием СС. Это взаимодействие реализуется системой команд управления работой процессов и агрегатов (пары команд запуска процесса), и затем по окончании имитации функций оборудования обратный запуск агрегатов или процессов, активизировавших СТЭЛ_{*j*} типа 2.

На входы трех агрегатов синхронизации поступают с выходом агрегатов исполнителей следующие динамические элементы:

- на СТЭЛ₇ поступают ДЭЛ_{26,7}, ДЭЛ_{3,7}, ДЭЛ_{4,7} соответственно с выходом элементов исполнителей СТЭЛ₂₆, СТЭЛ₃, СТЭЛ₄;
- на СТЭЛ₁₄ поступают ДЭЛ_{9,14}, ДЭЛ_{10,14}, ДЭЛ_{11,14} соответственно с выходом элементов исполнителей СТЭЛ₉, СТЭЛ₁₀, СТЭЛ₁₁;
- на СТЭЛ₂₂ поступают ДЭЛ_{18,22}, ДЭЛ_{19,22}, ДЭЛ_{20,22}, с ДЭЛ_{21,22} с выходом элементов исполнителей СТЭЛ₁₈, СТЭЛ₁₉, СТЭЛ₂₀, СТЭЛ₂₁.

Все перечисленные ДЭЛ_{*ij*} относятся к 5-му типу SSig_{*ij*}. Как видим, верхняя часть GR ИМ СС представляет собой попытку формализации части компонентов СС агрегатно-процессным способом имитации. Нижняя часть GR ИМ СС представляет собой формализацию другой части компонентов СС транзактно-процессным способом. В нее входят следующие группы процессов, представляющие собой обслуживающие устройства ОУ_{*j*} типа СМО_{*j*} со сложным алгоритмом имитации выделения ресурсов СС по заказу элементов четвертого типа ДЭЛ_{*ij*}, которые также информационно «подкрашены». Во множество {ОУ_{*j*}} входят следующие статические элементы: СТЭЛ₁₂, СТЭЛ₁₃, СТЭЛ₁₆. Каждый из этих СТЭЛ_{*ij*} имеет на входе по одной очереди (ОЧ₁₂, ОЧ₁₃, ОЧ₁₆), на которых осуществляется накопление сложных транзактов STR_{*ij*} в тех случаях, когда СТЭЛ_{*j*} занят обслуживанием предыдущего STR_{*ij*}.

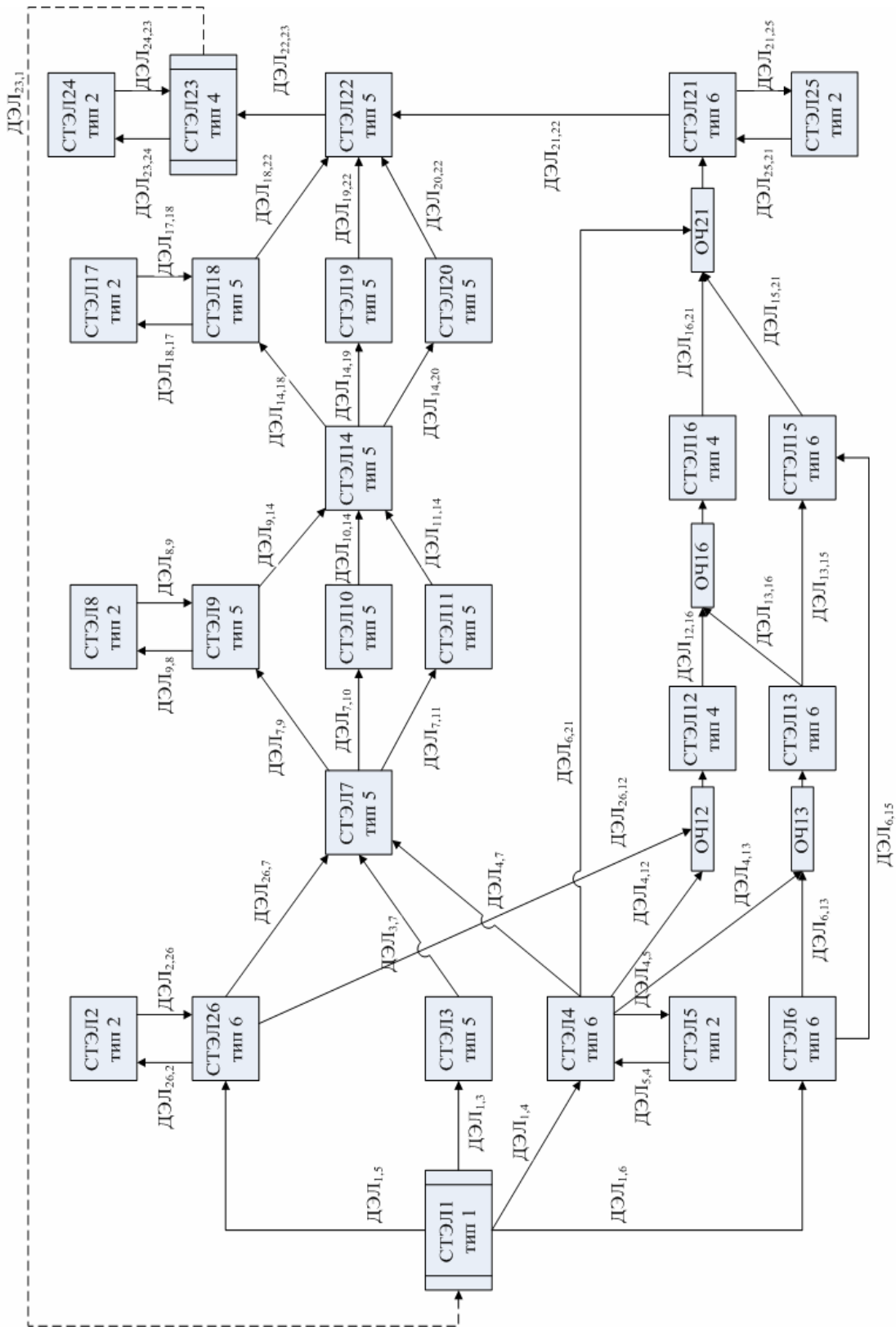


Рис. 1. Пример графа ИМ ВПП, состоящего из $\{СТЭЛ_i\}$ и $\{ДЭЛ_{ij}\}$ в среде SAM UNIVERSAL

Можно выделить СТЭЛ_j 6-го типа, поскольку их входы одновременно могут поступать ДЭЛ_{ij} двух типов, а на выходе они также формируют несколько типов ДЭЛ_{ij}. Это означает, что на входе исполнительных процессов или агрегатов типа 6 может поступать любая комбинация SSig_{ij} и STR_{ij}. После имитации операции расхода ресурсов соответствующим агрегатом или процессом формируется комбинация сигналов и транзактов сложной структуры. Из рис. 1 видно, что в состав СТЭМ_j типа 6 входят агрегаты (СТЭЛ₂₆, СТЭЛ₆) и процессы (СТЭЛ₆, СТЭЛ₆, СТЭЛ₂₁). Данный пример ИМ ВТП демонстрирует такие возможности универсального способа имитации, как сочетание в одном тексте ИМ ВТП двух подмоделей, представленных двумя способами формализации: верхняя часть графа — это, по своей сути, является сетевым графиком выполнения операций в ВТП [14], а нижняя часть графа — это блок-схема представления части ВТП транзактно-процессным способом [6]. Кроме того, за счет использования ДЭЛ_{23.1} обеспечивается очередной цикл имитации согласно процедуре Монте–Карло. А без этого цикла нельзя имитировать ВТП. Еще одна возможность реализована в этой ИМ СС: организация управления оборудованием с помощью агрегатов (СТЭЛ₂₆, СТЭЛ₉, СТЭЛ₁₈) и процессов (СТЭЛ₄, СТЭЛ₆, СТЭЛ₂₅, СТЭЛ₂₃) на основе использования ДЭЛ_{ij} 2-го типа.

Система автоматизации моделирования UNIVERSAL

Для реализации нового способа имитации СС реализуется САМ UNIVERSAL (рис. 2), состоящая из следующих компонентов:

- множества библиотек программ статических элементов 6 типов (LIB.СТЭЛ_j 1-го типа ÷ LIB.СТЭЛ_j 6-го типа);
- множества библиотек программ динамических элементов (LIB.ДЭЛ_{ij} 1-го типа ÷ LIB ДЭЛ_{ij} 6-го типа);
- блока формирования структуры ИМ СС;
- блока задания начальных условий (БНУ) в базах данных ОБД₁ и ОБД₂;
- оперативного блока программы множества {СТЭЛ_j} (БЛОПСТЭЛ);
- блока окончания имитации (БОКИ);
- подпрограмм вторичной обработки статистики имитации (ПП ВТОР. ОБРАБ СТАТИМ);
- блок сбора статистик (БССТ₁), формируемой самими СТЭЛ_j и ДЭЛ_{ij} в каждой реализации процедур Монте–Карло;
- поля сбора статистики имитации (ПСТАТ);
- общей базы данных (ОБД₁), в которой хранятся информационные части СТЭЛ_j (для каждого типа СТЭЛ_j выделяется своя область 1–6);
- общей базы данных (ОБД₂), в которой хранятся информационные части ДЭЛ_{ij} (для каждого типа ДЭЛ_{ij} выделяется своя область 1–6);
- управляющей программы моделирования (УПМ).

Каждая библиотека программы СТЭЛ_j любого типа разработана по одной и той же технологии, используя принцип реентерабельности выполнения программы и ее информационной части в соответствующей области ОБД₁. Одна программа СТЭЛ_j любого типа может использовать одновременно в модельном времени столько информационных частей, сколько имеется в ИМ СС копий СТЭЛ_j данно-

го типа. В любой момент времени модельного времени t_0 выполняется на процессоре ЭВМ только одна активность $АКТ_{kj}$ (здесь k — номера активностей в программе $СТЭЛ_j$). Эти активности квазипараллельно обслуживаются УПМ в порядке

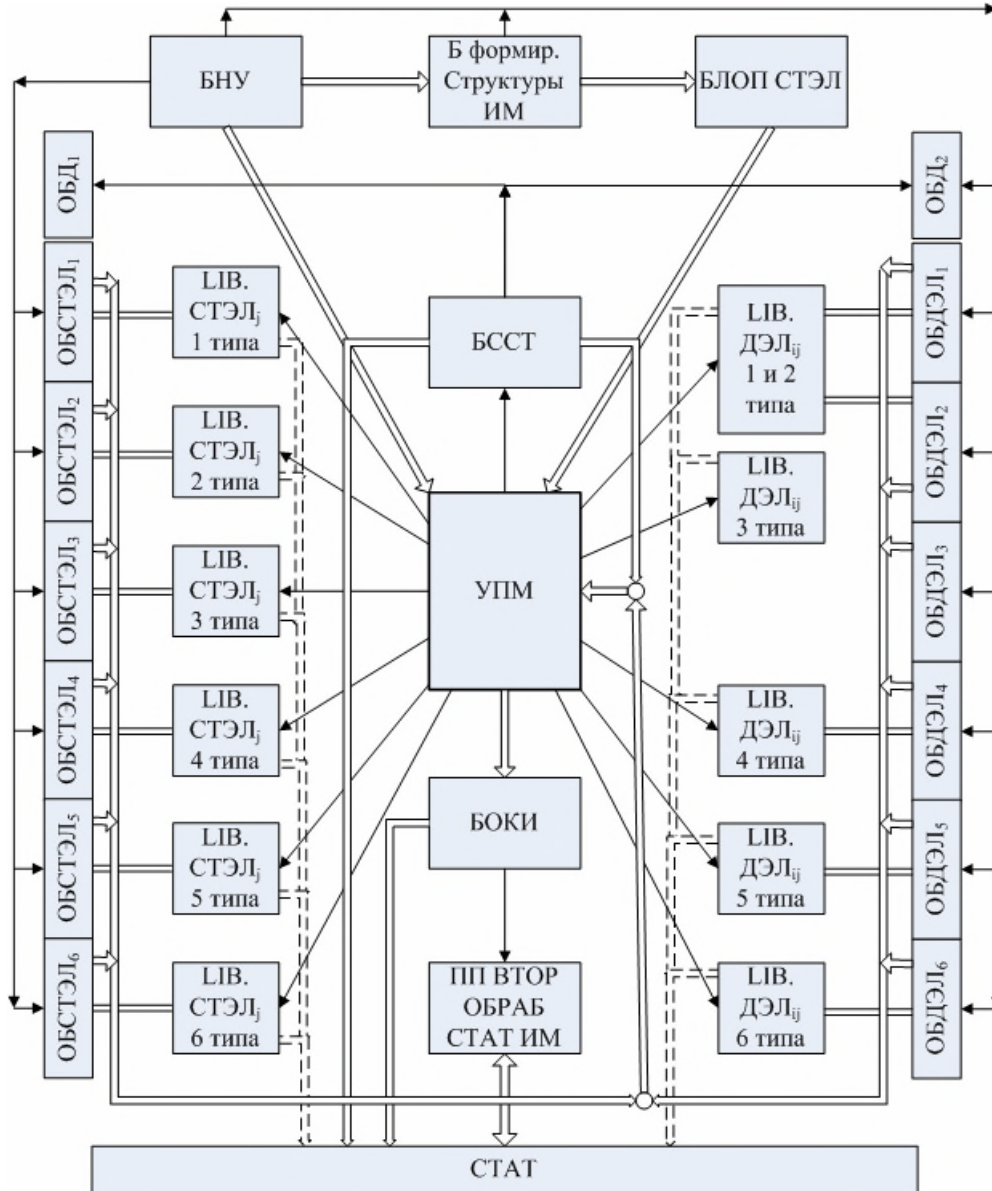


Рис. 2. Структура SAM UNIVERSAL

приоритета. Выбрав k -й номер $АКТ_{kj}$, УПМ активизирует алгоритм этой активности, имитируя выполнения алгоритма той компоненты СС, которой соответствует $СТЭЛ_j$. Алгоритм $АКТ_{kj}$ в соответствии со способом формализации компонент СС меняет переменные, расходует ресурсы ИМ СС, формирует статистику использования ресурсов, статистику использования $СТЭЛ_j$ в $ОБД_1$ и статистику обслуживания $ДЭЛ_{ij}$ в $ОБД_2$. В конце концов, завершение $АКТ_{kj}$ управления возвращается на УПМ. Далее, согласно своему алгоритму, все $СТЭЛ_j$ и $ДЭЛ_{ij}$, обслуживают в

соответствии с приоритетами π_{2j} динамических и статических элементов. УПМ активизирует другую либо $СТЭЛ_j$, либо $ДЭЛ_{ij}$ (иногда даже той же самой программы, если число копий $СТЭЛ_j$ и $ДЭЛ_{ij}$ в $ОБД_1$ и $ОБД_2$ большое). Связь $СТЭЛ_j$ с другими $СТЭЛ_j$ осуществляет только через УПМ. Для каждого типа $СТЭЛ_j$ имеет свое подмножество $ДЭЛ_{ij}$ соответствующего типа.

Очевидно, что в $ОБД_2$ имеется своя область для каждого типа $ДЭЛ_{ij}$. Таким образом, за счет реентерабельности $ДЭЛ_{ij}$ одна и та же программа $ДЭЛ_{ij}$ из соответствующей библиотеки может обслуживать множество копий информационных частей. При этом одновременно в каждой из областей $ОБД_2$ может использоваться часть программы $ДЭЛ_{ij}$, формироваться статистика использования копий $ДЭЛ_{ij}$ и накапливаться статистика расхода ресурсов и изменения локальных переменных $ДЭЛ_{ij}$. Как видим, одна программа каждого типа $СТЭЛ_j$ обслуживает множество активностей $АКТ_{kj}$, и одна программа каждого типа $ДЭЛ_{ij}$ также обслуживает соответствующие ей множество информационных частей. Принцип реентерабельности $СТЭЛ_j$ и $ДЭЛ_{ij}$ существенно сокращает размеры ИМ СС и, кроме того, позволяет их разработать навсегда на языке программирования СИ разработчиками *SAM UNIVERSAL*, являющимися специалистами по системному программированию и прикладной математике. Поэтому после отладки программ $СТЭЛ_j$ и $ДЭЛ_{ij}$ они каталогизируются в соответствующие библиотеки (см. рис. 2) $ЛІВ СТЭЛ_j$ типа 1÷6 и $ЛІВ ДЭЛ_{ij}$ типа 1÷6.

Блок формирования структуры ИМ СС по исходной информации о составе и структуре компонентов ИМ СС, используя таблицы коммутации $СТЭЛ_j$ друг с другом, формирует структуру баз данных $ОБД_1$ и $ОБД_2$. Кроме того, по таблице исходного состава ресурсов в варианте ИМ СС в таблице коммутации составляется блок операционных программ БЛОПСТЭЛ множества $\{СТЭЛ_j\}$, в который входит по одной программе для каждого типа $СТЭЛ_j$.

Блок БНУ обеспечивает заполнение в $ОБД_1$ таблиц заказов ресурсов ИМ СС исходной информации, указывая при этом необходимое число детерминированных ресурсов (n_1-n_6) и записывает функции распределения вероятностей ресурсов ($F_{1j}(\tau) \div F_{4kj}(КО)$) для каждого $СТЭЛ_j$. Другой функцией БНУ является задание начальной структуры информационной части $ДЭЛ_{ij}$ в $ОБД_2$.

Блок окончания имитации инициируется УПМ при выполнении условий завершения варианта имитации СС, формирует статистику имитации из буферных зон статистики использования $\{СТЭЛ_j\}$ в $ОБД_1$ и статистики использования $\{ДЭЛ_{ij}\}$ в $ОБД_2$. Кроме того, стандартная статистика имитации, формируемая блоком $БССТ_2$, записывается в поле статистик также блоком окончания имитации (БОКИ).

Состав стандартной статистики фиксирован, и блок сбора статистики ($БССТ$), взаимодействуя конкретно с УПМ, формирует обычный набор статистик использования $\{СТЭЛ_j\}$ и $\{ДЭЛ_{ij}\}$ за время имитации одной реализации процедуры Монте-Карло, записывая ее в поле статистики (ПСТАТ).

УПМ является координатором взаимодействий $\{СТЭЛ_j\}$ друг с другом и использования ими $\{ДЭЛ_{ij}\}$. Каждые $ДЭЛ_{ij}$ и $СТЭЛ_j$ взаимодействуют с УПМ с помощью своего набора операторов моделирования. Каждая активность $АКТ_{kj}$ $СТЭЛ_j$ завершается каким-либо оператором взаимодействия с УПМ. Из-за необходимости обеспечить одновременно квазипараллелизм обслуживания УПМ этих

операторов и использования принципа реентерабельности программ СТЭЛ_j и программ ДЭЛ_{ij} в каждом операторе указывается следующая информация: номера j для СТЭЛ_j или ij для ДЭЛ_{ij}; номера k активностей АКТ_{kj} адреса β_i информационной части СТАЛ_j в ОБД₁; адреса α_i информационной части ДЭЛ_{ij} в ОБД₂; адреса возврата А на программу СТЭЛ_j; адреса А₁ на ту часть УПМ, которая обслуживает данную активность или программу ДЭЛ_{ij}, заказы которых необходимо выполнить. Наконец, в операторах обращения к УПМ указывается: время имитации АКТ_{kj} (τ_{kj}) или время имитации выполнения ДЭЛ_{ij} (τ_{ij}); адрес условия завершения оператора (АУ_k), где k — номер процедуры пользователя в поле заказа на имитацию хранящемся в ОБД₁. Столь сложный алгоритм обращения АКТ_{kj} и ДЭЛ_{ij} к УПМ необходим для обеспечения квазипараллелизма по событийному способу имитации и для организации реентерабельности программ каждого типа СТЭЛ_j и ДЭЛ_{ij}.

Безусловно, все программы СТЭЛ_j и ДЭЛ_{ij} стандартизованы по типам агрегатов или процессов. В первую очередь, перед активизацией УПМ агрегата или процесса в алгоритме СТЭЛ_j формируется заказ на его выполнение. При этом возможен режим захвата ресурсов СТЭЛ_j и, при отсутствии свободных ресурсов, возможно ожидание выполнения СТЭЛ_j до появления требуемых ресурсов ИМ СС.

Заключение

Итак, САМ UNIVERSAL реализует возможности имитации дискретных сложных систем (ДСС) за счет совмещения в «теле» одной программы ИМ ДСС нескольких способов формализации ее компонентов. Как было показано на примере ВТП (рис. 1), возможно сочетание двух подмоделей, основанных на разных принципах имитации: агрегатно-процессного способа имитации (верхняя часть рис. 1) и транзактно-процессного способа имитации (нижняя часть рис. 1). Кроме того, использование принципа реентерабельности программ СТЭЛ_j и программ ДЭЛ_{ij} позволяет существенно сократить объем оперативной памяти ЭВМ, занимаемой множеством элементов ИМ ДСС. Тот факт, что увеличивается собственно время имитации на ЭВМ, не имеет существенного значения из-за возросшей производительности современных компьютеров. С нашей точки зрения, это допустимая плата расхода дополнительных ресурсов ЭВМ за существенное сокращение размеров программ.

Весьма важное достоинство данной САМ состоит в том, что программы СТЭЛ_j и программы ДЭЛ_{ij} разработаны на языке СИ навсегда специалистами по системному программированию и имитации, и поэтому исследователю, согласно инструкции, нужно только составить граф GR ДСС, заменив при этом узлы на соответствующие типы СТЭЛ_j и выбрав при этом типы ДЭЛ_{ij} для каждой ветви этого графа. В таблице заказа расхода ресурсов каждым СТЭЛ_j указывается требуемое количество детерминированных ресурсов (n_1-n_6) и средств функций распределения вероятностей значений расхода вероятностных ресурсов ($F_{1j}(\tau) \div F_{4kj}(k_0)$). Никакого программирования специалистам предметной области не требуется. Необходимо только правильно указать таблицу коммутации СТЭЛ_j и ДЭЛ_{ij} и задать начальное значение 10-ти типов ресурсов ИМ ДСС.

Наконец, существенным достоинством САМ UNIVERSAL является органическое объединение известных способов формализации разных частей ИМ ДСС с

процедурой Монте–Карло метода статистических испытаний. Перечисленные достоинства SAM UNIVERSAL, по нашему мнению, обеспечат перспективу развития использования и дальнейшего развития ее возможностей на основе результатов ее апробации.

1. *Бусленко Н.П.* Моделирование сложных систем / Н.П. Бусленко. — М.: Наука, 1978. — 400 с.
2. *Бусленко Н.П.* Автоматизация имитационного моделирования сложных систем / Н.П. Бусленко. — М.: Наука, 1977. — 240 с.
3. *Праневичус Г.* Модели и методы исследования вычислительных систем / Г. Праневичус. — Вильнюс: Моклас, 1982.—227с.
4. Система моделирования на базе языка Симуна–67 для БЭСМ–6 и ЕСЭВА / [А.И. Андрианов, С.П. Бычков, Г.В. Попкова и др.] // Моделирование дискретных управляющих и вычислительных систем. Тез. 3-го Всесоюзного семинара. — Свердловск, 1981. — С. 44–45.
5. *Шрайбер Т.Дж.* Моделирование на DPSS / Т.Дж. Шрайбер. — М.: Машиностроение. — 519 с.
6. *Левчук В.Д.* Программно-технологические комплексы имитации дискретных систем: монография / В.Д. Левчук, И.В. Максимей. — М-во образования, Гомельский Госуниверситет им. Ф. Скорины. — Гомель, 2006. — 261 с.
7. *Максимей И.В.* Имитационное моделирование на ЭВМ / И.В. Максимей. — М.: Радио и связь, 1988. — 232 с.
8. Программно-технологический комплекс имитационного моделирования многопроцессорных вычислительных систем / [О.В. Быченко, В.А. Никишаев, М.В. Потрашкова и др.] // «Веснік»/ — Гродненский дзярж. ун-т ім. Янкі Купалы. — 2005. — № 1(31). — С. 96–106.
9. *Никишаев В.А.* Рациональный выбор организации узла ЛВС с помощью средств мониторинга и имитационного моделирования вычислительного процесса и рабочей нагрузки: автореф. дис. на соискание учен. степени канд. техн. наук: спец. 05.13.13 «Телекоммуникационные системы и компьютерные сети» // Никишаев В.А. — Гомель: Гомельский Госуниверситет им. Ф. Скорины, 2004. — 22 с.
10. *Быченко О.В.* Метод, средства и технология адаптации рабочей нагрузки к составу ресурсов узла ЛВС: автореф. дис. на соискание учен. степени канд. техн. наук: спец. 05.13.13 «Телекоммуникационные системы и компьютерные сети». — Гомель: Гомельский Госуниверситет им. Ф. Скорины, 2006. — 22 с.
11. *Демиденко О.М.* Проектное моделирование вычислительного процесса в локальных вычислительных сетях / О.М. Демиденко, И.В. Максимей. — Мн.: Белорусская наука, 2001. — 252 с.
12. *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.
13. *Демиденко О.М.* Методы и средства исследования и адаптации вычислительного процесса под рабочую нагрузку на локальную вычислительную сеть: автореф. дис. на соискание учен. степени докт. техн. наук: спец. 05.13.13 «Телекоммуникационные системы и компьютерные сети». — Минск: БГУ, 2004.
14. *Сморозин В.С.* Методы и средства имитационного моделирования технологических процессов производства / В.С. Смородин, И.В. Максимей. — М-во образования, Гомельский Госуниверситет им. Ф. Скорины. — Гомель, 2007. — 369 с.

Поступила в редакцию 10.06.2009