

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УДК 681.324

Н.В. Скворцов, С.В. Скворцов**АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ
ПРОЦЕССОВ САМОДИАГНОСТИКИ
ДЛЯ МНОГОПРОЦЕССОРНЫХ СИСТЕМ
С АКТИВНОЙ ОТКАЗОУСТОЙЧИВОСТЬЮ**

Разработаны алгоритмы динамического планирования параллельных диагностических вычислений в отказоустойчивых многопроцессорных системах реального времени. Показано, что применение коллективных элементарных проверок в процессе вычислений позволяет повысить эффективность самодиагностики за счет более равномерной загрузки вычислительных модулей по сравнению с парными элементарными проверками модулей.

Ключевые слова: многопроцессорная система, отказоустойчивость, диагностический граф, элементарная проверка.

Введение. Современные вычислительные системы параллельной обработки данных содержат достаточно большое число вычислительных модулей (ВМ) и по сути своего построения являются отказоустойчивыми [1, 2], поскольку при работе в режиме реального времени часть ВМ периодически бездействует из-за алгоритмических особенностей прикладных задач, не позволяющих подготовить для них исходные данные к моментам окончания решения предыдущих частей решаемой задачи [2, 3]. Это дает возможность использовать незагруженные решением основной задачи ВМ для организации активной отказоустойчивости системы путем решения некоторых частей задачи одновременно парами ВМ.

Активная отказоустойчивость представляет собой один из методов построения самодиагностируемых многопроцессорных систем, который базируется на отдельно выделенных процессах автоматического обнаружения отказа, его локализации и реконфигурации системы с последующим восстановлением вычислительного процесса в реальном времени [3, 4].

Активная отказоустойчивость основана на динамическом перераспределении вычислительных ресурсов многопроцессорной системы в процессе решения прикладной задачи. В системе выделяется некоторое число дополнительных вычислительных модулей (ВМ), которые в соот-

ветствии с определенной дисциплиной [2, 5] подключаются к основным модулям на некоторые периоды времени, называемые тактами контроля, и образуют пары ВМ, дублирующие вычисления. В каждой паре выделяют контролируемый (основной) и контролирующий (дополнительный) ВМ, а сравнение результатов, полученных этими модулями (например, некоторых контрольных сумм или непосредственно промежуточных данных), реализует одну элементарную проверку (ЭП).

Заметим, что состав основных и дополнительных ВМ может изменяться в соответствии с реализуемой дисциплиной активной отказоустойчивости или вследствие деградации системы из-за отказов отдельных ВМ [5].

Структура всех проверочных связей, реализуемых за цикл контроля, т.е. за период времени между повторением ЭП для совпадающих пар модулей, описывается с помощью диагностического графа (ДГ). Вершинам ДГ $G=(U, D)$ соответствует множество ВМ системы $U = \{u_1, u_2, \dots, u_n\}$, а ребра задают пары модулей, дублирующих вычисления. С каждым ребром $[u_i, u_j] \in D$ сопоставляется результат $s_{ij} \in \{0, 1\}$ одной ЭП контролирующим модулем u_i контролируемого модуля u_j . Совокупность всех значений s_{ij} , полученных за цикл контроля, образует синдром S системы.

Дешифрация синдрома выполняется в рамках выбранной диагностической модели [6, 7], которая определяет зависимость результатов взаимной проверки модулей от их состояний и позволяет идентифицировать отказавшие ВМ системы [8, 9].

Отдельную задачу составляет синтез ДГ, обеспечивающих в рамках используемой диагностической модели необходимый уровень самодиагностируемости системы при минимальном числе выполняемых ЭП. Частные решения этой задачи для симметричной диагностической модели даны в работах [6, 7, 10 - 12].

В данной статье рассматриваются вопросы реализации активной защиты многопроцессорной системы в режиме реального времени с учетом текущей загрузки ВМ при выполнении прикладной задачи.

Цель работы заключается в создании алгоритмов динамического планирования диагностических вычислительных процессов, которые обеспечивают выполнение множества ЭП, заданных диагностическим графом, для известного алгоритма работы многопроцессорной системы, представленного диаграммой загрузки [13] вычислительных модулей, с учетом незанятых в текущий момент ВМ системы.

Теоретические исследования. Диаграмма загрузки описывает расписание выполнения вычислительными модулями $u_i \in U$ системы некоторых логических частей $a_j \in A$ исходной задачи, возможно, имеющих разное время τ_j реализации и называемых функциональными операторами [13, 14].

Предположим, что система работает непрерывно с циклическим повторением основных вычислений, а продолжительность τ^* такта контроля соответствует длительности шага планирования вычислений и определяется минимальным временем $\tau^* = \min \tau_j, a_j \in A$, исполнения оператора исходной задачи. Тогда оператор $a_j \in A$ может быть разбит на $m_j = \tau_j / \tau^*$ фрагментов $a_j^{(k)}, k = \overline{1, m_j}$, каждый из которых выполняется в одном такте работы системы, соответствующем по длительности такту контроля.

Интенсивность использования ВМ в процессе функционирования системы количественно можно оценить функцией плотности загрузки

[1], которая определяется как $F(t) = \sum_{i=1}^n f_i(t)$, где

$f_i(t) = 0$, если в момент времени t вычислительный модуль $u_i \in U$ простаивает, и $f_i(t) = 1$, если этот ВМ выполняет решение некоторого фрагмента задачи.

Нетрудно заметить, что в каждый момент

времени t разность $n - F(t)$ показывает число незагруженных ВМ, которые могут использоваться для выполнения взаимных ЭП путем решения копий фрагментов задачи параллельно с другими ВМ таким образом, чтобы образовать пары, определяемые ребрами ДГ.

В качестве примера рассмотрим отказоустойчивую систему, в состав которой входит $n = 4$ идентичных ВМ, причем для решения основной задачи в каждый момент времени используется не более трех модулей, диаграмма загрузки которых, заимствованная из работы [1], показана на рисунке 1,а. Каждая строка диаграммы описывает расписание работы одного ВМ, где числа в прямоугольниках указывают номера операторов задачи, время решения которых соответствует длине этих прямоугольников, а символом "x" обозначены моменты простоя ВМ. График соответствующей функции плотности загрузки $F(t)$ приведен на рисунке 1,б.

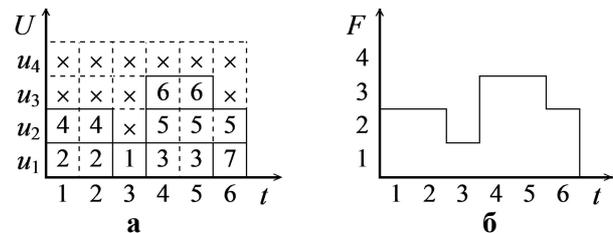


Рисунок 1 – Исходный план работы системы:
а – диаграмма загрузки ВМ;
б – функция плотности загрузки

Простейший алгоритм динамического планирования вычислений для очередного такта работы системы, обеспечивающий выполнение ЭП парами ВМ, производит следующие действия. Сначала определяется множество A_t фрагментов операторов задачи, которые должны быть выполнены в момент времени t , задаваемый номером такта работы системы, а затем с использованием диагностического графа $G = (U, D)$ выбираются необходимые пары ВМ для проведения ЭП и производится назначение выбранных фрагментов и их копий на вычислительные модули (процессоры) системы.

При этом каждое ребро $[u_i, u_j] \in D$ графа G может находиться в двух состояниях - просмотрено, если соответствующая ЭП для текущего цикла контроля уже выполнена, и не просмотрено – в противном случае, т.е. для момента времени t задано разбиение $D = D_t \cup \overline{D}_t$. Подмножество D_t содержит просмотренные, а \overline{D}_t - непросмотренные ребра диагностического графа G . В исходном состоянии $D_t = \emptyset$ и $\overline{D}_t = D$.

Для определения состава множества A_t ис-

пользуются известные из диаграммы загрузки значения $t_j \geq 1$ номеров тактов, в которых начинается выполнение операторов $a_j \in A$. Результат работы алгоритма планирования представляется набором переменных $p(u_i)$, фиксирующих номера фрагментов задачи, назначенных на соответствующие ВМ, причем $p(u_i) = 0$, если модуль u_i в очередном такте будет простаивать.

Алгоритм 1. Планирование параллельных вычислений при выполнении парных ЭП

Данные: значения переменных t_j для всех операторов $a_j \in A$; номер текущего такта $t \in \{1, 2, \dots, T\}$ работы системы, где $T = \max \{t_j + m_j\}$, $a_j \in A$; диагностический граф $G = (U, D)$, для которого $D = D_t \cup \overline{D}_t$.

Результаты: значения переменных $p(u_i)$ для всех $u_i \in U$.

1. Определить множество A_t по правилу $A_t = \{a_j^{(k)} : (t_j \leq t \leq t_j + m_j) \& (k = t - t_j + 1) = \text{true}\}$.
2. Присвоить $p(u_i) = 0$ для всех $u_i \in U$.
3. Определить количество $r = n - |A_t|$ модулей, которые могут использоваться для дублирования вычислений.
4. Если $\overline{D}_t = \emptyset$, то $\overline{D}_t = D$ и $D_t = \emptyset$.
5. Если $r = 0$ или $\overline{D}_t = \emptyset$, то выполнить переход к шагу 9.
6. Выбрать ребро $[u_i, u_l] \in \overline{D}_t$, такое, что $p(u_i) = 0$ и $p(u_l) = 0$. Если такое ребро не найдено, то переход к шагу 9.
7. Присвоить $p(u_i) = a_j^{(k)}$, $p(u_l) = a_j^{(k)}$, где $a_j^{(k)} \in A_t$, и изменить состав множеств: $A_t = A_t \setminus \{a_j^{(k)}\}$; $D_t = D_t \cup \{[u_i, u_l]\}$; $\overline{D}_t = \overline{D}_t \setminus \{[u_i, u_l]\}$.
8. Присвоить $r = r - 1$ и выполнить переход к шагу 5.
9. Если $A_t = \emptyset$, то переход к шагу 11.
10. Выбрать $a_j^{(k)} \in A_t$ и определить модуль $u_i \in U$, для которого $p(u_i) = 0$. Присвоить $p(u_i) = a_j^{(k)}$, $A_t = A_t \setminus \{a_j^{(k)}\}$, и переход к шагу 9.
11. Если $t = T$, то $t = 1$, иначе $t = t + 1$.
12. Конец алгоритма.

Для оценки эффективности использования ВМ в цикле контроля введем в рассмотрение функцию

$$L(k) = \sum_{x=1}^{N(k)} \sum_{i=1}^n I(u_i, x) / g, \quad (1)$$

которая показывает среднее количество фрагментов задачи (или их копий), решенных за k -й цикл контроля при проведении одной ЭП. В

формуле (1) использованы следующие обозначения: $g = |D|$ – общее число ЭП в цикле контроля; $I(u_i, x) = 1$, если в момент времени (такт) x ВМ u_i был задействован в некоторой ЭП и $I(u_i, x) = 0$ – в противном случае; $N(k)$ – число тактов в k -м цикле контроля.

Если в каждой ЭП используется пара ВМ, то функция $L(k) = L_p$ всегда принимает значение $L_p = 2$. Если же в некотором такте x один и тот же фрагмент задачи получают $s_x \geq 2$ ВМ, то можно провести $g_x = 0,5s_x(s_x - 1)$ сравнений полученных результатов, что будет соответствовать выполнению всех возможных ЭП между этими s_x ВМ. В этом случае функция $L(k) = L_c(k)$ принимает значение

$$L_c(k) = \sum_{x=1}^{N(k)} \frac{s_x}{g_x} = \sum_{x=1}^{N(k)} \frac{2}{s_x - 1}. \quad (2)$$

Нетрудно заметить, что для случая $s_x = 2$, соответствующего реализации парных ЭП, при любых значениях $N(k)$ по формуле (2) также получаем $L_c(k) = 2$. Однако, если $s_x > 2$, то $L_c(k) < 2$, что означает снижение трудоемкости диагностических процедур, поскольку при равном числе проверок в k -м цикле контроля, т. е. для $g = \sum_{x=1}^{N(k)} g_x$, всегда $L_c(k) < L_p$.

Такие ЭП, в которых $s_x > 2$ модулей одновременно решают один и тот же фрагмент задачи, будем называть *коллективными* элементарными проверками. На ДГ каждая коллективная ЭП отражается полным подграфом, построенным на s_x вершинах. При равных показателях $L_p = L_c$ загрузки системы дублирующими вычислениями применение коллективных ЭП позволяет повысить качество диагностики за счет увеличения общего числа проверок в цикле контроля, что следует из неравенства $g < \sum_{x=1}^{N(k)} g_x$.

Планирование загрузки ВМ на каждом такте работы системы при реализации коллективных ЭП может быть выполнено с помощью алгоритма 2, где использованы такие же обозначения, как и при описании алгоритма 1.

Алгоритм 2. Планирование параллельных вычислений при выполнении коллективных ЭП

Данные: значения переменных t_j для всех операторов $a_j \in A$; номер текущего такта $t \in \{1, 2, \dots, T\}$ работы системы, где $T = \max \{t_j + m_j\}$, $a_j \in A$; диагностический граф $G = (U, D)$, для которого $D = D_t \cup \overline{D}_t$.

Результаты: значения переменных $p(u_i)$ для всех $u_i \in U$.

1. Определить состав множества A_t .
2. Присвоить $p(u_i) = 0$ для всех $u_i \in U$.

3. Определить количество $r = n - |A_t|$ модулей, которые могут использоваться для дублирования вычислений в такте t .

4. Если $\overline{D}_t = \emptyset$, то $\overline{D}_t = D$ и $D_t = \emptyset$.

5. Если $r = 0$ или $\overline{D}_t = \emptyset$, то выполнить переход к шагу 9.

6. Выделить в графе $G = (U, D)$ связный подграф $G_0 = (U_0, D_0)$, удовлетворяющий следующим условиям:

$$U_0 = \{u_i \in U : p(u_i) = 0\}; |U_0| \leq r + 1;$$

$$D_0 \subseteq \overline{D}_t; |D_0| \rightarrow \max.$$

Если $D_0 = \emptyset$, то переход к шагу 9.

7. Для всех $u_i \in U_0$ присвоить $p(u_i) = a_j^{(k)}$, где $a_j^{(k)} \in A_t$, и изменить состав множеств:

$$A_t = A_t \setminus \{a_j^{(k)}\}; D_t = D_t \cup D_0; \overline{D}_t = \overline{D}_t \setminus D_0.$$

8. Присвоить $r = r - |U_0|$, и переход к шагу 5.

9. Если $A_t = \emptyset$, то переход к шагу 11.

10. Выбрать $a_j^{(k)} \in A_t$ и определить модуль $u_i \in U$, для которого $p(u_i) = 0$. Присвоить $p(u_i) = a_j^{(k)}$, $A_t = A_t \setminus \{a_j^{(k)}\}$, и переход к шагу 9.

11. Если $t = T$, то $t = 1$, иначе $t = t + 1$.

12. Конец алгоритма.

Следует отметить, что представленные алгоритмы 1 и 2 предполагают возможность исполнения фрагментов одного функционального оператора различными ВМ системы. Это является их общим недостатком, поскольку в данном случае возникает дополнительная нагрузка на информационные магистрали за счет пересылок промежуточных результатов и данных о текущих состояниях модулей. При этом очевидно, что такие фрагменты задачи $a_j^{(k)}$, $k = \overline{1, m_j}$, которые составляют один оператор $a_j \in A$, должны быть назначены на один и тот же ВМ, возможно, отличающийся от указанного в диаграмме загрузки системы.

В качестве примера реализации такого подхода к устранению указанного недостатка разработан алгоритм 3, который выполняет планирование параллельных вычислений для парных ЭП при жестком назначении основных ВМ, когда каждый оператор исходной задачи исполняется некоторым модулем в целом без прерываний и перезагрузки фрагментов.

В основе алгоритма лежит следующее правило. Оператор $a_j \in A$ жестко закрепляется за некоторым ВМ $u_i \in U$ сразу же, как только на этот модуль будет назначен первый фрагмент $a_j^{(1)}$ оператора a_j . Если таких модулей два (выполняется ЭП), то закрепляется ВМ с меньшим

номером. Следует отметить, что такое правило является достаточно простым и в более сложных случаях желательно анализировать свойства ДГ, например локальные степени и характер связей соответствующих вершин.

При описании алгоритма 3 дополнительно используются переменные $vm(a_j)$, принимающие значения $vm(a_j) = u_i$ и фиксирующие ВМ системы, жестко закрепляемые для исполнения соответствующих операторов $a_j \in A$ на период T выполнения прикладного алгоритма. Его общая длительность T определяется диаграммой загрузки модулей системы. При этом $vm(a_j) = 0$, если оператор a_j еще не назначен на исполнение. В исходном состоянии $vm(a_j) = 0$ для всех $a_j \in A$. Кроме того, все ВМ, жестко закрепленные для исполнения некоторых операторов до начала планирования в такте t , обозначены как подмножество $U_t \subseteq U$.

Алгоритм 3. Планирование параллельных вычислений при выполнении парных ЭП с жестким закреплением ВМ

Данные: значения переменных t_j для всех операторов $a_j \in A$; номер текущего такта $t \in \{1, 2, \dots, T\}$ работы системы, где $T = \max \{t_j + m_j\}$, $a_j \in A$; диагностический граф $G = (U, D)$, для которого $D = D_t \cup \overline{D}_t$.

Результаты: значения переменных $p(u_i)$ для всех $u_i \in U$.

1. Определить состав множества A_t .

2. Присвоить $U_t = \emptyset$, $p(u_i) = 0$ для всех $u_i \in U$.

3. Для каждого фрагмента $a_j^{(k)} \in A_t$, такого, что $k > 1$ ($t_j > t$), определить закрепленный модуль $u_i = vm(a_j)$, включить u_i в множество U_t и присвоить $p(u_i) = a_j^{(k)}$.

4. Определить количество $r = n - |A_t|$ модулей, которые могут использоваться для дублирования вычислений.

5. Если $\overline{D}_t = \emptyset$, то $\overline{D}_t = D$ и $D_t = \emptyset$.

6. Если $r = 0$ или $\overline{D}_t = \emptyset$, то выполнить переход к шагу 13.

7. Выбрать ребро $[u_i, u_l] \in \overline{D}_t$, такое, что $u_i \in U_t$ и $p(u_l) = 0$. Если такое ребро не найдено, то переход к шагу 9.

8. Определить фрагмент $a_j^{(k)} = p(u_i)$, закрепленный за ВМ $u_i \in U_t$, присвоить $p(u_l) = a_j^{(k)}$ и выполнить переход к шагу 11.

9. Выбрать ребро $[u_i, u_l] \in \overline{D}_t$, такое, что $p(u_i) = 0$ и $p(u_l) = 0$. Если такое ребро не найдено, то переход к шагу 13.

10. Присвоить $p(u_i) = a_j^{(k)}$, $p(u_l) = a_j^{(k)}$, где $a_j^{(k)} \in A_t$ и $k = 1$ ($t_j = t$). Если $i < l$, то присвоить $vm(a_j) = u_i$, иначе $vm(a_j) = u_l$.

11. Изменить состав множеств:

$$A_t = A_t \setminus \{a_j^{(k)}\}; D_t = D_t \cup \{[u_i, u_l]\};$$

$$\overline{D}_t = \overline{D}_t \setminus \{[u_i, u_l]\}.$$

12. Присвоить $r = r - 1$, и переход к шагу 5.

13. Если $A_t = \emptyset$, то переход к шагу 16.

14. Выбрать $a_j^{(k)} \in A_t$. Если $vm(a_j) = 0$, то определить модуль $u_i \in U$, для которого $p(u_i) = 0$, и присвоить $p(u_i) = a_j^{(k)}$, $vm(a_j) = u_i$.

15. Присвоить $A_t = A_t \setminus \{a_j^{(k)}\}$ и выполнить переход к шагу 13.

16. Если $t = T$, то $t = 1$ и $vm(a_j) = 0$ для всех $a_j \in A$, иначе $t = t + 1$.

17. Конец алгоритма.

Анализ предложенных алгоритмов. Оценим вычислительную сложность разработанных алгоритмов и покажем примеры их работы с целью анализа получаемых расписаний загрузки ВМ многопроцессорной системы.

Пример 1. Рассмотрим применение предложенного алгоритма 1 для организации вычислений в отказоустойчивой системе, исходная диаграмма загрузки которой представлена на рисунке 1,а. Пусть заданный диагностический граф имеет вид, приведенный на рисунке 2.

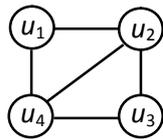


Рисунок 2 – Диагностический граф

Тогда результаты работы алгоритма 1 показаны на рисунке 3. Несложно убедиться, что все ЭП, заданные исходным ДГ, выполняются за 4 такта, а далее цикл контроля повторяется.

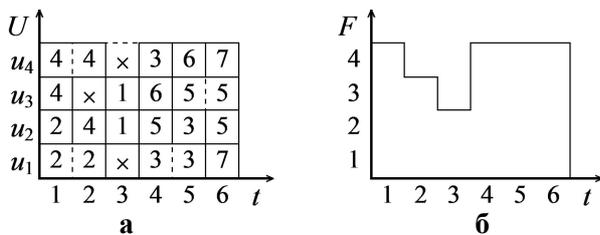


Рисунок 3 – Результаты работы алгоритма 1:
а – диаграмма загрузки ВМ;
б – функция плотности загрузки

В таблице 1 показаны процессы выполнения ЭП и синтеза ДГ во времени. За время решения задачи в целом выполняется восемь ЭП, но

$F(2) < 4$ и $F(3) < 4$, что говорит о неполной загрузке ВМ системы.

Таблица 1 – Планирование парных ЭП по алгоритму 1

Такты работы	Пары ВМ для выполнения ЭП	Процесс синтеза ДГ
$t = 1$	$[u_1, u_2]; [u_3, u_4]$	
$t = 2$	$[u_2, u_4]$	
$t = 3$	$[u_2, u_3]$	
$t = 4$	$[u_1, u_4]$	
$t = 5$	$[u_1, u_2]$	
$t = 6$	$[u_1, u_4]; [u_2, u_3]$	

Вычислительная сложность алгоритма 1 определяется, главным образом, трудоемкостью шага 6, на котором в предельном случае требуется просмотреть все множество ребер D диагностического графа, и оценивается как $O(ng)$ действий, поскольку $g = |D|$ и всегда $r < n$.

Пример 2. Применение алгоритма 2 для организации вычислений в отказоустойчивой системе рассмотрено для исходных данных из примера 1. Результат планирования вычислений с применением коллективных ЭП показан на рисунке 4. График функции плотности загрузки (рисунок 4,б), в котором $F(t) = n$ для любого значения $t = 1, 2, \dots, 6$, свидетельствует о полной загрузке ВМ системы в данном примере.

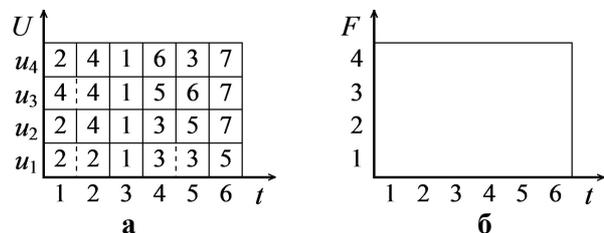
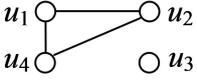
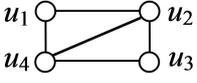
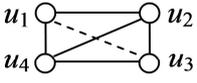
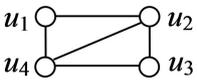


Рисунок 4 – Результаты работы алгоритма 2:
а – диаграмма загрузки ВМ;
б – функция плотности загрузки

Процесс выполнения коллективных ЭП во времени показан в таблице 2, из которой видно, что цикл контроля повторяется три раза и всего выполняется 15 сравнений результатов, соответствующих парным ЭП.

Таблица 2 – Планирование коллективных ЭП по алгоритму 2

Такты работы	Группы ВМ для коллективных ЭП	Процесс синтеза ДГ
$t = 1$	$[u_1, u_2, u_4]$	
$t = 2$	$[u_2, u_3, u_4]$	
$t = 3$	$[u_1, u_2, u_3, u_4]$	
$t = 4$	$[u_1, u_2]$	
$t = 5$	$[u_1, u_4]$	
$t = 6$	$[u_2, u_3, u_4]$	

Каждая коллективная ЭП задается полным подграфом, в котором избыточные ребра, соответствующие уже выполненным в текущем цикле контролю (при $t = 2$) или не предусмотренным исходным ДГ (для $t = 3$) парным ЭП, показаны штриховой линией.

Нетрудно заметить, что первый цикл контроля выполняется за два такта, т. е. при $k = 1$ имеем $N(1) = 2$ и функция $L(k)$ принимает значение $L_c(1) = 6/5$. Аналогично для второго и третьего циклов, где $N(2) = 1$ и $N(3) = 3$, получаем $L_c(2) = 4/5$ и $L_c(3) = 7/5$. Эти числовые значения показывают, что для данного примера почти в два раза уменьшается нагрузка ВМ дублирующими вычислениями, что существенно сокращает среднюю длительность цикла контроля. Заметим, что на третьем такте работы системы ($t = 3$) образуется избыточная проверочная связь между ВМ u_1 и u_3 , которая показана штриховой линией и не предусмотрена исходным ДГ (рисунок 2).

Вычислительная сложность алгоритма 2 также определяется трудоемкостью шага 6, на котором решается задача выделения связного подграфа G_0 , построенного не более чем на $r + 1$ вершинах, соответствующих еще незагруженным ВМ $u_i \in U_0$, таким, что $p(u_i) = 0$, и содержащего наибольшее число непрсмотренных ребер, задающих одну коллективную ЭП.

В худшем случае здесь выделяется максимальный полный подграф (клика) G_0 и задача является NP -полной [15]. Поиск клики обычно производится с помощью построения макси-

мальных внутренне устойчивых множеств на дополнительном к G графе \bar{G} . Алгоритм поиска максимального внутренне устойчивого множества, основанный на систематическом переборе решений и обеспечивающий получение результата за почти постоянное время, не зависящее от размера графа \bar{G} , можно найти в работе [16].

В инженерной практике получение максимального внутренне устойчивого множества производится за полиномиальное время на основе приближенного метода, который используется в алгоритме расслоения Эйкера [17].

Здесь для определения максимального внутренне устойчивого множества в графе отыскивается вершина с максимальной локальной степенью и исключается из графа вместе с инцидентными ей ребрами. Такие вершины исключаются до тех пор, пока в графе не останется ни одного ребра. Оставшиеся несвязные вершины образуют внутренне устойчивое множество, которое считается максимальным. Трудоемкость такого метода составляет $O(n^2)$ действий, и при его использовании (на шаге 6) вычислительная сложность алгоритма 2 также становится полиномиальной и имеет оценку $O(n^3)$.

Пример 3. Возможный вариант организации параллельных вычислений по алгоритму 3 для исходных данных из примера 1 показан на рисунке 5.

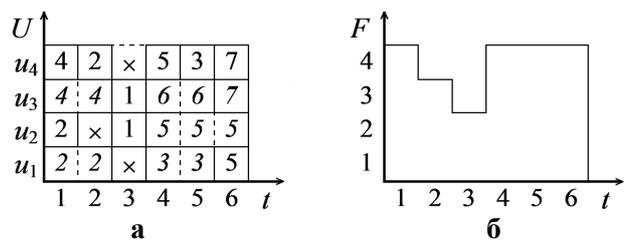
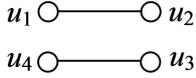
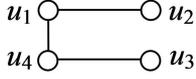
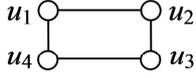
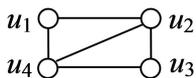
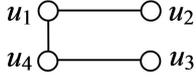


Рисунок 5 – Результаты работы алгоритма 3:
а – диаграмма загрузки ВМ;
б – функция плотности загрузки

Здесь так же, как и для алгоритма 1, за период решения основной задачи выполнено восемь ЭП. При этом номера исполняемых операторов с жестким закреплением ВМ выделены наклонным шрифтом (рисунок 5,а). Соответствующий процесс выбора ребер ДГ, определяющих парные ЭП, представлен в таблице 3.

Поскольку в предельном случае требуется дважды просмотреть все множество ребер ДГ (шаг 7 и шаг 9 алгоритма 3) и всегда $r < n$, то оценка вычислительной сложности этого алгоритма определяется как $O(2ng)$ действий.

Таблица 3 – Планирование парных ЭП по алгоритму 3 (с жестким закреплением ВМ)

Такты работы	Пары ВМ для выполнения ЭП	Процесс синтеза ДГ
$t = 1$	$[u_1, u_2]; [u_3, u_4]$	
$t = 2$	$[u_1, u_4]$	
$t = 3$	$[u_2, u_3]$	
$t = 4$	$[u_2, u_4]$	
$t = 5$	$[u_1, u_4]$	
$t = 6$	$[u_1, u_2]; [u_3, u_4]$	

Заключение. В статье предложен подход к организации параллельных диагностических вычислений в специализированных многопроцессорных системах реального времени, который на основе известной диаграммы загрузки вычислительных модулей обеспечивает проведение множества элементарных проверок, заданных диагностическим графом.

Основным теоретическим результатом работы является способ повышения эффективности использования вычислительных ресурсов многопроцессорной системы за счет применения коллективных элементарных проверок, в которых один и тот же фрагмент прикладной задачи решают не пара, а группа из трех или более вычислительных модулей. Во-первых, при использовании коллективных ЭП среднее число ВМ, необходимое для получения одного элемента s_{ij} синдрома системы S , становится явно меньше двух, что имеет место для парных ЭП. Во-вторых, достигается более полная и равномерная загрузка ВМ без увеличения общего времени решения основной задачи.

С практической точки зрения основным достоинством разработанных алгоритмов является возможность их использования для динамического планирования параллельных диагностических процессов в отказоустойчивых системах в режиме реального времени.

Библиографический список

1. Головкин Б.А. Вычислительные системы с большим числом процессоров. – М.: Радио и связь, 1995. – 320 с.

2. Скворцов С.В. Организация отказоустойчивых вычислений в магистрально-модульных многопроцессорных системах // Вестник Рязанской государственной радиотехнической академии. – 1996. – Вып. 1. – С. 27–32.

3. Ибиду К.А. Надежность, контроль и диагностика вычислительных машин и систем. – М.: Высшая школа, 1989. – 216 с.

4. Пархоменко П.П., Согомонян Е.С. Основы технической диагностики. – М.: Энергия, 1981. – 320 с.

5. Гершанов В.И., Скворцов С.В., Телков И.А. Методы повышения отказоустойчивости вычислительных систем, основанных на принципе ассоциативной селекции потоков данных // Вопросы радиоэлектроники. Серия Электронная вычислительная техника – 1992. – Вып. 7. – С. 50–58.

6. Скворцов С.В. Применение симметричной диагностической модели при организации активной отказоустойчивости многопроцессорных систем // Вестник Рязанской государственной радиотехнической академии. – 1998. – Вып. 4. – С. 47–64.

7. Скворцов С.В. Решение задачи характеристики для одной симметричной диагностической модели // Вестник Рязанской государственной радиотехнической академии. – 1998. – Вып. 5. – С. 23–26.

8. Буслов А.В., Скворцов С.В. Дешифрация результатов самодиагностирования многопроцессорных систем в режиме реального времени // Вестник Рязанской государственной радиотехнической академии. – 1999. – Вып. 6. – С. 34–37.

9. Скворцов Н.В., Скворцов С.В., Хрюкин В.И. Дешифрация диагностического синдрома многопроцессорной системы в реальном времени // Системы управления и информационные технологии. – 2010. № 1 (39). – С. 49–53.

10. Корячко В.П., Скворцов С.В., Шувиков В.И. Характеризация диагностических графов для симметричной модели дешифрации синдрома // Информационные технологии. – 1999. – № 6. – С. 18–22.

11. Корячко В.П., Скворцов С.В., Шувиков В.И. Синтез оптимальных диагностических графов для симметричной модели дешифрации синдрома // Информационные технологии. – 1999. – № 12. – С. 32–37.

12. Скворцов Н.В., Скворцов С.В., Хрюкин В.И. Синтез диагностических графов для многопроцессорных систем с активной отказоустойчивостью // Вестник Рязанского государственного радиотехнического университета. – 2012. № 1 (Вып. 39). – Ч. 2. – С. 83–89.

13. Барский А.Б. Параллельные процессы в вычислительных системах. Планирование и организация. – М.: Радио и связь, 1990. – 256 с.

14. Корячко В.П. Конструирование микропроцессорных систем контроля РЭА. – М.: Радио и связь, 1987. – 160 с.

15. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985. – 512 с.

16. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.

17. Теория и методы автоматизации проектирования вычислительных систем / под ред. М. Брейера. – М.: Мир, 1977. – 284 с.

УДК 681.3.06,658.512.2.011.56

К.С. Кульга

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ТЕХНИЧЕСКОЙ ПОДГОТОВКИ И ОПЕРАТИВНОГО УПРАВЛЕНИЯ ПОЗАКАЗНЫМ ПРОИЗВОДСТВОМ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНТЕГРИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ

Обобщается 20-летний опыт автора, связанный с разработкой и внедрением программного обеспечения (ПО) интегрированной автоматизированной информационной системы (ИАИС) машиностроительного предприятия. Вниманию читателей предлагается методология создания ПО ИАИС, предназначенной для автоматизации в едином информационном пространстве (ЕИП) предприятия бизнес-процессов (БП) технической подготовки и оперативного управления позаказным производством наукоемких изделий. Описываемая методология, воплощенная в виде ПО ИАИС Stalker PLM v7.x, экспериментально апробирована на десяти машиностроительных предприятиях РФ.

Ключевые слова: методология создания ПО ИАИС, автоматизация предприятия.

Введение. К началу XXI века стало ясно, что достаточно дорогостоящие АИС CAD/CAM/PDM/FRP/MRP/MES (*Computer Aided Design/Computer Aided Manufacturing/Product Data Management/Finance Requirements Planning/Material Requirements Planning/Management Execution System*) не оправдывают возлагающихся на них надежд. Дело в том, что вышеуказанные АИС, ориентированные на локальную автоматизацию и формирование традиционных баз данных, не решают проблему создания ЕИП, предназначенного для синхронизированного обмена данными между всеми участниками, вовлеченными в жизненный цикл изделий (ЖЦИ).

На основании анализа реализованных проектов автоматизации управления машиностроительными предприятиями (далее, предприятия) выявлено, что контуры ERP-системы (*Enterprise Resource Planning*), соответствующие классификации APICS (*American Production and Inventory Control Society*) [1], не обеспечивают решения интегрированных задач технической подготовки и оперативного управления позаказным производством предприятия, ограничиваясь стратегическим планированием. С другой стороны, контуры CAD/CAM/CAE-систем не имеют прямых информационных связей с ERP-системой. Все это определяет существование значительного

функционального и информационного разрыва между этими АИС [4]. Подтверждением этой тенденции является тот факт, что начиная с 2008 года к разработке АИС класса PLM приступили компании, разрабатывающие ПО ERP-системы: SAP, Oracle Corporation.

Для решения задачи взаимодействия БП технической подготовки производства и БП, которые обеспечиваются средствами ERP-систем, в настоящее время разрабатывается ПО АИС предприятия на основе концепции PLM (*Product Lifecycle Management*). Изучение возможностей PLM-систем показало, что в них в той или иной степени реализована интеграция в ЕИП предприятия CAD/CAM/CAE/PDM-систем, то есть БП конструкторской и технологической подготовки производства. В то же время для осуществления взаимодействия PDM- и ERP-систем, которые реализуют БП основных стадий ЖЦИ, определяющих эффективность работы предприятия, в настоящее время используются методы интеграции на основе бумажной технической документации (БТД), программного обмена через структурированные файлы данных или API (*Application Programming Interface*). Применение таких методов интеграции приводит к отсутствию единых процедур администрирования, к многочисленным ошибкам и потере актуально-

сти наборов данных, существенному затруднению процесса параллельного проектирования и производства изделий, увеличению стоимости внедрения и сопровождения.

Проведенный анализ позволил сформулировать **постановку научной проблемы**, имеющей важное значение для российских предприятий в условиях постоянного роста конкуренции на мировом рынке наукоемкой продукции – разработка и практическая апробация в реальных производственных условиях моделей и методов создания ИАИС, обеспечивающих автоматизацию ключевых БП предприятия в ЕИП на основе CAD/CAM/CAE/PDM/FRP/MRP/MES-систем, включая информационное и программное взаимодействие с ERP-системой (учётно-хозяйственной АИС).

Разработка концептуальной модели ИАИС. Согласно статистическим данным, собранным аналитической компанией *Standish Group* (США), из 30 000 проектов АИС, обследованных в США в период 1994÷2013 гг., успешными оказались не более 24 % (были выполнены в срок и в рамках заданного объема финансирования) [2]. Проведенный анализ показал, что большинство неудач связано с отсутствием или неправильным применением методологии создания АИС, отвечающей требованиям предприятий.

С точки зрения системного анализа описание создаваемой ИАИС предприятия может быть представлено следующей шестеркой компонентов:

$$DSys = \{NSys, PSys, ASys, ISys, ESys, SSys\},$$

где *NSys* – наименование; *PSys* – цели; *ASys* – общесистемные характеристики; *ISys* – вход; *ESys* – выход; *SSys* – методология создания.

Основной компонентой концептуальной модели является методология создания ИАИС предприятия, которая реализуется через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства, которые обеспечивают реализацию жизненного цикла проекта (ЖЦП или *ALM – Application Lifecycle Management*) системы.

Рассмотрим последовательность реализации предлагаемой методологии создания ИАИС предприятия.

Определение платформы ЖЦП ИАИС предприятия. При выборе платформы ЖЦП ИАИС учитывались, прежде всего, следующие характеристики создаваемого проекта:

- уровень качества по стандарту *CMM/СММ(Сapability Maturity Model/Capability Maturity Model Integration)*;
- область применения (наукоемкие проек-

ты ИАИС или бизнес-приложения);

- методы разработки (итеративные и каскадные);
- возможность распределенной работы и администрирования прав участников;
- документированность;
- возможность управления требованиями;
- масштабируемость;
- организация и стоимость тестирования;
- стоимость внедрения и сопровождения.

На основании сравнения вышеуказанных характеристик платформ ЖЦП в работе [4] обоснован выбор для разработки проекта ИАИС предприятия платформы *RUP (Rational Unified Process)*.

Обоснование технологии создания проекта ИАИС. Выбор CASE-технологии основывался на следующих её возможностях: повышение качества создаваемого ПО за счет применения графических средств моделирования предметной области, формирования и контроля исходного кода; уменьшение времени создания проекта.

Обоснование метода проектирования функциональной модели проекта ИАИС. Объектно-ориентированный (ОО) метод проектирования функциональной модели (ФМ) был выбран на основании следующих его возможностей: реализация структурной декомпозиции БП предприятия; моделирование динамического поведения ИАИС в зависимости от возникающих в ней событий. ОО ФМ ИАИС рассматривается как совокупность взаимодействующих во времени объектов. Для ОО проектирования ФМ ИАИС обоснован выбор унифицированного языка моделирования *UML* [3].

Определение базовых информационных технологий ИАИС. Для проектирования ИАИС необходимы базовые информационные технологии, которые должны обеспечивать [4]:

- 1) реинжиниринг БП технической подготовки и оперативного управления в ЕИП предприятия для обеспечения перехода от функционально-ориентированной (на основе плановой экономики) к процессорной организации управления предприятием;
 - 2) описание единой информационной модели ИАИС предприятия;
 - 3) управление данными электронной структуры изделия (ЭСИ), БП и ресурсами, а также интеграцию информации из наборов данных ИАИС и ERP-системы;
 - 4) алгоритмическое и программное обеспечение CAD/CAM/CAE/PDM/FRP/MRP/MES-систем в ЕИП предприятия.
- 1. Базовая информационная технология реинжиниринга БП предприятия**

1.1. Синтез структуры ИАИС предприятия. Формирование новой структуры ИАИС предприятия осуществлялось с помощью разработанной методики синтеза, базирующейся на результатах системного анализа.

В рамках этой методики проводились: исследование существующих реализаций АИС; анализ взаимодействия в ЕИП предприятия БП технической подготовки и оперативного управления производством; исследование направлений развития ИАИС.

Исследование существующих реализаций АИС предприятия. Были рассмотрены три структуры ИАИС, внедряемые на предприятиях.

Первая структура основана на использовании локальных АИС, автоматизирующих отдельные БП средних и малых предприятий. Обмен данными осуществляется через БТД, локальные файлы или файлы БД.

Вторая структура характеризуется преимущественной автоматизацией финансовых и учетно-хозяйственных БП предприятия. Методы интеграции систем, основанные на организации специальных групп сотрудников для обработки данных БТД или на программном обмене через файлы данных, приводят к многочисленным ошибкам и потере актуальности информации.

Третья структура основана на интеграции в ЕИП предприятия только БП конструкторской и технологической подготовки производства наукоемких изделий. Передача информации из PDM-системы в ERP-систему осуществляется на основе структурированных файлов данных (ИСО 10303-21/ГОСТ Р ИСО 10303, ИСО 8879/10744) или с помощью прикладного API-интерфейса (ИСО 10303-22).

Исследование вышеуказанных методов интеграции АИС выявило следующие их недостатки:

- невозможна реализация параллельных БП технической подготовки и оперативного управления позаказным производством;

- отсутствуют единые процедуры администрирования и одновременного доступа пользователей к набору данных изделий на стадиях ЖЦИ;

- для актуализации набора данных изделий на стадиях ЖЦИ необходимы многочисленные рутинные процедуры синхронизации и контроля (проверки) записей структурированных файлов обмена.

Следствием этих недостатков являются многочисленные ошибки и потеря доверия пользователей к АИС, ограничения возможностей

масштабируемости структуры АИС, увеличение сроков, стоимости внедрения и сопровождения. Кроме того, исключается интеграция с системами реального времени SCADA (*Supervisory Control And Data Acquisition*).

1.2. Анализ взаимодействия БП технической подготовки и управления производством в ЕИП предприятия. Анализ взаимодействия БП АИС выявил необходимость комплексной интеграции в ЕИП предприятия БП технической подготовки производства (CAD/CAM/CAE/PDM-системы) и многих финансовых и учетно-хозяйственных БП, которые обеспечиваются средствами FRP/MRP/MES (входят в состав ERP-системы). В свою очередь, БП управления персоналом, бухгалтерского учета (фактический учет), экономического анализа и прогнозирования, электронной коммерции, перспективных исследований рынков промышленной продукции не требуют оперативного управления в рамках ЕИП предприятия, так как они необходимы для реализации стратегических задач. Для организации взаимодействия в ЕИП предприятия БП CAD/CAM/CAE/PDM/FRP/MRP/MES- и ERP-систем предлагается использование программных методов интеграции наборов данных.

1.3. Исследование направлений развития ИАИС предприятия. Наиболее важными направлениями развития являются:

- интеграция в ЕИП предприятия БП технической подготовки и оперативного управления производством;

- развитие единых процедур администрирования и управления документооборотом на стадиях ЖЦИ;

- поставка комплексных решений автоматизации предприятий на основе интеграции ИАИС и ERP-систем;

- повышение эффективности и снижение стоимости внедрения ПО ИАИС для средних и малых предприятий.

На основании результатов реализации методики синтеза разработана структура ИАИС предприятия, представленная на рисунке 1.

1.4. Разработка ФМ ИАИС предприятия. Этапы разработки функциональной модели предприятия: бизнес-моделирование; определение функциональных и нефункциональных требований; анализ и проектирование; реализация; тестирование; развертывание.

Каждый этап включал выполнение задач для достижения конечной цели функционального моделирования – разработки ПО ИАИС предприятия.

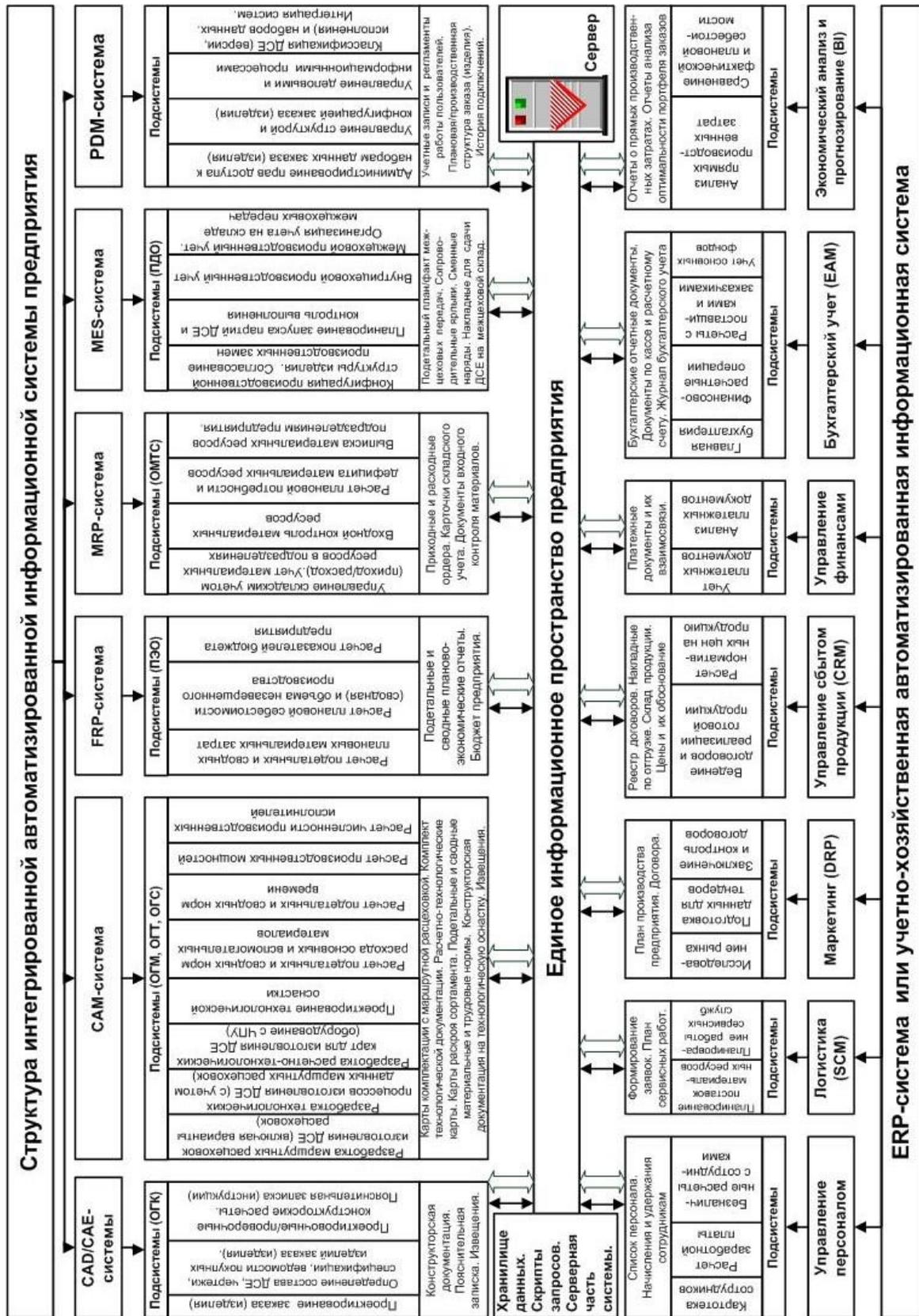


Рисунок 1 – Структура ИАИС предприятия

Примеры диаграмм ФМ ИАИС приведены на рисунке 2. Подробное описание этапов разработки и диаграмм ФМ приведено в работе [4].

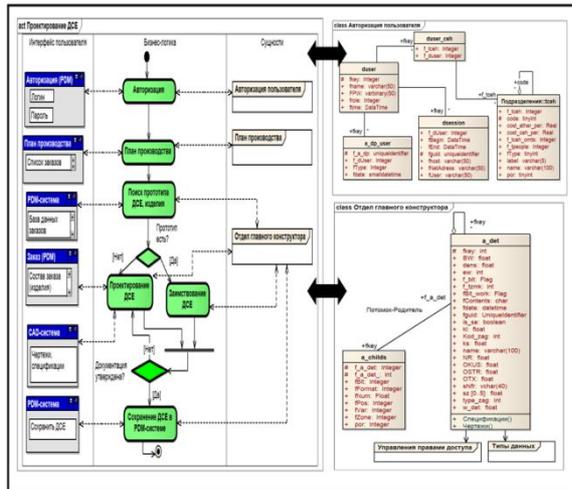


Рисунок 2 – Примеры диаграмм деятельности и классов ФМ ИАИС

2. Базовая информационная технология описания единой информационной модели ИАИС. При построении единой информационной модели (ЕИМ) ИАИС необходимо установить взаимосвязь свойств материальных объектов с характеристиками их функциональных структурных элементов, которые, в свою очередь, зависят от свойств этих объектов. Указанная взаимосвязь является основным признаком целостности ЕИМ ИАИС. Отношения между структурными элементами в ЕИМ ИАИС предлагается фиксировать в виде иерархических (древовидных) представлений. Это позволяет при объединении структурных элементов с формированием системных связей отражать одновременно как структурные, так и параметрические отношения, что исключает необходимость аналитического описания связей посредством уравнений [5].

Предложенная технология древовидного представления данных основана на разработанной ФМ ИАИС предприятия и реализована в виде ЕИМ. Она удовлетворяет следующим основным требованиям, которые можно разделить на три группы:

- структура: максимальное количество уровней – не менее 50; общее количество записей – нет существенных ограничений; формирование осуществляется по событию в ИАИС; построение множества альтернативных деревьев, описывающих иерархические представления предметной области ИАИС;
- реализация: вхождение узла в разные уровни иерархического представления с различным значением количественного параметра; ас-

социативная связь атрибутов дочернего узла с атрибутами родительского узла; различное сочетание атрибутов узла, например, детали/сборочные единицы (ДСЕ) и их возможные технологические маршрутные расщеповки;

- высокая производительность операций: определение количества всех потомков узла; навигации и корректирования поддерева; прямой выборки всех потомков узла, поддерева, пути от узла до корня дерева (всех предков узла); выбора узла (например, изделие, ДСЕ, технологическая маршрутная расщеповка, документ и т.п.), удовлетворяющего условиям отбора с группированием отобранных записей по изделию, ДСЕ, технологическому маршруту и операции, рабочему центру и другим наборам данных.

Предлагаемая технология позволила исключить моноцентризм иерархической модели, вводимой в теории графов, которая накладывает слишком жесткие ограничения на сценарий обработки содержащейся в ней информации, так как предусматривает единственный вход в ее структуру. Каждый смысловой атрибут объявлен как потенциально корневой, даже если он является промежуточной вершиной основного дерева. Для каждой такой вершины формируется альтернативная иерархия атрибутов, в которой подграф, расположенный ниже вершины, остается без изменений, а остальная часть дерева будет инвертирована. Таким образом, из одного основного дерева может быть порождено множество альтернативных деревьев, описывающих иерархические модели наборов данных предметной области. Тем самым достигается множественность возможных точек входа в ЕИМ ИАИС предприятия (рисунок 3).

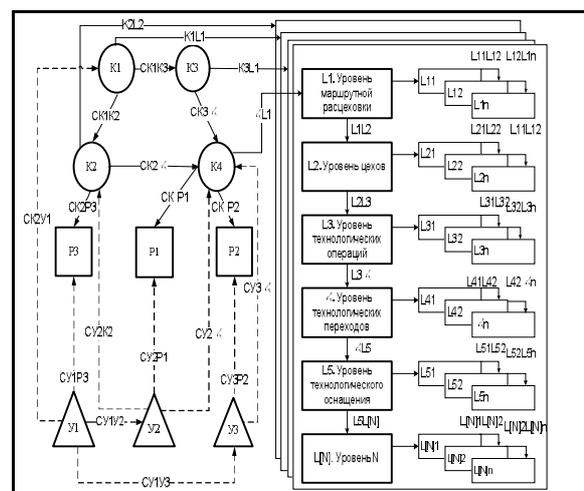


Рисунок 3 – ЕИМ ИАИС предприятия

Сплошные линии соответствуют связям, представленным в таблицах реляционной схемы данных ЕИМ, а пунктирными линиями обозна-

чены связи, которые являются ограничениями внешнего ключа в схеме данных. В ЕИМ выделены следующие иерархические представления: ДСЕ изделия: У1...У3; топологии вхождения ДСЕ и их наборы данных: К1...К4; технологические маршруты изготовления ДСЕ: Р1...Р3; уровни взаимодействия наборов данных на стадиях ЖЦИ L1...L[N].

3. Базовая информационная технология управление данными ЭСИ. В отличие от представления данных, в виде БТД, одновременно могут отображаться несколько уровней иерархических представлений наборов данных на стадиях ЖЦИ. Разработанная ЕИМ ИАИС является основой для реализации PDM-технологии, предназначенной для управления наборами данных на стадиях ЖЦИ. Функции PDM-технологии (рисунок 4):

- управление описанием, хранением, конфигурацией и правами доступа к набору данных и документам ЭСИ. ЭСИ – это не менее сложный продукт, чем само изделие, разработка которого требует совершенно новых технологий для своего создания и управления. Все наборы данных и документы ЭСИ сохраняются в хранилище данных ИАИС, которое обеспечивает их целостность, организует совместный доступ пользователей к ним в соответствии с правами доступа. Данные документы называются электронными техническими документами, которые включают электронные цифровые подписи;

- управление деловыми БП, включая отслеживание и сохранение всех операций пользователей с наборами данных ЭСИ. Деловой БП с информационной точки зрения – это последовательность действий над объектами, изменяющих их состояние и направленная на обеспечение ЖЦИ. Объектами ИАИС являются, например, документы, геометрические модели в виде файлов, технологические процессы и т.п.;

- классификация объектов. Смысл классификации состоит в том, что схожая информация сгруппирована в классы, имена которых отражают суть объектов, причем реализованная классификация является более гибкой, чем ее бумажный аналог. Чем больше количество компонентов в проектируемом изделии, тем выше потребность предприятия в функциях классификации (снижение стоимости и сроков разработки ЭСИ).

4. Алгоритмическое и программное обеспечение CAD/CAM/CAE/PDM/FRP/MRP/MES-систем в ЕИП предприятия. Данные базовые информационные технологии реализованы в ви-

де ИАИС Stalker PLM v7.x (рисунок 4). В состав ИАИС Stalker PLM v7.x входит следующее ПО, подробно описанное в работе [4].

4.1. Средства интеграции CAD-системы и ИАИС Stalker PLM v7.x. Разработаны и реализованы следующие методы программной и информационной интеграции с CAD-системами: на основе импорта данных из внешних файлов; с помощью API-функций; на уровне схемы данных в ЕИП предприятия.

4.2. Система Stalker CAM. Задачи:

- создание технологических маршрутов (расцеховок);

- проектирование технологических процессов, включающее следующие расчеты и информационное обеспечение: режимов резания, сварки, резки; нормирования трудовых затрат; баз данных технологического оснащения, типовых операций и трудовых нормативов;

- нормирование расхода основных и вспомогательных материалов (включая оптимальных раскрой штангового и листового сортамента).

4.3. Система Stalker FRP. Рассмотрим подробнее задачи этой системы: проведение предварительных расчетов плановой себестоимости изделий для открытых аукционов (опросные листы тендеров) и плана производства.

Результаты предварительных расчетов плановой себестоимости изделий закладываются в разделы заключаемых контрактов (договоров) предприятия. Рассмотрим особенности таких расчетов: ограниченность исходного набора данных изделий (например, только чертеж общего вида, эскиз или текстовое описание); значительная ответственность за результаты и высокие требования к квалификации сотрудников конструкторских, технологических и экономических отделов; значительный объем (например, на предприятиях химического машиностроения России, количество расчетов в месяц превышает одну тысячу наименований и существует тенденция к постоянному росту) и предельно сжатые сроки выполнения.

В настоящее время большинство предприятий применяют приближенные методы расчета предварительной плановой себестоимости изделий на основе локальных АИС:

- использования наборов данных изделия-прототипа (конструкторская и технологическая документация);

- использования удельных показателей изделия: масса основных материалов, типоразмер или значения основных технических характеристик.

Недостатки существующих методов предварительных расчётов плановой себестоимости изделий:

- использование только сводного уровня наборов исходных и результирующих данных. Это приводит к тому, что при последующем открытии производственного заказа для изготовления изделия отсутствует возможность использования вышеуказанных наборов данных;
- низкая производительность и значительное увеличение штата сотрудников инженерных и экономических подразделений предприятия, занимающихся этими расчётами;
- ошибки из-за отсутствия обоснованных величин прямых и косвенных затрат;
- увеличение плана производства изделий не приводит к увеличению прибыли предприятия;
- потери позиций предприятия на рынке ликвидной продукции.

В результате применения вышерассмотренных методов в плане производства предприятия остается не более 2-3 % от общего количества предварительно рассчитанных изделий. В настоящее время стало очевидно, что решение задачи расчета предварительной плановой себестоимости изделий, основанной на приближенных методах, является не эффективным, что в конечном итоге приводит к значительному ухудшению фактических экономических показателей предприятия.

На рисунке 4 приведены результаты расчёта предварительной плановой себестоимости ДСЕ (входит в ЭСИ) в системе *Stalker FRP*. Проведение таких расчётов на поддетальном (ДСЕ) и сводном уровнях (изделие) в дальнейшем позволяет эффективно использовать их результаты в ходе технической подготовки производства.

4.4. *Системы Stalker MES/MRP*. Эти системы предназначены для планирования запуска ДСЕ с учётом следующих наборов данных: ЭСИ; ресурсы и план производства предприятия.

Выводы

На основе предложенной методологии разработано ПО ИАИС *Stalker PLM v7.x* для реализации базовых информационных технологий *CAD/CAM/CAE/PDM/FRP/MRP/MES*-систем и программной интеграции ИАИС и *ERP*-системы в ЕИП предприятия. Экспериментальные исследования и практическая апробация ПО ИАИС *Stalker PLM v7.x* на предприятиях России с заказным характером производства (например,

ОАО «Уралтехнострой–Туймазыхиммаш», г. Туймазы; ООО «Зенит-Химмаш», г. Дмитровград и др.) позволили получить следующие технико-экономические показатели:

- повышение качества проектирования изделий на основе использования опыта ранее созданных проектов (от 15 до 30 %);
- реализация параллельных БП технической подготовки и оперативного управления производством (от 10 до 50 %);
- автоматизация предварительных планово-экономических расчетов для открытых аукционов (до 100 %);
- масштабируемость возможностей системы и сокращение затрат на сопровождение проекта ИАИС;
- сокращение сроков технической подготовки и освоения производства новых конкурентоспособных изделий (от 15 до 30 %);
- сокращение затрат, связанных с внесением изменений в первоначальный проект изделия (от 20 до 70 %);
- уменьшение объема незавершенного производства и эффективное использование деловых отходов материальных ресурсов;
- сокращение затрат на приобретение, внедрение и сопровождение ПО ИАИС за счёт комплексного решения производственных задач предприятия;
- автоматизация план-фактного анализа деятельности предприятия (до 100 %);
- увеличение средней прибыли предприятия (от 5 до 25 %).

Библиографический список

1. APICS [Электронный ресурс]: сайт компании. – Электрон. дан. – [Б.м.], 2013. – Режим доступа: <http://www.apics.org> – Загл. с экрана.
2. The Standish Group International, Inc. [Электронный ресурс]: сайт компании. – Электрон. дан. – [Б.м.], 2013. – Режим доступа: www.standishgroup.com – Загл. с экрана.
3. *Booch G.* Object-oriented analysis and design with applications. – Boston: Addison-Wesley, 2004.
4. Модели и методы создания интегрированной информационной системы для автоматизации технической подготовки и управления авиационным и машиностроительным производством / К. С. Кульга, И.А. Кривошеев М.: Машиностроение, 2011. – 377 с.
5. *Кульга К.С.* Единая информационная модель интегрированной информационной системы предприятия. – 2013. – №3; URL: www.science-education.ru/109-9302 (дата обращения: 01.11.2013).

УДК 681.3

А.В. Бакулев, М.А. Бакулева

ПОСТРОЕНИЕ АССОЦИАТИВНЫХ ПРАВИЛ НА ОСНОВЕ ДИФФЕРЕНЦИРОВАНИЯ ГРАФОВОЙ МОДЕЛИ АНАЛИЗИРУЕМОЙ ВЫБОРКИ

Рассматриваются методы интеллектуального анализа данных, основанные на алгоритмах построения ассоциативных правил. Предложен подход к построению ассоциативных правил на основе построения и дифференцирования модельного графа анализируемой выборки.

Ключевые слова: ассоциативные правила, дифференцирование модельного графа, Data Mining.

Введение. Технология интеллектуального анализа данных (*Data Mining*) применяется для обнаружения (дословно «добычи», от английского «mining») новых, ранее неизвестных знаний, представляющих практический интерес с точки зрения оптимизации процесса принятия решений [1, 2].

Data Mining включает различные методы и алгоритмы автоматизированной аналитической обработки данных, в том числе генетические алгоритмы, нейронные сети, кластеризацию, построение ассоциативных правил.

Ассоциативные правила позволяют находить закономерности между связанными событиями. Первый алгоритм поиска ассоциативных правил, называвшийся AIS, был разработан в 1993 году сотрудниками исследовательского центра IBM Almaden [3].

В статье рассматривается алгоритм поиска ассоциативных правил на основе построения и дифференцирования модельного графа анализируемой выборки.

Цель работы. Приоритетной целью исследования является разработка новых эффективных алгоритмов аналитического исследования числовых данных, в частности построения обоснованных ассоциативных правил.

Постановка задачи. Пусть в хранилище данных содержится информация о значениях параметра какого-то объекта (например, наблюдения за состоянием здоровья пациента). Каждая транзакция, отражаемая в хранилище данных, – это набор значений параметров за одно измерение [4].

Введем следующие обозначения: $T = \{t_1, t_2, \dots, t_n\}$ — множество транзакций, отраженных в хранилище данных, $P = \{p_1, p_2, \dots, p_k\}$ — множество значений наблюдаемых парамет-

ров. В данном случае элементы множества $T = \{t_1, t_2, \dots, t_n\}$ являются псевдодобулевыми, то есть принимают значение 0 или 1 в зависимости от того, присутствует ли значение p_i в данной транзакции или нет.

В математической постановке задачи ассоциативное правило представляет собой продукцию вида $P_a \rightarrow P_k$. При этом основными параметрами практической значимости построенного правила являются поддержка (*support*) и достоверность (*confidence*).

Поддержка отображает приведенное количество транзакций, подтверждающих правило, к общему числу транзакций в процентном соотношении:

$$S(P_a \rightarrow P_k) = \frac{|\{P_a \cup P_k\}|}{n} \times 100\% .$$

Достоверность отображает приведенное количество транзакций, подтверждающих правило к количеству транзакций, содержащих antecedent данного правила (P_a) в процентном соотношении:

$$C(P_a \rightarrow P_k) = \frac{|\{P_a \cup P_k\}|}{|\{P_a\}|} \times 100\% .$$

Пример: пусть проводится наблюдение за состоянием здоровья (таблица 1).

Если на основании приведенных данных сформировать правило:

Учащённый пульс \rightarrow *Повышенное давление*
(если учащен пульс, то давление повышено), то

его поддержка $S = \frac{2}{7} \times 100\% \approx 30\%$,

а достоверность $C = \frac{2}{4} \times 100\% \approx 50\%$.

Очевидно, что именно эти величины позволяют отнести сформулированную продукцию к разряду ассоциативных правил, то есть под-

держка и достоверность должны превышать некоторое пороговое значение (минимальную поддержку и минимальную достоверность).

Таблица 1 – Зафиксированные наблюдения за состоянием здоровья

t_1	Повышенная температура	Учащенный пульс	Повышенное давление	Избыточный вес
t_2	Повышенная температура	Учащенный пульс	Нормальное давление	Нормальный вес
t_3	Нормальная температура	Слабый пульс	-	Нормальный вес
t_4	Повышенная температура	Учащенный пульс	Повышенное давление	-
t_5	Нормальная температура	-	Пониженное давление	Нормальный вес
t_6	Повышенная температура	Учащенный пульс	Пониженное давление	-
t_7	-	Слабый пульс	Нормальное давление	Избыточный вес

То есть классический алгоритм поиска ассоциативных правил (например, алгоритм *Apriori*) включает два основных этапа:

- 1) поиск часто встречающихся наборов, удовлетворяющих минимальным пороговым значениям;
- 2) генерацию ассоциативных правил из часто встречающихся наборов, отвечающих минимальным требованиям достоверности [3].

Основная проблема поиска ассоциативных правил – большая алгоритмическая сложность нахождения часто встречающихся наборов, поскольку с ростом числа анализируемых параметров экспоненциально растет число возможных наборов.

Решение задачи. Для эффективного нахождения ассоциативных правил данные приводятся к нормализованному виду, то есть строится матрица, элементы (m_{ik}) которой принимают следующие значения:

$$m_{ij} = \begin{cases} 1, & k\text{-й элемент в } i\text{-й транзакции,} \\ 0, & \text{в противном случае.} \end{cases}$$

Например, таблица 1 примет следующий нормализованный вид (таблица 2):

Таблица 2 – Нормализованная таблица транзакций

<i>T</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>
t_1	0	1	1	0	1	0	0	1	0
t_2	0	1	1	0	0	0	1	0	1
t_3	1	0	0	1	0	0	0	0	1
t_4	0	1	1	0	1	0	0	0	0
t_5	1	0	0	0	0	1	0	0	1
t_6	0	1	1	0	0	1	0	0	0
t_7	0	0	0	1	0	0	1	1	0

В представленной таблице заголовкам столбцов соответствуют следующие элементы:

- A* – нормальная температура,
- B* – повышенная температура,
- C* – учащенный пульс,
- D* – слабый пульс,
- E* – повышенное давление,
- F* – пониженное давление,
- G* – нормальное давление,
- H* – избыточный вес,
- I* – нормальный вес.

Рассматривая данную таблицу как матрицу смежности модельного графа, можно установить «интенсивность» участия анализируемых элементов в ассоциативных правилах как вес производных по каждой паре элементов. В рассматриваемом подходе событием будет формулируемое ассоциативное правило.

Производной $\frac{\partial G}{\partial S}$ графа *G* по событию *S*

называется неориентированный взвешенный граф $G(V, (U, P))$, носитель *V* которого совпадает с носителем модели, определяемой этим событием; *U* – множество дуг, соединяющих пары вершин, взвешенных отношением их частоты к частоте их совместного участия в событии *S* [5].

$$\frac{\partial G}{\partial S} = \frac{f_i - 2 * f_{ij} + f_j}{f_{ij}},$$

где f_i, f_j – собственные частоты дифференцируемых элементов, f_{ij} – взаимная частота.

Для нахождения производной модельного графа строится частотная матрица *F*:

$$F = M^T \times M,$$

где *M* – исходная матрица смежности (нормали-

зованная таблица транзакций).

Для рассматриваемого примера матрица F имеет вид:

$$F = \begin{array}{c|cccccccc|c} & A & B & C & D & E & F & G & H & I \\ \hline A & 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 2 \\ B & 0 & 4 & 4 & 0 & 2 & 1 & 1 & 1 & 1 \\ C & 0 & 4 & 4 & 0 & 2 & 1 & 1 & 1 & 1 \\ D & 1 & 0 & 0 & 2 & 0 & 0 & 1 & 1 & 1 \\ E & 0 & 2 & 2 & 0 & 2 & 0 & 0 & 1 & 0 \\ F & 1 & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 1 \\ G & 0 & 1 & 1 & 1 & 0 & 0 & 2 & 1 & 1 \\ H & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 2 & 0 \\ I & 2 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 3 \end{array}$$

Наибольшие элементы данной матрицы, расположенные выше главной диагонали (очевидно, что данная матрица симметрична), определяют наиболее частые наборы. Нахождение производных для этих наборов позволит судить об интенсивности участия элементов в образовании ассоциативного правила.

$$\text{Производная } \frac{\partial G}{\partial S}(B, C) = \frac{4 - 2 * 4 + 4}{4} = 0,$$

следовательно, элементы B и C равномерно участвуют в образовании ассоциации, и соответствующее правило можно формулировать и как «если повышенная температура, то учащенный пульс», и как «если учащенный пульс, то повышенная температура».

$$\text{Аналогично } \frac{\partial G}{\partial S}(A, I) = \frac{2 - 2 * 2 + 2}{2} = 0,$$

следовательно, правило $A \rightarrow I$ равнозначно правилу $I \rightarrow A$, то есть антецедентом может быть как высказывание A – нормальная температура, так и высказывание I – нормальный вес. Однако в силу небольшого значения взаимной частоты $f_{AI} = 2$ эти высказывания не следует принимать за правило.

В проводимых исследованиях установлена

закономерность между значениями производных и значимостью формируемых правил. Отмечено, что чем больше значение производной, тем больше степень «разброса» элементов по транзакциям, следовательно, построенные продукции становятся ассоциативными правилами при значениях производных, близких к нулю.

Закключение. В представленной работе рассмотрен подход к построению ассоциативных правил, основанный на матричном представлении модельного графа с последующим дифференцированием. Данный подход позволяет пренебречь трудоемким этапом подсчитывания поддержки и достоверности для каждой пары анализируемых параметров, а также исследовать практическую значимость формулируемого правила. Однозначно определяется возможность обращения ассоциативного правила. То есть предложенный подход позволяет выявить случай, когда один и тот же элемент сформированного правила может быть как антецедентом, так и консеквентом.

Библиографический список

1. Филиппович А.Ю. Автоматизированная система научных исследований ассоциативных экспериментов (АСНИ АЭ) // Вопросы психолингвистики. 2007. № 6.
2. Бакулева М.А., Бакулев А.В. Применение вейвлет-преобразования для анализа данных хранилища // Вестник РГРТУ: науч.-техн. журнал. Выпуск 21. Рязань: РГРТУ, 2007. С. 57-60.
3. Филиппович А.Ю., Нейский И.М. Методика адаптивной кластеризации фактографических данных на основе интеграции алгоритмов MST и FUZZY C-MEANS // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. 2009. № 3.
4. Бакулева М.А., Бакулев А.В., Телков И.А. Алгоритм автоматизации проектирования хранилищ данных // Вестник РГРТУ: науч.-техн. журнал. Выпуск 23. Рязань: РГРТУ, 2008. С. 90-93.
5. Корячко В.П., Бакулева М.А., Бакулев А.В., Гостин А.М. Дискретная математика: учеб. пособие. Рязань: РГРТУ, 2011. 178 с.