

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЕ МЕТОДЫ И КОМПЛЕКСЫ ПРОГРАММ

УДК 621.377

Б.В. Круглов

ОПТИМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ ПРИ ПРОЕКТИРОВАНИИ ФУНКЦИОНАЛЬНЫХ ПОДСИСТЕМ

Круглов Борис Владимирович, окончил механико-математический факультет Ульяновского государственного университета. Ведущий инженер-программист ФНПЦ ОАО «НПО «Марс», соискатель Ульяновского государственного технического университета. Имеет публикации в области автоматизации проектирования АСУ. [e-mail: boris_kruglov@mail.ru].

Аннотация

В статье представлены постановки задач оптимизации работ по проектированию АСУ на примере разработки функциональных подсистем (ФП). Описан алгоритм оптимального распределения ресурсов методами имитационного моделирования и случайного поиска. Разработано программное обеспечение, которое позволяет сократить сроки и стоимость разработки АСУ.

Ключевые слова: проектирование АСУ, электронный документооборот, теория расписаний, имитационное моделирование.

Boris Vladimirovich Kruglov, graduated from the Faculty of Mechanics and Mathematics at Ulyanovsk State University, leading programmer at FRPC OJSC 'RPA 'Mars', applicant for Candidate of Sciences at Ulyanovsk State Technical University; author of publications in the field of computer-aided design of computer-aided control systems. e-mail: boris_kruglov@mail.ru.

Abstract

The article presents problems of optimization concerning design of computer-aided control system using an example of development of operational subsystems. It also describes an algorithm for optimal allocation of resources by simulation and random search. The developed software contributes to decrease of time and costs of development of computer-aided control system.

Key words: design of computer-aided control systems, e-document management, scheduling theory, simulation.

ВВЕДЕНИЕ

Особенностью проектирования иерархических систем управления является разработка проектов большого числа (несколько десятков) функциональных подсистем и протоколов взаимодействия с внешними системами. Достаточно сложные и объемные проекты ФП и протоколов взаимодействия состоят из нескольких разделов (до 10), выполняемых последовательно различными группами разработчиков.

Применяя теорию расписаний [1, 2], можно сократить время всей работы за счет выбора оптимального порядка выполнения проектов. В [3–5] рассмотрены различные стороны применения теории расписаний к организации работ по проектированию ФП и протоколов.

В данной статье предлагается новый способ сокращения времени выполнения за счет оптимального распределения ресурсов в сочетании с оптимальным порядком выполнения.

1 МАТЕМАТИЧЕСКИЕ МОДЕЛИ ОПТИМАЛЬНОЙ ОРГАНИЗАЦИИ РАБОТ

Рассмотрим m функциональных подсистем $\Phi\Pi_i$, которые разрабатываются в n отделах с числом исполнителей R_1, R_2, \dots, R_n . В каждом отделе этапы разработки выполняются одним ресурсом. Проектирование каждой функциональной подсистемы $\Phi\Pi_i$ состоит из последовательности этапов проектирования $\Phi\Pi_{i1} \rightarrow \Phi\Pi_{i2} \rightarrow \dots \rightarrow \Phi\Pi_{in}$.

Время выполнения j -го этапа разработки i -й функциональной подсистемы:

$$t_{ij} = \alpha_{ij} \cdot V_{ij} / R_j; \quad i = \overline{1, m}; \quad j = \overline{1, n}, \quad (1)$$

где α_{ij} – коэффициенты связи (трудоемкость выполнения единицы объема работ на j -м этапе разработки i -й ФП);

V_{ij} – объем j -го этапа разработки i -й ФП;

R_j – ресурс j -го отдела, используемый при выполнении j -го этапа разработки всех ФП.

Время выполнения j -го этапа всех ФП:

$$T_j = \sum_{i=1}^m \alpha_{ij} \cdot V_{ij} / R_j. \quad (2)$$

Время полной разработки i -й ФП:

$$t_i = \sum_{j=1}^n \alpha_{ij} \cdot V_{ij} / R_j. \quad (3)$$

Полное время разработки всех m функциональных подсистем сложным образом зависит от времени разработки каждой из ФП с учетом запаздываний и ожиданий на каждом уровне и последовательности разработки π :

$$T = (t_{ij}, \pi); \quad i = \overline{1, m}; \quad j = \overline{1, n}. \quad (4)$$

Выбирая различные последовательности разработки ФП из множества возможных $\Omega = (\pi_1, \pi_2, \dots, \pi_M)$, можно попытаться определить оптимальное время разработки всех проектов.

Можно находить $T = (t_{ij}, \pi)$ различными способами. Например, в [1] приведена формула для приближенного расчета времени разработки всех ФП с учетом того, что очередной этап выполнения отдельной ФП начинается после выполнения предыдущего этапа разработки данной ФП:

$$T = \sum_{i=1}^m t_{i1} + \sum_{j=2}^n t_{mj} + \begin{cases} t'_{m-1j} - t''_{m-1j}, & \text{если } t'_{m-1j} > t''_{m-1j} \\ 0, & \text{если } t'_{m-1j} \leq t''_{m-1j} \end{cases}, \quad (5)$$

$$t'_{m-1j} = t_{m-1j} + \begin{cases} t'_{m-2j} - t''_{m-1j-1}, & \text{если } t'_{m-2j} > t''_{m-1j-1} \\ 0, & \text{если } t'_{m-2j} \leq t''_{m-1j-1} \end{cases},$$

$$t''_{m-1j} = \begin{cases} t'_{m-2j} + (t_{m-1j} - t'_{m-1j-1}), & \\ \text{если } t'_{m-2j} + (t_{m-1j} - t'_{m-1j-1}) > t_{m-1j} \\ t_{m-1j}, & \\ \text{если } t'_{m-2j} + (t_{m-1j} - t'_{m-1j-1}) \leq t_{m-1j} \end{cases}.$$

Для повышения точности алгоритма расчета времени полной разработки используется формула, известная из теории расписаний [2]:

$$T_{\phi}(\pi) = \max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{m-1} \leq n} \left\{ \sum_{k=1}^{u_1} t_{i_k 1} + \sum_{k=u_1}^{u_2} t_{i_k 2} + \dots + \sum_{k=u_{m-1}}^n t_{i_k m} \right\}, \quad (6)$$

где максимум определяется по всем возможным наборам индексов $u_k, k = \overline{1, m}, 1 \leq u_k \leq n; \pi = (i_1, i_2, \dots, i_n)$ – последовательность разработки ФП. При больших значениях n и m вычисление может оказаться неприемлемо долгим.

В статьях [3] и [4] описан алгоритм расчета оптимальной последовательности разработки при помощи имитационного моделирования. В статье [5] также показано преимущество имитационного моделирования по сравнению с расчетами по формуле из теории расписаний.

Время полной разработки для заданной очередности функциональных подсистем в операторной форме имеет вид:

$$T_{\pi}(\pi) = \left(A_1 \rightarrow A_2 \rightarrow B \rightarrow C \rightarrow \begin{cases} D_1 \uparrow^B \\ D_2 \uparrow^B \\ \dots \\ D_{(2nm)} \uparrow^E \end{cases} \rightarrow E \right), \quad (7)$$

где A_1 – оператор задания начальных условий для моделируемого варианта (объем работы и коэффициент трудоемкости для каждой части каждой ФП);

A_2 – оператор задания начальных условий для одного имитационного эксперимента;

B – оператор определения очередного момента изменения системы;

C – логический оператор перехода к соответствующему оператору D_i , имитирующему реакцию системы на события;

E – оператор, определяющий условие выхода из программы.

Минимизация времени разработки функциональных подсистем при заданном общем ресурсе достигается распределением ресурса по этапам и установлением оптимального расписания выполнения работ:

$$T = \max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{m-1} \leq n} \left\{ \sum_{k=1}^{u_1} t_{i_k 1} + \sum_{k=u_1}^{u_2} t_{i_k 2} + \dots + \sum_{k=u_{m-1}}^n t_{i_k m} \right\} \xrightarrow{t_{ij}, \Omega} \min \quad (8)$$

$$(или T = \left(A_1 \rightarrow A_2 \rightarrow B \rightarrow C \rightarrow \begin{cases} D_1 \uparrow^B \\ D_2 \uparrow^B \\ \dots \\ D_{(2nm)} \uparrow^E \end{cases} \rightarrow E \right) \xrightarrow{t_{ij}, \Omega} \min \quad (9)$$

для процедуры имитационного моделирования),

$$R = \sum_{j=1}^n R_j, \quad \Omega = (\pi_1, \pi_2, \dots, \pi_M), \quad (10)$$

где Ω – множество расписаний, последовательностей разработки документов на этапах.

Минимизация общего ресурса при заданном времени реализации ФП достигается установлением времени выполнения отдельных этапов каждого документа и расписанием работ:

$$\sum_{j=1}^n R_j(t_{ij}, \Omega) \rightarrow \min, \quad (11)$$

$$\Omega = (\pi_1, \pi_2, \dots, \pi_M),$$

$$\max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{m-1} \leq u_m} \left\{ \sum_{k=1}^{u_1} t_{ik1} + \sum_{k=u_1}^{u_2} t_{ik2} + \dots + \sum_{k=u_{m-1}}^n t_{ikm} \right\} = T_0 \quad (12)$$

$$(или \left(A_1 \rightarrow A_2 \rightarrow B \rightarrow C \rightarrow \begin{cases} D_1 \uparrow^B \\ D_2 \uparrow^B \\ \dots \\ D_{(2nm)} \uparrow^E \end{cases} \rightarrow E \right) = T_0 \quad (13)$$

для процедуры имитационного моделирования).

2 ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ

Для решения задачи оптимального распределения ресурсов предлагается алгоритм и программные средства его реализации.

Исходными данными алгоритма являются данные о количестве сотрудников R_{ucx} , групп m и объеме работ V_{ij} . Из ресурсов R_{ucx} берется некоторое начальное распределение по группам: $R_{01}, R_{02}, \dots, R_{0m}$ – например, по одному сотруднику в каждой группе.

На первом шаге алгоритма увеличиваем число сотрудников на ΔR в первой группе, получая распределение ресурсов:

$$R_{11}, R_{12}, \dots, R_{1m}$$

где $R_{11} = R_{01} + \Delta R,$

$$R_{12} = R_{02}, \dots, R_{1m} = R_{0m}. \quad (14)$$

При этом должно выполняться ограничение на ресурсы:

$$\sum_{j=1}^n R_{1j} \leq R_{ucx}. \quad (15)$$

Время на выполнение $\Phi\Pi_{ij}$ теперь вычисляется по формуле:

$$t_{ij}^1 = \alpha_{ij} \cdot V_{ij} / R_{j1}. \quad (16)$$

Выполняется расчет оптимального времени разработки $T_o(I)$ по формуле теории расписаний (6) либо при помощи имитационного моделирования (7).

Второй шаг повторяет то же самое для второй группы: для распределения ресурсов $R_{21}, R_{22}, \dots, R_{2m}$,

$$где R_{21} = R_{01}, R_{22} = R_{02} + \Delta R, \dots, R_{2m} = R_{0m} \quad (17)$$

вычисляем оптимальное время $T_o(2)$. На шаге i для распределения ресурсов $R_{i1}, R_{i2}, \dots, R_{im}$ где $R_{i1} = R_{01},$

$$R_{i2} = R_{02}, \dots, R_{ii} = R_{0i} + \Delta R, \dots, R_{im} = R_{0m} \quad (18)$$

вычисляем оптимальное время $T_o(i)$. Наконец, на шаге m находим время $T_o(m)$.

Среди найденных значений $T_o(1), T_o(2), \dots, T_o(m)$ выбирается некоторое наименьшее значение времени $T_o(k)$. Параметр k определяет группу, в которой следует увеличивать число сотрудников.

Выполняя описанную процедуру для всех R_{ucx} сотрудников, получим в итоге оптимальное распределение ресурсов по группам.

С помощью визуального редактора пользователь может ввести исходные данные задачи (рис. 1): объем работ и трудоемкость (в условных единицах); число сотрудников и подразделений; способ расчета времени: по формуле

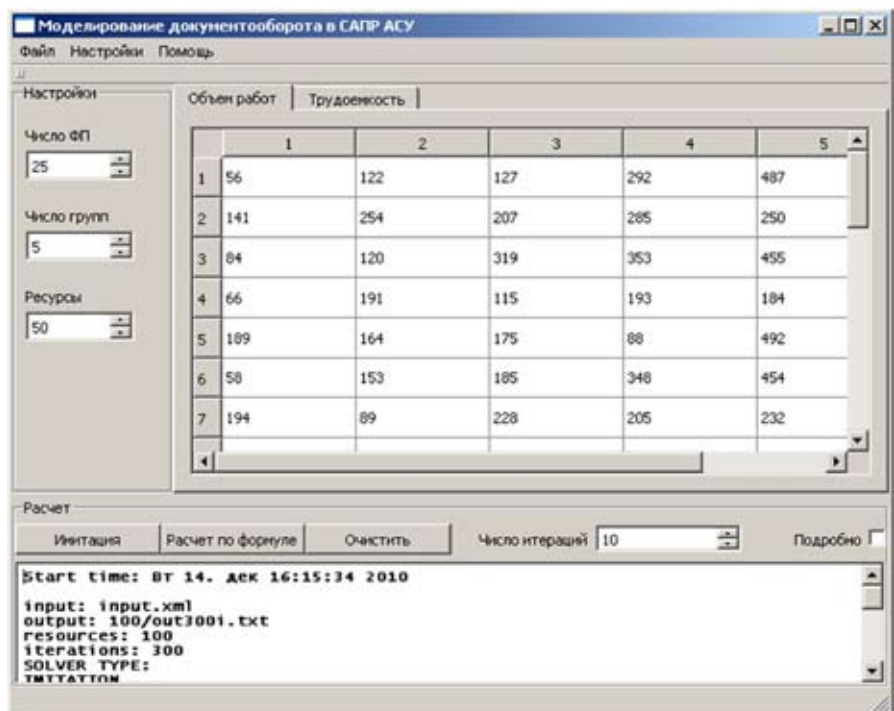


Рис. 1. Главное окно визуального редактора

теории расписаний либо при помощи имитационного алгоритма; число итераций алгоритма.

На выходе программы (рис. 2) отображаются оптимальная последовательность разработки ФП («Optimized sequence»), соответствующее распределение сотрудников («Optimized resources») и время разработки («Topt»).

3 ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

В качестве тестовой рассмотрим задачу распределения 100 человек для проектирования 25 ФП. Объем работ задается случайными числами от 50 до 100 листов на нижнем уровне и от 200 до 500 листов на верхнем уровне управления по линейному закону:

$$V_{ij} = rand(a_i + j \cdot b_j) \quad (19)$$

Результаты расчетов приведены в таблице 1.

Исходные данные по объему работ показаны на рисунке 3, а время выполнения ФП при оптимальном распределении ресурсов – на рисунке 4.

Полученная оптимальная последовательность разработки ФП: 1, 17, 21, 13, 3, 12, 23, 18, 14, 19, 5, 6, 16, 11, 2, 20, 9, 22, 7, 15, 25, 10, 4, 8, 24 – дает время разработки 292 условных единицы. При равномерном распределении ресурсов требуется уже 431 единица времени.

В таблице 2 приведен пример решения более объемной задачи для 50 ФП. В этом случае получается оптимальная последовательность разработки: 46, 16, 7, 3, 22, 23, 37, 17, 19, 15, 33, 18, 26, 36, 10, 34, 9, 44, 49, 39, 38, 45, 2, 13, 14, 24, 40, 29, 50, 6, 41, 8, 42, 28, 35, 48, 5 – со временем разработки 586. При равномерном распределении ресурсов время разработки было бы равно 917.

Полученные результаты расчетов (таблицы 1 и 2) подтверждают, что алгоритм на основе формулы (2) неустойчив: его результаты все больше отклоняются с увеличением числа итераций. Имитационный алгоритм устойчив, т. е. при увеличении числа итераций результат сходится к какому-то определенному значению. Однако и этот алгоритм обладает тем недостатком, что поиск может завершиться в области локального минимума, хотя даже в этом случае мы получаем последовательность разработки функциональных подсистем, экономящую 40% времени по сравнению с равномерным распределением сотрудников по группам.

```

Start time: Вт 14. дек 16:15:34 2010
input: input.xml
output: 100/out300i.txt
resources: 100
iterations: 300
SOLVER TYPE:
IMITATION
DATA:
N  1  2  3  4  5
1  36 122 127 292 487
2  141 254 207 285 250
3  84 120 319 353 455
4  66 191 115 193 184
5  69 124 175 88 492
6  98 163 185 948 454
7  194 89 223 205 232
8  167 111 265 228 156
9  174 111 193 305 444
10 174 287 186 187 429
11 79 182 191 122 158
12 134 257 125 298 465
13 76 168 269 323 265
14 64 214 223 242 142
15 181 256 281 145 253
16 155 74 272 136 160
17 185 66 184 488 376
18 110 212 199 155 407
19 179 72 93 249 485
20 99 173 272 165 485
21 67 92 335 151 477
22 92 235 143 330 278
23 125 184 85 247 317
24 73 145 283 89 104
25 188 118 167 320 195
OPTIMIZED SEQUENCE:
[1, 17, 3, 21, 18, 14, 9, 11, 12, 6, 13, 20, 5, 7, 19, 22, 2, 25, 16, 15, 10, 4, 23, 8, 24]
OPTIMIZED RESOURCES:
[12, 15, 19, 22, 32]
Topt=294
Speed up: 96.5686 %
End time: Вт 14. дек 16:21:05 2010
Elapsed time: 331 seconds
    
```

Рис. 2. Вывод результата программы

Таблица 1
Результаты оптимизации ресурсов для проектирования ФП
($m = 25$), $R_{исх} = 100$

Число итераций	Оптимальное время разработки		Распределение ресурсов	
	Модель	Формула	Модель	Формула
50	300	284	12, 15, 20, 22, 31	14, 19, 20, 21, 26
100	301	320	13, 15, 18, 23, 31	23, 16, 22, 18, 21
200	298	283	12, 16, 19, 22, 31	13, 18, 23, 21, 25
300	294	286	12, 15, 19, 22, 32	15, 19, 20, 22, 24
500	298	288	12, 16, 19, 22, 31	17, 19, 19, 20, 25
700	293	269	12, 15, 19, 22, 32	18, 17, 20, 21, 24
1000	292	272	12, 15, 19, 22, 32	15, 16, 24, 21, 24

Таблица 2
Результаты оптимизации ресурсов для проектирования ФП
($m = 50$), $R_{исх} = 100$

Число итераций	Оптимальное время разработки		Распределение ресурсов	
	Модель	Формула	Модель	Формула
50	612	623	8, 16, 21, 24, 31	13, 14, 20, 24, 29
100	597	599	8, 14, 21, 25, 32	9, 15, 21, 24, 31
200	597	596	9, 13, 21, 25, 32	9, 14, 22, 25, 30
300	594	604	9, 13, 21, 25, 32	11, 13, 23, 23, 30
500	592	580	7, 13, 22, 26, 32	10, 14, 21, 25, 30
700	592	578	7, 14, 21, 26, 32	9, 14, 21, 25, 31
1000	586	578	8, 13, 21, 25, 33	10, 14, 21, 25, 30

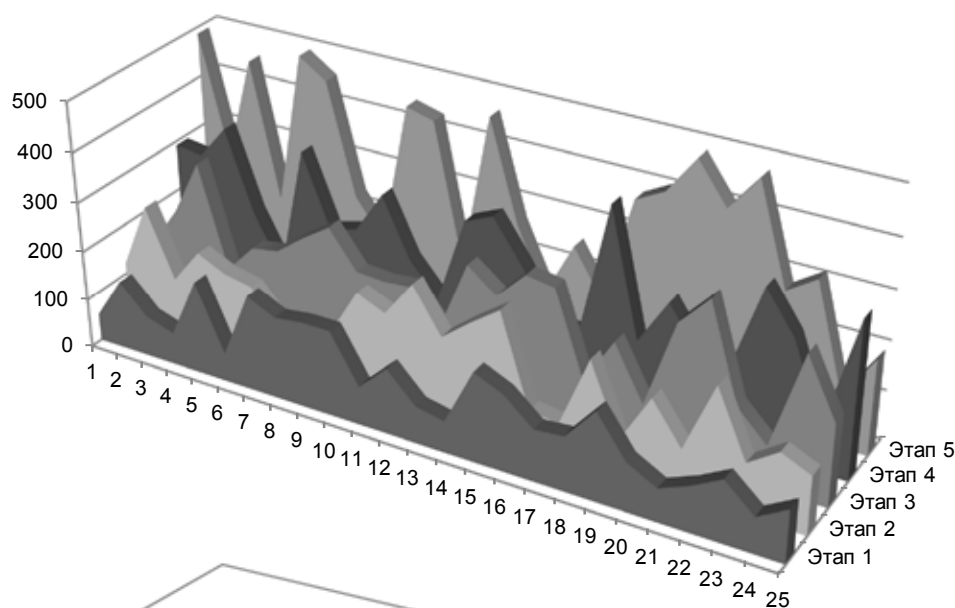


Рис. 3. Объем работ на этапах разработки функциональных подсистем

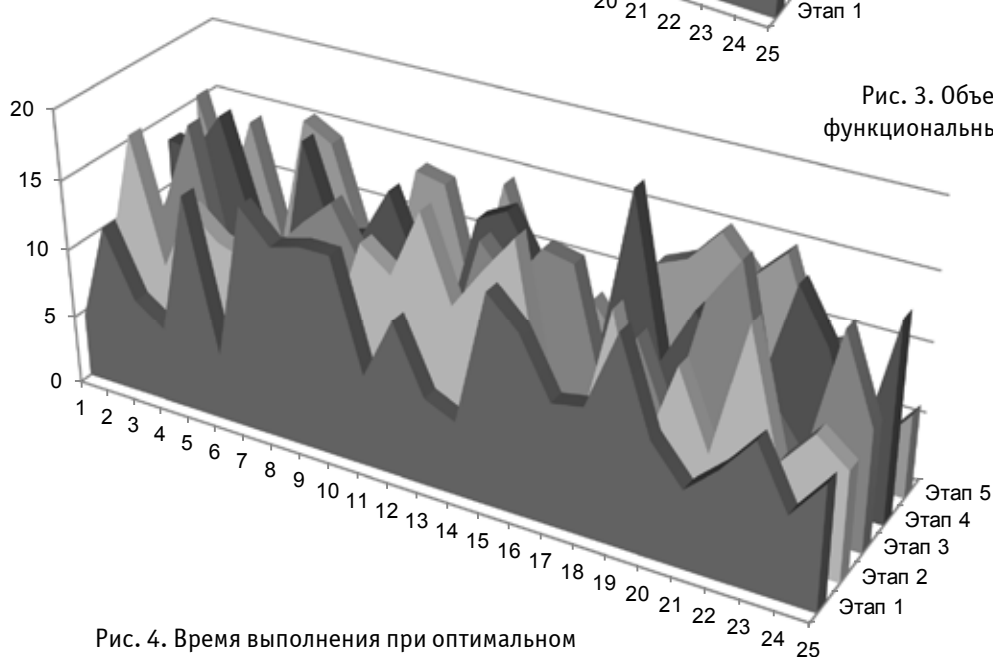


Рис. 4. Время выполнения при оптимальном распределении ресурсов

ЗАКЛЮЧЕНИЕ

Разработанный автором алгоритм имитационного моделирования для решения задачи нахождения оптимальной последовательности и оптимального времени разработки функциональных подсистем был успешно применен в сочетании с оптимизацией распределения ресурсов. Практическая значимость этой дополнительной оптимизации заключается в еще большем выигрыше во времени при организации работ проектирования АСУ.

Принцип инкапсуляции объектно-ориентированного программирования позволяет изменять имитационную модель, не затрагивая алгоритма оптимизации ресурсов методом случайного поиска, получая таким образом решение задачи оптимизации для различных видов организации разработки. Например, можно ввести нелинейную зависимость времени разработки ФП от числа сотрудников и другие усложнения модели, приближающие ее к реальности.

СПИСОК ЛИТЕРАТУРЫ

1. Иванов А.К. Математическое моделирование в технологии проектирования АСУ. – Ульяновск : УлГТУ, 2007. – 290 с.
2. Танаев В.С., Шкурба В.В. Введение в теорию расписаний. – М. : Наука, 1975. – 256 с.
3. Круглов Б.В. Модель последовательной разработки документации в САПР АСУ // Автоматизация процессов управления. – 2010. – № 1 (19). – С. 64–68.
4. Круглов Б.В. Программная реализация имитационной модели последовательной разработки документации в САПР АСУ // Автоматизация процессов управления. – 2010. – № 3 (21). – С. 76–80.
5. Круглов Б.В. Оптимизационное моделирование последовательной разработки функциональных подсистем // Автоматизация процессов управления. – 2010. – № 4 (22). – С. 57–60.