

АЛГОРИТМИЧЕСКИЕ МОДЕЛИ ОРГАНИЗАЦИОННОГО УПРАВЛЕНИЯ В СЛОЖНЫХ СОЦИОТЕХНИЧЕСКИХ СИСТЕМАХ

к.т.н. Е.А. Дружинин, В.А. Бек, Д.А. Горлов
(представил д.т.н., проф. О.Е. Федорович)

Проанализированы основные схемы взаимодействия между узлами организационной структуры в процессе выполнения работ проекта. Предложены формализованные модели протоколов управления в языке регулярных схем алгоритмов. Разработана модель обобщенного протокола управления, как универсального инструментария для проведения исследований систем организационного управления проектами методами имитационного моделирования.

Для получения моделей процессов управления проектами важным вопросом является выбор методов и языков формализованного описания протоколов взаимодействия между элементами организационной структуры сложной социотехнической системы (ССТС).

Анализ объекта исследования показал, что модели протоколов управления (ПУ) в ССТС носят событийный характер.

Для описания событийных процессов предлагается использовать аппарат регулярных схем алгоритмов (РСА), предложенный В.М. Глушковым [1]. Язык РСА позволяет наглядно, однозначно и строго описывать действия алгоритмов, осуществлять равносильные преобразования алгоритмов из любых языков описания в РСА, реализовывать тождественные преобразования алгоритмов с целью их минимизации, как по условиям, так и по операторам на основании аксиом П.М. Иванова [2].

Достоинства РСА определяются наличием функциональной полноты сигнатуры базовых операций для описания регулярных алгоритмов, контекстно-свободным алфавитом, формализованностью и компактностью записи, наличием аксиом тождественных и слабо эквивалентных преобразований структур алгоритмов, возможностью автоматизации перевода на машинные алгоритмы.

В общем случае алгоритмы в РСА можно представить в виде:

$$R = f \left(y_i, x_k, e, \emptyset, 1, 0, y, y, y, y \right),$$

где y_i – основные операторы, описывающие действия (операции); x_k – описывают условия переходов по алгоритму; e – описывает переход по алгоритму без выполнения основных операторов; \emptyset – пустой оператор,

который может выполнять роль индикатора останова алгоритма; $\mathbf{1,0}$ – тождественно-истинное и тождественно-ложное условия; $\overset{\bullet}{Y}, \hat{Y}, \overset{\vee}{Y}, \overset{*}{Y}$ – сигнатура базовых операций РСА, описывающая правила перехода по алгоритму (умножение, конъюнкция, дизъюнкция, итерация).

Сигнатура базовых операций позволяет адекватно описать все возможные переходы в алгоритмах с помощью конечного множества операций. РСА – контекстно-свободный язык, имеет системы аксиом тождественных преобразований алгоритмов; правила равносильных преобразований алгоритмов из графических, логических и матричных схем алгоритмов в РСА [3]; операции свертки и расчленения алгоритмов; построение минимизированного обобщенного алгоритма [4]. Для описания протоколов взаимодействия в организационных структурах управления проектами под оператором РСА представляется задача по функционированию, передачи данных, управлению. Для описания структурных моделей в качестве оператора РСА используется структурный элемент, а связи между ними задаются базовыми операторами.

Проанализируем действия, которые имеют место в протоколах управления ССТС, и сформируем базис основных операторов РСА для описания процессов функционирования организационных структур управления проектами. Результаты анализа представлены в табл. 1.

Опишем алгоритмы функционирования и управления типовых узлов ОС в процессе решения ФЗ в языке РСА.

1. $\mathbf{R_1}$ – алгоритм действий узла начальника (инициатора) при решении задачи подчиненными нижних уровней:

$$\mathbf{R_1 = CEP \cdot IE \downarrow^{OP} \cdot AW \cdot IE \uparrow^{cl} \cdot TR.}$$

2. $\mathbf{R_2}$ – алгоритм, описывающий действия исполнителя задачи, получившего УВ с верхнего уровня:

$$\mathbf{R_2 = IE \downarrow^{OP} \cdot PE \cdot EXE \cdot PA \cdot IE \uparrow^{cl} .}$$

Алгоритмы $\mathbf{R_1}$ и $\mathbf{R_2}$ описывают взаимодействие между узлами начальник (инициатор) – подчиненный исполнитель.

3. $\mathbf{R_3}$ – алгоритм действий узла подчиненного (инициатора) решения задачи на верхнем уровне):

$$\mathbf{R_3 = PR \cdot IE \uparrow^{OP} \cdot AW \cdot IE \downarrow^{cl} \cdot TR.}$$

4. $\mathbf{R_4}$ – алгоритм, описывающий действия начальника-исполнителя задачи управления, необходимость решения которой затребовано с нижнего уровня:

$$\mathbf{R_4 = IE \uparrow^{OP} \cdot PE \cdot EXE \cdot PA \cdot IE \downarrow^{cl} .}$$

5. Если узел совмещает в себе все роли, т.е. сам является и инициатором и исполнителем, то алгоритм его работы представляется как $\mathbf{R_5}$:

$$R_5 = PE \cdot EXE \cdot PA$$

Таблица 1

Базис основных операторов описания протоколов
взаимодействия узлов ОС в языке PСA

№ п/п	Функции управления и взаимодействия узлов ОС	Базис основных операторов ПВ в PСA
1	Подготовка управляющего воздействия (приказа) (ПУВ) (control effect preparation) начальником или узлом посредником для исполнителя более низкого уровня	CEP
2	Подготовка требования на управление (ПТ) (preparation of requirement). Исполнитель нижнего уровня является инициатором задачи управления, которая должна быть решена на верхних уровнях	PR
3	Информационный обмен (ИО) (information exchange), связанный с передачей УВ узлу нижнего уровня с генерацией типового ПВ и формированием контура управления	IE ↓^{op}
4	Информационный обмен, связанный с передачей требования на решение задачи управления узлу верхнего уровня с генерацией типового ПВ и формированием контура управления	IE ↑^{op}
5	Информационный обмен, связанный с передачей результатов решения задачи исполнителем начальнику с замыканием контура управления	IE ↑^{cl}
6	Информационный обмен, связанный с передачей узлу нижнего уровня результатов решения задач управления с замыканием контура управления	IE ↓^{cl}
7	Ожидание ответа (ОО) (доклада о выполнении) (answer waiting) узлами инициаторами или посредниками	AW
8	Пассивное ожидание (passive waiting) – состояние ожидающего узла при котором он может обслуживать или решать другие задачи управления или функционирования	PW
9	Активное ожидание (active waiting) – состояние ожидающего узла, при котором он не может обслуживать или решать другие задачи управления или функционирования, отслеживая процесс решения поставленной задачи исполнителями	ActW
10	Подготовка исполнителя (ПИ) (preparation of executor) к решению задачи	PE
11	Исполнение (И) (execution) задачи функционирования или управления	EXE
12	Подготовка ответа (ПО) (доклада о выполнении) (preparation of answer). Оформление результатов решения задачи функционирования и управления узлами исполнителями	PA
13	Обработка результата (ОР) (treatment of result) выполняется	TR

Следует отметить, что, в общем случае, каждый узел ОС для множества задач может выполнять различные роли с разнообразными алгоритмами действий.

Поэтому перед решением любой задачи осуществляется динамическая настройка узла на решение конкретной задачи. Для этого необходимо построить обобщенную модель работы узла организационной структуры. Эта модель формируется методом построения минимизированного обобщенного алгоритма для множества частных алгоритмов управления [4].

Исходное множество составляют алгоритмы $\mathbf{R}_1 - \mathbf{R}_4$. Для удобства преобразований множества исходных алгоритмов заменим однозначно повторяющиеся цепочки на операторы общностей первого рода:

$$\begin{aligned} \mathbf{X}_1 &= \text{PE} \cdot \text{EXE} \cdot \text{PA}; \\ \mathbf{R}_1 &= \text{CEP} \cdot \text{IE} \downarrow^{\text{OP}} \cdot \text{AW} \cdot \text{IE} \uparrow^{\text{cl}} \cdot \text{TR}; \mathbf{R}_2 = \text{IE} \downarrow^{\text{OP}} \cdot \mathbf{X}_1 \cdot \text{IE} \uparrow^{\text{cl}}; \\ \mathbf{R}_3 &= \text{PR} \cdot \text{IE} \uparrow^{\text{OP}} \cdot \text{AW} \cdot \text{IE} \downarrow^{\text{cl}} \cdot \text{TR}; \mathbf{R}_4 = \text{IE} \uparrow^{\text{OP}} \cdot \mathbf{X}_1 \cdot \text{IE} \downarrow^{\text{cl}}; \\ \mathbf{R}_5 &= \mathbf{X}_1. \end{aligned}$$

Выделим массив линейных цепочек и операторов, имеющих повторяемость (применяемость) больше единицы (общности второго рода) $\{\mathbf{RO}_i\}$:

$$\begin{aligned} \mathbf{R}_1 &:= \{\text{IE} \downarrow^{\text{OP}}; \text{AW}; \text{IE} \uparrow^{\text{cl}}; \text{TR}\}; \\ \mathbf{R}_2 &:= \{\text{IE} \downarrow^{\text{OP}}; \mathbf{X}_1; \text{IE} \uparrow^{\text{cl}}\}; \\ \mathbf{R}_3 &:= \{\text{IE} \uparrow^{\text{OP}}; \text{AW}; \text{IE} \downarrow^{\text{cl}}; \text{TR}\}; \\ \mathbf{R}_4 &:= \{\text{IE} \uparrow^{\text{OP}}; \mathbf{X}_1; \text{IE} \downarrow^{\text{cl}}\}; \\ \mathbf{R}_5 &:= \{\mathbf{X}_1\}; \\ \mathbf{RO} &= \{\text{IE} \downarrow^{\text{OP}}; \text{AW}; \text{IE} \uparrow^{\text{cl}}; \text{TR}; \mathbf{X}_1; \text{IE} \uparrow^{\text{OP}}; \text{IE} \downarrow^{\text{cl}}\}; \\ \mathbf{RO}_1 &= \text{IE} \downarrow^{\text{OP}}; \quad \mathbf{RO}_4 = \text{TR}; \quad \mathbf{RO}_7 = \text{IE} \downarrow^{\text{cl}}. \\ \mathbf{RO}_2 &= \text{AW}; \quad \mathbf{RO}_5 = \mathbf{X}_1; \\ \mathbf{RO}_3 &= \text{IE} \uparrow^{\text{cl}}; \quad \mathbf{RO}_6 = \text{IE} \uparrow^{\text{OP}}; \end{aligned}$$

После проведения замены $\mathbf{b}_k \rightarrow \mathbf{RO}_i$ множество алгоритмов приобретает вид:

$$\begin{aligned} \mathbf{R}_1 &= \text{CEP} \cdot \mathbf{RO}_1 \cdot \mathbf{RO}_2 \cdot \mathbf{RO}_3 \cdot \mathbf{RO}_4; \\ \mathbf{R}_2 &= \mathbf{RO}_1 \cdot \mathbf{RO}_5 \cdot \mathbf{RO}_3; \\ \mathbf{R}_3 &= \text{PR} \cdot \mathbf{RO}_6 \cdot \mathbf{RO}_2 \cdot \mathbf{RO}_7 \cdot \mathbf{RO}_4; \\ \mathbf{R}_4 &= \mathbf{RO}_6 \cdot \mathbf{RO}_5 \cdot \mathbf{RO}_7; \end{aligned}$$

$$\mathbf{R}_5 = \mathbf{R}\mathbf{O}_5.$$

Построение обобщенного алгоритма (ОА) производится путем применения модифицированной свертки PCA. Свертки столбцов представляются следующими выражениями:

$$\mathbf{RS}_1 = \alpha_1 (\text{CEP} \vee_{\alpha_2} (\text{TR} \vee e)^{\alpha_2})^{\alpha_1};$$

$$\mathbf{RS}_2 = \alpha_3 (\mathbf{RO}_1 \vee_{\alpha_4} (\mathbf{RO}_6 \vee e)^{\alpha_4})^{\alpha_3};$$

$$\mathbf{RS}_3 = \alpha_5 (\mathbf{RO}_2 \vee \mathbf{R}_5)^{\alpha_5};$$

$$\mathbf{RS}_4 = \alpha_6 (\mathbf{RO}_3 \vee_{\alpha_7} (\mathbf{RO}_7 \vee e)^{\alpha_7})^{\alpha_6};$$

$$\mathbf{RS}_5 = \alpha_8 (\mathbf{RO}_4 \vee e)^{\alpha_8}.$$

Обобщенный алгоритм \mathbf{R}_0 формируется путем объединения сверток столбцов \mathbf{RS}_i при помощи операции логического умножения PCA:

$$\mathbf{R}_0 = \prod_{i=1}^n \mathbf{RS}_i,$$

где n – количество сверток столбцов.

Проведем обратную замену $\mathbf{RO}_i \rightarrow \mathbf{b}_k \subset \mathbf{B}$. Обобщенный алгоритм работы узла ОС \mathbf{R}_0 в базисе основных операторов представляется следующим образом:

$$\mathbf{R}_0 = \alpha_1 (\text{CEP} \vee_{\alpha_2} (\text{TR} \vee e)^{\alpha_2})^{\alpha_1} \cdot \alpha_3 (\text{IE} \downarrow^{\text{OP}} \vee_{\alpha_4} (\text{IE} \uparrow^{\text{OP}} \vee e)^{\alpha_4})^{\alpha_3} \times \\ \times \alpha_5 (\text{AW} \vee \text{PE} \cdot \text{EXE} \cdot \text{PA})^{\alpha_5} \alpha_6 (\text{IE} \uparrow^{\text{cl}} \vee_{\alpha_7} (\text{IE} \downarrow^{\text{cl}} \vee e)^{\alpha_7})^{\alpha_6} \cdot \alpha_8 (\text{TR} \vee e)^{\alpha_8}.$$

Значения условий переходов по ОА приведены в табл. 2.

Настройка узла ОС на выполнение своей роли в решении задачи проводится путем определения условий α_j .

Опишем смысловое значение условий перехода:

$$\alpha_1 = \begin{cases} 1 - \text{если узел ОС является начальником (посредник ом),} \\ \quad \text{который готовит УВ для подчиненных узлов нижнего уровня;} \\ 0 - \text{если нет;} \end{cases}$$

$$\alpha_2 = \begin{cases} 1 - \text{если подчиненный узел требует решения задачи управления} \\ \quad \text{от начальника (узла верхнего уровня);} \\ 0 - \text{если нет;} \end{cases}$$

$$\alpha_3 = \begin{cases} 1 - \text{если УВ передается на нижний уровень} \\ \quad \text{подчиненно му узлу (с генерацией протокола);} \\ 0 - \text{если нет;} \end{cases}$$

$$\alpha_4 = \begin{cases} 1 - \text{если требование на решение задачи управления} \\ \quad \text{передается на верхний уровень ОС (с генерацией протокола);} \\ 0 - \text{если нет;} \end{cases}$$

$$\alpha_5 = \begin{cases} 1 - \text{если узел ОС находится в режиме ожидания ответа;} \\ 0 - \text{если узел ОС является непосредственным исполнителем} \\ \quad \text{задачи функционирования или управления;} \end{cases}$$

$$\alpha_6 = \begin{cases} 1 - \text{если узел получает результаты решения задачи} \\ \quad \text{от узла исполнителя нижнего уровня;} \\ 0 - \text{если нет;} \end{cases}$$

$$\alpha_7 = \begin{cases} 1 - \text{если узел получает решение задачи управления} \\ \quad \text{от узла верхнего уровня;} \\ 0 - \text{если нет;} \end{cases}$$

$$\alpha_8 = \begin{cases} 1 - \text{если производится обработка результатов решения задачи;} \\ 0 - \text{если нет.} \end{cases}$$

Таблица 2

Значение условий переходов по ОА

α_j R_i	R_1	R_2	R_3	R_4	R_5
α_1	1	0	0	0	0
α_2	0*	0	1	0	0
$\alpha_3 = \alpha_6$	1	1	0	0	0
$\alpha_4 = \alpha_7$	0	0	1	1	0
$\alpha_5 = \alpha_8$	1	0	1	0	0

Настройка обобщенного протокола взаимодействия (ОПВ) осуществляется пользователем при формировании ответов на четко поставленные вопросы. Ответы на вопросы могут принимать два значения 1 («да») и 0 («нет»). Для решения этой задачи необходимо для каждой задачи сформировать опросный лист, отвечающий требованиям однозначности.

Если в узле ОС объединены все роли и он самостоятельно решает задачу, то вектор значений условий будет иметь вид

$$\bar{\alpha} = (0, 0, 0, 0, 0, 0, 0, 0).$$

Если инициатором решения задачи является начальник (узел верхнего уровня), то

$$\bar{\alpha} = (1, 0, 1, 0, 1, 1, 0, 1).$$

Если необходимо решать задачу управления на верхнем уровне (инициатор – подчиненный), то

$$\bar{\alpha} = (0, 1, 0, 1, 1, 0, 1, 1).$$

Если решение задачи выполняется узлом нижнего уровня (относительно инициатора), то

$$\bar{\alpha} = (0, 0, 1, 0, 0, 1, 0, 0).$$

Если решение задачи выполняется на более высоком уровне ОС, то

$$\bar{\alpha} = (0, 0, 0, 1, 0, 0, 1, 0).$$

Узлы, отдавшие указания на решение задач, могут находиться в режимах пассивного или активного ожидания, т.е.

$$AW = \overline{\alpha_{PO}} (PW \vee ActW)^{\alpha_{PO}},$$

где $\overline{\alpha_{PO}} = \begin{cases} 1 - \text{режим пассивного ожидания;} \\ 0 - \text{режим активного ожидания.} \end{cases}$

Выводы. Использование ОПВ позволяет существенно сократить время настройки системы имитационного моделирования в процессе проведения экспериментов по исследованию влияния механизмов взаимодействия между элементами ОС на выполнение проектов и программ.

ЛИТЕРАТУРА

1. Глушков В.М., Цейтлин Г.Е., Юценко Е.Л. Алгебра, языки программирования – К.: Наука. думка, 1974. – 327 с.
2. Иванов П.М. Аксиоматизация микропрограммных алгебр // IV Республиканская научная конференция молодых исследователей по системотехнике – К. – 1969. – Т. 2.
3. Илюшко В.М., Попов В.А. Об одной модификации регулярных схем алгоритмов // Математические методы анализа динамических систем – Х.: ХАИ, 1977.
4. Дружинин Е.А. Методика построения минимизированного обобщенного алгоритма для множества частных линейных алгоритмов // Проектирование комплексно-автоматизированных производств. – Х.: ХАИ. – 1990. – С. 91 – 99.

Поступила 20.08.02

ДРУЖИНИН Евгений Анатольевич, канд. техн. наук, доцент Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ». Область научных интересов – автоматизированные системы обработки информации и управления.

E-mail: druzhinin@xai.edu.ua

БЕК Владимир Александрович, соискатель Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ». Область научных интересов – автоматизированные систе-

мы обработки информации и управления.

E-mail: Vladimir Bek@telesensKSCL.com.ua

ГОРЛОВ Дмитрий Александрович, аспирант Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ». Область научных интересов – автоматизированные системы обработки информации и управления.

E-mail: gorlov@megabank.net
