

М. Г. Доррер, В. В. Курохтин

РЕШЕНИЕ ЗАДАЧИ ПРЯМОГО И ОБРАТНОГО ПРЕОБРАЗОВАНИЯ МЕЖДУ ЦВЕТНОЙ СЕТЬЮ ПЕТРИ И МОДЕЛЬЮ БИЗНЕС-ПРОЦЕССОВ

Рассмотрена методика транслирования моделей бизнес-процессов в раскрашенные сети Петри, предназначенная для исследования исходного процесса путем имитационного моделирования.

Имитационное моделирование – это метод исследования, основанный на подмене реальной системы на ее модель. С моделью можно производить различные эксперименты, которые в действительности проводить было бы рискованно или невозможно.

Следует отметить, что данный в сфере моделирования бизнеса построение модели проводит эксперт (либо группа экспертов), опираясь на свои знания и опыт. Несмотря на то, что в дальнейшем эта модель может быть проверена другими специалистами, но в любом случае речь будет идти о субъективном мнении, а не об объективных результатах, в то время как реализовать на практике различные варианты работы системы очень часто невозможно.

Адаптация имитационного моделирования к бизнес-моделям позволит проводить различные эксперименты за короткое время для обнаружения и устранения различных недостатков конкретной реализации бизнес-системы.

В качестве объекта исследования авторами были выбраны организационные системы. Метод исследования – структурный анализ систем при помощи методик моделирования бизнес-процессов [1; 2; 3; 4]. Имитационное моделирование динамических систем осуществлялось при помощи цветных сетей Петри (*Colored Petri Nets*, CPN) [5, 6].

Главная цель, которую ставили перед собой авторы, – опираясь на стандартные инструменты, формализовать процесс имитационного моделирования бизнес-систем. Эта цель достигается путем прямого преобразования между нотациями моделирования систем (в данном случае это ARIS eEPC [2] и цветные сети Петри в нотации, предложенной К. Йенсенем в [5] и реализованной в программном комплексе CPNTools¹). При этом в трансляции из нотации CPN в нотацию ARIS в первую очередь интересны результаты имитационного моделирования.

Диаграмма в нотации ARIS eEPC может быть описана трехдольным графом D . К типам вершин данного графа будут относиться следующие события, функции и ресурсы:

$$D = (S, F, R, J, L_p, L_r, L_j, \Phi_p, \Phi_r, \Phi_j, I, Q, W,$$

где S – множество вершин типа «состояние»; F – множество вершин типа «функция»; R – множество вершин типа «ресурс»; J – множество логических развязок, используемых для расширения множества моделируемых систем; L_p – множество ребер, между вершинами множеств S и F (процессные дуги); L_r – множество ребер, соединяющих вершины множеств F и R (ресурсные дуги); L_j – множество ребер, соединяющих вершины множеств S, F и J ; Φ_p – отношение инцидентности, определяющее направленные связи между вершинами множества S и F : $\Phi_p : l_p \rightarrow i_p$, здесь $i_p \in (S \times F) \cup (F \times S)$; Φ_r – отношение инцидентности, определяющее не направленные связи между вершинами множества F и R : $\Phi_r : l_r \rightarrow \{f, r\}$, здесь $l_r \in L_r, f \in F, r \in R$; Φ_j – отношение инцидентности, определяющее направленные связи между вершинами множества S, F и J : $\Phi_j : l_j \rightarrow i_j$, здесь $i_j \in \varphi \subset (S \times J) \cup (F \times J) \cup (J \times S) \cup (J \times F) \cup (J \times J)$; для данного множества справедливы следующие ограничения:

$$\begin{aligned} \forall (j, s) \in \varphi \mid j \in J, s \in S &\Rightarrow (j, f) \notin \varphi \quad \forall f \in F, \\ \forall (j, f) \in \varphi \mid j \in J, f \in F &\Rightarrow (j, s) \notin \varphi \quad \forall s \in S, \\ \forall (s, f) \in \varphi \mid j \in J, s \in S &\Rightarrow (f, j) \notin \varphi \quad \forall f \in F, \\ \forall (f, j) \in \varphi \mid j \in J, f \in F &\Rightarrow (s, j) \notin \varphi \quad \forall s \in S, \\ \forall (j_1, j_2) \in \varphi &\Rightarrow j_1 \neq j_2. \end{aligned}$$

$$\forall (j_1, j_2) \in \varphi \Rightarrow \left\{ \begin{array}{l} (\sigma, j_1) \in \varphi \Rightarrow \left\{ \begin{array}{l} (\varphi, j_2) \notin \varphi \quad \forall \varphi \in \Phi \\ (j_2, \sigma) \notin \varphi \quad \forall \sigma \in \Sigma \end{array} \right\} \\ (\varphi, j_1) \in \varphi \Rightarrow \left\{ \begin{array}{l} (\sigma, j_2) \notin \varphi \quad \forall \sigma \in \Sigma \\ (j_2, \varphi) \notin \varphi \quad \forall \varphi \in \Phi \end{array} \right\} \\ (j_1, \sigma) \in \varphi \Rightarrow \left\{ \begin{array}{l} (\sigma, j_2) \notin \varphi \quad \forall \sigma \in \Sigma \\ (j_2, \varphi) \notin \varphi \quad \forall \varphi \in \Phi \end{array} \right\} \\ (j_1, \varphi) \in \varphi \Rightarrow \left\{ \begin{array}{l} (\varphi, j_2) \notin \varphi \quad \forall \varphi \in \Phi \\ (j_2, \sigma) \notin \varphi \quad \forall \sigma \in \Sigma \end{array} \right\} \end{array} \right\},$$

т. е. для логической развязки входными и выходными могут быть только элементы одного типа; I определяет время работы функции: $I : f \rightarrow i$, где $f \in F, i$ – целое положительное число (как и везде далее по тексту); Q – начальное количество ресурсов: $Q : r \rightarrow i$, здесь $r \in R$; W – веса дуг множества L_r , определяющие количество потребляемых либо производимых ресурсов: $W : l_r \rightarrow i$, здесь $l_r \in L_r$.

Раскрашенную временную сеть Петри можно задать следующим образом:

$$\begin{aligned} N &= (P, T, L, C, \Phi_t, \Phi_c, \Phi_w, \Phi_i) N = \\ &= (P, T, L, C, \Phi_1, \Phi_c, \Phi_w, \Phi_i), \end{aligned}$$

¹ Подробную информацию о программном комплексе CPNTools можно получить на сайте: <http://wiki.daimi.au.dk/spntools/spntools.wiki>. (см. п. 3 библиографического списка на С. 5).

где P – множество позиций сети; T – множество переходов сети; L – множество ребер сети; C – множество цветов сети; $\Phi_l : l \rightarrow i_l$ – функция инцидентности, определяющая связи между позициями и переходами, здесь $l \in L, i_l \in \varepsilon \subset (P \times T) \cup (T \times P)$; $\Phi_c : c \rightarrow i_c$ – функция, определяющие разрешенные для позиции цвета, здесь $c \in C, i_c \subseteq C$; $\Phi_w : l \rightarrow i_w$ – функция, определяющая выражения на дугах, здесь $l \in L, i_w$ – мультимножество цветов (элементов множества C); $\Phi_t : t \rightarrow i$ – функция, определяющая время срабатывания перехода, здесь $t \in T$.

Данное подмножество сети Петри достаточно для описания трансляции моделей.

Введем набор транслирующих функций (грамматик), позволяющих произвести преобразование элементов исходной модели в элементы целевой модели. Для этого в сеть Петри необходимо добавить следующие дополнительные элементы:

- цвет «Текущий процесс», который будем обозначать c_p . Фишки этого цвета будут перемещаться по ребрам, соответствующим связям между событиями и функциями, т. е. будут обозначать протекание процесса во времени;

- позиция «Хранилище ресурсов», которую будем обозначать как p_r . В данной позиции изначально будут находиться все фишки, соответствующие состоянию ресурсов системы как стартовому:

$$\Phi_r^t(r) \in \Phi_c(p_r) \quad \forall r \in R.$$

Введем понятие *кластер сети Петри*. Кластер сети Петри является ее подмножеством, т. е. множества P, T, L и C являются подмножествами сети более высокого уровня, а функции Φ_t, Φ_c и Φ_w равны функциям сети более высокого уровня. Это понятие понадобится нам в дальнейшем.

Введем следующие функции:

- $\Phi_s^t : s \rightarrow p$ – функция трансляции состояний системы, где $s \in S, p \in P$, причем

$$\Phi_s^t(s_1) \neq \Phi_s^t(s_2) \quad \forall s_1, s_2 \in S, s_1 \neq s_2,$$

- $\Phi_f^t : f \rightarrow t$ – функция трансляции функций системы, где $f \in F, t \in T$, причем

$$\Phi_f^t(f_1) \neq \Phi_f^t(f_2) \quad \forall f_1, f_2 \in F, f_1 \neq f_2,$$

кроме того $\Phi_t(t) = I(f)$;

- $\Phi_r^t : r \rightarrow c$ – функция трансляции ресурсов системы, где $r \in R, c \in C$, причем

$$\Phi_r^t(r_1) \neq \Phi_r^t(r_2) \quad \forall r_1, r_2 \in R, r_1 \neq r_2,$$

- $\Phi_j^t : j \rightarrow N_c$ – функция трансляции логических развязок системы, где $j \in J$; N_c – кластер сети Петри, обладающий следующими свойствами:

$$\forall (j, s) \in L_j \exists (N_c, p) \in L,$$

$$\forall (j, f) \in L_j \exists (N_c, t) \in L,$$

$$\forall (s, j) \in L_j \exists (p, N_c) \in L,$$

$$\forall (f, j) \in L_j \exists (t, N_c) \in L,$$

$$\forall (j_1, j_2) \in L_j \exists (N_{c1}, N_{c2}) \in L$$

где $s \in S; f \in F; p \in P; t \in T; p = \Phi_s^t(s); t = \Phi_f^t(f)$;

- $\Phi_{lp}^t : l_p \rightarrow l$ – функция трансляции процессных дуг системы, где $l_p \in L_p, l \in L$, причем

$$\Phi_{lp}^t(l_{p1}) \neq \Phi_{lp}^t(l_{p2}) \quad \forall l_{p1}, l_{p2} \in L_p, l_{p1} \neq l_{p2}, \Phi_w(l) = 1c_p;$$

- $\Phi_{lr}^t : l_r \rightarrow \{l_i, l_o\}$ – функция трансляции ресурсных дуг системы, где $l_r \in L_r; l_i, l_o \in L$, для l_i и l_o справедливо следующее: $\Phi_l(l_i) = (p_r, \Phi_f^t(f)), \Phi_l(l_o) = (\Phi_f^t(f), p_r)$, где $f \in F; f \in \Phi_r(l_r); \Phi_w(l_i) = \Phi_w(l_o) = m_r$, m_r равно мультимножеству с количеством фишек цвета $\Phi_r^t(r)$, где $r \in R$ и $r \in \Phi_r(l_r)$, равным $W(l_r)$;

- $\Phi_{lj}^t : l_j \rightarrow l$ – функция трансляции дуг логических развязок системы, где $l_j \in L_j; l \in L$, причем l связывает кластер сети Петри $\Phi_j^t(j)$ с другой вершиной сети Петри.

Используя определенные выше функции, выполним трансляцию моделей в нотации ARIS eEPC в цветные сети Петри. Кластеры будем использовать для моделирования логических развязок.

По определению логической развязки «Или» функция Function может выполняться в семи различных комбинациях, соответствующих значению «Истина» дизъюнкции трех высказываний, принимая за «Истину» развитие процесса по каждой из ветвей Event 1...3 (рис. 1). В соответствующем кластере сети Петри будет по одному переходу для каждого случая. Эти переходы связывают три входные вершины с выходной. Связи между вершинами и переходами определяют, из какой позиции забирать фишки, т. е. какие события уже наступили.

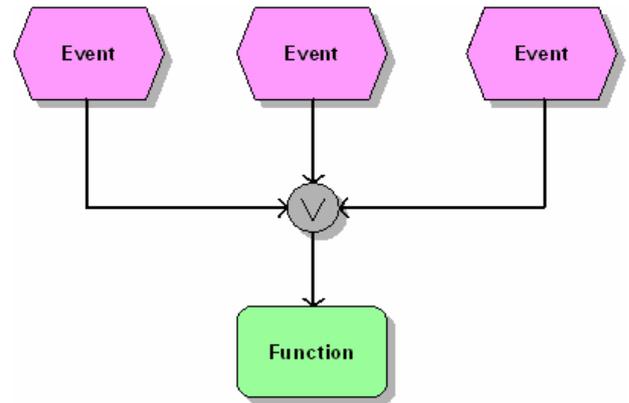


Рис. 1

Но применение кластера в таком виде не удобно, так как присутствует необходимость в соответствии сложной схемы соединения исходных элементов (событий) и результирующего (функции). Для решения данной проблемы в кластер были добавлены еще два интерфейсных слоя (рис. 2).

Аналогично решается задача преобразования для развязок, соответствующих логическим функциям «И» и «Исключающее или», а также для кластеров, соответствующих разветвлению логики процесса.

После построения дерева маркировок полученной сети Петри производим ее отображение на исходные объекты бизнес-модели, т. е. обратную трансляцию. Для этого определим функции обратной трансляции, обратные описанным выше:

- Φ_s^{-1} – функция трансляции позиций сети Петри в состояния системы;

– Φ^{-1}_f – функция трансляции переходов сети Петри в функции системы;

– Φ^{-1}_r – функция трансляции цветов сети Петри в ресурсы.

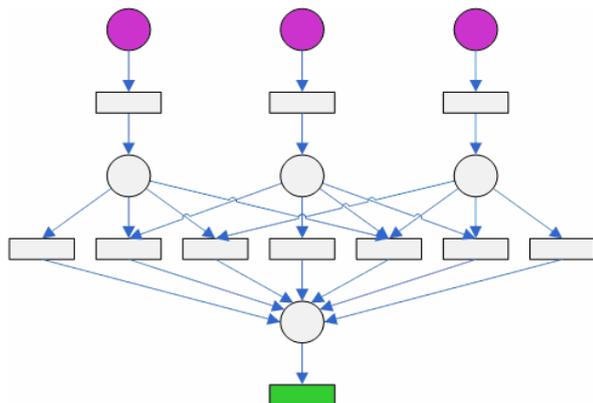


Рис. 2

Процесс обратной трансляции выглядит следующим образом:

1. Исходное состояние для каждой вершины дерева получаем на основе функции Φ^{-1}_s .

2. За счет применения функции Φ^{-1}_f и информации о переходе, которому соответствует дуга, производим преобразование каждой дуги дерева маркировок в дополнительную вершину дерева состояний системы.

3. Для каждой вершины определим ресурсы, потребляемые и производимые ею:

– с помощью функции Φ_l находим связи перехода с хранилищами ресурсов (p_r);

– если такая связь существует, то при помощи функции Φ_w определяем выражения на дугах, связывающих переход с хранилищем, которые затем с помощью функции Φ^{-1}_r транслируем в потребляемые и производимые ресурсы.

Таким образом получаем расширенное дерево всех возможных состояний и функций системы.

Анализ минимального и максимального количества фишек в позиции сети Петри, примененный к позиции «Хранилище ресурсов», позволяет определить минимальное и максимальное наличие ресурсов в системе за анализируемый период. Использование функции Φ^{-1}_r дает возможность получить данную информацию в терминах бизнес-системы.

Исследуя переходы сети Петри на живость и применяя к результатам функцию Φ^{-1}_f , можно

определить мертвые (никогда не выполняющиеся) функции бизнес-системы.

Кроме того, информация о типах листьев дерева состояний и функций системы позволяет определить тупиковые состояния системы. Это особенно важно для циклических процессов. Для таких процессов любой лист дерева состояний и функций системы должен быть либо циклом, либо повтором. Если же он является просто листом (тупиковым элементом), то цепочка событий (путь от вершины дерева к данному листу) является цепочкой событий, блокирующих систему.

При трансляции исходной диаграммы получается сеть Петри, расширенная временем, в связи с чем можно проанализировать время работы процесса в целом. Минимальное и максимальное время работы процесса определяют как минимальное и максимальное время сети на момент остановки при различных комбинациях срабатывания переходов. При этом следует отметить, что данная информация не может быть получена для циклических процессов.

Таким образом, разработанные авторами методика и программное обеспечение обеспечивают решение задачи динамического моделирования бизнес-систем на основе результатов регламентации бизнес-процессов.

Библиографический список

1. Калянов, Г. Н. Теория и практика реорганизации бизнес-процессов / Г. Н. Калянов. М. : СИНТЕГ, 2000. 212 с.
2. Моделирование бизнеса. Методология ARIS. Практическое руководство / М. Каменнова, А. Громов, М. Ферапонтов, А. Шматалюк. М. : Весть-Метатехнология, 2001. 327 с.
3. Шеер, А. В. Бизнес-процессы. Основные понятия. Теория. Методы : пер. с англ. / А. В. Шеер. 2-е изд., испр. и доп. М. : АОЗТ «Просветитель», 1999. 153 с.
4. Шеер, А. В. Моделирование бизнес-процессов : пер. с англ. / А. В. Шеер. 2-е изд., испр. и доп. М. : ООО «Издательство „Серебряные нити“», 2000. 205 с.
5. Jensen, K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use / K. Jensen. Berlin : Spingler, 1996. 1997.
6. Котов, В. Е. Сети Петри / В. Е. Котов. М. : Наука, 1984.

M. G. Dorrer, V. V. Kurokhtin

DECISION OF TASK OF DIRECT AND REVERSE TRANSFORMATION BETWEEN COLORED PETRI NETS AND BUSINESSMODELS

It is considered the methods of translating business models into colored Petri nets for the purpose of source process analysis by the imitation modeling.