

ПРИМЕНЕНИЕ ИЕРАРХИЧЕСКИХ КОНЕЧНЫХ АВТОМАТОВ В ПОВЕДЕНЧЕСКИХ МОДЕЛЯХ ГРУПП

Круковский М.Ю., аспирант

(Институт проблем математических машин и систем НАНУ, г.Киев)

Современное общество все интенсивнее использует решения информационных технологий [1]. Внедрение систем автоматизации совместной работы делает услуги управления более качественными. Известны примеры применения теории графов и теории автоматов к поведенческим моделям [2-4]. В данной работе предложены формальная модель документооборота, построенная на графах [5] и автоматах [6]. Модель связности поведенческих единиц на графах. Для построения графа модели документооборота предлагается использовать способ отображения документооборота графами, предложенный автором в работе [5]. Для задания множества вершин графа будем использовать множество возможных состояний Φ . Ребра графа зададим с помощью множества действий D . Установим это соответствие таким образом, чтобы выполнялись следующие правила: одной вершине графа соответствует один и только один элемент множества Φ ; одному ребру графа соответствует один и только один элемент множества D ; одному элементу множества Φ соответствует одна и только одна вершина графа; одному элементу множества D соответствует одно и только одно ребро графа.

Такое отображение множеств состояний Φ в множество вершин v и множества состояний D в множество ребер e можно математически определить следующим образом: для любого i справедливо утверждение $v(i) = \Phi(i)$ и $e(i) = D(i)$, где $i \in I$, $I=1,2,3..n$. Определяются две парных грамматики – первая грамматика для установления перевода Φ в v , вторая грамматика – для установления перевода D в e . Таким образом, связи между вершинами тождественно соответствуют связям состояний моделируемого документооборота. В графе документооборота вершины графа соединяют ребра только в том случае, если соответствующие вершинам состояния связаны действием, соответствующим ребру, то есть $e = \{e, \text{ если ребро существует; } 0, \text{ если ребро отсутствует}\}$. Направленность ребер устанавливается так, чтобы отображать логику последовательности смены состояний документооборота. Вершина i является входящей для вершины j через ребро k только в том случае, если состояние i сменяется на состояние j после совершения действия k . Состояниям y_1, y_2, \dots, y_n сопоставляются вершины графа v_1, v_2, \dots, v_n , и каждая пара вершин v_i и v_j соединена дугой e_{ij} , от v_i к v_j только в том случае, когда состояние v_i является входным для v_j .

Модель поведенческой единицы на автоматах В работе [6] система представляется в виде детерминированного конечного автомата. Автомат, представляющий поведенческую единицу документооборота задается обще-

принятой нотацией конечного автомата. В соответствии с ней автомат представляется следующим образом: $D_T = (Q, \Sigma, \delta, q_0, F)$,

где Q – конечное множество состояний документов, тождественное множеству $\{\Phi\}$ из определения композитного документооборота; Σ – конечное множество входных символов, образующих входной алфавит и представляющее собой данные, которые поступают на вход системы документооборота; δ – функция переходов, аргументами которой являются текущее состояние и входной символ, а значением – новое состояние; q_0 – начальное состояние (или множество начальных состояний) из множества Q ; F – множество заключительных, или допускающих состояний из множества Q .

Множество состояний автоматов $\{Q\}$ получается из множества состояний документов. Завершающие состояния выделяются из общего множества состояний путем анализа каждого из состояний. Если при анализе выявляется состояние, которое имеет одну или несколько входящих связей, но не имеет ни одной исходящей, то оно помечается как завершающее. Состояния моделирующего автомата упорядочиваются таким образом, чтобы документооборот был представлен состояниями документа в порядке от начального состояния документа к конечному состоянию.

Автомат исполняет функции переходов для принятия решения о выборе следующего состояния. Функции переходов программируются с помощью анализа действий участников документооборота. Производимое действие определяет результирующее состояние, для которого входными данными для определения выбора являются текущее состояние документа и участник процесса. Автомат реализует документооборот, в котором на каждом шаге происходит действие на основании процесса и на основании анализа текущего состояния документа (исполнителя), принимается решение о следующем состоянии документа. Функция перехода F_i автоматной модели является i -м элементом множества действий $\{D\}$ документооборота, после выполнения которого, происходит смена состояния s_i на состояние s_{i-1} .

В реализованной автоматной модели документооборота в качестве алфавита автомата использован список участников. Символами обозначены ролевые участники производственных сценариев. Из этих символов образуются последовательности, которые обрабатываются автоматом формальной модели. Последовательности символов, которые обрабатываются автоматом документооборота, являются допустимыми для модели. Такие последовательности символов приводят к корректному выполнению процесса документооборота и получению конечного результирующего документа. Последовательности символов документооборота, которые не принимаются автоматом, реализующим модель документооборота, являются недопустимыми для моделируемого процесса. Последовательности, принимаемые автоматом модели документооборота, являются его языком. Слова этого языка являются возможными последовательностями участников документооборота, участвующих в работе над документами.

Программирование автомата осуществляется путем установления соответствия между нотацией конечного автомата и композитного документооборота. То есть, после проведения декомпозиции процессов и синтеза модели $\{Y, D, \Phi\}$ строится автоматная модель документооборота, которая определяется пятеркой (A, S, s_0, T, F) , где $\{A\} \equiv \{Y\}$, $\{S\} \equiv \{\Phi\}$, $s_0 = \phi_0$, $\{S\} \equiv \{\Phi_n\}$, $\{F\} \equiv \{D\}$. При этом на каждом шаге автомата, множества, определяющие его состояния, соответствуют множествам, описывающим жизненный цикл модели документооборота. На каждом шаге существует соответствие A_i и F_i : $\forall i, j \in N \cap i - j \Rightarrow \{(s_0 = \phi_0), (s_i = \phi_j), A_i = Y_j, F_i = D_j\}$.

Иерархический конечный автомат Для представления иерархического конечного автомата документооборота будем использовать описание ИКА из работы [7]. В соответствии с ним ИКА представляется пятеркой (I, O, S, T, r) , где I и O описывают множества входных и выходных алфавитов, S представляет множество состояний, функции переходов $T: S \times B^{|I|} \times B^{|O|} \times S \rightarrow B$ $B = \{0, 1\}$, r задает начальное состояние. Детерминированный ИКА может быть представлен шестеркой $(I, O, S, \lambda, \delta, r)$, где $\lambda: S \times B^{|I|} \rightarrow B^{|O|}$ описывает выходную функцию и $\delta: S \times B^{|I|} \rightarrow S$ описывает функции переходов.

Про данную пару входных и выходных последовательностей (σ_i, σ_o) , где $\sigma_i = (i_0, \dots, i_{t-1})$ и $\sigma_o = (o_0, \dots, o_{t-1})$ говорят, что она принимается ИКА $T=(I, O, S, T, r)$, если существует последовательность состояний (s_0, \dots, s_t) такая, что $T(s_j, i_j, o_j, s_{j+1}) = 1$ для всех $j=0, \dots, t-1$ и $s_0 = r$.

В настоящей работе рассматривается поведение ИКА, поведение которого определяется отношением элементов входа и выхода. Иными словами, отношение между алфавитами входа и выхода есть набор пар входов и выходов, которые принимает детерминированный ИКА. Для заданного автомата $T=(I, O, S, T, r)$ поведение между входом I и выходом O содержится в функциях переходов T , если каждая пара последовательности входов и выходов реализуется в T . Рассмотрим реализацию ИКА, управляющего конечными автоматами на примере рис. 1. Заданы автоматы $M=(I, O, S, T_M, r)$ и $M2 = (X \cup V, U \cup Z, S_2, (\lambda_2^{U|}, \lambda_2^{Z|}), \delta_2, r_2)$. Предположим, что система документооборота может принимать сразу несколько состояний, в то время как один исполнитель производит смену состояния только на одно из возможных. Таким образом, M может быть НДКА, в то время, как $M2$ является только ДКА. Выходная функция λ_2 автомата $M2$ состоит из $\lambda_2^{U|}$ и $\lambda_2^{Z|}$, которые соответственно определяют выходные функции U и Z .

Заданы подмножество s^* из множества состояний S и вход x автомата M , зададим $\Delta(s^*, x)$ как множества всех возможных выходов. То есть $z \in \Delta(s^*, x)$ в том и только том случае, если существует $(\tilde{s}, s) \in s^* \times S$ такое, что $T_M(\tilde{s}, x, z, s) = 1$. Аналогично, $\Delta(\tilde{s}, x)$ будет множеством всех возможных состояний, то есть $s \in \Delta(\tilde{s}, x)$ в том и только том случае, если $\exists(\tilde{s}, z) \in s^* \times B^{|I|}$ такое, что $T_M(\tilde{s}, x, z, s) = 1$.

В рамках определения иерархического конечного автомата, который реализует комплексную систему документооборота, рассмотрим реализуемость и

допустимость возможных моделей документооборота. При заданном ДКА $M1 = (V, U, S_1, \lambda_1, \delta_1, r_1)$ считается реализуемым на $M1$ если существует хотя бы одна пара циклических реализаций $M1$ и $M2$ таких, что их соединение не вызывает цикла между U и V .

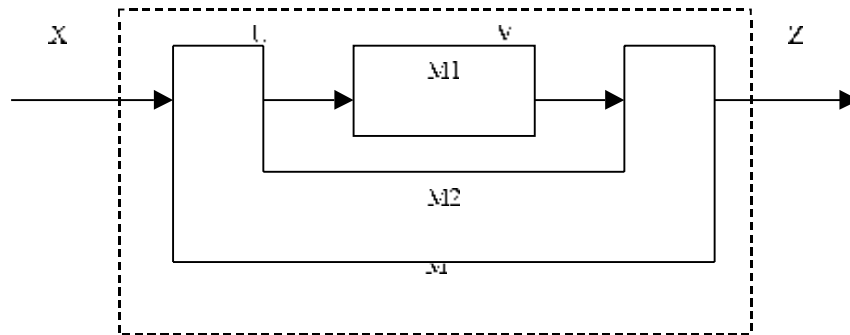


Рисунок 1. Пример иерархического конечного автомата

При заданных автоматах $M = (Y, Z, S, T_M, F)$ и $M2 = (Y \cup V, U \cup Z, S_2, \lambda_2, \delta_2, r_2)$ ДКА $M1 = (V, U, S_1, \lambda_1, \delta_1, r_1)$ является допустимым автоматом если автомат $M1$ реализуем и поведение $M1 \times M2$ содержится в M , где $M1 \times M2$ является выходным результатом M . Поведение, которое реализуется допустимым автоматом является допустимым.

Представление модели документооборота в виде ИКА позволяет применить апробированный аппарат. В результате сложные процессы представлены в виде единого автомата, который оперирует автоматами моделирования единицы поведения системы. Каждый из автоматов моделирует поведение участника, который обрабатывает изменения состояний документов. Последовательность обработки автоматов центральным автоматом определяется связями, описанными графом, который определяет связность системы. Такое представление позволило реализовать формальную модель двигателя управленческих процессов.

Перечень источников

1. Теслер Г.С. Новая кибернетика.- Киев: Логос, 2004. – 401с.
2. Алферова З.В. Математическое обеспечение экономических расчетов с использованием теории графов.:М. Статистика.- 1974. – 208с
3. Clarence Ellis. Team Automata for Groupware Systems. Phoenix, Arizona:ACM SIGGROUP. P.415-424
4. Marc Hoffman, David Shute, Mike Ebbers. Advanced Workflow Solutions. New York: Redbooks IBM, 1999, 141 p.
5. Круковский М.Ю. Графовая модель композитного документооборота// Математичні машини і системи. – 2005. – № 3. – С. 149с – 163с.
6. Круковский М.Ю. Автоматная модель композитного документооборота. // Математические машины и системы.-2004.-№4.-С.37-50.
7. Yosinori Watanabe, Robert Brayton. The maximum set of permissible behaviors of FSM networks.-Los Alamitos CA USA: IEEE Computer society press.-1993.-pp.316-420