**В.А.Филатов, О.Ф. Козырь**

# Модель поведения автономного сценария в задачах управления распределенными информационными ресурсами

## Введение

Важную роль в повышении эффективности функционирования любого предприятия, интенсификации и развитии управленческих и инновационных процессов играет качество управления информационными ресурсами (УИР), под которыми будем понимать информацию и инструменты работы с нею.

В условиях постоянного развития информационных систем и технологий становится невозможно эффективно решать вручную множество сложных задач управления информационными ресурсами. Поэтому, учитывая все возрастающий спрос на программные средства, которые облегчают администрирование в распределенных системах и выполняют те или иные задачи информационной поддержки, разработчики предлагают все новые решения для управления информационными ресурсами, расширяют возможности существующих программных продуктов.

Однако в настоящее время не существует универсального класса систем УИР, которые отвечали бы потребностям предприятий различных масштабов и разного уровня автоматизации процессов управления. Поэтому поиск доступных, гибких и универсальных средств управления всем многообразием информационных ресурсов распределенных систем является одной из важнейших задач ИТ-индустрии.

Такие свойства программных сценариев, как автономность, целенаправленность, мобильность и адаптивность, а также независимость от сред и квалификации разработчика, способность решать и простые, и сложные задачи, как на сервере, так и на стороне клиента, позволяют рассматривать их как одно из перспективных направлений развития УИР.

Для системной проработки вопросов возможности и целесообразности применения автономных сценариев в автоматизации процессов УИР необходимы формализация и исследование их структуры и поведения при решении различных классов задач УИР.

## *1Разработка концептуальной модели автономного сценария*

Рассмотрим один из подходов к созданию модели универсального автономного сценария, ориентированного на решение задач управления информационными ресурсами вычислительных систем, на основе концепции фреймов [1,2]. В общем случае она может быть записана в виде:

$$FR\{< R_1, C_{11,} C_{12,..} C_{1m,} >, < R_2, C_{21}, C_{22,} ...C_{2m,} >, ..., < R_k, C_{k1}, ..., C_{km} >\},$$

(1)

где FR – имя фрейма;

совокупность $<R_i,C_{i1},\ldots,C_{ij},\ldots,C_{im}>$ – описание i-го слота фрейма;

$R_i$ – имя i-го слота, $C_{ij}$ – j-ое значение i-го слота.

На основании фрейма (1), может быть реализована модель программного сценария в терминах <объекты>, <условия>, <действия>, <результаты>. Фрейм выступает в виде универсального каркаса или типовой оболочки, в которую могут добавляться функциональные модули-слоты для решения конкретных задач управления информационными ресурсами.

Каждый слот фрейма связан с конкретным объектом информационного пространства и выполняет с ним заданное действие. Математическое описание слота имеет вид:

$$\mathrm{Slot} = \langle \mathrm{U}, \mathrm{D}, \mathrm{dom}, \mathrm{r_i}, \theta, \Omega \rangle, \qquad (2)$$

где $\mathrm{U}$ – множество имен атрибутов, $\mathrm{D}$ – множество доменов, $\mathrm{dom}$ – отображение $\mathrm{U} \Rightarrow \mathrm{D}$, $\theta$ – множество, определяющее начальные условия и признаки выполнения действий в структуре задания, $\Omega$ – множество операций, при этом $\Omega = \{\Omega_1, \Omega_2, \Omega_3\}$, где $\Omega_1$ – операции над слотами-кортежами, $\Omega_2$ – операции над состояниями кортежей, $\Omega_3$ – операции над значениями типовых атрибутов. При этом $\Omega_3 = \{\Omega_{31}, \Omega_{32}\}$, где $\Omega_{31}$ – операции над данными одного типа, $\Omega_{32}$ – межтиповые операции, $\mathrm{r_i}$ – модель-кортеж i-го задания автономного сценария.

Кортеж $\mathrm{r_i}$ в модели (2) может быть представлен в виде:

$$r_i = \{\{R\}_{ij}, \Omega_i, V_i\}, \qquad (3)$$

где $\{R\}_{ij}$ – множество состояний кортежа $r_i$, $V_i$ – множество ограничений целостности, $\Omega_i \subset \Omega$ – множество операций, заданных на $\{\mathrm{R}\}_{\mathrm{ij}}$.

С учетом (2) и (3) логическую модель автономного сценария можно рассматривать как двумерный объект, имеющий реляционную структуру, слоты-кортежи которой описаны с помощью типового набора атрибутов $\{< OBG >, < CON >, < ACT >, < FLAG >\}$ [3,4]:

$$Slot < [OBG],[CON],[ACT],[FLAG] > \qquad (4)$$

Где в качестве базовых представлены следующие типы:

OBG (object-объект)={база данных, файл, папка, сценарий, том, диск};

ACT(action-действие)={записать, копировать, читать, удалить, искать, наблюдать, защищать, ссылаться, выполнять};

CON(condition-условие)={ЕСЛИ <условие> ТО <предикат> };

FLAG(признак выполнения задания)={0; 1}.

Слот также может иметь ключевой атрибут – идентификатор *ID* (index-индекс) и атрибут *PRI* (priority – приоритет), значение которого определяется пользователем и используется для взаимодействия сценариев.

Модель слота автономного сценария показана на рис.1.

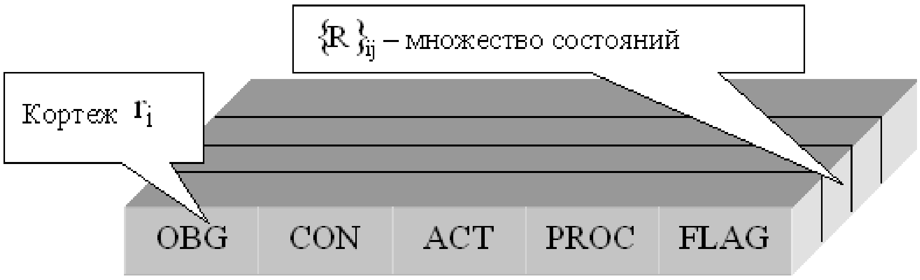


Рис.1. - Модель слота автономного сценария

Состояния-кортежи $\{R\}_{ij}$ могут быть представлены, например, в виде реляционной таблицы, атрибутами которой являются $\{ID, PRI, OBG, CON, ACT, FLAG\}$.

### *2 Классификация автономных сценариев на основе фреймовой структуры*

Так как во многих реальных ситуациях автономные сценарии должны решать возложенные на них задачи в условиях априорной неопределенности, то достижение ими заданной цели возможно лишь на основе применения адаптивного подхода. Суть такого подхода состоит в использовании текущей информации, получаемой в результате выбора конкретных действий, для обоснования выполнения последующих действий. В задачах адаптивного выбора вариантов такой текущей информацией являются значения потерь, получаемые в результате выбора конкретных вариантов. Это позволяет компенсировать недостаток информации и реализовать оптимальную на классе систем стратегию управления [5].

Рассмотрим общую постановку задачи адаптивного выбора вариантов, представленную на рисунке 2.

Смысл подхода состоит в следующем – в каждый из последовательных моментов времени $t_n (n = \overline{1;N})$ необходимо выбирать вариант $\mathrm{v}_n$ из конечного множества возможных вариантов $\mathrm{V}$.

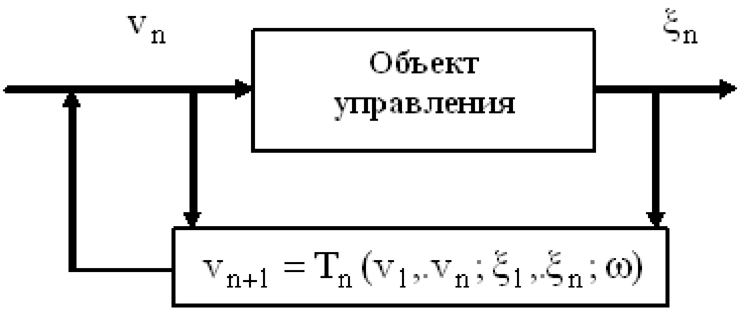


Рис.2. - Схема адаптивного выбора вариантов

Потери системы $\xi_n$ представляют собой функцию элементарного исхода $\omega$ (имеет бинарные значения «штраф» и «отсутствие штрафа») и зависят от выбранного варианта $\mathrm{v}_n$, а также, возможно, от состояния системы. Реализуемая при этом последовательность вариантов $\{\mathrm{v}_n\}$ должна быть такой, чтобы достигалась заданная цель, формулируемая в терминах предельных значений текущих средних потерь.

Выбор очередного варианта $\mathrm{v}_{n+1}$ производится на основе полученной к данному моменту времени совокупности потерь $\xi_1, \xi_2, ..., \xi_n$, которая соответствует реализованной последовательности вариантов $v_1, v_2, ..., v_n$. Это означает, что $\mathrm{v}_{n+1}$ является функцией от $v_1, v_2, ..., v_n$, $\xi_1, \xi_2, ..., \xi_n$ и, возможно, от момента времени $t_n (n = \overline{1;N})$ и элементарного исхода $\omega$. Эту функцию $T_n$ назовем правилом выбора варианта $\mathrm{v}_{n+1}$:

$$v_{n+1} = T_n(v_1, v_2, ..., v_n; \xi_1, \xi_2, ..., \xi_n; \omega),\ n = \overline{1;N}\ , \qquad (5)$$

где $\xi_n$ в зависимости от задачи – либо скаляр, либо вектор.

Функция $T_n$ может быть как детерминированной, так и случайной (рандомизированной). Последовательность $\{\mathrm{T}_n\}$ правил выбора определяет стратегию выбора вариантов или стратегию управления информационным пространством [6,7].

Неопределенность исхода приводит к необходимости использовать более сложные рандомизированные стратегии. Большинство из них реализуют рандомизированные правила выбора следующего вида:

$$p_{n+1} = R_n(v_1, v_2, ..., v_n; p_1, p_2, ..., p_n; \xi_1, \xi_2, ..., \xi_n),\ n = \overline{1;N}, \qquad (6)$$

где $R_n$ – вектор-функция,

$p_n$ – вектор условных вероятностей выбора вариантов $v(1), v(2), ..., v(n)$ в момент времени $t_n$.

Выбору очередного варианта $v_{n+1}$ предшествует вычисление в соответствии с (6), вектора $p_{n+1}$. Вариант $v_{n+1}$ представляет собой случайную дискретную величину, принимающую значения $v(1), v(2), ..., v(N)$ с условными вероятностями $p_{n+1}(1), p_{n+1}(2), ..., p_{n+1}(N)$ при фиксированной предыстории $(v_1, v_2, ..., v_n; \xi_1, \xi_2, ..., \xi_n)$.

Рандомизированные правила выбора (7) включают и так называемые марковские правила, которые можно описать как:

$$p_{n+1} = Q_n(v_n, p_n, \xi_n),\ n = 1,2,... \qquad (7)$$

Рандомизированные стратегии, определяемые последовательностью правил вида (7) относятся к классу рекуррентных алгоритмов адаптивного выбора вариантов. Эти алгоритмы достаточно просто реализуются, поскольку они на каждом шаге $n$ используют минимальную информацию о предыстории процесса.

Применение рандомизированных стратегий позволит решать широкий класс задач адаптивного выбора вариантов, включая задачи с небинарными и с неограниченными потерями $\xi_n$, более того, единообразно формировать алгоритм адаптивного выбора вариантов для всех рассматриваемых задач.

В условиях полной информации о системе оптимальная стратегия всегда принадлежит классу детерминированных стратегий:

$$v_{n+1} = T_n(\omega),\ n = \overline{1;N}. \qquad (8)$$

С помощью детерминированных стратегий может быть решено большинство задач УИР, возникающих в распределенных информационных системах.

Более простая реализация детерминированных стратегий возможна с помощью детерминированных конечных автоматов [8], которые в основном ориентированы на задачи с бинарными потерями, хотя могут применяться и в других случаях. Кроме того, для них характерно обеспечение приемлемого поведения, близость которого к оптимальному возрастает с увеличением глубины памяти автомата. Однако это влечет за собой уменьшение скорости достижения цели и увеличивает сложность, а именно число состояний, соответствующего автомата. Это же свойственно и стохастическим автоматам с постоянной структурой [9], которые реализуют рандомизированные стратегии выбора. Сложным рандомизированным стратегиям (6) и (7) в теории поведения автоматов соответствуют стохастические автоматы с переменной структурой.

Анализ наиболее распространенных стратегий адаптивного выбора вариантов позволил сформировать общий подход к созданию модели поведения автономного сценария в информационном пространстве с использованием конечных автоматов [9]. В соответствии с этим была предложена классификация автономных сценариев на основе фреймовой структуры. Для каждого класса используется бинарная функция потерь и определены стратегия поведения, условия функционирования и тип конечного автомата (см. табл.1).

Таблица № 1

Классы автономных сценариев

| **Характеристика** | **Классы автономных сценариев** | | |
|---|---|---|---|
| | **Класс А** | **Класс В** | **Класс С** |
| Наличие информации о состоянии системы | полная информация | априорная неопределенность | априорная неопределенность |
| Функция потерь | $\xi_n = \{1{,}0\}$ бинарная | $\xi_n = \{1{,}0\}$ бинарная | $\xi_n = \{1{,}0\}$ бинарная |
| Стратегия поведения | $v_{n+1} = T_n(\omega)$ | $v_{n+1} = T_n(v_1,...,v_n;\xi_1, ...,\xi_n;\omega)$ | $p_{n+1} = R_n(v_1,..., v_n; p_1,..., p_n; \xi_1,..., \xi_n)$ |
| Модель поведения | автоматная модель поведения типа «автомат-строка» | автоматная модель поведения | автоматная модель поведения |
| Тип автомата | детерминированный, стохастический с | детерминированный, стохастичесиий с | детерминированный, стохастическиий с |

| | | | |
|---|---|---|---|
| | постоянной структурой | переменной структурой | переменной структурой |
| Тип сценария | рефлексивный, автономный | автономный | интеллектуальный |

Исследование выделенных классов позволяет формализовать их поведение в терминах теории конечных автоматов.

### *3 Модель поведения автономного сценария в терминах теории конечных автоматов*

Конечный автомат рассматривается как некоторый объект [7 – 9], способный в каждый момент времени $t = 1,2,\ldots,N$ воспринимать конечное число сигналов $s \in (s_1, s_2, \ldots s_N)$ и изменять в зависимости от них свое внутреннее состояние. Автомат может производить конечное число действий $f \in (f_1, f_2, \ldots, f_n)$, выбор действия определяется внутренним состоянием автомата. Автомат имеет конечное число $m$ внутренних состояний $\varphi \in (\varphi_1, \varphi_2, \ldots, \varphi_m)$, которое называется емкостью памяти автомата.

Предполагается, что автомат находится в некоторой среде и что действия f автомата вызывают ответные реакции s среды $E$. Эти реакции, в свою очередь, являются для автомата входными сигналами, которые он использует для принятия решения о дальнейших действиях (рис. 3).

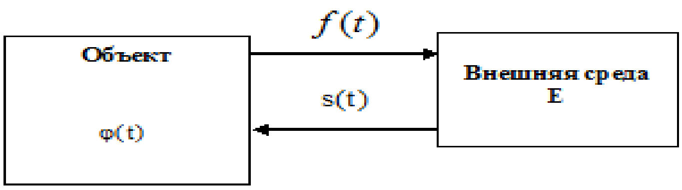


Рис.3 - Схема взаимодействия объекта с внешней средой

Рассмотрим простейший случай, когда все возможные реакции среды $s \in (s_1, s_2, \ldots s_N)$ воспринимаются автоматом как относящиеся к одному из двух классов – классу благоприятных реакций (выигрыш, $s = 0$) и классу реакций неблагоприятных (проигрыш, $s = 1$). Внутри каждого из этих классов реакции среды являются для автоматов неразличимыми. Целесообразность поведения автомата в некоторой среде заключается в увеличении числа благоприятных реакций и уменьшении числа реакций неблагоприятных.

Ограничим наше исследование рассмотрением детерминированных и стохастических автоматов.

Автомат задается уравнением $f(t) = F(\varphi(t))$, показывающим зависимость действия $\mathrm{f(t)}$ автомата в момент времени $t$ от его состояния $\varphi(t)$, и стохастической матрицей $\|a_{ij}(s)\|, i, j = 1,2,...,m$. При этом $a_{ij}(s)$ равно вероятности перехода состояния $\varphi(t) = \varphi_i$ в состояние $\varphi(t+1) = \varphi_j$ под воздействием входа $\mathrm{s(t+1)}$. Для детерминированных автоматов матрицы $\|a_{ij}(s)\|$ состоят из нулей и единиц. Так как рассматриваются автоматы, воспринимающие лишь два сигнала $s = 0$ и $s = 1$, то достаточно задать две такие матрицы $\|a_{ij}(0)\|$ и $\|a_{ij}(1)\|$. Таким образом, детерминированный автомат $\mathrm{U}$ может быть задан каноническими уравнениями:

$$\varphi(t+1) = \Phi(\varphi(t), s(t+1)), \tag{9}$$

$$f(t) = F(\varphi(t)). \tag{10}$$

Уравнение (10) описывает зависимость действий автомата от его состояний, а уравнение (9) – изменения его состояний под воздействием входной переменной $s(t)$. Каждая строка матрицы состояний детерминированного автомата при любом фиксированном значении $\mathrm{s}$ содержит один элемент, равный 1, а остальные элементы равны 0. Смена состояний детерминированного автомата осуществляются в соответствии с правилом: если в момент $\mathrm{t}$ автомат находится в состоянии $\varphi_i$, то в момент $t+1$ он перейдет в такое состояние $\varphi_j$, для которого $a_{ij}(s(t+1)) = 1$.

Стохастический автомат также имеет конечное число состояний $\varphi \in (\varphi_1, \varphi_2, ..., \varphi_m)$ и конечное число действий $f \in (f_1, f_2, ..., f_n)$. Действия стохастического автомата однозначно определяются его состоянием: $f(t) = F(\varphi(t))$, а матрицы состояний $\|a_{ij}(s)\|$, $s \in \{0, 1\}$ являются стохастическими. При этом $\mathrm{a_{ij}(s)}$ имеет смысл вероятности перехода из *i*-го состояния в *j*-е при заданном значении входной переменной $\mathrm{s}$. Пусть в момент $t$ автомат находится в состоянии $\varphi_i$, $i = 1,2,...m$, которому соответствует действие $f_\alpha = F(\varphi_i)$. Тогда вероятность $p_{ij}$ перехода автомата из состояния $\varphi_i$ в состояние $\varphi_j$ определяется формулой:

$$p_{ij} = p_\alpha a_{ij}(1) + q_\alpha a_{ij}(0), \qquad i, j = 1,2,...,m. \tag{11}$$

Очевидно, что матрица $P = \|p_{ij}\|$ является стохастической.

С учетом общей модели поведения конечного автомата (10) и модели слота (2) и (3) представим автономный сценарий в терминах модели конечного автомата, тогда n слотов $r_i$ соответствуют n типам действий $f_i$:

$$r_i = \{\{R\}_{ij}, \Omega_i, V_i\} \approx f_i (i = \overline{1, n}) \quad (12)$$

Каждый слот $\mathrm{r_i}$ по аналогии с автоматом для i-го действия обладает конечным числом внутренних состояний $\{\mathrm{R}\}_{\mathrm{ij}}$:

$$\{\mathrm{R}\}_{\mathrm{ij}} \approx \varphi_{ij} (i = \overline{1, n}; j = \overline{1, m}) \quad (13)$$

где $n$ – количество слотов-заданий, $m$ – количество состояний $i$-го кортежа-задания.

Тогда логическая модель автономного сценария примет вид, представленный на рис.4.

При выполнении условия, заданного форматом атрибута [CON], для объекта [OBG] выполняется встроенная процедура [PROC] или действие, определенное спецификацией [ACT]. На каждое действие среда отвечает сигналом $\mathrm{s(t)}$, значение которого {1,0} отображается в поле [FLAG]. Накопленные в течение определенного периода результаты выполнения заданий, сформулированных в слотах-кортежах, могут быть использованы для моделирования адаптивного поведения сценариев. Ориентация на реакцию среды, в которой функционирует автономный сценарий, позволяет ему достичь поставленной цели. Конфликтные ситуации между сценариями разрешаются на основе приоритетов, заданных в поле [PRI].

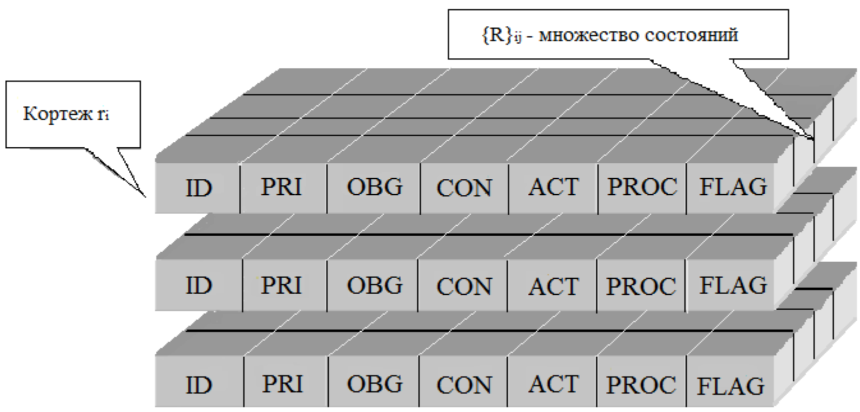


Рис.4. - Двумерная логическая модель автономного сценария

Таким образом, модель автономного сценария представляет собой сложную логическую структуру и как обязательный атрибут должна содержать имя фрейма-сценария, который включает слоты-задания $r_i \ (i=\overline{1,n})$ (см. рис. 5).

| Имя автономного сценария ____________ | | | | | | |
|---|---|---|---|---|---|---|
| ID | PRI | OBG | CON | ACT | PROC | FLAG |
| ID | PRI | OBG | CON | ACT | PROC | FLAG |
| **** | | | | | | |
| ID | PRI | OBG | CON | ACT | PROC | FLAG |

Рис.5. - Структура фрейма автономного сценария

Поведение автономного сценария определяется матрицей переходов, которая имеет вид:

| | | | |
|---|---|---|---|
| $r_1$ | $r_2$ | $\cdots$ | $r_n$ |
| $k_1$ | $k_2$ | $\cdots$ | $k_n$ |

Алгоритм автономного сценария состоит в следующем: задание, сформулированное в первом слоте-кортеже, выполняется $k_1$ раз, после чего управление передается на второй слот-кортеж. Задача считается полностью выполненной тогда, когда действие $r_n$ слота-кортежа выполнится $k_n$ раз.

В качестве примера рассмотрим автономный сценарий класса А, состоящий из одного слота и имеющий три состояния [10]. Сценарий активируется при наступлении конкретной даты (21 сентября 2011 года) и времени (17.00 часов системного времени) и копирует содержимое папки d:\arhiv\### на диск е:\ в одноименную папку (см.рис.6). Сценарий трижды выполняется в системе 21, 22 и 23 сентября. Результаты копирования заносятся либо в соответствующую таблицу БД, либо в текстовый файл (журнал).

| **OBG:**<br>**D:\arhiv\###** | **CON:**<br>**Date Time=17:00** | **ACT:**<br>**Copy from D:\arhiv\ to E:\arhiv** | **FLAG:** |
|---|---|---|---|
| D:\arhiv\### | 21.09.11 17:00 | Copy from D:\arhiv to E:\arhiv | 1 |
| D:\arhiv\### | 22.09.11 17:00 | Copy from D:\arhiv to E:\arhiv | 1 |
| D:\arhiv\### | 23.09.11 17:00 | Copy from D:\arhiv to E:\arhiv | 1 |

Рис.6.- Структура автономного сценария класса А

Если несколько изменить задание этого сценария, то есть не указать дату, а количество состояний задать равным 1, то копирование папки будет производиться ежедневно в 17.00 часов.

**Выводы**

Одним из перспективных направлений автоматизации процесса управления информационными ресурсами вычислительной системы является технология автономных сценариев, обеспечивающая решение широкого класса задач, таких как интеграция гетерогенных информационных структур и распределенных баз данных, мониторинг и автономный аудит информационных ресурсов. В статье предложена логическая модель автономных сценариев на основе фреймов, позволяющая формализовать как детерминированные, так и стохастические сценарии.

Для моделирования поведения автономных сценариев при взаимодействии с информационной средой предложено использовать аппарат конечных автоматов, позволяющий описать широкий спектр алгоритмов поведения, в том числе адаптивные и интеллектуальные. Разработанные модели повышают эффективность проектирования и сопровождения систем управления информационными ресурсами, являются основой для создания инструментального программного средства автоматизированного генерирования автономных сценариев.

**Список литературы:**


1. Minsky, Marvin. A framework for representing knowledge. [Electronic resource] // MIT AI Laboratory Memo 306. June, 1974. -. Режим доступа: http://web.media.mit.edu/~minsky/papers/Frames/frames.html (доступ свободный) – Загл. с экрана. – Яз. англ.
2. Chaib-draa, B., Moulin, B., Mandiau, R. & Millot, P. Chapter 1 - Trends in Distributed Artificial Intelligence, Foundations of Distributed Artificial Intelligence [Text] // G. M. P. O'Hare and N. R. Jennings (eds.), John Wiley & Sonsmc, 1996. – p. 3-55.
3. Аксенов К.А. Коалиционная модель мультиагентного процесса преобразования ресурсов [Электронный ресурс] // «Инженерный вестник Дона», 2012, №4 (часть 2). – Режим доступа: http://ivdon.ru/magazine/issue/106 (доступ свободный) – Загл. с экрана. – Яз. рус.
4. Филатов В.А. Модель поведения автономного агента на основе

теории автоматов [Текст] // Вестник Херсонского государственного технического университета.- Херсон: ХГТУ, 2004. - № 1 (19) - с.108 - 111.

5. Трахтенброт Б.А., Барздинь Я.М. Конечные автоматы (поведение и синтез) [Текст] // Борис Трахтенброт, Ян Барздинь - М.: Мир, 1970. - с.400

6. Кудрявцев В.Б., Введение в теорию автоматов [Текст] // В.Б. Кудрявцев, С.В. Алешин, А.С. Подколзин - М.: Наука, 1985. - с.319

7. Назин А.В., Позняк А.С. Адаптивный выбор вариантов: рекуррентные алгоритмы [Текст] //А.В. Назин, С.В. Алешин - М.: Наука, Глав. ред. физико-математической лит-ры, 1986. - с.288, ил., 21 см.

8. Цетлин М. Л. Исследования по теории автоматов и моделированию биологических систем [Текст] // М.Л. Цетлин – М.: "Наука", 1969 – с.316

9. Филатов В.А., Козырь О.Ф. Мультиагентный подход к идентификации пользователей в системе дистанционного образования [Текст] // Сборник трудов региональной научной конференции,- Старый Оскол ООО "ТНТ", 2005. – т.1- с. 284-290.

10. Ананьев А.С., Бутенко Д.В., Попов К.В. Интеллектуальные технологии проектирования информационных систем. Методика проектирования программных продуктов в условиях наличия прототипа [Электронный ресурс] // «Инженерный вестник Дона», 2012, №2. – Режим доступа: http://ivdon.ru/magazine/issue/103 (доступ свободный) – Загл. с экрана. – Яз. рус.