

## **РАСПРЕДЕЛЕННЫЕ МЕХАНИЗМЫ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ В ЗАДАЧАХ МОНИТОРИНГА СОСТОЯНИЯ ОБОРУДОВАНИЯ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ**

В.В. Лежебоков

Волгоградский Государственный Технический Университет (ВолгГТУ)  
Тел. +41(22) 7330703, e-mail: valery.lezhebokov@gmail.com

**Аннотация.** В работе описывается подход к обработке массивов данных на основе иерархических структур с применением положений теории многоагентных систем. Определены механизмы управления процессами предварительной обработки данных в многопоточной среде выполнения. Предложена гибридная модель системы предварительной обработки данных.

**Ключевые понятия:** система управления, обработка данных, многоагентная система.

### **Введение**

Одной из характерных тенденций развития научных исследований является появление и использование информационно-технических систем большой сложности. Важным классом данных систем являются автоматизированные системы мониторинга, обеспечивающие сбор и предварительную обработку данных отражающих определенные характеристики состояния объекта наблюдения с последующей их передачей конечному множеству заинтересованных клиентов. В данном контексте одной из основных задач является построение модели высокопроизводительной системы, обеспечивающей автоматизацию процессов предварительной обработки данных, что подразумевает разработку модели соответствующей автоматизированной системы управления данными процессами.

Необходимость обработки больших объемов информации в ограниченное время обуславливает создание распределенных систем осуществляющих препроцессинг данных в масштабах реального времени. В данной статье предложена гибридная модель системы автоматизации предварительной обработки данных, в основе которой использован классический подход к созданию информационных систем, наряду со структурными элементами многоагентной системы, осуществляющей предварительную обработку данных.

Зачастую процесс предварительной обработки данных представляет собой вычисление значений параметров «верхнего уровня» на основании массивов

данных получаемых от технических устройств. Тогда автоматизация предварительной обработки данных может быть достигнута за счет построения многоагентной системы (МАС), элементы которой осуществляют исчисление значений параметров на основе иерархических структур. В то же время, в целях централизованного управления препроцессингом данных необходимо предусмотреть возможность управления процессами обработки на основе системы событий.

## 1. Автоматизированная система управления информационными процессами

На верхнем уровне управление процессами предварительной обработки данных сводится к обеспечению достижения совокупностью данных процессов заданного поведения, что обуславливает необходимость разработки модели автоматизированной системы управления информационными процессами (АСУИП). В рамках предлагаемой модели функция управления достигается за счет инкапсуляции алгоритмов внутренних информационных преобразований ИП в модуле автоматизации информационного процесса (МАИП), управляемом на основе системы событий.

Будем рассматривать автоматизированную систему в виде совокупности основных подсистем обеспечивающих выполнение определенных задач в соответствии с функциональной структурой и множества модулей автоматизации ИП:

$$S_N = \{s^m, s^c, s^d, s^s, H\}, \quad (1)$$

где:  $s^m$  – подсистема обработки сообщений;  $s^c$  – подсистема управления модулями автоматизации ИП;  $s^d$  – подсистема распределения нагрузки;  $s^s$  – подсистема доступа к конфигурации МАИП;  $H$  – множество модулей автоматизации ИП.

Подсистему обработки сообщений определим в виде:

$$s^m = (X^m, y^m, \Phi, M, E), \quad (2)$$

где:  $X^m$  – множество входных информационных потоков;

$y^m$  – выходной информационный поток;

$\Phi$  – множество функций отображения множества сообщений на множестве событий;

$M$  – множество сообщений;

$E$  – множество событий.

В основе информационных преобразований на верхнем уровне для подсистемы  $s^m$  лежит отображение вида:

$$\varphi_i \in \Phi: M_i \rightarrow E_i; \quad (3)$$

подразумевающее анализ входного информационного потока содержащего элементы множества  $M_i \subseteq M$  с последующим отображением на множестве событий  $E_i$ . Подсистему управления информационными процессами представим как:

$$s^c = (X^c, Y^c, \Omega, E, A), \quad (4)$$

где:  $X^c$  и  $Y^c$  – множество входных и выходных информационных потоков;  
 $\Omega$  – модель выработки управлений;  
 $E$  – множество событий;  
 $A$  – множество управляющих воздействий.

Модель выработки управлений позволяет перейти от теоретико-множественного описания входной для системы  $s^c$  информации к выработке воздействий направленных изменение информационного состояния объекта управления, посредством сюръективного отображения вида:

$$\Omega: E \rightarrow A; \quad (5)$$

Конечное множество определенных в рамках данной модели управляющих воздействий имеет вид:

$$A = \{ a^s, a^i, a^r, a^c \} \quad (6)$$

где:  $a^s$  – синтез нового модуля автоматизации ИП;  
 $a^i$  – инициализация модуля автоматизации ИП;  
 $a^r$  – выполнение алгоритмов модуля автоматизации ИП;  
 $a^c$  – завершение выполнения алгоритмов модуля автоматизации ИП.

Основной задачей подсистемы доступа к конфигурации МАИП ( $s^s$ ) является представление описания свойств и параметров модуля автоматизации включающего политику жизненного цикла  $p^m$ . Под **политикой жизненного цикла модуля** понимается совокупность продукций, определяющих управляющие воздействия подлежащие выработке в ответ на поступившее событие, в целях достижения желаемого поведения.

Представим модуль автоматизации ИП в следующем виде:

$$h_i = (X_i^h, Y_i^h, t^m, \psi_i, \theta_i, w_i); h_i \in H \quad (7)$$

где:  $X_i^h$  и  $Y_i^h$  – множество входных и выходных информационных потоков;  
 $t^m$  – тип модуля автоматизации ИП;  
 $\psi_i$  – комплексный алгоритм информационных преобразований ИП;  
 $\theta_i$  – функция оценки состояния ИП;  
 $w_i$  – текущее состояние МАИП,  $w_i \in W$ .

Текущее состояние ИП может принимать одно из следующих предопределенных значений: модуль синтезирован ( $w^s$ ), модуль инициализирован ( $w^i$ ), состояние ожидания ( $w^a$ ), состояние выполнения ( $w^e$ ) и выполнение завершено ( $w^c$ ). Множество переходов между данными состояниями определяет жизненный цикл модуля автоматизации ИП.

Концептуальное представление внутренней структуры АСУИП на уровне взаимодействия подсистем представлено на рисунке 1:

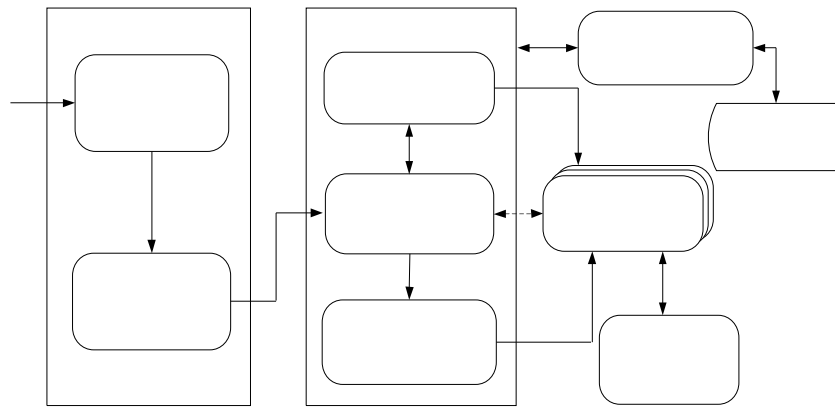


Рис. 1. Архитектура автоматизированной системы управления ИП

Множество сообщений поступающих из внешних по отношению к системе источников, отображается на множестве событий посредством подсистем синтеза событий. Подсистемой управления МАИП осуществляется выполнение периодических запросов к подсистеме буферизации на предмет появления новых событий. В случае наличия непустого множества событий, подсистемой анализа выполняется определение управляющих воздействий подлежащих выработке. Подсистема управления модулями автоматизации взаимодействует с подсистемой доступа к конфигурации МАИП, в рамках процессов анализа событий и синтеза новых модулей. Подсистемой выполнения алгоритмов МАИП обеспечивается выработка управляющих воздействий. В контексте рассматриваемой подсистемы под *выработкой управления* понимается выполнение алгоритмов модуля автоматизации ИП направленных на:

- инициализацию;
- выполнение алгоритмов информационных преобразований автоматизируемого ИП;
- завершение функционирования модуля.

Каждое из вырабатываемых в рамках отдельного потока управлений потенциально ведет к изменению текущего состояния модуля автоматизации. Тогда, существует вероятность возникновения ситуации, при которой выработка некоторого управления  $a_i$  является недопустимым, в силу того, что состояние модуля было изменено в результате выработки в рамках параллельного потока другого управления  $a_j$ . Следовательно, перед непосредственной выработкой управления подсистема выполнения алгоритмов должна оценить допустимость выработки посредством анализа текущего состояния модуля. Процессы выработки управлений в многопоточной среде выполнения могут быть смоделированы с помощью представленной раскрашенной сети Петри (см. Рис. 2), начальная маркировка которой, определяет множество управляющих воздействий подлежащих выработке, а функции  $x(t)$ ,  $y(t)$ ,  $z(t)$  имитируют временные задержки при выработке управлений.

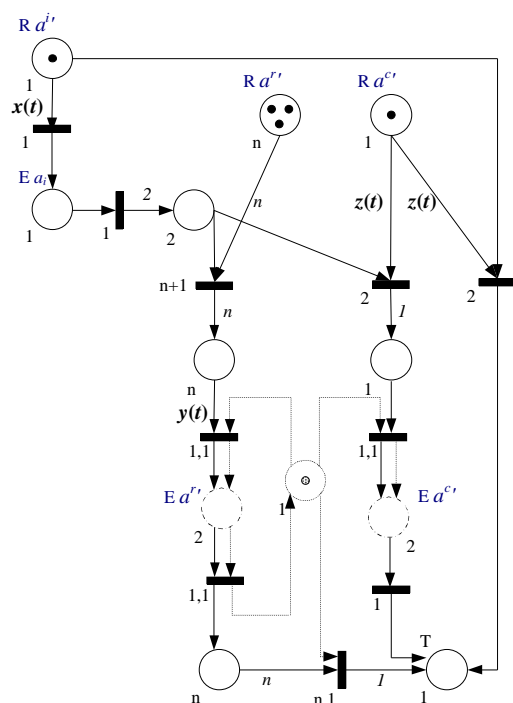


Рис. 2. Имитационное моделирование процессов выработки управляющих воздействий

На практике оценка допустимости выработки управлений может быть осуществлена посредством разработки соответствующих механизмов с использованием основных положений теории автоматов и стандартных средств синхронизации потоков.

Данная модель ориентирована на автоматизацию процессов имеющих итерационный характер в реальном времени и позволяет управлять ими на основе системы событий.

## 2. Иерархические структуры исчисления значений параметров

В любой технической системе некоторое устройство характеризуется совокупностью параметров, значения которых определяет его текущее состояние. Множество параметров, значения которых, в процессе мониторинга, считываются непосредственно с технического устройства являются параметрами нижнего уровня. В общем случае в виду высокой степени сложности современных технических устройств (а следовательно большого количества параметров ассоциированных с данными устройствами), восприятие и анализ конечным пользователем больших массивов получаемых данных представляет собой нетривиальную задачу. С другой стороны, значения части параметров нижнего уровня сами по себе не являются информативными и не представляют большой ценности, в то время как среднее (максимальное, минимальное, и т.д.) значение, полученное в результате выполнения некоторых алгоритмов над выборкой значений, может оказаться намного более полезным для конечного пользователя. Полученное значение является значением

параметра верхнего уровня первой степени. В свою очередь данное значение может быть использовано в вычислениях значения параметра верхнего уровня второй степени, и т.д.

Учитывая вышесказанное наиболее целесообразным будет являться представление зависимостей между параметрами в виде иерархической структуры следующего вида:

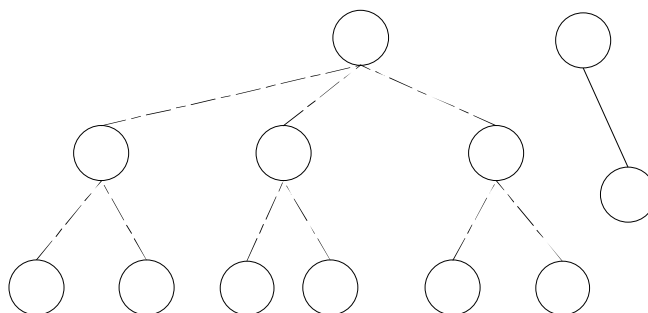


Рис. 3. Представление зависимостей параметров в виде иерархической структуры

Формально данная структура представима в виде кортежа  $H(P, \Xi)$ , где  $P$  – множество параметров;  $\Xi$  – множество правил исчисления значения параметра. Необходимо отметить, что для любого параметра в иерархии допустимо определение не более одного правила исчисления в рамках заданной иерархии:  $\forall p_i \in P \mid \exists (p^i \times p_j^{i-1}) - \{0, 1\}$ . Каждый параметр описывает некоторую характеристику  $v$  устройства  $d$ :  $p(d, v)$ .

Иерархии параметров позволяют организовать вычисления на основе общего механизма, абстрагированного от специфики применяемых в процессе исчисления алгоритмов. В то же время с ростом количества технических устройств, размерность множества иерархий параметров возрастает согласно закону арифметической прогрессии:

$$a_n = a_1 + (n - 1)d \mid \forall n \geq 1,$$

где:  $n$  – количество устройств в системе;

$d$  – среднее количество параметров определенных для технического устройства.

При достаточно больших размерах системы описание иерархий параметров для каждого технического устройства будет представлять собой довольно ресурсоемкую задачу. Кроме того, создание большого числа иерархических структур, также, скажется на производительности системы предварительной обработки данных в целом. В целях сокращения общего количества иерархических структур, рассмотрим подход предполагающий определение зависимостей на уровне мета-параметров определенных для каждого класса технических устройств  $C^d$ .

В общем случае состав оборудования любой сложной технической системы включает в себя совокупность множеств однотипных ТУ. Общие характеристики и состав параметров устройств принадлежащих одному множеству может быть заданы на уровне класса ТУ посредством множества

мета-параметров  $P^?$ . Тогда вместо описания двух идентичных иерархических структур для исчисления значений параметров  $p_i(d_j, v)$  и  $p_n(d_m, v)$  для двух однотипных устройств, достаточно определить зависимости на уровне класса ТУ, используя элементы  $p^?(c^d, v)$ ;  $c^d \in C^d$ ,  $c^d = \{d_i\}$ ,  $i \geq 1$ ;  $p^? \in P^?$ . В то же время иногда возникает необходимость в определении специфических правил исчисления и иерархии для некоторого параметра ТУ, отличных от определенных на уровне мета-параметров. Следовательно, в целях вычисления значения некоторого параметра  $p_i$  необходимо определить алгоритм синтеза иерархии для данного параметра на основании иерархических структур уровней параметров и мета-параметров. Логические и структурные взаимосвязи параметров данных уровней представлены на Рис. 4.

Предлагаемый алгоритм синтеза основан на использовании иерархии мета-параметров в целях построения иерархий параметров путем проекции общих зависимостей и правил, определенных для уровня классов технических устройств на частные случаи исчисления значений параметров. Более формально – на верхнем уровне упрощенный алгоритм синтеза иерархии для заданного параметра  $p^i$  состоит из следующих основных шагов:

1. Поиск иерархии для заданного параметра на уровне  $L_p$ .
2. Выявление конечных узлов иерархии определенных для  $p^i \Rightarrow \{p_j^k\}$ .
3. Синтез иерархии для каждого параметра  $p_j^i$ .
  - 3.1. Поиск иерархии для мета-параметра  $p_j^?(c^d, v) \mid d_j \in c^d$  на уровне  $L_p$ !
  - 3.2. Синтез одного уровня иерархии для параметра  $p_j^k$  для технического устройства  $d_j$ .  $p_j^k \Rightarrow \{p_j^{k-1}\}$
  - 3.3. Повтор пунктов 1-3 для каждого из параметров множества  $\{p_j^{k-1}\}$ .

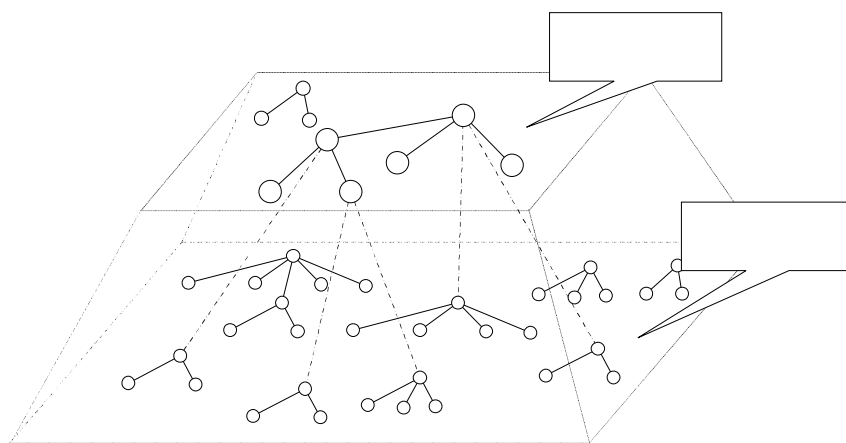


Рис. 4. Взаимосвязи параметров различных уровней

На основании полученных иерархий производятся вычисления значений параметров результаты, которых передаются заинтересованным клиентам. В рамках предлагаемой модели построение иерархии и исчисление значений параметров осуществляется на основе многоагентной системы, как теории с наиболее гибким и эффективным в настоящий момент подходом к созданию распределенных систем.

### 3. Применение многоагентных технологий к задаче исчисления значений параметров

Использование многоагентной системы в качестве базиса при построении систем предварительной обработки данных имеет ряд преимуществ, среди которых необходимо отметить:

- масштабируемость системы;
- возможность организации распределенных вычислений;
- возможность реализации интеллектуальных методов обработки информации.

Предлагаемая модель многоагентной системы (МАС) предполагает описание логики функционирования агентов, и их взаимодействия в процессе исчисления значений параметров.

Учитывая модель представления зависимостей параметров определим два класса агентов взаимодействующих между собой в целях осуществления предварительной обработки данных:  $A^P$  – класс агентов иерархии параметров (АИП) и  $A^{mh}$  – класс агентов иерархии мета-параметров (АИМП). Кроме того, зачастую возникает необходимость в использовании «виртуальных параметров», отличительной особенностью которых является отсутствие их декларативных описаний во внешних источниках информации (БД), что также подразумевает отсутствие явного определения иерархий исчисления значений параметров данного типа. Синтез «виртуальных» иерархий является основной задачей агентов третьего класса  $A^{vh}$  – агентов иерархий виртуальных параметров (АИВП).

Взаимодействие агентов данных классов отображено на Рис. 5.

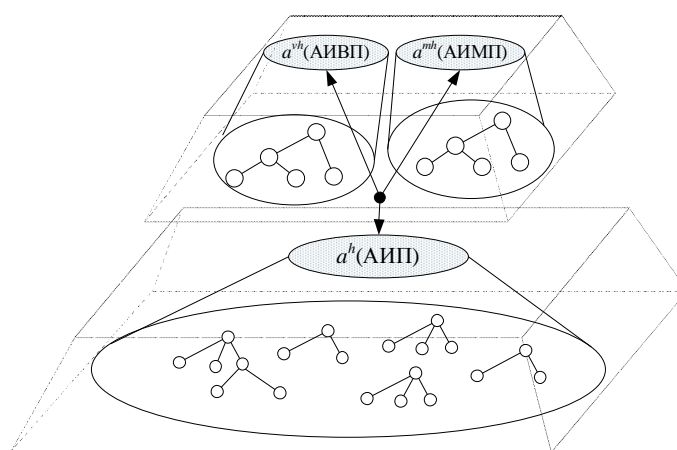


Рис. 5. Взаимодействие агентов различных уровней

Согласно классификации, приведенной в Тарасовым<sup>1</sup>, агенты перечисленных типов могут быть отнесены к классу реактивных в силу отсутствия у них представления о картине мира, а функционирование агента сводится к его реакции на внешние воздействия. В комплексе рассмотренные агенты реализуют алгоритм синтеза иерархий для множества параметров  $P$  рассмотренный в предыдущем параграфе.



Располагая знаниями о виде иерархий параметров агент предварительной обработки (АПО) данных способен осуществлять исчисление значений параметров. Алгоритм функционирования агента данного класса зависит от состояния внешнего по отношению к агенту мира, а именно от наличия всех необходимых данных (значений параметров) в момент выполнения предварительной обработки. Также необходимо отметить, что зачастую существует необходимость в осуществлении процесса предварительной обработки данных согласно некоторой стратегии, определяемой пользователем системы. Создание и использование различных стратегий подразумевает реализацию различных систем целей, к которым стремится агент.

Принимая во внимание вышеперечисленные факторы, построим формализованное описание агента предварительной обработки  $a^p \in A^p$  на основе делиберативной модели BDI архитектуры агента<sup>2</sup>.

Данный подход опирается на использование СТЛ логики вычислительных деревьев, в которой каждому моменту времени приписывается некоторая ситуация, характеризующаяся тремя множествами достижимых миров: миров убеждений  $W^b$ , желаний  $W^d$  и намерений  $W^i$ . В целях определения элементов перечисленных множеств построим визуализацию альтернативных ветвлений во времени в виде дерева следующего вида:

$$\Lambda = (S^\lambda, \Xi^\lambda, Y^\lambda, R^\lambda), \quad (8)$$

где:  $S^\lambda$  – непустое множество состояний в мире агента, состоящее из элементов двух непересекающихся конечных множеств: множества состояний принятия решений  $S^d$  и множества состояний определяемых событиями окружающего мира  $S^c$ , т.е.  $S^\lambda: S^\lambda = S^d \cup S^c, S^d \cap S^c = \emptyset$ ;

$\Xi^\lambda$  – непустое множество действий предпринимаемых агентом;

$Y^\lambda$  – множество исходов событий окружающего мира;

$R^\lambda$  – отношение достижимости состояний, ставящее в соответствие каждому переходу между состояниями некоторый элемент объединения непересекающихся множеств действий и исходов событий, т.е.  $R^\lambda: S^\lambda \times S^\lambda \rightarrow \Xi^\lambda \cup Y^\lambda \mid \Xi^\lambda \cap Y^\lambda = \emptyset$ .

Определим множества действий и возможных исходов событий для процесса исчисления значения параметра на основе иерархической структуры. Совокупность действий реализуемых агентом ограничена следующим множеством:

$$\Xi^\lambda = \{\xi^v, \xi^h, \xi^a, \xi^c, \xi^w, \xi^x\}, \quad (9)$$

где:  $\xi^v$  – извлечение данных из буфера:  $\xi^v(p^i)$  – для заданного параметра,  $\xi^v(P)$  – для множества параметров,  $\xi^v(h(p^i))$  – для всех параметров заданной иерархической структуры;

$\xi^h$  – получение иерархической структуры  $h$  для заданного параметра,  $\xi^v(p^i)$ ;

$\xi^a$  – проверка наличия значений:  $\xi^a(p^i)$  – для заданного параметра,  $\xi^a(P)$  – для множества параметров;

$\xi^c$  – вычисление значения параметра на основе заданной иерархической структуры,  $\xi^c(p^i, h(p^i))$ ;

$\xi^w$  – сохранение множества параметров для заданной иерархической структуры в буфере данных,  $\xi^w(h(p^i))$ ;

$\xi^x$  – экспорт значения параметра заинтересованным клиентам,  $\xi^x(p^i)$ .

Множество типов событий окружающего мира состоит из следующих элементов:

$$\Delta = \{ \delta^v, \delta^h \}, \quad (10)$$

где:  $\delta^v$  – тип события, характеризующий извлечение буфером данных информации для заданного множества параметров некоторого уровня;

$\delta^h$  – тип события, характеризующий поиск (синтез) иерархической структуры  $h$  для заданного параметра.

Тогда множество исходов событий окружающего мира представим в виде совокупности элементов двух подмножеств  $Y^\lambda = Y^v \cup Y^h$ , в соответствии с типами выделенных событий. Для конечного множества  $Y^v$  исходов события  $\delta^v$ , получим:

$$Y^v = \{ v_0^v, v_1^v, v_2^v, \dots, v_n^v \}, n = |P^i|, \quad (11)$$

элементы которого, описывают результаты извлечения значений параметров из буфера данных. Индекс элемента множества соответствует количеству параметров, для которых значения не были найдены, а сам элемент множества представим в виде структуры содержащей идентификаторы данных параметров.

Множество исходов события  $\delta^h$  содержит два элемента соответствующих двум взаимоисключающим ситуациям:

$$Y^h = \{ v_0^h, v_1^h \}, \quad (12)$$

где:  $v_0^h$  – иерархия для заданного параметра не найдена (не может быть синтезирована);

$v_1^h$  – иерархия для заданного параметра найдена (синтезирована).

Учитывая приведенные формализмы описания базовых понятий используемых при описании возможных миров в агентных системах, построим дерево принятия решений для процесса исчисления значения параметра на основе иерархической структуры:

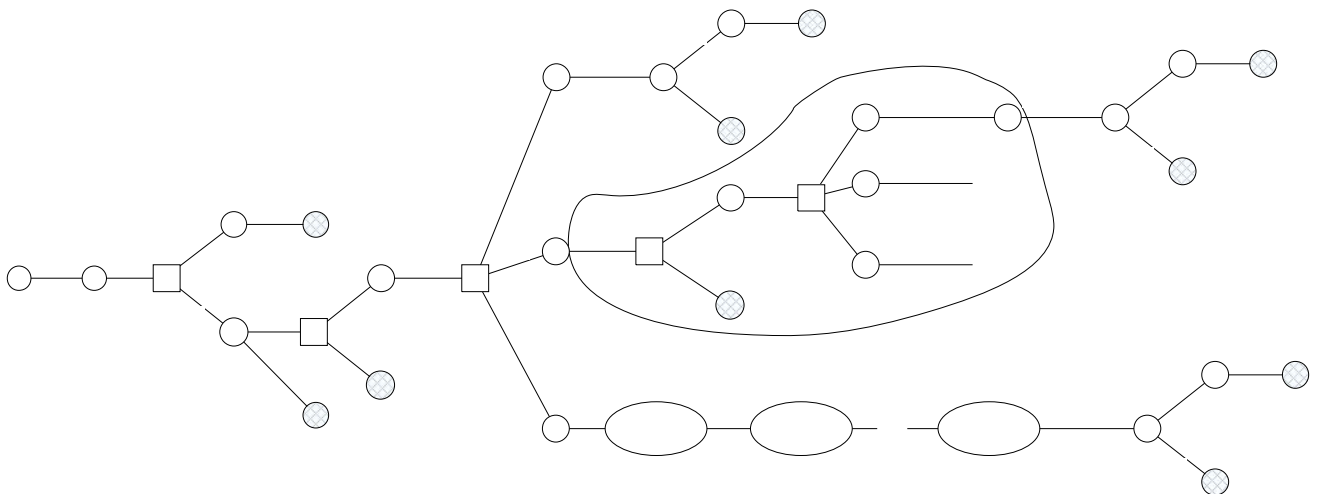


Рис. 6. Дерево принятия решений агента предварительной обработки данных

Согласно алгоритму, описанному Рао и Джорджефом<sup>2</sup>, построим множество миров описывающих убеждения (beliefs) агента, основываясь на построенном дереве принятия решений. Для этого необходимо:

1. исключить из дерева принятия решений все вершины соответствующие состояниям подмножества  $S^c$  (см. (8));
2. для каждого исхода события  $v \in Y$  определенного для исключенной вершины  $s^c$  создается отдельный мир  $w_i^b$ , в котором данный исход события  $v$  истинен;
3. из созданного мира  $w_i^b$  удаляются все альтернативные (по отношению к  $v$ ) ответвления определенные для вершины  $s^c$ ;
4. в месте разрыва между двумя смежными вершинами устанавливается отношение достижимости;
5. на получившуюся структуру проецируется система целей агента, путем присвоения элементам заданного подмножества состояний различных весов, задаваемых целевыми функциями  $g(s) \in G$ .

В результате выполнения описанных преобразований получим множество возможных миров определяющие убеждения  $W^b$  агента, которые, согласно BDI модели интеллектуального агента, характеризуют его представления об окружающем мире. К примеру, приведенные на Рис. 7 элементы множества  $W^b$  описывают возможные состояния окружающего мира, в которых:

- a) значение параметра  $p^i$  уже вычислено и его необходимо экспортировать;
- b) значение параметра  $p^i$  не вычислено, и нет возможности его вычислить, т.к. не существует соответствующей иерархии;
- c) значение параметра  $p^i$  не вычислено, однако существует иерархия параметров позволяющая произвести необходимые вычисления, причем все необходимые информационные ресурсы (значения параметров  $P^{i-1}$ ) присутствуют.

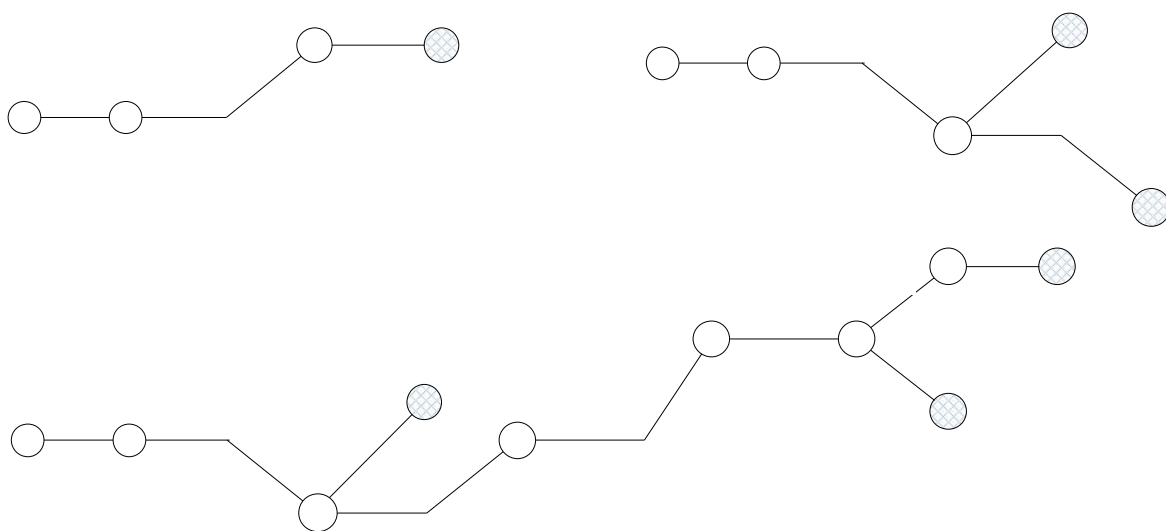


Рис. 7. Пример элементов множества убеждений агента предварительной обработки данных.

Далее необходимо определить множество предпочтений (desires) агента, путем отсечения путей в графе для которых значение целевой функции  $g_i(s) \leq 0$ , тем самым устраняя явно нерациональные решения, которые потенциально могли бы быть выбраны агентом. К примеру, агент предварительной обработки данных знает, что в условиях отсутствия значения параметра  $p^i$  существует вариант развития событий, при котором агент может не предпринимать каких-либо активных действий, направленных на исчисление значения  $p^i$ . Вместо этого агент может сообщить клиенту о невозможности (или нежелании) осуществления данной операции ( $\xi^x(\emptyset)$ ) после чего завершить выполнение алгоритмов предварительной обработки данных для данного параметра. Однако, не смотря на то, что данная возможность существует и агент осведомлен о наличии данного варианта развития событий, приведенное поведение агента не является предпочтительным в любом случае.

Следовательно, на этапе формирования предпочтений агента, получим множество достижимых миров убеждений  $W^d$  элементы которого, в общем случае, будут содержать поддеревья элементов множества убеждений  $W^b$ . В рамках рассматриваемого нами примера получим следующее древовидные структуры, характеризующие убеждения агента предварительной обработки данных:

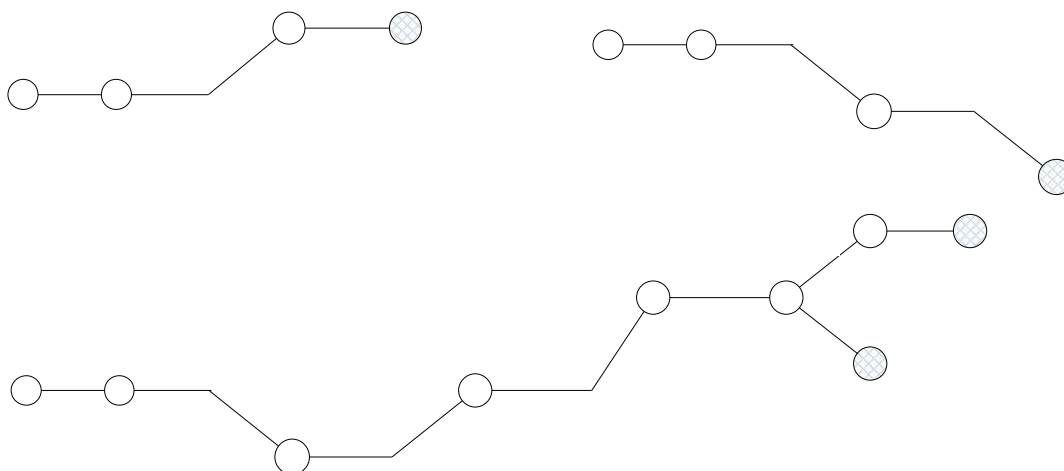


Рис. 8. Пример элементов множества предпочтений агента предварительной обработки данных.

В зависимости от целей агента, формируется множество намерений (intentions) агента, определяющих совокупность действий, подлежащих реализации. Формирование данного множества происходит с учетом целей достигаемых агентом. В соответствии с Рао и Джорджефом<sup>2</sup> агент анализирует различные варианты развития событий и выбирает единственный путь согласно максимальному критерию:

$$\text{INT}(\text{DES}, g(s)) = \max_{\text{DES}} (g(s)), \quad (13)$$

С течением времени система целей агента может изменяться. Таким образом, агент может принимать различные решения, находясь в одном и том же состоянии в различные моменты времени с различными текущими

приоритетами (целями). Возвращаясь к рассматриваемому примеру, получим множество намерений агента, изображенное на Рис. 9. Намерения агента были сформированы на базе множества предпочтений, полагая что следующее неравенство истинно:

$$g_i(s_{k5}) > g_i(s_{k4}); \quad (14)$$

где:  $g_i(s_{k5})$  – значение целевой функции агента в состоянии  $s_{k5} \in S^?$ ,  $S^? \subseteq S^d$ ;

$g_i(s_{k4})$  – значение целевой функции агента в состоянии  $s_{k4} \in S^?$ ,  $S^? \subseteq S^d$ .

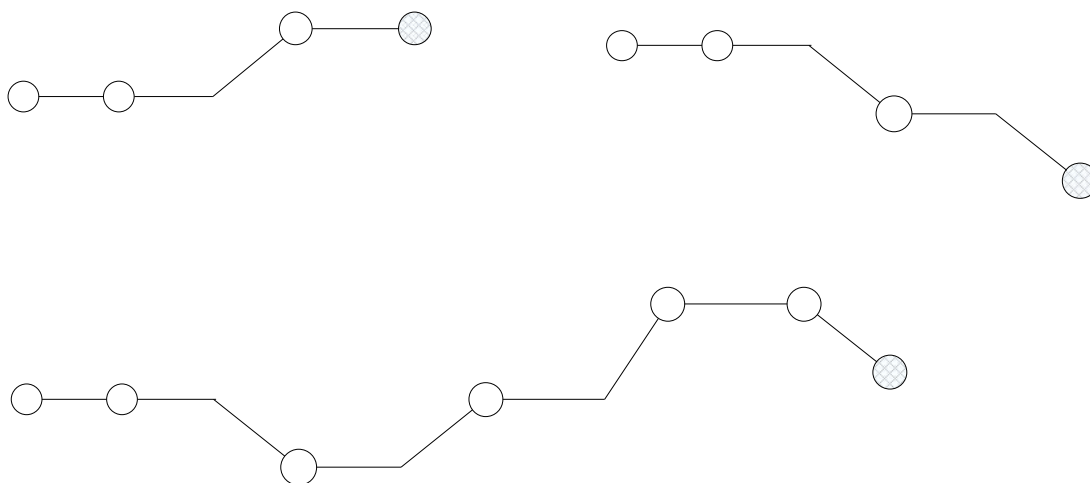


Рис. 9. Пример элементов множества намерений агента предварительной обработки данных

Дальнейшая формализация рассматриваемого подхода предусматривает использование структуры Крипке:

$$M = (W, \{S_w: w \in W\}, \{R_w: w \in W\}, L, B, D, I), \quad (15)$$

где:  $W$  – множество возможных достижимых миров убеждений  $W^b$ , предпочтений  $W^d$  и намерений  $W^i$ , т.е.  $W = W^b \cup W^d \cup W^i$ ;  $S_w$  – множество состояний для каждого мира  $W$ ;  $R_w$  – бинарное отношение достижимости определенное на множестве состояний  $S_w$ , т.е.  $R_w \subseteq S_w \times S_w$ ;  $L$  – функция назначения принадлежности атомарных высказываний из множества  $\Phi$  каждому миру  $w \in W$ , для каждого состояния  $s \in S_w$ , т.е.  $L(w,s): \Phi \rightarrow \{\text{True}, \text{False}\}$ ;  $B, D, I$  – отношения на множествах миров и состояний (в рамках представления убеждений агента получим  $B \subseteq W \times S \times W$ ).

Подробное рассмотрение процесса функционирования агента на основе структуры описанной данной моделью приведено в трудах Арразола<sup>3</sup> и Рао<sup>2</sup>.

#### 4. Распределенная многоагентная система предварительной обработки данных

Управление процессами предварительной обработки данных в многоагентной среде может быть автоматизировано за счет использования модели управления ИП предложенной ранее (см. (1)). В рамках данной было

спроектировано ограниченное множество модулей автоматизации ИП, каждый из которых агрегирует агент модуля автоматизации ИП (АМАИП), выполняя, таким образом, роль «промежуточного звена» в гибридной системе предварительной обработки данных (см. Рис. 10).

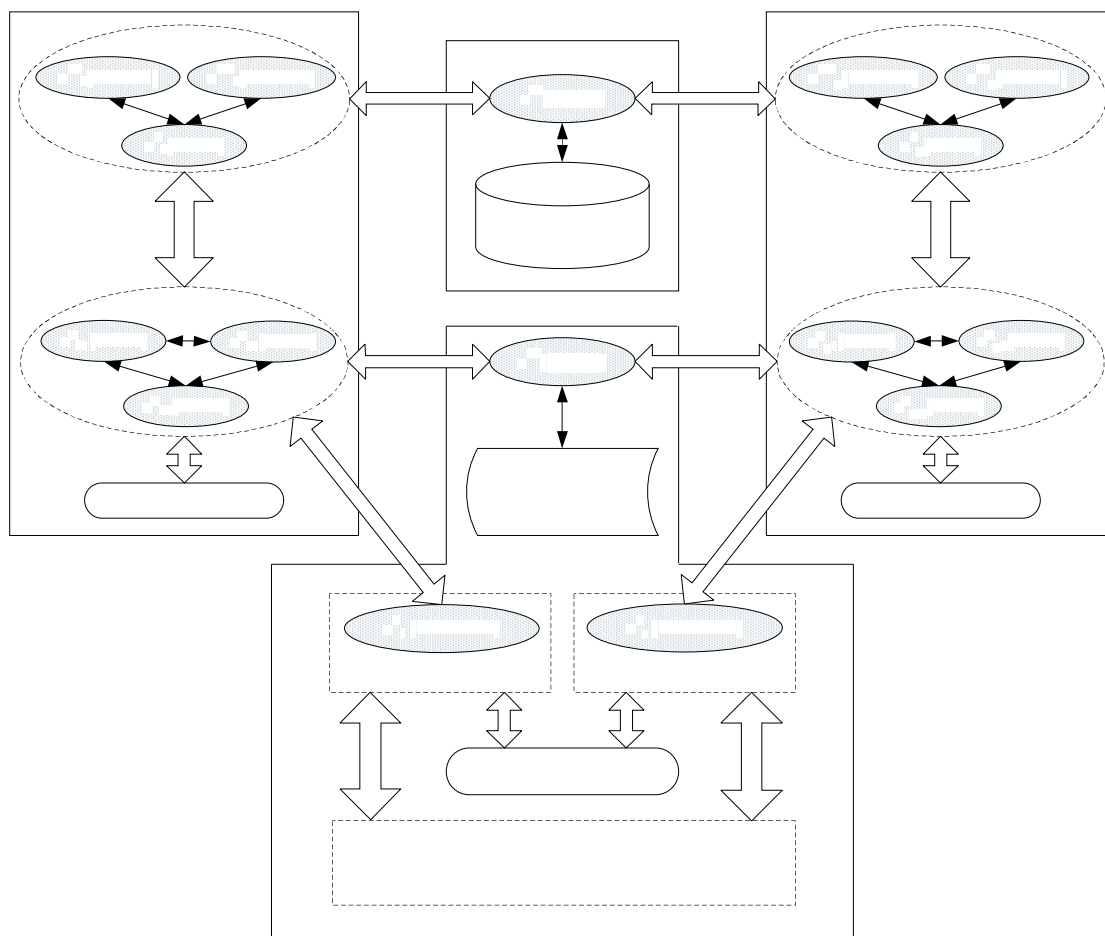


Рис. 10. Модель гибридной системы предварительной обработки данных

Функциональная структура агента модуля автоматизации ИП содержит следующие основные элементы:

- распределение множества параметров, значения которых являются объектами предварительной обработки данных между агентами предварительной обработки данных  $a_g^p$ , функционирующими на различных вычислительных платформах;
- взаимодействие с агентами  $a_g^p$  в целях их уведомления о наступлении событий в системе;
- сбор обработанных агентами системы данных для последующего экспорта заинтересованным клиентам.

В целях пояснения взаимодействия элементов системы рассмотрим процесс функционирования системы на верхнем уровне. При получении модулем автоматизации ИП управляющего воздействия направленного на инициализацию модуля автоматизации, агент модуля получает множество параметров, значения которых должны быть обработаны. Агент анализирует

элементы множества на предмет определения подходящего агента АПО и отправляет сообщение данному агенту содержащее запрос на обработку значения параметра. В случае отсутствия у АМАИП сведений об агентах предварительной обработки данных, АМАИП обращается к службе каталога содержащей всю информацию верхнего уровня о существующих в системе агентах. Агенты предварительной обработки осуществляют препроцессинг данных находящихся в соответствующем буфере, во взаимодействии с агентами иерархий параметров различных типов, после чего обработанное значение параметра передается агенту модуля автоматизации ИП.

## **Заключение**

Основные положения предложенной модели были использованы при разработке системы мониторинга данных в рамках проекта обновления программного обеспечения мониторинга и управления оборудованием – InCA (Injection Control Architecture). Система прошла успешные предварительные испытания в рамках централизованного тестирования систем мониторинга и управления оборудованием 13 августа и 14 ноября 2008 в Европейской организации ядерных исследований (ЦЕРН). В целях проверки работоспособности системы в условиях высоких нагрузок, операторами были запущены более 40 различных графических интерфейсов обеспечивающих визуализацию данных получаемых от синхротрона PS. Синхротрон PS представляет собой циклический резонансный ускоритель частиц с орбитой постоянного радиуса, с растущим во времени управляющим магнитным полем и постоянной величиной ускоряющего напряжения. Количество устройств подлежащих процедуре мониторинга и управления приближается к 1500, что соответствует примерно 12000 параметрам.

В рамках проведенных тестов одновременно системой обеспечивалось наблюдение в реальном времени более чем за 11000 параметрами PS ускорителя, при этом объем обрабатываемой информации достигал 50 Мб/с. В целом полученные результаты подтвердили состоятельность рассматриваемой модели с точки зрения решения задачи предварительной обработки данных.

Введение единой системы мониторинга и управления в эксплуатацию намечено на середину 2010 года.

### Список использованной литературы

<sup>1</sup>*Тарасов В. Б.* От многоагентных систем к интеллектуальным организациям: философия, психология, информатика.– М.: Эдиториал УРСС, 2002.– 352 с.

<sup>2</sup>*Rao A. S., Georgeff M.* Formal models and decision procedures for multi-agent systems./ Technical Report 61, Australian Artificial Intelligence Institute.  
<http://citeseer.ist.psu.edu/rao95formal.html>

<sup>3</sup>*Arrazola H., Pelletier F. , Korta K.* Discourse, Interaction and Communication Recueil Des Cours, 2007. 210 p.

### Сведения об авторе

*Лежебоков Валерий Валерьевич* – аспирант кафедры САПР и ПК, Волгоградский Государственный Технический Университет (ВолгГТУ), научный руководитель: д.т.н., проф. каф. САПР и ПК Камаев В.А.  
тел. +41(22) 7330703, e-mail: [valery.lezhebokov@gmail.com](mailto:valery.lezhebokov@gmail.com)