УДК 519.72

МОДЕЛИ ПРИНЯТИЯ РЕШЕНИЙ В ЗАДАЧАХ АНАЛИЗА СЛОЖНЫХ СИСТЕМ НА ОСНОВЕ СЕТЕЙ ВЫСОКОГО УРОВНЯ

к.т.н. Е.И. Кучеренко, к.т.н. В.А. Филатов
(представил д.т.н., проф. Е.В. Бодянский)

Статья посвящена проблеме управления информационными ресурсами вычислительных систем на основе технологии программных агентов. Учитывая динамические свойства управляемых объектов, для прогнозирования состояния всей системы предложен подход на основе модифицированного аппарата сетей Петри.

Введение

Проектирование и эксплуатация автоматизированных систем поддержки принятия решений, функционирующих в условиях жесткого ограничения на ресурсы, вызывает трудности, связанные с необходимостью обеспечения заданных функциональных и эксплуатационных свойств системы. Существующие средства диагностики и тестирования, как правило, обеспечивают лишь контроль состояния системы и, в меньшей степени, контроль выполнения заданий конкретными приложениями. Функционирование системы на основе серийных программно-аппаратных средств в значительной мере определяется качеством приложений. Положение усложняется распределенным характером обработки данных и значительным влиянием субъективного фактора на этапе проектирования.

Представление модели системы в виде целенаправленного взаимодействия сети фреймов позволяет реализовать сложные системные задачи. Однако использование фреймов имеет известные ограничения в их компьютерной реализации. В связи с этим возникает необходимость разработки и применения эффективного аппарата моделирования и анализа поведенческих свойств системы в процессе обработки данных. Учитывая значительные затраты на проектирование и эксплуатацию таких систем, высокие требования к качеству получаемых решений, задача моделирования, комплексного анализа и обеспечения качества поведенческих свойств автоматизированных систем в конкретных приложениях является достаточно актуальной, и в настоящее время не имеет приемлемых решений [1].

Постановка задачи

Пусть задана логическая схема функционирования автоматизированной системы управления сложным объектом. Схема представлена в виде сети взаимодействующих фреймов, содержащих слоты. Для слотов определены их значения в виде некоторых атрибутов, включающих данные, ссылки на другие фреймы, слоты, процедуры вычислений. Содержание атрибутов носит динамический характер, определяемый реальным временем.

В процессе функционирования системы возникают динамические тупиковые ситуации в полученных решениях при обработке данных.

Необходимо предложить и обосновать:

- аппарат формализации и анализа процессов обработки данных;
- формальные критерии выявления на модели динамических тупиковых ситуаций в полученных решениях при обработке данных.

Решения должны быть ориентированы на современные информационные технологии.

Состояние проблемы

Существующие подходы в значительной мере основываются на использовании сетевых моделей для исследования и моделирования процессов обработки данных.

Известно, что сеть фреймов может быть однозначно представлена семантической сетью, что дает принципиальные возможности ее использования для решения поставленной задачи. Однако в практических приложениях семантические сети достаточно громоздки, относительно плохо формализуемы. При росте количества фреймов и их размерности, соответствующая семантическая сеть теряет свою удобную интерпретацию, наглядность и становится мало пригодной для решения практических задач.

Перспективным направлением моделирования и анализа процедур обработки данных является применение сетей Петри и их расширений [2]. Так, предикатные сети Петри типа предикат / переход позволяют исследовать широкий класс задач по моделированию реальных процессов. Существующие модели и решения во многом ориентированы на исследование формальной структуры выполнения задач, например работы [3,4], и в меньшей степени учитывают собственно сами данные, их динамику в процессе обработки.

В связи с этим можно утверждать, что для реализации поставленных целей необходимо разработать сетевые модели, основанные на расширениях сетей Петри, позволяющих учитывать достоинства существующих решений и осуществлять моделирование и анализ объектов в реальном времени.

Построение моделей сложных процессов на основе сетей высокого уровня

Сеть Петри высокого уровня типа предикат / переход, которая по сути является раскрашенной сетью Петри, может быть представлена следующим образом:

$$\mathbf{S = < P, T, F, C, V, K, M_0 >}, \qquad (1)$$

где $\mathbf{P}$ – множество позиций; $\mathbf{T}$ – множество переходов;

$\mathbf{F = (P \times T) \cup (T \times P)}$ – функция инцидентностей позиций и переходов;

$\mathbf{C}$ – функция цвета маркера;

$\mathbf{V}$ – условия срабатывания переходов;

$\mathbf{K}$ – ёмкость маркеров в позициях с учетом $\mathbf{C}$;

$\mathbf{M_0}$ – вектор начальной маркировки.

Условие срабатывания переходов $\mathbf{V}$ отнесено к входным и выходным дугам переходов сети и представлено некоторым предикатом от переменных $\mathbf{C}$, $\mathbf{K}$.

Существенным ограничением моделей, построенных на основе сетей (1), является следующее:

- не учитывается фактор времени;
- не учитывается на содержательном уровне динамика взаимодействия данных и процедур;
- отнесение предиката $\mathbf{V}$ к дугам сети, а не к переходам не позволяет учитывать при моделировании объекта исследования такие свойства, как надежность, сложность, степень достоверности конкретного события, что создает определенные трудности в формализации процедур и анализе объекта исследования.

С целью расширения возможностей модели (1) рассмотрим следующую архитектуру предикатной сети:

$$\mathbf{S_{PR} = < P, T, F, A, C, V_s, K, M_0 >}, \quad (2)$$

где $\mathbf{A}$ – параметр времени, отнесенный ко всем компонентам сети: $\mathbf{P}$, $\mathbf{T}$, $\mathbf{F}$, $\mathbf{M_0}$; $\mathbf{V_s}$ – условия выполнения переходов $\mathbf{t_k \in T, k \in K}$.

$$\mathbf{V_{sk} = V_{p_k I}\ and\ V_{p_k O}\ and\ V_{t_k}\ and\ V_{M_{pl}}\ and\ V_F}, \quad (3)$$

где $\mathbf{V_{p_k I}}$ – условие выполнения $\mathbf{t_k \in T, k \in K}$, отнесенное к его входным позициям; $\mathbf{V_{p_k O}}$ – условие выполнения $\mathbf{t_k \in T, k \in K}$, отнесенное к его выходным позициям; $\mathbf{V_{t_k}}$ – условие выполнения $\mathbf{t_k \in T, k \in K}$, отнесенное непосредственно к этому переходу; $\mathbf{V_{Mp_l}}$ – условие выполнения $\mathbf{t_k \in T, k \in K}$, отнесенное к маркированию его инцидентным позициям $\mathbf{l \in L}$; $\mathbf{V_F}$ – условие выполнения $\mathbf{t_k \in T, k \in K}$, отнесенное к его входным и выходным дугам инцидентных позиций.

Следует отметить, что условие выполнения перехода $\mathbf{V_s}$ включает также такие свойства, как надежность, сложность, степень достоверности конкретного события. Некоторый переход будет запрещен, если, по крайней мере, одна из компонент (3) не будет удовлетворена.

Утверждение 1. Сеть $\mathbf{S_{PR}}$ (2) будет разрешена, если

$$\mathbf{\forall t_k, t_k \in T \mid V_{sk} = true}. \quad (4)$$

Для целей построения сетевых моделей введем интерпретацию компонент сети высокого уровня (2) в терминах объекта моделирования и исследования. Рассмотрим два иерархических уровня – уровень фрейма; уровень сети фреймов.

Фрейм обобщенно определим в виде [имя фрейма (<имя слота> <значение слота>), …, (<имя слота> <значение слота>)]. Значение слота

определяют данные, условия, действия, следствия, ссылки, присоединенные вычислительные процедуры.

Рассмотрим интерпретацию на уровне фреймов:

- множеству позиций **P** сети $\mathbf{S_{PR}}$ поставим в соответствие множество условий выполнения слотом действий;
- множеству переходов **T** сети $\mathbf{S_{PR}}$ поставим в соответствие множество действий, реализуемых слотом;
- функция инцидентностей **F** позиций и переходов интерпретирует причинно-следственные связи условий и действий, ссылок слота и между слотами в составе фрейма, причем $\mathbf{F = F^I \cup F^O}$, где $\mathbf{F^I, F^O}$ – соответственно входные и выходные инцидентности;
- параметр времени **A** на сети отнесен ко всем компонентам и определяет временные характеристики реализации исследуемых процедур;
- параметры $\mathbf{C, K}$, характеризующие свойства маркеров в позициях сети $\mathbf{S_{PR}}$, определяют условия выполнения действий, результат выполнения действий слота и между слотами фрейма;
- условия выполнения перехода $\mathbf{V_{sk}}$ определяют наличие характерных данному действию свойств, обеспечивающих выполнение слотом действий. Компоненты $\mathbf{V_{sk}}$ (3) определяют условия выполнения действий, отнесенных к соответствующей составляющей сети и реальных процессов;
- вектор начальной маркировки $\mathbf{M_0}$ определяет маркирование позиций сети и определяет выполнение (наличие) соответствующих условий реального процесса слота с учетом функции цвета.

Замечание 1. Для каждого фрейма определен формат слота, атрибуты и их содержимое составляют значение слота.

Рассмотрим интерпретацию на уровне сети фреймов:

- множеству позиций **P** сети $\mathbf{S_{PR}}$ поставим в соответствие множество условий выполнения фреймов действий;
- множеству переходов **T** сети $\mathbf{S_{PR}}$ поставим в соответствие множество действий, реализуемых фреймом;
- функция инцидентностей **F** позиций и переходов интерпретирует причинно-следственные связи условий и действий, ссылок между фреймами, причем $\mathbf{F = F^I \cup F^O}$, где $\mathbf{F^I, F^O}$ – соответственно входные и выходные инцидентности;
- параметр времени **A** на сети отнесен ко всем компонентам и определяет временные характеристики реализации исследуемых фреймами процедур;
- параметры $\mathbf{C, K}$, характеризующие свойства маркеров в позициях сети $\mathbf{S_{PR}}$, определяют условия выполнения действий, следствия выполнения фреймом действий;

- условия выполнения перехода $\mathbf{V_{sk}}$ определяют наличие характерных данному действию свойств, обеспечивающих выполнение фреймом действий. Компоненты $\mathbf{V_{sk}}$ (3) определяют условия выполнения действий, отнесенных к соответствующей составляющей сети и реальных процессов;
- вектор начальной маркировки $\mathbf{M_0}$ определяет маркирование позиций сети и определяет выполнение (наличие) соответствующих условий реального процесса фрейма с учетом функции цвета.

Замечание 2. Для каждого фрейма определен его формат, атрибуты и их содержимое.

Необходимо также отметить, что в содержимом замечаний 1 и 2 определяется как аналогия, так и существенное различие в представлении процедур обработки данных на уровне фрейма и сети фреймов. Это связано с различным уровнем детализации объекта исследования.

Формализация критериев анализа взаимосвязанных фреймовых структур

В ряде работ, например в [3], приведены ситуации, приводящие к динамическим тупиковым и конфликтных ситуациям в условиях распределенной обработки данных сетей ЭВМ. Однако до настоящего времени задача достаточно не формализована и не нашла удовлетворительных решений в практических приложениях.

В терминах классических сетей Петри тупиковая ситуация определена множеством переходов, которые не могут быть разрешены при заданной начальной маркировке [4].

При рассмотрении реальных процессов обработки данных необходимо дополнительно учитывать динамику и содержательную их составляющую. Условие (3) представим в виде некоторого динамического объекта, отнесенного к компонентам модели. Отметим, что в сети необходимо отсутствуют конфликтные ситуации. Как известно, конфликты в сети определены отсутствием безопасности, наличием более одной метки в некоторой позиции $\mathbf{M(p_l) > 1}$.

Введем понятие фактического $\mathbf{V_{sk}}'$ и ожидаемого $\mathbf{V_{sk}}''$ условий, фактического $\mathbf{M_\alpha}'$ и ожидаемого $\mathbf{M_\alpha}''$ векторов текущего маркирования, фактического $\mathbf{M_\beta}'$ и ожидаемого $\mathbf{M_\beta}''$ терминального маркирования.

Утверждение 2. Если в модели на основе сети (2) справедливо при заданном векторе начальной маркировки $\mathbf{M_0}$, по крайней мере, одно условие:

$$\mathbf{M_\alpha}' \neq \mathbf{M_\alpha}'' \mid \mathbf{A > A^*}; \quad (5)$$

$$\mathbf{M_\beta}' \neq \mathbf{M_\beta}'' \mid \mathbf{A > A^*}, \quad (6)$$

где $\mathbf{A}^*$ – верхняя оценка времени реализации процесса, то в сети существует тупиковая ситуация.

Доказательство утверждения 2 непосредственно следует из сущности вектора маркировки и определения динамической тупиковой ситуации.

Утверждение 3. Если в модели на основе сети (2) справедливо при заданном векторе начальной маркировки $\mathbf{M_0}$ условие:

$$\exists \mathbf{t_k} \in \mathbf{T} \,|\, ((\mathbf{A} \le \mathbf{A}^*) \mathbf{and} (\mathbf{V_k}' \ne \mathbf{V_k}'')), \qquad (7)$$

то в сети существует тупиковая ситуация.

Доказательство утверждения 3 непосредственно следует из сущности условия (3) и определения динамической тупиковой ситуации.

Следствие 1. Коньюнкция (7) и дизьюнкции (5), (6) определяет достаточные условия существования динамических тупиковых ситуаций.

Следствие 2. Устранение динамических тупиковых ситуаций может быть осуществлено путем следующих целенаправленных процедур:

- моделирования и анализа динамических тупиковых ситуаций;
- выявления факта существования тупиковой ситуации согласно (5), (6), (7);
- локализации тупиковой ситуации;
- устранения тупиковой ситуации путем целенаправленных действий по изменению условий (5), (6), (7);
- модификации модели;
- модификации процессов предметной области [5].

Рассмотренный подход может быть применен для оперативного анализа задачи автономного администрирования и управления потоком задач на основе технологии программных агентов.

Для формального представления задачи автономного управления информационными ресурсами вычислительной системы введем следующие определения.

Мультиагентное пространство определим как систему

$$\mathbf{MP} = < \mathbf{Z}, \mathbf{M}, \mathbf{S_z}, \mathbf{G}, \mathbf{\Omega_z} >, \qquad (8)$$

где Z – множество задач информационного пространства МР;

$\mathbf{M}$ – множество программных модулей, реализации решений задач;

$\mathbf{S_z}$ – множество отношений между задачами;

$\mathbf{G}$ – множество программных агентов, реализующих функцию автономного управления и контроля выполнения задач в соответствии с $\mathbf{S_z}$;

$\mathbf{\Omega_z}$ – множество операций по управлению $\mathbf{Z}, \mathbf{M}, \mathbf{G}$ в мультиагентном пространстве.

Одной из известных структур представления данных и знаний, удовлетворяющей концептуальной модели мультиагентного пространства (8), является фрейм.

Для реализации технологии взаимодействия программного агента с объектами вычислительной среды в качестве модели программного агента также использована фреймовая структура [6].

В общем случае модель программного агента может быть записана в виде:

$$FR\{< R_1, C_{11}, C_{12}, \ldots C_{1m}, >, \ldots < R_2, C_{21}, C_{22}, \ldots C_{2m}, >, < R_{k1}, C_{km} >\}, \qquad (9)$$

где **FR** – имя фрейма (агента);

пара **<Ri,Ci>** – i -й слот фрейма;

Ri, – имя слота,

Ci – значение слота.

Слот в модели (9) является логической конструкцией для реализации конкретных заданий "фрейму - программному агенту". Слоты из фрейма можно удалять, добавлять, изменять функциональное назначение слота-задания. Однако в виде (9) классическое представление слота в виде <имя>,<значение> не может в полной мере отражать требования концептуальной модели (8).

Модифицируем структуру слота и приведем его к виду

$$\mathbf{Slot} = \langle \mathbf{Y}, \mathbf{D}, \mathbf{dom}, \mathbf{r_i}, \theta, \Sigma \rangle , \qquad (10)$$

где $\mathbf{Y}$ – множество имен атрибутов,

$\mathbf{D}$ – множество доменов, *dom* – отображение $\mathbf{Y} \Rightarrow \mathbf{D}$,

$\mathbf{r_i}$ – модель-кортеж i-го задания агента,

Σ – множество операций над отношениями.

$$\mathbf{r_i} = \{\{\mathbf{R}\}_\mathbf{i}, \Omega_{\mathbf{ij}}, \mathbf{V_i}\}, \qquad (11)$$

где $\{\mathbf{R}\}_\mathbf{i}$ – множество состояний кортежа $\mathbf{r_i}$,

$\mathbf{V_i}$ – множество ограничений целостности,

$\Omega_{\mathbf{ij}}$ – множество операций заданных на $\{\mathbf{R}\}_\mathbf{i}$,

θ – множество, определяющее начальные условия и признаки выполнения действий в структуре задания.

Пример

Проиллюстрировать эффективность разработанного подхода к задаче оперативного анализа поведения программного агента в информационной среде на основе аппарата сетей Петри можно на следующем примере.

Постановка задачи: в регионе произошла чрезвычайная ситуация, возникает необходимость размещения в профилакториях, пионерских лагерях, гостиницах попавших в экстремальную ситуацию людей. Существуют и поддерживаются в актуальном состоянии базы данных соответствующих объектов – B1, B2, B3. Необходимо последовательно заполнять свободные места: сначала – в профилакториях, затем в пионерских лагерях и, наконец, в гостиницах. Общее количество людей, попавших в беду известно. Так как информационные ресурсы в виде баз данных региональной системы реагирования на чрезвычайные ситуации для решения данной задачи

существуют, решить ее можно автономно при помощи агентного подхода. На рис.1 представим универсальную модель: граф-схему взаимосвязанных задач $\mathbf{Q1, Q2, Q3, Q4}$.

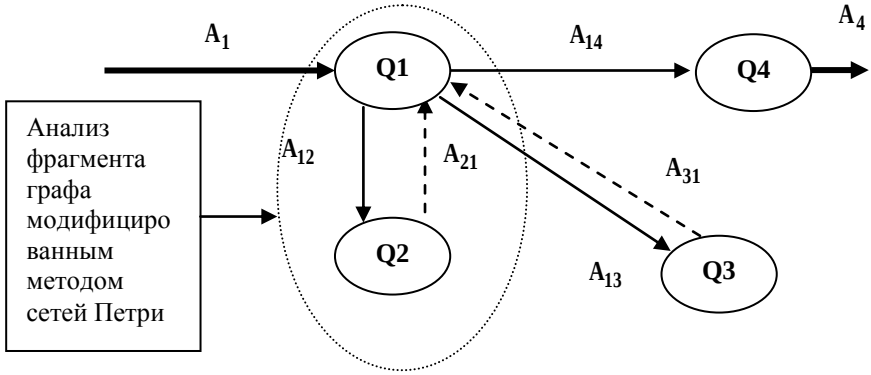

Рис.1 Схема взаимодействия задач.

Подробное описание задач в потоке работ приведено в табл.1.

Таблица 1

Описание функциональных задач

Идентифика-тор, описание	Объект	Описание задачи	Действие	Логические ограничения на выполнение действий
Q1, выбрать из базы данных B1 количество свободных мест. Условие [mest]>= M^*	D:\B1. dbf	Если количество свободных мест больше заданного ограничения, перейти к задаче Q4, если условие не выполнено, перейти к задаче Q2	Select sum([mes t]), from b1.dbf	Уровень доступа к диску "D" – общий
Q2, добавить к базе данных B1 базу данных B2	D:\B2. dbf	Если операция добавления прошла корректно, перейти к задаче Q1 и повторить ее.	Insert into b1… Select b2…. From b2	Обновления базы B1 каждые 7 дней [DATE_new] >=[DATE()]-7
Q3, добавить к базе данных B1 базу	D:\B3. dbf	Если операция добавления прошла корректно,	Insert into b1… Select	

данных B3		перейти к задаче Q1 и повторить ее.	b3…. From b3	
Q4, вывести данные на печать				

Для выделенного на рис.1. фрагмента сформируем сеть Петри. Переход $\mathbf{t_0}$ соответствует описанию входных условий (дуга $\mathbf{A_1}$) на выполнение действия по задаче **Q1**. Переход $\mathbf{t_1}$ моделирует особенности решения задачи Q1, которая выполняется при входном условии $\mathbf{M_{p1} = 1 | V = \{0,1\}}$ – уровень доступа к диску "D" – общий. Начальные условия для логических ограничений: уровень доступа к ресурсам на диске "D" – общий, последнее обновление базы данных B2 было 11 дней назад. На рис.2 приведен фрагмент сети Петри для динамического анализа поведения программного агента в информационной среде.

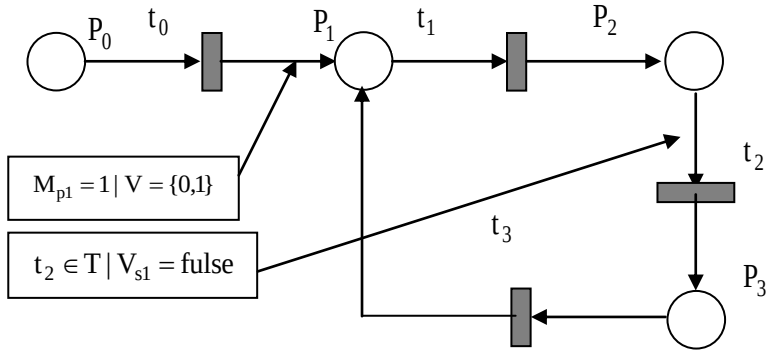

Рис.2 Фрагмент сети Петри

Путем моделирования и анализа определено, что произошла ошибка в обработке ситуации, $\mathbf{t_2 \in T | V_{s1} = fulse}$, дата $\mathbf{B_4}$ последнего обновления базы данных не соответствует входным условиям, что нарушает логическое ограничение на выполнение операции.

Таким образом, моделирование процессов обработки данных на основе модифицированной модели сетей Петри, позволяет эффективно оценивать логическую организацию сложных распределенных вычислительных процессов.

Выводы

В статье сформулирована постановка задачи анализа, выявления и устранения динамических тупиковых ситуаций в информационно-управляющих системах, функционирующих в условиях реального времени.

Определено, что эффективным аппаратом моделирования и анализа систем с целью анализа, выявления и устранения динамических тупиковых ситуаций являются сети Петри и их расширения.

Предложены и обоснованы новые расширения классов сетей Петри высокого уровня путем совершенствования предикатных сетей Петри, ввода предикатов и временных характеристик для пространства состояний и компонентов сети. Сформулированы правила построения и интерпретации сетевых моделей на основе новых расширений классов сетей Петри высокого уровня в задачах анализа процедур обработки данных на иерархических уровнях фреймовых структур.

Сформулированы утверждения, определяющие формальные условия существования динамических тупиковых ситуаций. Определено, что формальные критерии существования динамических тупиковых ситуаций являются основой их выявления, локализации и устранения в практических реализациях на основе модификации сетевых моделей и реальных процессов предметных областей.

Приведен пример практического применения полученных теоретических и практических результатов в конкретных разработках.

ЛИТЕРАТУРА

1. *Дмитриев А.К., Мальцев П.А. Основы теории построения и контроля сложных систем. – Л.: Энергоатомиздат. Ленингр. отделение, 1988. – 192 с.*
2. *Мурата Т. Сети Петри: Свойства, анализ, приложения // ТИИЭР, апрель 1989 г. – № 4. – с. 41 - 85.*
3. *Rokyta P., Fengler W., Hummel T. Electronic system design automation using high level Petri nets // Workshop for Hardware Design and Petri Nets, Lisboa, June 22 - 26, 1998. – 1998. – p. 129 -138.*
4. *Yakovlev A., Gomes L., Lavagno L. Hardware Design and Petri Nets // Kluwer Academic Publishers. – Boston: Kluwer Academic Publishers, 2000. – p. 193-204.*
5. *Кучеренко Є.І. Ефективна структура засобів опису та моделювання нечітких взаємодіючих процесів в інтелектуальних системах // Проблемы бионики. – 2001. – Вып. 55. – с. 98 - 101.*
6. *Филатов В.А. Модели и методы защиты информационных ресурсов на основе агентных технологий // Вестник Национального технического университета "ХПИ". – Харьков: НТУ "ХПИ".– 2003. – с.179-186.*

***Кучеренко Евгений Иванович**, канд. техн. наук, доцент кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники. В 1973г. окончил Харьковский институт радиоэлектроники. Область научных интересов – интеллектуальные системы поддержки принятия решений.*

***Филатов Валентин Александрович**, канд. техн. наук, доцент кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники. В 1980г. окончил Харьковский институт*

радиоэлектроники. Область научных интересов – базы данных и знаний, агентные технологии и мультиагентные системы. Filatov_val@ukr.net

УДК 519.72

Моделі прийняття рішень у задачах аналізу складних систем на основі мереж високого рівня / Є. І. Кучеренко, В.О. Філатов // Системи обробки інформації. – 2004. – Вип. 0(00). – С. 000-000.

У статті розглянута проблема управління інформаційними ресурсами за допомогою технології програмних агентів. Вважаючи динамічні властивості об'єктів, що управляються, запропоновано підхід щодо побудови моделі поведінки програмних агентів на основі модифікованого апарату мереж Петрі.

Табл.1. Іл.2. Бібліогр. 6 назв.

UDC 519.72

Decision support models in the problems of complex system analysis based on high level networks / E.I. Kucherenko, V.A. Filatov // Sistemi obrobki informacii. – 2004. – Issue 0(00). – P. 000-000.

The paper deals with the problem of control of information resources by means of software agent technology. Taking dynamic properties of the controled objects into account, an approach to the construction of behavior model based on modified Petri nets. The offered models can be used for development the systems for administration of information resources in the allocated computing systems. The given example illustrates the offered approach.

Tab. 1. Fig. 2. Ref. 6 items.