

Розглянуто питання опису закономірностей, що виникають у роботі програмних і апаратних компонентів розподілених інформаційних систем. Введені базові терміни і ряд тверджень, що розширюють багатосортну мову прикладної логіки для опису знань цієї предметної області. Розроблено модель онтології діагностики компонентів розподілених інформаційних систем, що базується на цьому розширенні

Ключові слова: онтологія, сервіси, розподілені інформаційні системи, багатосортна прикладна логіка

Рассмотрен вопрос описания закономерностей, возникающих в работе программных и аппаратных компонентов распределенных информационных систем. Введены базовые термины и ряд утверждений, расширяющие многосортный язык прикладной логики для описания знаний этой предметной области. Разработана модель онтологии диагностики компонентов распределенных информационных систем, основанной на этом расширении

Ключевые слова: онтология, сервисы, распределенные информационные системы, многосортная прикладная логика

РАЗРАБОТКА МОДЕЛИ ОНТОЛОГИИ ДИАГНОСТИКИ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ МНОГОСОРТНОГО ЯЗЫКА ПРИКЛАДНОЙ ЛОГИКИ

П. М. Тишин

Кандидат физико-математических наук, доцент
Компьютерные интеллектуальные системы и сети*

E-mail: tik88@mail.ru

А. С. Маковецкий

Старший преподаватель
Кафедра информационных систем*

E-mail: drnewman@mail.ru

*Одесский национальный политехнический университет
пр. Шевченка, 1, г. Одесса, Украина, 65044

1. Введение

В настоящее время существует два варианта разработки систем обработки сложно-структурированной информации. В первом варианте разработчику предлагается готовый модуль, который должен сформировать базу знаний о предметной области, при этом часть знаний о решении задачи встраивается в базу знаний. В качестве способа представления знаний в этом случае используются базы правил. Для диагностики распределенных информационных систем, рассуждения на основе правил (RBR -системы) применил [1]. При этом коммерческие системы диагностики, такие как IBM Tivoli Enterprise Console [2], HP OpenView Event Correlation Services [3], Micromuse Netcool Impact [4], основаны на правилах.

Уже первый опыт применения систем, основанных на правилах, показал исключительную сложность их разработки и сопровождения [5]. Модификация и расширение знаний в процессе эксплуатации системы, осталась трудоемким процессом [6]. В частности, в работе [7] RBR-системы, предназначенные для диагностики распределенных информационных систем, классифицируются как относительно негибкие. Второй вариант разработки предполагает сначала формирование онтологии, по которой может быть создан класс различных баз знаний. В онтологии OWL [8] определяется иерархия классов для представления

основных ее объектов, множество слотов для описания их свойств и отношений между ними, а также множество экземпляров классов. Однако использование для описания предметной области аппарата дескриптивной логики ограничивает ее применение при работе с сложно-структурированной информацией.

2. Анализ литературных данных и постановка проблемы

Рост распределенных информационных систем требует все большей автоматизации работ по их обслуживанию, что, в свою очередь, требует средств для более эффективной диагностики. В работе [9] для диагностики распределенных информационных систем предлагается использовать гибридную интеллектуальную систему, включающую систему рассуждения на основе правил и систему рассуждения на основе прецедентов. Однако в построенной системе присутствует множество ограничений. К примеру, окно корреляции событий, порядок событий не участвуют в рассмотрении, а для описания состояния сервисов и ресурсов вводятся только бинарные величины, не позволяющие вводить ухудшение качества сервиса или ресурса.

Переход к представлению знаний с использованием многосортного языка прикладной логики уже внес свой вклад в сопровождаемость баз знаний. По-

строенные базы знаний не только отделены в самостоятельный переносимый компонент, но и являются концептуальными, понятными для сопровождающих их экспертов [10]. В процессе построения системы создаются модели предметной области, в числе которых онтология знаний и данных, модель деятельности и концептуальная модель системы [11]. В результате предложена концепция проектирования интеллектуальных систем, все компоненты (данные, знания, решатель задач с пользовательским интерфейсом) которого имеют единые принципы для их формирования, доступа и модифицирования [12]. Однако подход разрабатываемый данными авторами применялся при решении задач органической химии и медицинской диагностики, но не применялся для решения задач диагностики распределенных информационных систем.

3. Цель и задачи исследования

В данной работе, при построении систем диагностики, основанных на знаниях, предлагается подход с использованием многосортного языка прикладной логики [13, 14], который позволяет описывать закономерности в сложных предметных областях.

Для достижения поставленной цели были поставлены следующие задачи:

- ввести основные понятия онтологии диагностики распределенных информационных систем;
- разработать модель онтологии диагностики распределенных информационных систем.

4. Разработка модели онтологии диагностики распределенных информационных систем

Поскольку основными структурными элементами современных РИС являются службы, неисправное состояние программных и аппаратных элементов РИС ведет к невозможности выполнять службами заданные функции. Под службой в данной работе понимаем взаимодействующие посредством РИС функциональные компоненты, составные части которых могут выполняться в отдельных аппаратно-программных окружениях.

В основу разрабатываемой модели положим представление РИС в виде множества служб [15, 16]

$$S = \{s_l\}_{l=1}^L, \quad (1)$$

где L – общее количество служб.

В соответствии с работой [16] совокупность источников диагностической информации представляется в виде множества

$$A = \{a_i\}_{i=1}^I, \quad (2)$$

где I – общее количество источников диагностической информации в рассматриваемой РИС. Представим совокупность диагностических параметров некоторого источника диагностической информации $a \in A$ в виде

$$P(a) = \{p_j(a)\}_{j=1}^{J(a)}, \quad (3)$$

где $J(a)$ – общее количество диагностических параметров источника диагностической информации $a \in A$.

В соответствии с рассматриваемой диагностической моделью в процессе функционирования сети из множества $P(a)$ формируется множества диагностических параметров $\{B(s_l)\}_{l=1}^L$, значения которых в дальнейшем необходимо учитывать в процессе идентификации состояния некоторой службы $s_l, l=1, \dots, L$. Обозначим совокупность диагностических параметров $B(s)$ ($s \in S$) в виде

$$B(s) = \{b_m(s)\}_{m=1}^{M(s)}, \quad (4)$$

где $M(s)$ – общее количество диагностических параметров службы $s \in S$.

Модель онтологии диагностики корпоративной сети строится на основе многосортного языка прикладной логики. Каждый конкретный язык прикладной логики включает ядро, а также обычно стандартное расширение и некоторые специализированные расширения. Таким образом, каждый конкретный язык прикладной логики характеризуется некоторой совокупностью названий расширений, а не сигнатурой.

Сигнатура же вводится в каждой конкретной логической теории, заданной на таком языке. При этом предложения этой теории могут сопоставлять элементам сигнатуры (именам) их значения (интерпретацию) или сорта, либо ограничивать возможные функции интерпретации этих имен в зависимости от интерпретации других имен.

В настоящем работе вводятся базовые термины расширения, которое используется для описания знаний предметной области – диагностика РИС, а также ограничения на их значения (не зависящие от значений терминов для описания действительности). Действительность в диагностике рассматривается как множество ситуаций, каждая из которых соответствует диагностическому случаю.

В итоге сформулировано множество утверждений. Данное утверждение с применением многосортного языка прикладной логики представляется в виде:

(1) Термин *разбиения* обозначает множество всех возможных разбиений множества целых неотрицательных чисел; каждое разбиение представляет собой конечную строго возрастающую последовательность.

разбиения $\equiv (\cup$ (длина: $I[0, \infty)$) {(последовательность: $I \uparrow$ (длина+1)) (&(элемент: $I[1, \text{длина}]$) π (элемент, последовательность) < π (элемент+1, последовательность)}).

(2) Термин *el* обозначает функцию, аргументами которой являются некоторое разбиение и целое число в диапазоне от 0 до числа элементов в этом разбиении, а результатом – элемент этого разбиения, номер которого равен второму аргументу.

$el \equiv (\lambda$ (разбиение: разбиения) (элемент: $I[0, \text{length}(\text{разбиение})-1]$) π (элемент+1, разбиение)).

(3) Термин *службы* обозначает совокупность объектов которые определяются соотношениями (1).

сорт службы: $\{\}N \setminus \{\emptyset\}$.

(4) Термин *источники* обозначает совокупность источников диагностической информации которые определяются соотношениями (2). В знаниях должен быть описан хотя бы один источник диагностической информации.

сорт *источники*: $\{N \setminus \{\emptyset\}\}$.

(5) Термин *параметры* обозначает класс понятий, соответствующих наблюдаемым диагностическим параметрам. В знаниях должен быть описан хотя бы один параметр.

сорт *параметры*: $\{N \setminus \{\emptyset\}\}$.

(6) Термин *параметры_ источники* обозначает функцию, которая каждому источнику диагностической информации сопоставляет множество диагностических параметров в соответствии с формулой (3).

сорт *параметры_ источники*: $\text{источники} \rightarrow \{\text{параметры}\}$.

(7) Термин *особенности* обозначает класс понятий, соответствующих особенностям объектов, которые должны учитываться при диагностике.

сорт *особенности*: $\{N\}$.

(8) Термин *множества значений (мз)* обозначает множество всех допустимых множеств скалярных значений.

$\text{мз} \equiv \{N \setminus \{\emptyset\}\}$.

(9) Значения не совпадают с названиями параметров.

$\text{параметры} \cap (\cup (x: \text{мз}) x) = \emptyset$.

(10) Термин *возможные значения (вз)* обозначает функцию, которая параметрам сопоставляет их возможные значения.

сорт *вз*: $\text{параметры} \rightarrow \text{мз}$.

(11) У каждого параметра не менее двух возможных значений.

$(x: \text{параметры}) \mu(\text{вз}(x)) \geq 2$.

(12) Термин *события* обозначает класс понятий, соответствующих событиям, которые должны учитываться при решении задачи.

сорт *события*: $\{N \setminus \{\emptyset\}\}$.

(13) Названия всех параметров и событий различны.

$\text{параметры} \cap \text{события} = \emptyset$.

(14) Каждый термин, входящий в класс терминов события, обозначает структурное значение с тремя

атрибутами: *служба (s)*, *параметр (p)*, *источник (a)*. Значением первого является наименование службы, второй атрибут является наименованием параметра.

(*e*: события) $e: (s \rightarrow \text{службы}, p \rightarrow \text{параметры}, a \rightarrow \text{источники})$.

(15) Термин *условия* обозначает множество всех возможных условий. Это множество, состоящих из элементов – структурных значений. Каждое условие есть конечное множество структурных значений. Каждое структурное значение имеет атрибуты: особенность (*o*) и область значений (*оз*). Значением первого является имя события, а второго – собственное подмножество возможных значений (*вз*) этой особенности. Пустое множество представляет тождественно истинное условие.

$\text{условия} \equiv \{(\text{условие}: (o \rightarrow \text{особенности}, оз \rightarrow \text{мз})) \mid оз(\text{условие}) \subset \text{вз}(o(\text{условие}))\}$.

(16) Термин *условия события* обозначает функцию, которая каждому событию сопоставляет необходимые условия его существования

сорт *условия события*: $\text{события} \rightarrow \text{условия}$.

(17) Термин *причины отклонений* обозначает класс понятий, соответствующих причинам отклонений, описания которых представлены в знаниях. В знаниях должно присутствовать описание хотя бы одной причины отклонений.

сорт *причины отклонений*: $\{N \setminus \{\emptyset\}\}$.

(18) Каждый термин, входящий в класс терминов причины отклонений, обозначает структурное значение с тремя атрибутами: число периодов развития (*чпр*), периоды развития (*пр*), необходимое условие (*ну*). Значением первого является натуральное число, второй атрибут является функцией, которая номеру периода развития причины отклонений сопоставляет интервал, значение третьего атрибута – условие, необходимое для существования этого причины отклонений в ситуации (если значением атрибута является пустое множество, то условие считается истинным).

(причина: *причины отклонений*) сорт причина: $(\text{чпр} \rightarrow \mathbb{I}[1, \infty), \text{пр} \rightarrow (\mathbb{I}[1, \text{чпр}] \rightarrow \text{интервал}), \text{ну} \rightarrow \text{условия})$.

(19) Термин *периоды динамики* это множество структурных значений с атрибутами длительность и область значений следствия (ОЗС). Значением первого атрибута является интервал, а второго – множество значений.

$\text{периоды динамики} \equiv (\text{длительность} \rightarrow \text{интервал}, \text{ОЗС} \rightarrow \text{множества значений})$.

(20) Термин *интервал* это множество элементов структурных значений с атрибутами нижняя граница (*нг*) и верхняя граница (*вг*). Их значениями являются

натуральные числа – минимальная и максимальная длительности интервала, измеряемая целым числом, причём верхняя граница больше нижней.

$$\text{интервал} \equiv (\text{нг} \rightarrow \mathbb{I}[1, \infty), \text{вг} \rightarrow \mathbb{I}[\text{нг}+1, \infty)).$$

(21) Термин *моменты (t)* обозначает функцию, которая каждому параметру и событию сопоставляет множество целых неотрицательных чисел – моментов времени в ситуации, когда наблюдался этот параметр. Если для некоторого параметра значение этой функции есть пустое множество, то это означает, что этот параметр не наблюдался.

$$\text{сорт } t: \text{параметры} \cup \text{события} \rightarrow \{\mathbb{I}[0, \infty)\}.$$

(22) Каждый термин, входящий в класс терминов *параметры*, обозначает функцию, которая сопоставляет моментам наблюдения этого параметра (p) его значения в эти моменты.

$$(p: \text{параметры}) \text{ сорт } p: t(p) \rightarrow \text{вз}(p).$$

(23) Термин *наблюдавшиеся события* обозначает подмножество событий, наблюдавшихся в ситуации.

$$\text{наблюдавшиеся события: } \{\text{события}\}.$$

(24) Каждый термин, входящий в класс терминов *наблюдавшиеся события*, обозначает значение, которое принимает параметр (p).

$$(\text{событие: наблюдавшиеся события}) \text{ сорт событие: вз}(p(\text{событие})).$$

(25) Термин *выполнено* обозначает предикат, аргументом которого является элемент множества условия и который истинен тогда и только тогда, когда для каждой составляющей этого элемента, значение первого атрибута структурного значения в ситуации принадлежит второму атрибуту (область значений). Пустое условие тождественно истинно.

$$\text{выполнено} \equiv (\lambda (y: \text{условия}) y \neq \emptyset \Rightarrow (\& (x: y) \text{ событие}(x) \in \text{наблюдавшиеся события} \Rightarrow j(\text{параметр}(\text{событие}(x))) \notin \text{оз}(x))).$$

(26) Если в ситуации хотя бы один раз наблюдалось некоторое событие, то для него должно быть выполнено условие.

$$(x: \text{наблюдавшиеся события}) t(x) \neq \emptyset \Rightarrow \text{выполнено}(\text{условия события}(x))$$

(27) Термин *диагноз* обозначает множество причин отклонений. Если отклонений нет, его диагноз есть пустое множество.

$$\text{сорт диагноз: } \{\text{причины отклонений}\}.$$

(28) Если некоторая причина отклонения входит в диагноз, то для этой причины отклонений должно быть выполнено необходимое условие.

$$(x: \text{диагноз}) \text{ выполнено}(\text{ну}(x)).$$

(29) Термин действительности *развитие (Ip)* обозначает функцию, которая каждой причине отклонений, сопоставляет разбиение оси времени, каждый интервал которого соответствует некоторому периоду развития этой причины, и каждому наблюдавшемуся событию сопоставляет разбиение оси времени, внутри каждого интервала которого значения этого события определяются одной и той же связанной с этим интервалом причиной.

$$\text{сорт } Ip: \text{диагноз} \cup \{(x: \text{наблюдавшиеся события}) t(x) \neq \emptyset\} \rightarrow \text{разбиения}.$$

(30) Интервал, на котором наблюдается развитие некоторого параметра, покрывает все моменты наблюдения этого параметра.

$$(x: \text{параметры}) t(x) \neq \emptyset \Rightarrow \text{el}(Ip(x), 0) \leq \text{inf}(t(x)) \& \& \text{el}(Ip(x), \text{length}(Ip(x))) \geq \text{sup}(t(x)).$$

(31) Термин *интервалы развития параметра (U)* это множество структурных значений, состоящих из двух атрибутов: параметр (p) и номер интервала (N). Значением первого атрибута является имя параметра, второго – номер интервала его развития.

$$U \equiv (p \rightarrow \text{параметры}, N \rightarrow \mathbb{I}[1, \text{length}(Ip(p)) - 1]).$$

(32) Если причина отклонения входит в диагноз, то число периодов развития этой причины отклонений совпадает с числом периодов его развития в базе знаний, а длительность каждого периода развития лежит между нижней и верхней границами длительностями этого периода развития причины отклонений.

$$(x: \text{диагноз}) \text{ length}(Ip(x)) = \text{чпр}(x) + 1 \& \& (\& (n: \mathbb{I}[1, \text{length}(Ip(x)) - 1]) \text{ el}(Ip(x), n) - \text{el}(Ip(x), n - 1) \in \mathbb{I}[\text{нг}(\text{чпр}(x)(n), \text{вг}(\text{чпр}(x)(n))])).$$

5. Пример применения разработанной модели онтологии

В качестве примера введем в рассмотрение службу, которая будет работать под управлением веб сервера Apache HTTP Server. В качестве источника диагностической информации в данном случае выберем сетевую карту. В данной работе совокупность диагностических параметров определяется с помощью MIB деревьев. MIB дерева представляет собой иерархическую организацию соответствующих данных для устройств, в то время как для отдельной задачи необходимы только некоторые переменные из этих переменных. С точки зрения задачи диагностики значения каждой отдельной контролируемой величины определяется параметрами

$$\{H, T, O, V\}, \tag{5}$$

где H – имя хоста, T – набор времен получения данных, O – OID или имя переменной, V – значение переменной.

В соответствии с этим, одна MIB переменная имеет ряд значений (в соответствии с любыми контроли-

руемыми узлами), и каждый из них имеет свой собственный ряд значений во времени. Контролируемая модель может быть описана множеством всех MIB переменных, которые наблюдаются во времени. Изменяемыми параметрами для сетевой карты будут Object Identifier(OID) из Management Information Base(MIB). OID'ы которые, были использованы для диагностики сетевой карты, приведены в табл. 1.

Таблица 1

OID использованные для диагностики сетевой карты

Название OID	Описание
ipInDelivers	Полное число входных дейтограмм, успешно обработанных на IP-уровне.
ipInReceives	Число полученных дейтограмм.
tcpInSegs	Полное число полученных tcp-сегментов.

Диагностическими параметрами P(a), определяемыми формулами (3), в данном случае будут производные от отобранных параметров ipInDelivers, ipInReceives, tcpInSegs, которые представляет собой скорость изменения определенных OID'ов.

Эти же параметры образуют множество B(s) введенное соотношением (4). Таким образом, для формализованного использования параметров системы, в соответствии с [15], можно определить лингвистические переменные:

$$\begin{aligned} P1 &= V_{cp} (ipInDelivers); \\ P2 &= V_{cp} (ipInReceives); \\ P3 &= V_{cp} (tcpInSegs). \end{aligned} \quad (6)$$

Предполагая, что данные лингвистические переменные определены на множестве термов: малая, средняя, большая и обозначая через S веб сервер Apache HTTP Server можно ввести следующий набор предложений:

$$\begin{aligned} \text{службы} &\equiv \{ S \}, \\ \text{источники} &\equiv \{ \text{сетевая карта} \}, \\ \text{параметры} &\equiv \{ P1, P2, P3 \}, \\ \text{параметры_источники} &\equiv (\lambda (v: \{ \text{сетевая карта} \}) / \\ &(v \in \{ \text{сетевая карта} \} \Rightarrow \{ P1, P2, P3 \}) /), \\ \text{особенности} &\equiv \{ \text{host} \}, \\ \text{мз} &\equiv \{ \text{малая, средняя, большая} \}, \\ \text{вз} &\equiv (\lambda (v: \{ P1, P2, P3 \}) / v \in \{ P1, P2, P3 \} \Rightarrow \{ \text{малая,} \\ &\text{средняя, большая} \}) /), \\ \text{причины отклонений} &\equiv \{ \text{переполнение буфера сетевой} \\ &\text{карты} \} \end{aligned}$$

где P1, P2, P3 – величины определяемые формулами (6).

Приведем предложение, которое определяет число периодов динамики введенных параметров:

$$\begin{aligned} \text{чпд} &\equiv (\lambda (v: \langle \text{переполнение буфера} \\ &\text{сетевой карты, P1} \rangle, \\ &\langle \text{переполнение буфера сетевой карты, P2} \rangle, \\ &\langle \text{переполнение буфера сетевой карты, P3} \rangle) / (\pi \\ &(1,v) = \text{переполнение буфера сетевой карты} \ \& \ \pi(2,v) \in \\ &\{ P1, P2, P3 \} \Rightarrow 2) /). \end{aligned}$$

Данное предложение может интерпретироваться следующим образом при отклонении переполнение буфера сетевой карты, число периодов динамики параметров P1, P2, P3 и P3 равно 2.

Предложение:

$$\begin{aligned} \text{моменты} &\equiv (\lambda (v: \{ P1, P2, P3 \}) / (v=P1 \Rightarrow \{12,60\}), \\ &(v=P2 \Rightarrow \{12,60\}), (v=P3 \Rightarrow \{36,60\}) /) \end{aligned}$$

вводит, что параметр P1 наблюдался через 12, и 60 минут после начала наблюдения, параметр P2 наблюдался через 12 и 60 минут после начала наблюдения, параметр P3 наблюдался через 36 и 60 минут после начала наблюдения.

Предложения:

$$\begin{aligned} P1 &\equiv (\lambda (v: \{12, 60\}) / (v \in \{12\} \Rightarrow \text{малая}), \\ &(v=60 \Rightarrow \text{большая}) /), \end{aligned}$$

$$\begin{aligned} P2 &\equiv (\lambda (v: \{12, 60\}) / (v \in \{12\} \Rightarrow \text{малая}), \\ &(v=60 \Rightarrow \text{большая}) /), \end{aligned}$$

$$\begin{aligned} P3 &\equiv (\lambda (v: \{36, 60\}) / (v \in \{36\} \Rightarrow \text{малая}), \\ &(v=60 \Rightarrow \text{большая}) /), \end{aligned}$$

описывают результаты наблюдения за производными величинами от параметров ipInDelivers, ipInReceives, tcpInSegs определяемых соотношением (6). То есть, через 12 минут после начала наблюдения для параметра P1 было получено значение малая, через 60 минут – большая. Через 12 минут после начала наблюдения для параметра P2 было получено значение малая, через 60 минут – большая. Через 36 минут после начала наблюдения для параметра P3 было получено значение малая, через 60 минут – большая.

6. Выводы

В данной работе введены основные понятия и разработана модель онтологии диагностики распределенных информационных систем. Основные понятия, введенные в рамках онтологии, позволяют описывать предметную область, учитывая:

- особенности работы служб в рамках распределенной информационной системы,
- определять знания в зависимости от периодов динамики параметров,
- фиксировать для заданной причины отклонений выбранное множество диагностических параметров и возможные значения параметров.

В результате при разработке интеллектуальных систем для диагностики распределенных информационных систем может применяться концепция проектирования интеллектуальных систем, все компоненты (данные, знания, решатель задач с пользовательским интерфейсом) которого имеют единые принципы для их формирования, доступа и модифицирования.

Литература

1. Lewis, L. Service Level Management for Enterprise Networks [Text] / L. Lewis. – Norwood: Artech House Inc., 1999.

2. IBM Tivoli Enterprise Console, International Business Machines Corporation [Electronic resource] / Available at: <http://www-306.ibm.com/software/tivoli/products/enterprise-console/>
3. HP OpenView Event Correlation Services, Hewlett Packard Corporation [Electronic resource] / Available at: <http://www.managementsoftware.hp.com/products/ecs/>
4. Netcool, Micromuse Incorporated [Electronic resource] / Available at: <http://www.micromuse.com>
5. Bachant, J. R1 revisited: four years in the trenches [Text] / J. Bachant, J. McDermott // The AI Magazine., Fall. – 1984. – Vol. 5, Issue 3. – P. 21–32.
6. Compton, P. Maintaining an expert system [Text] / P. Compton, K. Horn, R. Quinlan, L. Lazarus // Proceedings of the fourth Australian Conference on Applications of Expert Systems, 1988. – P. 110–129.
7. Appleby, K. Yemanja – A Layered Event Correlation Engine for Multi-domain Server Farms. In Proceedings of the Seventh IFIP/IEEE [Text] / K. Appleby, G. Goldszmidt, M. Steinder // International Symposium on Integrated Network Management, Seattle, Washington, USA, 2001. – P. 329–344. doi: 10.1109/inm.2001.918051
8. W3C, OWL Web Ontology Language Reference. W3C Recommendation [Text] / M. Dean, G. Schreiber (Eds.). – 2004. – Available at: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>
9. Hanemann, A. A Framework for Failure Impact Analysis and Recovery with Respect to Service Level Agreements [Text] / A. Hanemann, M. Sailer, D. Schmitz // In Proceedings of the IEEE International Conference on Services Computing (SCC 2005), Orlando, Florida, USA. – 2005. – Vol. 2. – P. 49–56. doi: 10.1109/scc.2005.10
10. Грибова, В. В. Проблемы создания жизнеспособных интеллектуальных систем и методы их решения [Текст] / В. В. Грибова, А. С. Клещев // International Journal "Information Technologies & Knowledge". Bulgaria. Sofia: ITHEA. – 2011. – Vol. 5, Issue 3. – P. 250–258.
11. Клещев, А. С. Системный анализ при автоматизации интеллектуальной профессиональной деятельности [Текст] / А. С. Клещев, Е. А. Шалфеева // XIII Национальная конференция по искусственному интеллекту с международным участием «КИИ-2012». Труды конференции, т.2. Белгород: Изд-во БГТУ, 2012. – С. 128–135.
12. Клещев, А. С. Управление интеллектуальными системами [Текст] / А. С. Клещев, В. В. Грибова // Известия РАН. Теории и системы управления. – 2010. – № 6. – С. 122–137.
13. Клещев, А. С. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия „онтология” [Текст] / А. С. Клещев, И. Л. Артемьева // НТИ. – 2001. – Сер. 2, № 2. – С. 20–27.
14. Kleshchev, A. S. A mathematical apparatus for ontology simulation. Specialized extensions of the extendable language of applied logic [Text] / A. S. Kleshchev, I. L. Artemjeva // Inf. Theories and Appl. – 2005. – Vol. 12, Issue 3. – P. 265–271.
15. Нестеренко, С. А. Модель онтологии анализа тенденций для диагностики сложных вычислительных систем [Текст] / С. А. Нестеренко, П. М. Тишин, А. С. Маковецкий. – Современные информационные и электронные технологии. – Одесса, 2013. – С. 42–43.
16. Нестеренко, С. А. Разработка модели онтологии диагностики сервис-ориентированных сетевых структур на основе много-сортного языка прикладной логики [Текст] / С. А. Нестеренко, П. М. Тишин, А. С. Маковецкий // Электротехнические и компьютерные системы. – 2012. – № 07 (83). – С. 102–108.