УДК 004.9

С.А. НЕСТЕРЕНКО, П.М. ТИШИН, А.С. МАКОВЕЦКИЙ

Национальный политехнический университет «ОНПУ», Украина

РАЗРАБОТКА ФОРМАЛИЗОВАННОГО ЯЗЫКА ДИАГНОСТИКИ СОСТОЯНИЙ НА ОСНОВЕ ДЕСКРИПЦИОННОЙ ЛОГИКИ

В работе рассмотрен вопрос разработки формализованного языка представления знаний для диагностики состояний в сложной вычислительной системе с использованием CIM-метамодели и CDM-схемы. Предложено описание основных понятий CIM-метамодели и CDM-схемы на языке ALC. Указаны конструкторы, которые позволят описать любую CIM-схему или CIM-модель. Приведен пример описания понятия. Предложенный язык ALC может быть использован в процессе разработки экспертных систем контроля и диагностики состояний сложных вычислительных систем на основе применения методов искусственного интеллекта.

Ключевые слова:*CIM, CDM, дескрипционная логика, базы знаний, распределённые вычислительные системы.*

Введение

Растущая сложность, разнородность и динамика развития в течение жизненного цикла, присущие распределенным системам и информационным услугам, в частности корпоративным сетям, требуют использования все более сложных технологий управления, координации и интеграции для обеспечения необходимого уровня функциональности, производительности и надежности вычислительных систем.

Диагностика является одним из важнейших компонентов систем управления корпоративными сетями. Диагностические услуги используются для решения задач поддержания доступа к узлам сети, локализации неисправностей, восстановления после сбоя, обеспечения целостности во время загрузки и обеспечения заданного уровня надежности функционирования корпоративной сети.

Из имеющихся технологий, именно те решения, которые связаны с парадигмой агентных вычислений, имеют больше возможностей для взаимодействия в открытой среде и обладают лучшими возможностями для решения задач контроля и диагностики сложных распределенных систем. Технология многоагентных систем представляет ряд новых возможностей в области сетевых операций, таких как формально-семантический уровень представления знаний, автоматические рассуждения и способности к обучению, языки высокого уровня для связи и протоколов, оболочки автоматизированного ведения переговоров, для достижения поставленной цели или рационального принятия решений.

Одним из подходов, применяемых при разработке многоагентных систем, является формальный подход на основе онтологий. Онтологии могут быть определены одним из языков задания онтологий, таких как RDFS или OWL. При этом OWL в настоящее время фактически является стандартом для описания онтологий. OWL имеет три диалекта OWL Lite, OWL DL и OWL Full. При этом OWL DL эквивалентен одному из языков дескрипционной логики (ДЛ). Интерес к OWL онтологиям объясняется тем, что существующие редакторы OWL онтологий обеспечивают хороший баланс выразительных возможностей и вычислимости.

С другой стороны стандартным формализмом для моделирования управления информацией в распределенных вычислительных системах является Common Information Model (CIM) [1]. CIM разработана DMTF и в контексте ее WBEM предложений [2], предназначена для обеспечения концептуального представления управляемой среды. Необходимость точной семантики при описании CIM-схем отмечалась, в частности в работе [3].

Известны работы, в которых для решения рассматриваемого вопроса применялись различные подходы, в том числе на основе дескрипционных логик [4]. Авторы предлагают ввести формальное представление знаний с использованием ДЛ, для представления концептуальных объектов общей диагностической модели (Common Diagnostic Model (CDM)). CDM - это архитектура и методология для отображения системного диагностического инструментария через CIM-стандартные интерфейсы.

Это решение позволяет автоматически вести рассуждение о моделях, основанных на CIM-концептах, как на этапе проектирования, так и на этапе контроля функционирования корпоративных сетей.

1. Дескрипционные логики

В качестве формального описания ДЛ предлагается использовать язык *ALC* (*attributive language with complement*) [5], который является одним из базовых языков описания дескрипционных логик. В основе *ALC*, как и в основе ДЛ, лежат понятия «концепт» и «роль», которые задают множество индивидов и бинарные отношения между ними соответственно. Относительно предикатной логики концепты могут рассматриваться как одноместные, а роли - как двуместные предикаты. Для описания некоторой предметной области с помощью *ALC* вводятся атомарные концепты, на основе которых, используя конструкторы, можно получить более сложные описания реальных объектов. Существуют многочисленные расширения логики *ALC* дополнительными конструкторами для построения концептов, ролей, а также дополнительными видами аксиом в TBox. Далее в описании будем использовать символы A и R для описания атомарных концептов и атомарных ролей соответственно, символы C и D - для описания составных концептов. Выражения *top* и *bottom* обозначают универсальные концепты, включающие в себя все понятия предметной области и пустой концепт соответственно. Семантика ДЛ задается путем интерпретации ее атомарных концептов как множеств объектов (индивидов), выбираемых из некоторого фиксированного множества предметной области (домена), а семантика атомарных ролей - как множеств пар индивидов, т.е. бинарных отношений на домене.

Формально, интерпретацией I называется пара,

$$\mathrm{I} = (\Delta^{\mathrm{I}}, \bullet^{\mathrm{I}}), \qquad (1)$$

состоящая из непустого множества Δ^{I} (так называемый домен), а также интерпретирующей функции $\bullet^{\mathrm{I}}$, которая сопоставляет каждому атомарному концепту A некоторое подмножество $\mathrm{A}^{\mathrm{I}} \subseteq \Delta^{\mathrm{I}}$, а каждой атомарной роли R – некоторое подмножество $\mathrm{R}^{\mathrm{I}} \subseteq \Delta^{\mathrm{I}} \times \Delta^{\mathrm{I}}$.

В табл. 1 определены конструктора понятий и ролей, которые используются для создания предложений, описывающих рассматриваемую предметную область. Интерпретирующая функция I распространяется на составные концепты и роли *ALC* согласно правилам, изложенным в соответствующем столбце табл. 1.

Таблица 1

Конструкторы понятий и ролей

Конструктор	Синтакс	Семантика
Атомарный концепт	A	$\mathrm{A}^{\mathrm{I}} \subseteq \Delta^{\mathrm{I}}$
Пересечение	$\mathrm{C} \sqcap \mathrm{D}$	$\mathrm{C}^{\mathrm{I}} \cap \mathrm{D}^{\mathrm{I}}$
Квантор всеобщности	$\forall$R.C	$\left\{ \mathrm{x} \mid \forall \mathrm{y} : \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \to \mathrm{C}^{\mathrm{I}}(\mathrm{y}) \right\}$
Квантор существования	$\exists$R.top $\exists$R.C	$\left\{ \mathrm{x} \mid \exists \mathrm{y} : \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \right\}$ $\left\{ \mathrm{x} \mid \exists \mathrm{y} : \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \wedge \mathrm{C}^{\mathrm{I}}(\mathrm{y}) \right\}$
Концепт отрицание	$\neg$C	$\Delta^{\mathrm{I}} \setminus \mathrm{C}^{\mathrm{I}}$
Номиналы	$\mathrm{a}_1 \cdots \mathrm{a}_\mathrm{n}$	$\mathrm{a}_1^{\mathrm{I}} \cdots \mathrm{a}_\mathrm{n}^{\mathrm{I}}$
Объединение	$\mathrm{C} \sqcup \mathrm{D}$	$\mathrm{C}^{\mathrm{I}} \cup \mathrm{D}^{\mathrm{I}}$
Ограничение	$\geq$ nR $\leq$ nR = nR	$\left\{ \mathrm{x} \mid \#\left\{ \mathrm{y} \mid \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \right\} \geq \mathrm{n} \right\}$ $\left\{ \mathrm{x} \mid \#\left\{ \mathrm{y} \mid \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \right\} \leq \mathrm{n} \right\}$ $\left\{ \mathrm{x} \mid \#\left\{ \mathrm{y} \mid \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \right\} = \mathrm{n} \right\}$
Условное ограничение	$\geq$ nR.C $\leq$ nR.C = nR.C	$\left\{ \mathrm{x} \mid \#\left\{ \mathrm{y} \mid \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \wedge \mathrm{C}^{\mathrm{I}}(\mathrm{y}) \right\} \geq \mathrm{n} \right\}$ $\left\{ \mathrm{x} \mid \#\left\{ \mathrm{y} \mid \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \wedge \mathrm{C}^{\mathrm{I}}(\mathrm{y}) \right\} \leq \mathrm{n} \right\}$ $\left\{ \mathrm{x} \mid \#\left\{ \mathrm{y} \mid \mathrm{R}^{\mathrm{I}}(\mathrm{x},\mathrm{y}) \wedge \mathrm{C}^{\mathrm{I}}(\mathrm{y}) \right\} = \mathrm{n} \right\}$
Транзитивная роль	R^+	$\bigcup_{\mathrm{n} \geq 1} \left(\mathrm{R}^{\mathrm{I}} \right)^{\mathrm{n}}$
Обратная роль	R^-	$\left\{ (\mathrm{x},\mathrm{y}) \in \Delta^{\mathrm{I}} \times \Delta^{\mathrm{I}} \mid \mathrm{R}^{\mathrm{I}}(\mathrm{a},\mathrm{b}) \right\}$

Имеется неформальное соглашение об именовании получающихся при этом логик: обычно путем добавления к имени логики букв, соответствующих добавленным в язык конструкторам. Наиболее известными расширениями являются: $\mathcal{F}$, $\mathcal{N}$, $\mathcal{Q}$, $\mathcal{I}$, $\mathcal{O}$, $\mathcal{H}$, $\mathcal{S}$, $\mathcal{R}$. Такая система конструкторов реализует $\mathrm{ALCQHI}_{R^+}(D^-)$ дескрипционную логику. С другой стороны, полученная логика использована в сервере логического вывода RacerPro. RacerPro - это система представления знаний, которая реализует высокооптимизированные таблицы исчислений для выражения дескрипционной логики. Она предоставляет услугу рассуждений с множеством концептов (T-boxes) и множеством ролей (A-boxes). При этом используется базовая *ALC* логика, дополненная ограничениями кардинальности ролей, иерархичностью ролей, инверсией ролей, транзитивностью ролей [6].

2. CIM-метамодель

CIM-метамодель (далее CIM-метасхема), представляет собой базис, на котором описываются CIM-схемы и CIM-модели. CIM-метасхема описывает мета-элементы, имеющие атрибуты и связи. CIM-метасхема описывается при помощи четырех понятий языка UML (Unified Modeling Language): UML-класс, UML-обобщение, UML-свойство и UML-ассоциация, и может быть выражена в виде диаграммы UML.

Для представления CIM-элементов при помощи ДЛ предлагается рассмотреть их как концепты и роли, а также ввести специальные аксиомы, представленные в табл. 2.

С помощью приведенных в таблице аксиом описываются все элементы CIM-метасхемы. Так, например, элемент *CIM_Class*, согласно [1], должен расширять элемент *CIM_NamedElement* и быть связанным

Таблица 2

Представление CIM-элементов в ДЛ.

CIM-элементы	Концепты и роли	Введённые аксиомы ДЛ
Класс C	Концепт C	
Атрибут a у C типа T	Бинарная роль	$C \sqsubseteq \langle = 1a \rangle \sqcap \exists a.T$
Ключевой атрибут a у C	Бинарная роль a	$C \sqsubseteq \langle = 1a \rangle \sqcap \exists A.D \sqcap \langle \le 1A^- \rangle$
Атрибут a у C типа T	Бинарная роль a	$C \sqsubseteq \langle \ge 1a \rangle \sqcap \forall a.T$
Атрибут a в диапазоне $(n_i, \cdots, n_j)$	Бинарная роль a	$C \sqsubseteq \langle \ge n_i a \rangle \sqcap \langle \le n_j a \rangle \sqcap \forall a.T$
Зависимость A	Бинарная роль A Роли R1 и R2	$\top \sqsubseteq \forall A.C_2 \sqcap \forall A^-.C_1$ $C_1 \sqsubseteq \forall A.C_2 \sqcap \langle \ge n_i A \rangle \sqcap \langle \le n_j A \rangle$ $C_2 \sqsubseteq \forall A^-.C_1 \sqcap \langle \ge m_i A^- \rangle \sqcap \langle \le m_j A^- \rangle$ $A \sqsubseteq R_1,\ R_1 \sqsubseteq A,\ A^- \sqsubseteq R_2,\ R_2 \sqsubseteq A^-$
N-арная ассоциация	Концепт A Роли Ar, R1 … Rn	$A \sqsubseteq \exists R_1.C_1 \sqcap \cdots \sqcap \exists R_n.C_n \sqcap \langle \le 1R_1 \rangle \sqcap \cdots \sqcap \langle \le 1R_n \rangle$ $C_i \sqsubseteq \forall R_i^-.A \sqcap \langle \ge n_i R_i^- \rangle \sqcap \langle \le n_j R_i^- \rangle$ $i = 1, \cdots, n$
Бинарная ассоциация	Концепт A Роли Ar, R1 и R2	$A \sqsubseteq \exists R_1.C_1 \sqcap \exists R_2.C_2 \sqcap \langle \le 1R_1 \rangle \sqcap \langle \le 1R_2 \rangle$ $C_1 \sqsubseteq \forall R_1^-.A \sqcap \langle \ge m_i R_1^- \rangle \sqcap \langle \le m_j R_1^- \rangle$ $C_2 \sqsubseteq \forall R_2^-.A \sqcap \langle \ge n_i R_2^- \rangle \sqcap \langle \le n_j R_2^- \rangle$
Наследованные отношения (частичные и не различные)		$C_i \sqsubseteq C,\ i = 1, \cdots, n$
Наследованные отношения (частичные и различные)		$C_i \sqsubseteq C,\ i = 1, \cdots, n\ C_i \sqsubseteq \neg C_j,\ \forall i \ne j$
Наследованные отношения (общие и не различные)		$C_i \sqsubseteq C,\ i = 1, \cdots, n\ C \sqsubseteq \bigsqcup_{i=1}^{n} C_i$
Наследованные отношения (общие и различные)		$C_i \sqsubseteq C,\ i = 1, \cdots, n\ C_i \sqsubseteq \neg C_j,\ \forall i \ne j$ $C \sqsubseteq \bigsqcup_{i=1}^{n} C_i$

ассоциациями: *PropertyDomain* с *CIM_Property*, *MethodDomain* с *CIM_Method*, *ReferenceRange* с *CIM_ReferenceType*, *Generalization* с *CIM_Class* и *DefiningClass* с *CIM_Instance*. С учетом ограничения множественности, на ДЛ *CIM_Class* будет описан следующим образом:

$$\begin{aligned}&\mathrm{CIM\_Class} \sqsubseteq \mathrm{CIM\_NamedElement} \sqcap \\ &\neg\left(\mathrm{CIM\_TypedElement} \sqcup \mathrm{CIM\_Trigger} \sqcup \mathrm{CIM\_Schema} \sqcup \mathrm{CIM\_Qualifier}\right) \sqcap \\ &\forall \mathrm{PropertyDomain}.\mathrm{CIM\_Property} \sqcap \forall \mathrm{MethodDomain}.\mathrm{CIM\_Method} \sqcap \\ &\forall \mathrm{ReferenceRange}.\mathrm{CIM\_ReferenceType} \sqcap \forall \mathrm{Generalization}^{-}.\mathrm{CIM\_Class} \sqcap \\ &\left\langle \leq 1 \mathrm{Generalization}^{-} \right\rangle \sqcap \forall \mathrm{Generalization}.\mathrm{CIM\_Class} \sqcap \forall \mathrm{DefiningClass}.\mathrm{CIM\_Instance}\end{aligned} \quad (2)$$

Аналогично описываются все остальные CIM-метаэлементы:

CIM_NamedElement, CIM_TypedElement, CIM_Type, CIM_PrimitiveType, CIM_ReferenceType, CIM_Schema, CIM_Class, CIM_Property, CIM_Method, CIM_Parameter, CIM_Trigger, CIM_Indication, CIM_Association, CIM_Reference, CIM_QualifierType, CIM_Qualifier, CIM_Flavor, CIM_Instance, CIM_InstanceProperty, CIM_Value.

Таким образом, в данной работе в дальнейшем будут использоваться приведенные выше конструкторы, которые позволят описать любую CIM-схему или CIM-модель, в соответствии с CIM-стандартом.

3.CDM-модель.

Общая диагностическая модель (CDM), отображающая системный диагностический инструментарий через CIM-стандартные интерфейсы.

Таким образом, в работе используются заданные в табл. 3 соотношения при определении элементов CDM-модели через введенные концепты и роли на языке ДЛ. Это позволило создать базу знаний, основу сигнатуры которой составляют атомарные концепты роли, определяющие соответствующие элементы CDM-модели. Описание обязательных атомарных концептов приведено в табл. 3.

Таблица 3

Обязательные концепты CDM модели

CIM_AvailableDiagnosticService	Ассоциативная связь диагностической услуги, которая может быть выполнена по отношению к управляемому элементу.
CIM_ConcreteJob	Класс, представляющий конкретную задачу. Используется клиентской стороной для наблюдения и контроля за выполнением диагностической услуги.
CIM_DiagnosticCompletionRecord	Запись, содержащая итоговую информацию работы услуги.
CIM_DiagnosticLog	Журнал диагностики, агрегирующий все диагностические записи (существует несколько легитимных механизмов протоколирования)
CIM_DiagnosticServiceRecord	Используется в качестве сообщений-отчетов диагностических услуг, сообщая результаты, ошибки, предупреждения и статусы.
CIM_DiagnosticTest	Класс, который представляет диагностическую услугу, разработанный для проверки и наблюдения за поведением устройства, которое причастно к неисправности на каком-либо уровне системы.
CIM_ElementSoftwareIdentity	Связь диагностической услуги с информацией о версии
CIM_HostedService	Предназначен для связи конкретного диагностического теста и конкретной компьютерной системы, в пределах которой он работает, а также конкретной справочной услуги и конкретной компьютерной системы, на которую она распространяется.
CIM_LogManagesRecord	Связывает журнал с его записями
CIM_OwningJobElement	Связывает диагностическую услугу с конкретными задачами
CIM_RegisteredProfile	Идентифицирует Diagnostics Profile для определения клиентской стороной совместимости конкретной диагностической услуги.
CIM_ServiceAffectsElement	Связь диагностической услуги с любым управляемым элементом, на который она воздействует.
CIM_ServiceAvailableToElement	Связывает диагностическую услугу с его информацией справочной услуги.
CIM_SoftwareIdentity	Класс, использующийся для представления версии диагностической услуги.
CIM_UseOfLog	Связь журнала с управляемым элементом (устройство или диагностическая услуга), информация о котором хранится в журнале.

К необязательным концептам разработанной базы знаний относятся: *CIM_AffectedJobElement,, CIM_CorrespondingSettingDataRecord, CIM_DiagnosticCompletionRecord, CIM_DiagnosticLog, CIM_DiagnosticServiceCapabilities, CIM_DiagnosticServiceRecord, CIM_DiagnosticSettingData, CIM_DiagnosticSettingDataRecord, CIM_DiagnosticTest, CIM_ElementCapabilities, CIM_ElementSettingData, CIM_ElementSoftwareIdentity, CIM_HelpService, CIM_HostedService, CIM_JobSettingData, CIM_LogManagesRecord, CIM_OwningJobElement, CIM_RecordAppliesToElement, CIM_RegisteredProfile, CIM_ServiceAffectsElement, CIM_ServiceAvailableToElement, CIM_ServiceComponent, CIM_SoftwareIdentity, CIM_UseOfLog*.

В качестве примера, приведем также описание класса *CIM_DiagnosticTest* и ассоциации *CIM_HostedService*.

Согласно [7], класс*CIM_DiagnosticTest* содержит свойства: *SystemCreationClassName, SystemName, CreationClassName, Name, ElementName, Characteristics, OtherCharacteristicsDescriptions*, и метод *RunDiagnosticService()*.

Данное утверждение в ДЛ выглядит так:

$$
\begin{aligned}
&\mathrm{CIM\_DiagnosticTest} \sqsubseteq \mathrm{CIM\_Class} \sqcap \\
&\exists \mathrm{hasSystemCreationClassName.SystemCreationClassName\_DT} \sqcap \\
&\langle \leq 1 \mathrm{hasSystemCreationClassName} \rangle \sqcap \exists \mathrm{hasSystemName.SystemName\_DT} \sqcap \\
&\langle \leq 1 \mathrm{hasSystemName} \rangle \sqcap \exists \mathrm{hasCreationClassName.CreationClassName\_DT} \sqcap \\
&\langle \leq 1 \mathrm{hasCreationClassName} \rangle \sqcap \exists \mathrm{hasName.Name\_DT} \sqcap \langle \leq 1 \mathrm{hasName} \rangle \sqcap \\
&\exists \mathrm{hasElementName.ElementName\_DT} \sqcap \langle \leq 1 \mathrm{hasElementName} \rangle \sqcap \\
&\exists \mathrm{hasCharacteristics.Characteristics\_DT} \sqcap \langle \leq 1 \mathrm{hasCharacteristics} \rangle \sqcap \\
&\forall \mathrm{hasOtherCharacteristicsDescriptions.OtherCharacteristicsDescriptions\_DT} \sqcap \\
&\langle \leq 1 \mathrm{hasOtherCharacteristicsDescriptions} \rangle \sqcap \\
&\exists \mathrm{hasRunDiagnosticService.RunDiagnosticService\_DT} \sqcap \langle \leq 1 \mathrm{hasRunDiagnosticService} \rangle
\end{aligned}
\tag{3}
$$

Выводы

Научная новизна работы состоит в разработке формализованного языка представления знаний для диагностики состояний в сложной вычислительной системе с использованием CIM моделей и CDM схем.

Практическая ценность предложенного формализованного языка вытекает из возможности его использования в процессе разработки экспертных систем контроля и диагностики состояний сложных вычислительных систем на основе применения методов искусственного интеллекта. Использование многоагентного подхода и заложенные в языке возможности формального вывода диагностических правил позволяют автоматизировать процесс контроля сложных распределенных вычислительных систем, реализуемых в виде корпоративных компьютерных сетей.

Литература

1. Common Information Model (CIM) Infrastructure. Version 2.6.0. Specification, DMTF Standard [Text]. – Distributed Management Task Force. – 2010.

2. Web-Based Enterprise Management (WBEM): Technical report [Text]. – Distributed Management Task Force. – 2003.

3. Westerinen, A. What is policy and what can it be? [Text] / A. Westerinen // Proceedings of the IEEE Policy 2003 Conference. IEEE Computer Society Press. – 2003. – P. 345 – 348.

4. Enabling Automatic Reasoning on CIM-based Management Platforms [Text] / F. Alonso, R. Fernandex, S. Frutos, J. Soriano // World Academy of Science, Engineering and Technology 14. –2006. – P. 123 – 129.

5. Schmidt-Schauss, M. Attributive concept descriptions with complements [Text] / M. Schmidt-Schauss, G. Smolka // Artificial Intelligence, 48:1–26.–1991. – P. 222 – 228.

6. RacerPro User's Guide. Version 1.9 [Text]. – Racer Systems GmbH & Co. KG. – 2005.

7. Diagnostics Profile. Version: 2.0.0. Specification, DMTF Standard [Text]. – Distributed Management Task Force. – 2010.

Поступила в редакцию 23.02.2012

Рецензент: д-р техн. наук, проф. А.А. Баркалов, Зеленогурский университет, Зелена Гура, Польша.

РОЗРОБКА ФОРМАЛІЗОВАНОЇ МОВИ ДІАГНОСТИКИ СТАНІВ НА ОСНОВІ ДЕСКРИПЦІЙНОЇ ЛОГІКИ

С.А. Нестеренко, П.М. Тішин, О.С. Маковецький

В роботі розглянуто питання розробки формалізованої мови представлення знань для діагностики станів в складній розрахунковій системі з використанням дескрипційної логіки - CIM-метамоделі та CDM-схеми. Запропоновано опис основних понять CIM-метамоделі та CDM-схеми на мові ALC. Вказані конструктори, що дозволяють описати будь-яку CIM-схему чи CIM-модель . Наведено приклад опису поняття. Запропонована мова ALC може бути використана в процесі розробки експертних систем контролю та діагностики станів складних розрахункових систем на основі застосування методів штучного інтелекту.

Ключові слова: CIM, CDM, дескрипційна логіка, бази знань, розподілені обчислювальні системи.

DEVELOPMENT FORMAL LANGUAGE FOR DIAGNOSTIC STATES BASED ON DESCRIPTION LOGIC

S.A. Nesterenko, P.M. Tishin, A.S. Makovetskiy

The paper deals with the issue concerning the development of formal knowledge representation language for the diagnostics of states in a complex computing system with CIM meta schema and CDM model. The description of main terms of CIM meta schema and CDM model with th ALC language is proposed. The constructors, which allows describing any CIM meta schema and CDM model are specified. The example of term description is presented. The language ALC can be used during the development process of expert control system and diagnostics of states of complex computational systems based on methods of artificial intelligence.

Key words:CIM,CDM,description logic, knowledge bases, distributed computing systems.

Нестеренко Сергей Анатольевич – д-р техн. наук, профессор, зав.кафедры компьютерных интеллектуальных систем и сетейНационального политехнического университета «ОНПУ», Одесса, Украина.

Тишин Пётр Метталинович – канд. физ.-мат. наук, доцент кафедры компьютерных интеллектуальных систем и сетей Национального политехнического университета «ОНПУ», Одесса, Украина.

Маковецкий Александр Сергеевич – аспирант кафедры компьютерных интеллектуальных систем и сетей, ст. преп. кафедры информационных систем Национального политехнического университета «ОНПУ», Одесса, Украина.