OSTIS-2016
(Open Semantic Technologies for Intelligent Systems)

УДК004.8

ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ЗНАНИЙ ХРАНЯЩИХСЯ В МНОГОМЕРНЫХ ДАННЫХ ПОСРЕДСТВОМ РАСШИРЕННОГО БАЗОВОГО СЕМАНТИЧЕСКОГО ГИПЕРГРАФА

Шарипбай А.А., Барлыбаев А.Б., Сабыров Т.С.

Евразийский Национальный Университет им. Л.Н.Гумилева,
г. Астана, Республика Казахстан

Sabyrov.Talgat@mail.ru

В данной работе рассматриваются вопросы представления и обработки знаний. Главной новизной работы является вложенность знаний, представление знаний посредствомрасширенного базового семантического гиперграфа(РБСГ). Построена математическая модель представления знаний. Описаны технология хранения, обработки знаний. Представленаклиентскаячастьпредставлениязнаний.
Ключевые слова: глобалы; многомерные массивы; гиперграфы; семантическая сеть; дейкстра.

Введение

Одним из самых важных и сложных вопросов искусственного интеллекта является вопрос представления и обработки знаний. Основные модели представления знаний являются: продукционные, фреймовые, логические и семантические сети. Используя любую из этих моделей можно разработать интеллектуальную систему. Поэтому перед началом разработки стоит вопрос об выборе необходимой модели [PaoloGiudici, DavidHeckerman, JoeWhittaker].

Первую продукционную модель предложил Пост в 1943 году. Основанна на правилах, позволяет представить знание в виде предложений типа «Если (условие), то (действие)». Продукционная модель обладает тем недостатком, что при накоплении достаточно большого числа (порядка нескольких сотен) продукций они начинают противоречить друг другу [Ishida, Toru.].

Фрейм – способ представления знаний в искусственном интеллекте, представляющий собой схему действий в реальной ситуации. Первоначально термин «фрейм» ввёл Марвин Минский в 70-е годы XX века для обозначения структуры знаний для восприятия пространственных сцен. Является моделью абстрактного образа, минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса [MatthiasStrobbe , OlivierVanLaere, BartDhoedt, FilipDeTurck, PietDemeester].

Основная идея подхода при построении логических моделей представления знаний – вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода. В основе логических моделей представления знаний лежит понятие формальной теории [CHRISTINE W.CHAN].

Семанти́ческая сеть — информационная модель предметной области, имеющая вид ориентированного графа. В семантической сети роль вершин выполняют понятия базы знаний, а дуги (причем направленные) задают отношения между ними. Таким образом, семантическая сеть отражает семантику предметной области в виде понятий и отношений [IsaacLera, CarlosJuiz, RamonPuigjaner].

При обработке знаний обычно используются методы поиска решений на основе исчисления предикатов (правила модус-понус и т.д., конъюнкции дизъюнкции и отрицания и т.п.) [YukselUckan]. Также используются прямой и обратный вывод в экспертных системах продукционного типа (стратегия поиска в глубину, стратегия поиска в ширину, разбиение на подзадачи, a-b-алгоритм и т.п.) [BirgitVogel-Heuser, AlexanderFay, InaSchaefer, MatthiasTichy]. Все чаще стали использоватся метод обработки знаний в интеллектуальных системах с фреймовым представлением (демоны, присоединенные

процедуры, механизм наследования) [JahangirKarimi, M.K.Zand].

Очень интересным подходом было бы использование многомерных массивов как хранилища для базы знаний. Многомерные массивы это массивы, элементами которых являются массивы. Определение многомерного массива должно содержать информацию о типе, размерности и количестве элементов каждой размерности. Элементы многомерного массива располагаются в памяти в порядке возрастания самого правого индекса.

Переменные как локальные, так и глобальные могут существовать в виде простых или индексированных структур. Глобальные переменные или глобалы, являясь хранимыми данными, создают основу так называемого прямого доступа [Иванчева Н.А., Иваньчева Т.А.]. Глобалы это структуры данных, как правило, многомерные, которые хранятся в базе данных и могут обрабатываться в многопользовательской среде различными процессами. Многомерность данных реализована через индексы, поэтому говорят об индексированных переменных.

Глобальные данные сохраняются в B*-деревьях. Дерево, которое имеет одинаковое число подуровней в каждом своем поддереве, называется сбалансированным (balancedtree, отсюда и B-дерево). Дерево, у которого каждый ключ указывает на блок данных, содержащий требуемую запись, называется B*-деревом. Оно делает возможной интеграцию области указателей и области данных. B*-деревья можно рассматривать как сеть состоящая из графов.

Согласно [ЗЫКОВА.А.] гиперграф, $H(V, E)$ есть пара, где V – множество вершин $V = \{v_i\}$, $i \in I = \{1,2,\dots,n\}$, а E – множество ребер $E = \{e_j\}$, $j \in J = \{1,2,\dots,m\}$; каждое ребро представляет собой подмножество V. Вершина v и ребро e называются инцидентными, если $v \in e$. Для $v \in V$ через $d(v)$ обозначается число ребер, инцидентных вершине v; $d(v)$ называется степенью вершины v. Степень ребра e – число вершин, инцидентных этому ребру, обозначается через $r(e)$. Гиперграф H является r-однородным, если все его ребра имеют одинаковую степень r.

Онтология – это комплексный и детально формализованное представление о определенном домене с использованием концептуальных фреймворков состоящего из экземпляров, концептов (классов), атрибутов (свойств), функций (операции), аксиом (фактов) и связей.

Для построения онтологической модели будет использоваться расширенный базовый семантический гиперграф (РБСГ). Узлы в этих графах представляют семантические признаки (свойства и функции) объектов или субъектов, а дуги представляют отношения между ними.

Отметим, что структура РБСГ аналогична парадигме объектно-ориентированного программирования. Поэтому РБСГ могут быть использованы для описания прикладного программного обеспечения, которое может ответить на вопросы пользователей о базе знаний. Концепты в структуре гиперграфа описаны на деревьях, которые преобразуются в математические формулы.

Есть несколько статей, где описывается семантический граф [Lian R., Goertzel B., Ke S., O'Neill J., Sadeghi K., Shiu S., Wang D., Watkins O., Yu G.]описывает модель семантического гиперграфа как «гипер-граф основанный на семантической сети», который может представить более сложные семантические сети и более эффективные структуры данных для хранения знании в репозиториях.

Веса вершин БСГ: $K = \{ka\}, a \in A = \{1,2,\dots b\}$, где $ka = \{Sa,Va,Ea\}$, $Sa = \bigcup_{j=1,2,\dots}^{ks} s_j^a$ – множество свойств класса, $Va = \bigcup_{j=1,2,\dots}^{kv} e_j^a, j \in ka$ – множество экземпляров класса, $Ea = \bigcup_{j=1,2,\dots}^{ke} e_j, j \in ka$ – множество семантических дуг, которые инцидентны классу, ks – количество свойств класса, kv – количество экземпляров класса, ke – количество семантических дуг, которые инцидентны классу. Вершина-экземпляр класса может быть представлена в виде тройки $vi = \{ki,Si,Ei\}$, где ki – родительский класс, т.е. $vi \in ki, j$ где $Si = \bigcup_{j=1,2,\dots}^{ks} s_j$ множество экземпляров, $Ei = \bigcup_{j=1,2,\dots}^{ke} e_j \in vi$ – множество семантических дуг, которые инцидентны экземпляру, ks – количество свойств экземпляра класса, ke – количество семантических дуг, которые инцидентны экземпляру классу.

Эту тройку, ka = {Sa,Va,Ea}, мы поменяли на пятерку ka = {Sa,Fa,Ia,Va,Ea}, где Fa - множество функции класса, Ia – множество инкапсуляции в классе. Вершина-экземпляр класса может быть представлена в виде пятерке vi = {ki,Si,Fa,Ia,Ei}, где Fi – множество функций экземпляров, Ia – множество инкапсуляций экземпляра.

Расширение семантического гиперграфа имеет тот же смысл, что и расширение контекстно-свободной грамматики до атрибутной грамматики. То есть если G = (N, T, P, S) это контекстно-свободная грамматика, то атрибутная грамматика, определяется как AG = (N, T, P, S, AS, AI, R), где N – множество нетерминалов, T – множество терминалов (непересекающихся с N), P – множество правил, S – начальный нетерминал, AS – конечное множество синтезированных атрибутов, AI – конечное множество наследственных атрибутов (непересекающихся с AS), R – конечное множество семантических правил.

Здесь атрибутная грамматика совершенно новый математический инструмент, который позволяет описывать не только структуру языковых единиц, но их атрибуты (семантические свойства). Это хорошо известный классический научный результат Дональда Эрвина Кнута. Вот как следует понимать расширение семантического гиперграфа, мы

добавили два новых символов, которые позволяют описать множество функций класса и набор классов инкапсуляции. Но мы не имеем имя для расширенного семантического гиперграфа.

Знания записанные, в глобале, мы будем структурировать и обрабатывать как онтологию, которая основана на РБСГ.

1. Представление знаний

1.1. Описание использованных технологий и структурданных

Глобалы это структуры данных в ООСУБД IntersystemsCache`, как правило, многомерные, которые хранятся в базе данных и могут обрабатываться в многопользовательской среде различными процессами[WolfgangKirsten, MichaelIhringer, MathiasKühn, BernhardRöhrig.]. Для оценки всех значений глобалов, разработаем каталог электронных устройств. Для этого введем глобал ^device, который на первом уровне имеет артикул ^device (article), а на втором уровне глобал имеет уровень свойств устройств ^device (article,property), уровень свойства устройств может содержать n-количество подуровней свойств устройств ^device (article,property,subproperty,...,n). Также на третьем уровнем артикула могут быть функции ^device(article,*func*). Этот уровень определяется ключевой строкой *func*. Все подуровни *func* содержат в себе функции объекта (в данном случае устройства) ^device(article,*func*,function). Аналогично уровню свойства устройств, уровень функций может содержать n-количество подуровней функций устройств ^device(article,*func*,function, subfunction,...,n). Тем самым мы использовали многомерность данных предоставленную глобалами.

Таблица 1 – Примеры записей в глобале*device*

Узел	Значение
^device(111587)	“”
^device(111587,"name")	"ASUS GL752VW
^device(111587,"type")	"pc"
^device(111587,"type","subtype")	"laptop"
^device(111587,"processor")	"Intel Core i7"
^device(111587,"ram")	"12gb"
^device(111587,"os")	"Windows 10"
^device(111587,"video")	"GeForce GTX960M"
^device(111587,"*func*","run")	w 'Hello'"
^device(111587,"*func*","run1")	w 'Hello 1'"
^device(100997)	“”
^device(100997,"display")	"sensory,IPS"
^device(100997,"name")	"3Q Meta RC7802F"
^device(100997,"os")	"Android"
^device(100997,"price")	37000
^device(100997,"ram")	"1gb"
^device(100997,"type")	"mobile"
^device(100997,"type","subtype")	"tablets"
^device("rel",111587,100997)	"SameCompany/MadeOfIron"

После того как мы указали свойства и функции устройств, нам необходимо связать эти устройства по артикулам. Для этого используется ключевая строка “rel” ^device(rel, article1, article2). В уровнях этой строки содержатся артикулы связанных устройств, а в значениях этих уровней будут содержаться названия связей, разделенные символом “/”.

^device("rel",111587,111588)=“SameCompany/ MadeOfIron”

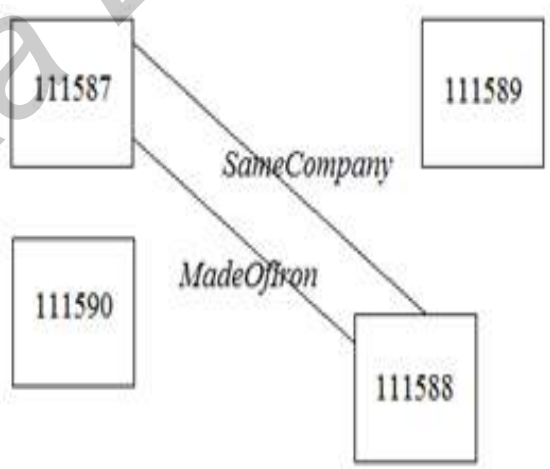

Рисунок 1 – Связи между устройствами

Функции представляются в глобале в виде записи:

^device(111587,"*func*","run") = "w 'Hello'"

Значения функций описаны ниже в таблице 2.

Команда	Действие	Сокращение
set	присвоитьзначение	s
if ...	выполнить все команды справа по строке - если ...	i ...
do	выполнитьподпрограмму	d

for ...	выполнять все команды справа по строке многократно (цикл)	f ...
quit	остановитьциклилипр ограмму	q
kill	уничтожитьпеременн уюилиузел	k
new	обьявить новые переменные с ограниченной областью	n
xecute	выполнит строку текста как отдельную программу	x
merge	обьединениеузлов и глобалей	m
$data	0 или 1 или 10 или 11	$d
$get	замену, если не существует такойпереиенной	$g
$select	одинизвариантов	$s
$length	(длинуиличисло_вхо ждений)+1	$l
$piece	возвращаетчастьстрок и	$p
$extract	возвращаетчастьстрок и	$e
$find	(номерсимвола в строке)+1	$f
$translate	перекодированнуюстр оку	$tr
$justify	округляетчисла	$j
$order	индекс узла дерева справа или слева на этом уровне	$o
$query	полный адрес ветви дерева справа/слева от текущей	$q
$qs	частьадресаветви (индекс)	$qs
$ql	длину полного адреса ветви дерева (размерность адреса)	$ql

Таким образом, мы создали глобалы, в которых хранятся знания об устройствах. Эти глобалы мы будем собирать в структурированные фреймы в семантической сети в нашей базе знаний об устройствах. На рисунке 2 мы соберем наши глобалы в сеть.

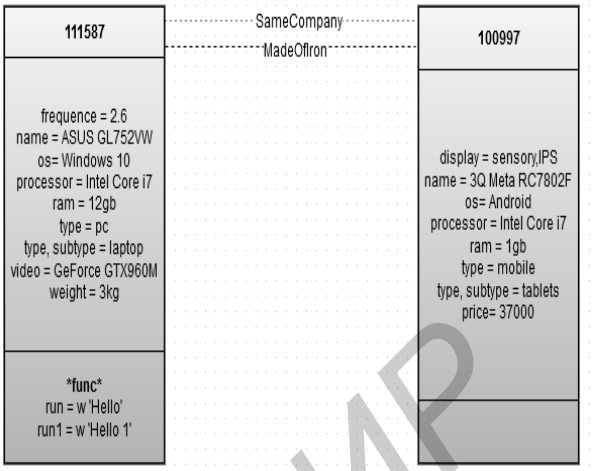

Рисунок 2 – Представление знаний об устройствах в структурной форме.

1.2. Описание гиперграфа и математической модели данных

Гиперграф является парой $H(V, E)$, но так как мы в знаниях используем помимо свойств и отношений функций, а также имеет свою внутреннюю вложенность данный вид гиперграфа нам не подходит. Поэтому мы предлагаем расширить гиперграф за счет добавления функций до расширенного базового семантического гиперграфа (РБСГ) [AltynbekSharipbayevand AlibekBarlybayev].

Будем использовать $H(A, P, F, R)$, где H – РБСГ, A – Id концепта, P – свойства, F – функции, R – отношения, семантические дуги. A = {A1, A2,…,An}, P = {P1,P2, …,Pn}, F = {F1,F2, …,Fn}, R = {R1,R2, …,Rn}.

$$P_{nm} = P_1^{1,2,\dots,m} + P_2^{1,2,\dots,m} + \dots + P_n^{1,2,\dots,m} \quad (1)$$

$$F_{nm} = F_1^{1,2,\dots,m} + F_2^{1,2,\dots,m} + \dots + F_n^{1,2,\dots,m} \quad (2)$$

Для поиска в графах мы использовали алгоритм дейкстра.
A – множество вершин графа
R - множество рёбер графа
w - вес (длина) всех ребер. В данном случае константа 1
a - вершина, расстояния от которой ищутся
U - множество посещённых вершин
d[u] - по окончании работы алгоритма равно длине кратчайшего пути из a до вершины u
l[u] - по окончании работы алгоритма содержит кратчайший путь из a в u
Присвоим $d[a] \leftarrow 0, p[a] \leftarrow 0$
Для всех $u \in A$ отличных от a
присвоим $d[u] \leftarrow \infty$
Пока $\exists v \notin U$
Пусть $v \notin U$ – вершина с минимальнымd[u]
занесем v в U
Для всех $u \notin U$ таких, что $\forall u \in R$
если $d[u] > d[v] + w$ то
изменим $d[u] \leftarrow d[v] + w$

изменим l[u] ← l[v], u

2. Программная реализация

2.1. Обработка данных

Для программной реализаций мы записали тестовые данные устройств в глобал ^device. Отображение можно увидеть в портале управления системой

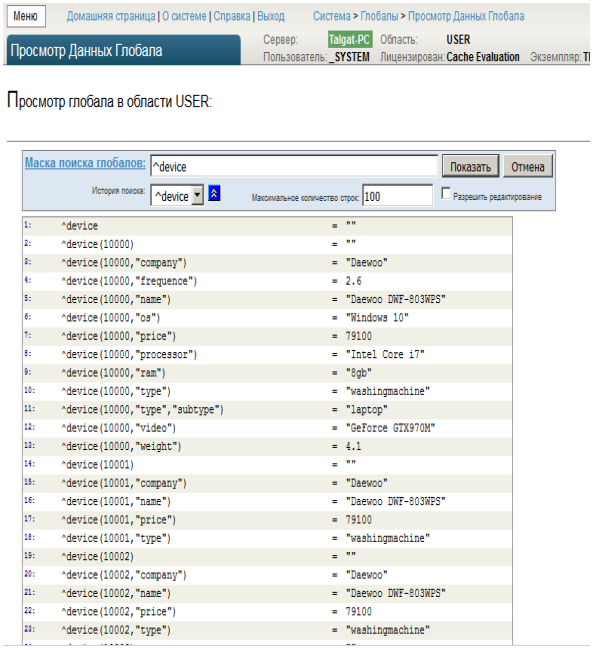

Рисунок 3 – Отображение глобалов

С помощью языка программирования Cache ObjectScript (COS) мы собрали данные глобала во фреймы и перевели их в данные формата JavaScriptObjectNotation (JSON) для вывода данных в клиентскую часть [JSON].

```
{
  "^device(58783)": {
    "^device(58783,\"name\")": "Data switch KVM D-Link DKVM-4K",
    "^device(58783,\"price\")": "10460",
    "^device(58783,\"type\")": "network"
  },
  "^device(75009)": {
    "^device(75009,\"interface\")": "usb",
    "^device(75009,\"name\")": "Defender Accent 930, Black, USB",
    "^device(75009,\"price\")": "2059",
    "^device(75009,\"type\")": "keyboard"
  },
  "^device(77115)": {
    "^device(77115,\"frequence\")": "20-16000h",
    "^device(77115,\"length\")": "1.37m",
    "^device(77115,\"name\")": "ACME MK-200",
    "^device(77115,\"price\")": "803",
    "^device(77115,\"type\")": "earphone"
  }
}
```

Рисунок 4 – Фрагмент данных JSON

Данные в виде фреймов были выведены в браузере. Для этого мы использовали язык программирования JavaScript с фреймворкомAngularJS (рисунок 5)

Рисунок 5 – Отображения данных в виде фреймов и их связи

2.2. Реализация пойска

Как мы упомянули ранее, мы использовали поиск методом Дейкстра. Блок схема алгоритма работы нашей программы показан на рисунке 6.

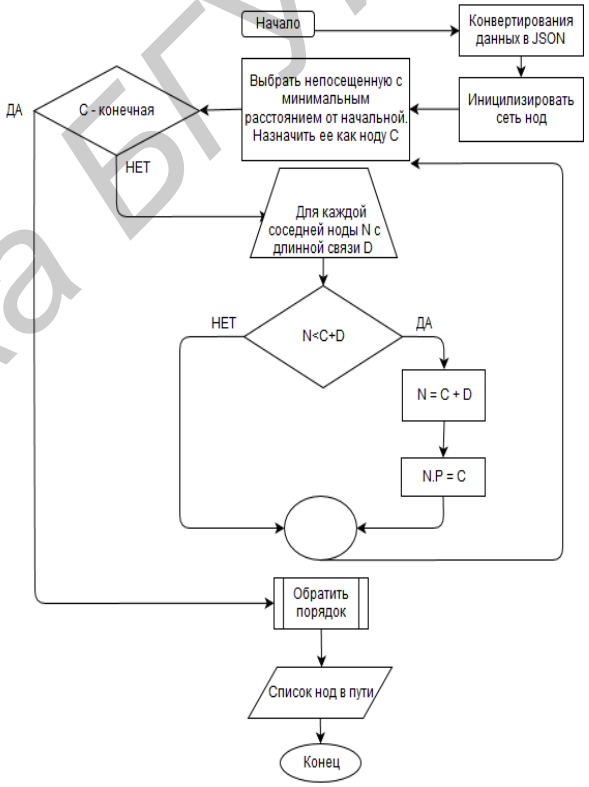

Рисунок 6 – Блок схема алгоритма поиска

В программе как входная данная пишется название вершины. Программа находит кратчайшие пути на каждые связанные вершины по вышеуказанному алгоритму. Результат показан на рисунке 7

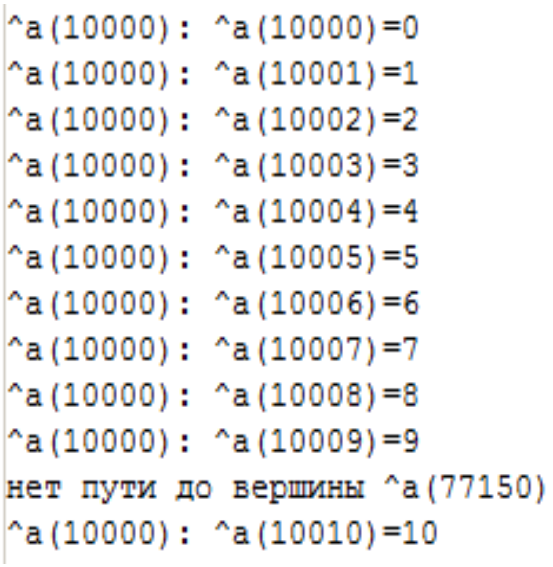

Рисунок 7 – Результат работы программы

Заключение

Описан метод представления и обработки знаний посредством РБСГ. Дальнейшая работа будет посвящена nlp, новому алгоритму обработки вложенных знаний, будут проведены эксперименты над производительностью обработки знаний.

Библиографический список

[PaoloGiudici, DavidHeckerman, JoeWhittaker.]Statistical Models for Data Mining.Data Mining and Knowledge Discovery. July 2001, Volume 5, Issue 3, pp 163-165.

[Ishida, Toru.]An optimization algorithm for production systems. Knowledge and Data Engineering, IEEE Transactions on (Volume:6 , Issue: 4). 549 – 558.

[Matthias Strobbe , Olivier Van Laere, Bart Dhoedt, Filip De Turck, Piet Demeester.] Hybrid reasoning technique for improving context-aware applications.Knowledge and Information Systems. June 2012, Volume 31, Issue 3, pp 581-616

[CHRISTINE W. CHAN.]FROM KNOWLEDGE MODELING TO ONTOLOGY CONSTRUCTION. International Journal of Software Engineering and Knowledge Engineering Vol. 14, No. 06, pp. 603-624 (2004)

[Isaac Lera, Carlos Juiz, Ramon Puigjaner.]Performance-related ontologies and semantic web applications for on-line performance assessment of intelligent systems. Science of Computer Programming, Volume 61, Issue 1, June 2006, Pages 27-37

[YukselUckan.]Knowledge representation using views in relational deductive data bases. Journal of Systems and Software, Volume 15, Issue 3, July 1991, Pages 217-232.

[Birgit Vogel-Heuser, Alexander Fay, Ina Schaefer, Matthias Tichy.]Evolution of software in automated production systems: Challenges and research directions. Journal of Systems and Software, Volume 110, December 2015, Pages 54-84.

[Jahangir Karimi, M.K. Zand.]Asset-based system and software system development – A frame-based approach. Information and Software Technology, Volume 40, Issue 2, 1998, Pages 69-78.

[Зыков А.А.]Гиперграфы // Успехиматематическихнаук. – 1974. – Т. 29. – Вып. 6. – С. 89–154.

[Иванчева Н.А., Иваньчева Т.А.]Постреляционная СУБД Caché. Новосибирск 2004, Новосибирский государственный университет, Высший колледж информатики, 83-93

[Wolfgang Kirsten, Michael Ihringer, Mathias Kühn, Bernhard Röhrig.]Object-Oriented Application Development Using the CachéPostrelational Database. Springer verlag. ISBN-13: 978-3540009603. 2003. 191-215

[AltynbekSharipbayevand AlibekBarlybayev.]Formal Models of the Intelligent Electronic University. International Journal of Information Technology & Computer Science (IJITCS) ISSN : 2091-1610. Volume 21 : Issue No : 1 / August 2015

[Lian R., Goertzel B., Ke S., O'Neill J., Sadeghi K., Shiu S., Wang D., Watkins O., Yu G.]Syntax-Semantic Mapping for General Intelligence: Language Comprehension as Hypergraph Homomorphism, Language Generation as Constraint Satisfaction. Artificial General Intelligence. Lecture Notes in Computer Science – 2012, Volume 7716 – P. 158-167.

[Zhen L., Jiang Z. Hy-SN:]Hyper-graph based semantic network. Knowledge-Based Systems – 2010, Vol 23, Issue 8 – P. 809-816.

[JSON]http://www.json.org/json-ru.html

PRESENTATION AND PROCESSING KNOWLEDGE STORED IN MULTIDIMENSIONAL DATA THOUGHT ENHANCED BASE SEMANTIC HYPERGRAPH

Sharipbay A.A.,Barlybayev A.B., Sabyrov T.S.

Eurasian National University L.N. Gumilyov, Astana, Kazakhstan

Sabyrov.Talgat@mail.ru

In this work we consider the presentation and processing knowledge. The main novelty of this research is the nesting of knowledge which is presented through enhanced base semantic hypergraph (EBSH). The mathematical model of presentation of knowledge is built. The storage technology and knowledge processing are described. The client side of knowledge presentation is presented.