

17. Zade L.A. Ponyatie lingvisticheskoy peremennoy i ego primeneniye k prinyatiyu priblizhennykh resheniy [The concept of a linguistic variable and its application to approximate reasoning]. Moscow: Mir, 1976, 165 p.
18. Prikladnye nechetkie sistemy [Applied fuzzy system]: Translation from Japanese K. Asai, D. Vatada, S. Iwai i dr. / Under ed. T. Terano, K. Asai, M. Sugeno. Moscow: Mir, 1993, 386 p.
19. Kofman A. Vvedeniye v teoriyu nechetkikh mnozhestv [Introduction to the theory of fuzzy sets]: Translated from French. Moscow: Radio i svyaz', 1982, 432 p.
20. Kureychik V.M., Lebedev B.K., Lebedev O.B., Chernyshev Yu.O. Adaptatsiya na osnove samoobucheniya [Adaptation based on learning]. Rostov-on-Don: RGASKhM GOU, 2004, 146 p.
21. Lebedev B.K. Adaptatsiya v SAPR: Monografiya [Adaptation in CAD: Monograph]. Taganrog: Izd-vo TRTU, 1999, 160 p.

Статью рекомендовал к опубликованию д.т.н., проф. Н.И. Витиска

Чернышев Юрий Олегович – Донской государственный технический университет; e-mail: myvnn@list.ru; 344000, г. Ростов-на-Дону, пл. Гагарина, 1; тел.: 88632738582; кафедра автоматизации производственных процессов; профессор.

Венцов Николай Николаевич – e-mail: vencov@list.ru; кафедра информационных технологий; доцент.

Панасенко Павел Александрович – Филиал военной академии связи (г. Краснодар); e-mail: we_panasenko_777@mail.ru; 350035, г. Краснодар, ул. Красина, 4; адъюнкт.

Chernyshev Yury Olegovich – Don State Technical University; e-mail: myvnn@list.ru; 1, Gagarina square, Rostov-on-Don, 344000, Russia; phone: +78632738582; the department of automation of productions; dr. of en. sc.; professor.

Ventsov Nikolay Nikolaevich – e-mail: vencov@list.ru; the department of information technologies; associate professor; cand. of eng. sc.

Panasenko Pavel Alexandrovich – Branch of the Military Academy of Telecommunications (Krasnodar); e-mail: we_panasenko_777@mail.ru; 4, Krasin street, Krasnodar, 350035, Russia; adjunct.

УДК 002.53:004.89

Ю.А. Кравченко, А.А. Новиков, В.В. Марков

МЕТОД СОЗДАНИЯ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ ГЛОССАРИЯ*

Рассмотрена разработка метода для создания онтологии предметной области из одного или нескольких глоссариев. Особенностью данного метода является параллельное построение онтологии и формирование полной терминологии предметной области. Метод позволяет определить основные этапы построения онтологии предметной области на основе итерационного отражения понятий глоссария в объектах онтологии. Данный процесс максимально автоматизирован, но на начальных этапах построения он должен вручную корректироваться разработчиками онтологии при помощи экспертов в данной предметной области. На этапе кластеризации термины глоссария группируются в кластеры на основе алгоритма кластеризации k-средних. Множество элементов глоссария разбивается на заранее известное количество кластеров k. Алгоритм заканчивает свою работу, когда кластеры больше не изменяются. Затем происходит добавление новых (неописанных в глоссарии) терминов и их определений. Для каждого кластера определяются отношения между всеми объектами, входящими в него. Предоставляется неформальное описание значения и особенности для каждой связи. На основе этих связей осуществляется поиск понятий, которые можно было бы выделить в подклассы кластера, в котором они находятся, и их группировка. Затем строится концептуальная модель

* Работа выполнена при финансовой поддержке РФФИ (проект № 13-07-00537).

онтологии, которая является обобщением представлений о данных. Результатом является отображение полученной онтологии в графическом виде посредством использования различных редакторов онтологий, например: *Ontolingua*, *OntoEdit*, *OilEd*, *WebOnto*, *ODE*, *Protégé*. Конечная онтология отражает в себе понятия с указанием точного определения и является законченной по отношению к конкретной предметной области.

Онтология предметной области; кластеризация; классификация; системы представления знаний.

Yu.A. Kravchenko, A.A. Novikov, V.V. Markov

METHOD OF CREATING A DOMAIN ONTOLOGY FROM GLOSSARY

*The article describes the development of a method to create the ontology of one or more glossaries. A feature of this method is the parallel construction of the ontology and the formation of the domain full terms. The method allows to identify the main stages of building ontology-based iterative reflection concepts glossary in the objects of the ontology. This process is automated, but in the early stages of building it must be adjusted manually using the ontology developers experts in the subject area. In step of clustering the glossary terms is grouped into clusters based on the algorithm of k-means clustering. The set of glossary elements is divided into a known k number of clusters. The algorithm finishes its work, when the clusters do not change. Then the addition of new (not described in the glossary) of terms and their definitions. For each clusters is define the relation between all objects in cluster. It provides an informal description of the values and characteristics for each relation. Then we get a conceptual model of ontology, which is a generalization of the concepts of data. The result is a display of the resulting ontology in a graphical form by using different ontologies editors e.g. *Ontolingua*, *OntoEdit*, *OilEd*, *WebOnto*, *ODE*, *Protégé*. The final ontology reflects the concepts, with indicating a precise definition and ontology is completed with respect to the specific subject area.*

Domain ontology; clustering; classification; knowledge representation system.

Введение. Онтологии предметной области (ПрО) в настоящее время находят основное применение в области построения поисковых систем, систем представления знаний, инженерии знаний и при решении задач семантической интеграции информационных ресурсов. Под онтологией понимается «формальная спецификация концептуализации, которая имеет место в некотором контексте предметной области» [1]. В свою очередь, концептуализация – представление предметной области через описание множества понятий (концептов) предметной области и связей (отношений) между ними.

Задача представления и интеграции информации и знаний связана с целым рядом подзадач разработки: баз знаний, содержащих блоки правил принятия решений и прецедентов; множеств объектных, онтологических, нечетких, семантических и аналитических моделей, реализующих процессы принятия решений; модулей выбора моделей и формирования решений на основе базы знаний, математического и имитационного моделирования.

Полученные ранее результаты не решают проблемы сложности идентификации и использования ключевой информации, что является препятствием обнаружения устойчивых междисциплинарных семантических отношений между объектами знания различных предметных областей. Основная цель работы – разработка метода создания онтологии, интегрирующей знания из разнородных предметных областей, на основе анализа глоссариев. Предлагаемые исследования базируются на гипотезе о том, что создание подобных моделей позволит адаптировать исследуемые информационные потоки к изменяющимся внешним условиям и интегрировать разнородные предметные области в рамках совместного решения задач управления знаниями в реальном времени.

На основе онтологических структур предполагается осуществлять преобразование знаний в хранилище данных, обеспечивающее возможность поддержки принятия решений с применением методов и алгоритмов биоинспирированного

поиска решений на основе семантической концентрации знаний. Несмотря на выраженную специфику предметных областей, онтология будет строиться как цепочка взаимосвязанных процессов накопления и обработки знаний из распределенных ресурсов, что позволит получить интегрированный характер интеллектуальной системы управления знаниями. Конкретным научным результатом исследования станет метод создания онтологии предметной области на основе глоссария. Будет предложена концепция разработки программного обеспечения для экспериментального исследования предложенного метода.

1. Постановка задачи. Рассмотрим модель онтологии предметной области, на основе которой могут быть построены описанные выше системы.

Модель онтологии может быть представлена кортежем [2]

$$O = \langle C, A, R \rangle,$$

где $C = \{c_i\}$ – множество понятий (концептов), образующих онтологию O , $i = \overline{1, I}$, т.е. $|C| = I$;

$A = \{a_1, \dots, a_d\}$ – множество атрибутов понятия c_i (d – количество атрибутов, описывающих данное понятие);

$R \subseteq C \times C$ – отношение непосредственного наследования.

Отношение R удобно задавать матрицей размером $I \times I$: если понятие c_k в данной онтологии непосредственно наследует понятию c_i , т.е. $(c_i, c_k) \in R$, то элемент $r_{ik} = 1$, в противном случае, если $(c_i, c_k) \notin R$, то $r_{ik} = 0$.

Дочерние понятия наследуют атрибуты, входящие в состав родительского понятия, расширяя его состав собственными атрибутами:

$$\exists (c_i, c_k) \in R \rightarrow A_i \subset A_k.$$

Онтология в общем случае поддерживает множественное наследование понятий в виде

$$\exists (c_j, c_k, c_l): r_{jk} = 1 \wedge r_{kl} = 1.$$

Корректно построенная онтология не должна содержать ложных отношений наследования, связывающих понятие с предком его же родительского понятия:

$$\exists (c_j, c_k, c_l): r_{jk} = 1 \wedge r_{kl} = 1 \wedge A_i = A_k \cap A_l.$$

Машинная модель понятия онтологии, используемая для построения систем, должна включать в себя поля, содержащие имя понятия, состав атрибутов понятия и родовидовые связи понятия.

$$O = \langle N, M, S, D \rangle,$$

где N – имя понятия; A – множество атрибутов понятия; S – множество родительских понятий (суперклассов); D – множество дочерних понятий (подклассов). При этом $M \subseteq A$.

Атрибут понятия, в свою очередь, характеризуется именем, типом и значением:

$$A = \langle N_A, T, V \rangle,$$

где N_A – имя атрибута; T – тип атрибута; V – значение атрибута. Атрибуты c_i понятий онтологии имеют определенный тип данных T . В качестве основных типов данных атрибутов онтологии можно выделить числовой, текстовый, логический и тип ссылки на объект.

Основным отношением, учитываемым при построении онтологии, является родовидовое отношение между понятиями (отношение гипоним-гипероним), на основе которого формируется таксономия понятий. Гипоним – понятие, выражающее частную сущность по отношению к другому, более общему понятию. Гипероним – слово с более широким значением, выражающее общее, родовое понятие, название класса (множества) предметов (свойств, признаков). Гипероним является результатом логической операции обобщения. Тогда как гипоним – ограничения.

Представление совокупности понятий предметной области и их отношений в основном реализуется в современных онтологических системах на основе модели семантической сети фреймов. Узлы сети представляют отдельные понятия предмет-

ной области, дуги – отношения между понятиями. Отдельное понятие в этой модели представляется фреймом, слоты которого содержат атрибуты понятия. Производные (дочерние) понятия наследуют атрибуты базовых (родительских) понятий. Рассмотрим метод создания онтологии ПрО из одного или нескольких глоссариев.

Фактически, процесс построения должен вручную корректироваться разработчиками онтологии при помощи экспертов в данной ПрО. Некоторые этапы очень сложно автоматизировать, так как требуется явная интерпретация терминов и понятий, например, в начальном этапе кластеризации, поэтому перевод должен контролироваться экспертами.

Описываемый метод дает возможность точного описания онтологии ПрО на основе глоссария. Главной особенностью является то, что при построении онтологии мы имеем возможность собрать полную терминологию интересующей предметной области. Однако, терминология не обязательно должна включать в себя понятия, являющиеся несущественными для рассматриваемой ПрО, в то время как онтология должна отражать в себе понятия, с указанием точного определения и быть законченной по отношению к предметной области. Другими словами, данный метод позволяет построить полную онтологию ПрО (т.е. без неопределенных объектов), начиная с неполного глоссария, в процессе добавляя в него новые определения.

Мы определяем глоссарий в виде списка (возможно, упорядоченных) лемм; каждая лемма состоит из термина и текстового определения, обеспечивающего смысл леммы; также в определениях могут появляться ссылки на другие леммы [3].

2. Описание метода. Этап 1. – Кластеризация. Задача кластеризации (или обучения без учителя) заключается в следующем. Имеется обучающая выборка $X_\ell = \{x_1, \dots, x_\ell\} \subset X$ и функция расстояния между объектами $\rho(x, x')$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ , а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X_\ell$ приписывается метка (номер) кластера y_i . Алгоритм кластеризации – это функция $a: X \rightarrow Y$, которая любому объекту $x \in X$ ставит в соответствие метку кластера $y \in Y$. Множество меток Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации [4]. Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин. Во-первых, не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд достаточно разумных критериев, а также ряд алгоритмов, не имеющих четко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты. Во-вторых, число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием. В-третьих, результат кластеризации существенно зависит от метрики ρ , выбор которой, как правило, также субъективен и определяется экспертом.

Для описываемого метода в начале необходимо сгруппировать термины глоссария в кластеры. Здесь используется модифицированный алгоритм кластеризации k -средних. Он разбивает множество элементов векторного пространства на заранее известное число кластеров k . То есть, алгоритм k -средних объединяет все точки данных так, что каждая точка x_i попадает в один и только один из k разделов. Можно отследить, какая точка находится в каком кластере, назначив каждой точке номер кластера. Точки с таким же номером кластера находятся в одном и том же кластере, в то время как точки с различными номерами кластера находятся в разных кластерах [5–8].

На первом шаге происходит присваивание данных. Каждой точке данных присваивается ее самый близкий представитель, при том что связи нарушаются произвольно. Это приводит к разделению данных.

Затем происходит перемещение «средних». Каждый представитель кластера перемещается к центру (т.е. среднее арифметическое) всех точек данных, присвоенных ему. Объяснение этого шага основано на наблюдении, что данное множество точек, единственный лучший представитель для этого множества (в смысле минимизации суммы квадрата Евклидова расстояния между каждой точкой и представителем) не что иное, как срединная точка данных. Именно поэтому представитель кластера часто взаимозаменяемо называют срединным элементом кластера или центроидом кластера.

Алгоритм сходится, когда присвоение (следовательно, и значения C_j) больше не изменяются. Целевая функция k -средних будет уменьшаться всякий раз, когда есть изменения в присвоении или шагах измерения, и сближение (сходимость в одной точке) гарантировано за конечное число итераций.

Улучшить работу алгоритма можно путем распараллеливания этапа распределения точек по кластерам. Если у нас имеется P процессоров, то мы можем создать P потоков. В этом случае исходное множество данных разбивается на P частей и каждый поток будет работать только со своим объемом информации, независимо от остальных данных.

Потенциальная проблема алгоритма – проблема «пустых» кластеров. При запуске k -средних, особенно с большим значением k или когда данные находятся в большом размерном пространстве, возможно, что в какой-то момент исполнения существует представитель кластера c_j , такой, что все точки x_i в пространстве ближе к некоторому другому представителю кластера, который не является c_j . Когда точки будут присвоены к ближайшему кластеру, j -му кластеру будут присвоены нулевые точки. Таким образом, кластер j является теперь пустым кластером. Стандартный алгоритм не принимает меры против пустых кластеров. В нашей модификации после каждой итерации происходит проверка на существование «пустых» кластеров. Если «пустые» кластеры существуют, то:

1. Алгоритм перезапускается (без учета последней итерации) и кластеры заново переформируются.
2. Если после перезапуска тот же кластер является «пустым», то он удаляется.

Входными данными здесь будет являться глоссарий. На выходе мы получаем таблицу сгруппированных по кластерам терминов. Она должна содержать в себе описание причин каждой группировки для кластеров.

Этап 2. – Добавление новых терминов. Если мы работаем с неполным глоссарием, то необходимо найти все термины, не являющиеся леммами. Добавить новые термины в глоссарий вместе с определением. В определении может присутствовать ссылка для описания термина. Затем следует распределить новые термины в ранее найденные кластеры [9, 10].

Входными данными на этом этапе будут исходный глоссарий и таблица кластеров. На выходе получаем обновленные глоссарий и таблицу. Во время данного шага исходный глоссарий расширяется новыми терминами.

Этап 3. – Идентификация отношений. Здесь осуществляется поиск отношений между терминами на основе их определений. Затем происходит оценка найденных отношений в рамках соответствующего кластера. Предоставляется неформальное описание значения и особенности для каждой связи (например, диапазон или отношение к предметной области).

Классическая задача прямой идентификации заключается в определении функциональной зависимости выходного сигнала y от входного – x ($y = F(x)$). Когда у исследователя нет непосредственного доступа к выходному сигналу, исполь-

зуют метод компараторной идентификации, позволяющей формулировать основные интеллектуальные выводы с помощью дедуктивного способа, исходя из физически имеющихся наблюдений.

Компараторная идентификация применяется для формального описания периферических механизмов интеллекта (восприятие, узнавание, понимание), обрабатываемые с ее помощью информационные объекты должны быть дискретными, конечными и детерминированными [9, 10].

На выходе получаем таблицу отношений. Так же как и таблица кластеров, она содержит в себе группы выявленных отношений для каждого кластера, а также описание их значений. Для каждого термина в кластере определяется синонимический ряд.

Этап 4. – Классификация. Классификация – упорядоченное по некоторому принципу множество объектов, которые имеют сходные классификационные признаки (одно или несколько свойств), выбранных для определения сходства или различия между этими объектами.

Методы классификации, можно разделить на несколько групп. По способу задания показателя качества классификации методы делятся на эвристические и оптимизационные. По способу объединения – на дивизимные, агломеративные и итеративные (рис. 1).

Эвристические алгоритмы основаны на опыте и интуиции человека. Показатель качества классификации, который необходимо обратить в экстремум, в этих алгоритмах в явном виде не задан. Эвристические алгоритмы реализуют процедуры, обладающие рациональным смыслом с точки зрения логики человека и приводящие во многих случаях к хорошим результатам на практике. К таким алгоритмам относятся, например, алгоритмы «Граф», «Спектр», «Форель».



Рис. 1. Методы классификации

К оптимизационным алгоритмам относятся методы классификации, в которых в явном виде задан показатель качества, который необходимо обратить в экстремум (максимум или минимум) по множеству допустимых разбиений. В отличие от алгоритмов первой группы, разбиения, получаемые оптимизационными алгоритмами классификации, являются наилучшими с точки зрения выбранного показателя качества. Выбор конкретного показателя зависит от специфики и ограничений решаемой задачи, а также принятых предложений. Следует отметить, что во многих случаях в эвристических алгоритмах показатель качества задан в неявном виде и они могут стать оптимизационными, если удастся его формализовать и сформулировать в явном виде.

В общем случае в любом оптимизационном алгоритме классификации можно выделить следующие элементы [11]:

- ◆ показатель качества классификации;
- ◆ ограничения;
- ◆ механизм поиска результирующего разбиения.

Ограничения в методах классификации в основном касаются типа исходных данных – множества допустимых разбиений, на котором ищется результирующее разбиение, и вида самого результирующего разбиения. Поиск результирующего разбиения осуществляется в соответствии с некоторым механизмом оптимизации. Это может быть механизм полного или частичного перебора, случайного перебора и т.д.

Агломеративные методы последовательно объединяют отдельные объекты в группы (классы), а дивизимные методы расчленяют группы на отдельные объекты. В свою очередь каждый метод классификации как объединяющего, так и разделяющего типа может быть реализован при помощи различных алгоритмов. Следует заметить, что как агломеративные, так и дивизимные алгоритмы трудоемки и их сложно использовать для больших совокупностей. Кроме того, результаты работы таких алгоритмов (их графическое изображение) трудно поддаются визуальному анализу [13].

В отличие от кластеризации, классификация относится к стратегии обучения с учителем, которое также именуют контролируемым или управляемым обучением [14].

На данном этапе происходит поиск понятий, которые можно было бы выделить в подклассы кластера, в котором они находятся, и их группировка. На выходе получаем обновленную таблицу понятий вместе с их разбиением на классы.

Этап 5. – Концептуальное моделирование. Концептуальная модель является собой обобщение представлений разных пользователей о данных. В связи с этим построение концептуальной модели, как правило, происходит в несколько этапов [13]:

1. На первом этапе происходит группировка найденных кластеров по областям. Здесь реализуется поиск абстрактных отношений между ними на основе обобщения ранее найденных связей.
2. Формирование каждой области в отдельности, начиная с подклассов, найденных на этапе 4.
3. Построение конечной модели, включающей в себя сгруппированные области и найденные отношения между ними.

На выходе получаем концептуальную модель онтологии. А именно графическое представление концептуальной структуры онтологии предметной области.

Этап 6. – Представление онтологии. Традиционным инструментом формирования онтологий являются редакторы онтологий. Существует целый ряд инструментов (Ontolingua, OntoEdit, OilEd, WebOnto, ODE), поддерживающих редактирование, документирование, визуализацию, импорт и экспорт онтологий, а также объединение и сравнение [15, 16, 17, 19]. Рассмотрим наиболее известные из них.

Protégé [18] – свободно распространяемый редактор, предназначенный для проектирования онтологий предметных областей. Редактор строит онтологии с использованием языка OWL. Такие инструментальные средства используются как для проектирования, так и для анализа онтологии, выполняя типовые операции, например:

- ◆ выравнивание – нахождение и установление соответствий в обеих онтологиях;
- ◆ отображение – нахождение семантических зависимостей между элементами разных онтологий;
- ◆ объединение – создание результирующей онтологии по двум другим.

PROMPT [20] является расширением системы Protégé и служит для объединения и группировки онтологий. PROMPT выводит пользователю список операций для объединения двух онтологий, а также список конфликтов и их возможные решения. «Пользователь выбирает необходимое действие, список формируется снова, пока не будет готова новая онтология» [20].

Также могут использоваться различные языки графического описания, например, унифицированный язык моделирования UML. С помощью него можно получить конечное представление иерархии классов и их свойств. Предыдущий этап должен быть формализован для того, чтобы объяснить, как кодировать определения, учитывая логические отношения.

Экспериментальные исследования. В данной статье представлен метод построения онтологии на основе одного или более (возможно неполных) глоссариев. Метод позволяет построить полную терминологию предметной области. Для проведения экспериментальных исследований за основу был взят глоссарий по дисциплине «Генетические алгоритмы и эволюционное моделирование». Для дальнейшего использования конечной онтологии, ее качество было проверено экспертами в данной предметной области. Для этого разработано web-приложение, которое позволяет контролировать процесс построения онтологии и корректировать ее структуру на каждом этапе.

На первом этапе случайным образом были выбраны центроиды для каждого кластера. Количество кластеров определяется заранее, путем анализа первоначального глоссария. В будущем предполагается задать механизм автоматического определения количества кластеров или генерировать его в процессе их формирования. После каждой итерации выполнялась проверка на наличие «пустых» кластеров. Центроиды в кластерах имеют мощность окрестности, так как, в самом начале центроид в каждом кластере выбирается случайным образом, и во время выполнения кластеризации он может быть заменен на другой объект в кластере с большим количеством связей и, следовательно, с большей мощностью окрестности. Алгоритм заканчивает свою работу, когда ни один из объектов не изменил своего местоположения. Для получения наиболее точной онтологии была проведена проверка результатов работы алгоритма кластеризации экспертами предметной области.

Этап добавления новых терминов будет доступен в том случае, если за основу будет взят неполный глоссарий. В нашем случае за основу был взят полный глоссарий, поэтому добавление новых терминов и определений не требуется, и данный этап был пропущен.

Для этапа классификации на основе полученных результатов кластеризации было описано формализованное представление для приведения текстовых определений понятий к их логическому виду. На основе определений были выявлены отношения между терминами предметной области.

На последнем этапе для представления онтологии в web ориентированном приложении был использован редактор Protégé, так как он доступен для свободного использования и имеет открытую, легко расширяемую архитектуру. Для упрощения реализации отображения результирующей онтологии в web-приложении использовался существующий унифицированный язык графического моделирования UML.

Автоматизация каждого этапа дает возможность сократить время построения онтологии, а их унификация позволяет применять описанный метод для различных предметных областей.

Заключение. В данной работе предложен метод построения онтологии предметной области на основе одного или нескольких глоссариев (которые могут быть неполными). Предложенный метод построения онтологии поддерживает динамическое формирование структуры входящих в состав онтологии ПрО моделей понятий и объектов, позволяя тем самым эффективно реализовать операции поэтапного формирования онтологии и редактирования ее структуры. Данная проблема является актуальной при построении поисковых систем, систем представления знаний, инженерии знаний и при решении задач семантической интеграции информационных ресурсов.

Весь процесс построения онтологии был разбит на 6 основных этапов, каждый из которых включает в себя различные механизмы преобразования и отражения начального набора данных в онтологии ПрО. Для этапа кластеризации использовался модифицированный алгоритм k -средних, разделяющий множество данных на k кластеров. По своей сути, алгоритм работает с помощью перебора в два этапа: (1) кластеризация всех точек данных в зависимости от расстояния между точкой и ее ближайшим представителем кластера и (2) переоценка представителей кластера.

Конкретные научные результаты проявляются в систематизации этапов построения онтологии предметной области, которые необходимо предпринять, для решения задачи получения полной онтологии ПрО без неопределенности за счет использования итерационного добавления объектов и обогащения начального глоссария новыми понятиями, относящимися к определенному кластеру.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Gruber T.R. A Translation Approach to Portable Ontology Specifications // Knowledge Acquisition. – 1993. – № 5 (2). – P. 199-220.
2. Кравченко Ю.А., Запорожец Д.Ю., Лежебоков А.А. Способы интеллектуального анализа данных в сложных системах // Известия КБНЦ РАН. – 2012. – № 3 (47). – С. 52-57.
3. Тузовский А.Ф., Чуриков С.В., Ямпольский В.З. Системы управления знаниями (методы и технологии) / Под общ. ред. В.З. Ямпольского. – Томск: Изд-во НТЛ, 2005. – 260 с.
4. Kravchenko Yu.A., Kureichik V.V. Knowledge management based on multi-agent simulation in informational systems // Conference proceedings. 8th IEEE International Conference “Application of Information and Communication Technologies – AICT 2014”. 15-17 October 2014, Astana, Kazakhstan. – P. 264-267.
5. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: Учеб. пособие. – М.: МГТУ им. Н.Э. Баумана, 2005. – 304 с.
6. Kureichik V.M., Rodzin S.I. Evolutionary algorithms: genetic programming // Journal of Computer and Systems Sciences International. – 2002. – Vol. 41, № 1. – P. 123-132.
7. Курейчик В.В., Сороколетов П.В. Концептуальная модель представления решений в генетических алгоритмах // Известия ЮФУ. Технические науки. – 2008. – № 9 (86). – С. 7-12.
8. Интеллектуальный анализ данных. Алгоритм кластеризации k-means. URL: <http://intellect-tver.ru/?p=265> (дата обращения: 14.05.2015).
9. Курейчик В.М. Особенности построения систем поддержки принятия решений // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 92-98.
10. Кравченко Ю.А., Бова В.В. Нечеткое моделирование разнородных знаний в интеллектуальных обучающих системах // Открытое образование. – 2013. – № 4 (99). – С. 70-74.
11. Искусственный интеллект. Методы классификации. URL: <http://www.aiportal.ru/articles/autoclassification/methods-class.html> (дата обращения: 14.05.2015).
12. Курейчик В.М. Особенности построения систем поддержки принятия решений // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 92-98.
13. Бова В.В., Курейчик В.В., Нужнов Е.В. Проблемы представления знаний в интегрированных системах поддержки управленческих решений // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 107-113.
14. Курейчик В.В., Родзин С.И. О правилах представления решений в эволюционных алгоритмах // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 13-21.
15. OntoEdit: Collaborative ontology development for the Semantic Web. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke // In Proc. of the Inter. Semantic Web Conference (ISWC 2002), Sardinia, Italia, June 2002.
16. ODE, WebODE [Электронный ресурс] URL: delicias.dia.fi.upm.es/webODE/ (дата обращения: 14.05.2015).
17. Ontolingua [Электронный ресурс] URL: <http://www.ksl.stanford.edu/software/ontolingua/> (дата обращения: 14.05.2015).
18. Protege. [Электронный ресурс]. URL: protege.stanford.edu/download/registered.html (дата обращения: 14.05.2015).

19. WebOnto [Электронный ресурс]. URL: <http://webonto.open.ac.uk> (дата обращения: 14.05.2015).
20. Noy N., Musen M. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping // Stanford Medical Informatics, Stanford Univ.

REFERENCES

1. Gruber T.R. A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, 1993, No. 5 (2), pp. 199-220.
2. Kravchenko Yu.A., Zaporozhets D.Yu., Lezhebokov A.A. Sposoby intellektual'nogo analiza dannykh v slozhnykh sistemakh [Methods data mining in complex systems], *Izvestiya KBNTs RAN* [Izvestiya of Kabardino-Balkar scientific centre of the RAS], 2012, No. 3 (47), pp. 52-57.
3. Tuzovskiy A.F., Chirikov S.V., Yampol'skiy V.Z. Sistemy upravleniya znaniyami (metody i tekhnologii) [The knowledge management system (methods and technologies)]. Under the General ed. V.Z. Yampol'skogo. Tomsk: Izd-vo NTL, 2005, 260 p.
4. Kravchenko Yu.A., Kureichik V.V. Knowledge management based on multi-agent simulation in informational systems, *Conference proceedings. 8th IEEE International Conference "Application of Information and Communication Technologies – AICT 2014". 15-17 October 2014, Astana, Kazakhstan*, pp. 264-267.
5. Bashmakov A.I., Bashmakov I.A. Intellektual'nye informatsionnye tekhnologii: Ucheb. Posobie [Intelligent information technologies: a tutorial]. Moscow: MGTU im. N.E. Baumana, 2005, 304 p.
6. Kureichik V.M., Rodzin S.I. Evolutionary algorithms: genetic programming, *Journal of Computer and Systems Sciences International*, 2002, Vol. 41, No. 1, pp. 123-132.
7. Kureychik V.V., Sorokoletov P.V. Kontseptual'naya model' predstavleniya resheniy v geneticheskikh algoritмах [A conceptual model of representation solutions in genetic algorithms], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 9 (86), pp. 7-12.
8. Intellektual'nyy analiz dannykh. Algoritm klasterizatsii k-means [Data mining. The clustering algorithm k-means]. Available at: <http://intellect-tver.ru/?p=265> (Accessed 14 May 2015).
9. Kureychik V.M. Osobennosti postroeniya sistem podderzhki prinyatiya resheniy [Features of decision making support system design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 92-98.
10. Kravchenko Yu.A., Bova V.V. Nechetkoe modelirovanie raznorodnykh znaniy v intellektual'nykh obuchayushchikh sistemakh [Fuzzy modeling heterogeneous knowledge in intelligent tutoring systems], *Otkrytoe obrazovanie* [Open Education], 2013, No. 4 (99), pp. 70-74.
11. Iskusstvennyy intellekt. Metody klassifikatsii [Artificial intelligence. Classification methods]. Available at: <http://www.aiportal.ru/articles/autoclassification/methods-class.html> (Accessed 14 May 2015).
12. Kureychik V.M. Osobennosti postroeniya sistem podderzhki prinyatiya resheniy [Features of decision making support system design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 92-98.
13. Bova V.V., Kureychik V.V., Nuzhnov E.V. Problemy predstavleniya znaniy v in-tegrirovannykh sistemakh podderzhki upravlencheskikh resheniy [Problems of knowledge presentation in management decision support of integrated systems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 7 (108), pp. 107-113.
14. Kureychik V.V., Rodzin S.I. O pravilakh predstavleniya resheniy v evolyutsionnykh algoritмах [On the rules for the submission decisions in evolutionary algorithm], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 7 (108), pp. 13-21.
15. OntoEdit: Collaborative ontology development for the Semantic Web. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke, In Proc. of the Inter. Semantic Web Conference (ISWC 2002), Sardinia, Italia, June 2002.
16. ODE, WebODE. Available at: <http://www.delicias.dia.fi.upm.es/webODE/> (Accessed 14 May 2015).
17. Ontolingua. Available at: <http://www.ksl.stanford.edu/software/ontolingua/> (Accessed 14 May 2015).

18. Protege. Available at: <http://www.protege.stanford.edu/download/registered.html> (Accessed 14 May 2015).
19. WebOnto. Available at: <http://webonto.open.ac.uk> (Accessed 14.05.2015).
20. Noy N., Musen M. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping, *Stanford Medical Informatics, Stanford Univ.*

Статью рекомендовал к опубликованию д.т.н., профессор М.М. Ошхунов.

Кравченко Юрий Алексеевич – Южный федеральный университет; e-mail: krav-jura@yandex.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент.

Новиков Антон Александрович – e-mail: anton.a.novikov@gmail.com; кафедра систем автоматизированного проектирования; аспирант.

Марков Владимир Васильевич – e-mail: v_v_mar@mail.ru; кафедра систем автоматизированного проектирования; доцент.

Kravchenko Yury Alekseevich – Southern Federal University; e-mail: krav-jura@yandex.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; associate professor.

Novikov Anton Alexandrovich – e-mail: anton.a.novikov@gmail.com; the department of computer aided design; postgraduate.

Markov Vladimir Vasilyevich – e-mail: v_v_mar@mail.ru; the department of computer aided Design; associate professor.

УДК 004.891

Н.А. Полковникова

ГИБРИДНАЯ ЭКСПЕРТНАЯ СИСТЕМА НА ОСНОВЕ ВЕРОЯТНОСТНО-ДЕТЕРМИНИРОВАННЫХ МОДЕЛЕЙ

Разработка и внедрение интеллектуальных гибридных экспертных систем является важным направлением повышения надёжности и эффективности технической эксплуатации сложных объектов и позволит использовать результаты мониторинга и автоматизировать процедуры решения эксплуатационных задач для информационной поддержки оператора. Концепция комплексного использования объекта управления с экспертной системой позволяет повысить эффективность результатов мониторинга и искусственную компетентность оператора для решения эксплуатационных задач. Рассмотрена разработка базы знаний гибридной экспертной системы с расчётом эталонных и текущих стохастических моделей по контролируемым параметрам, а также алгоритмов для автоматизированной интеллектуальной обработки информации на основе системного подхода для поддержки оператора в принятии правильных решений при эксплуатации сложных технических объектов на примере главного судового двигателя. Для реализации диагностико-прогнозирующих процедур в экспертной системе разработаны: модели, алгоритмы и программные модули, позволяющие на качественно новом уровне использовать оптимизационные методы и резервы фактического технического состояния для предотвращения отказов и увеличения межремонтного периода. Для формализации процедур эволюции данных в базе знаний экспертной системы разработана методика использования стохастических моделей различных типов и уровней: эталонных и текущих (адаптивных), локальных и интегральных, для получения которых требуется база данных измеряемых параметров, программное обеспечение и база данных моделей. Разработана модель и алгоритм многокритериальной оптимизации на основе эволюционных алгоритмов для поддержки оператора по выбору режима работы главного судового двигателя. Целью алгоритма является опре-