УДК 004.89
DOI: 10.15827/0236-235X.135.381-389
Дата подачи статьи: 03.06.21
2021. Т. 34. № 3. С. 381–389

Программная реализация модуля анализа данных на основе прецедентов для распределенных интеллектуальных систем

А.П. Еремеев [1], *д.т.н., профессор, eremeev@appmat.ru*
П.Р. Варшавский [1], *к.т.н., доцент, зав. кафедрой, VarshavskyPR@mpei.ru*
С.А. Поляков [1], *аспирант, PoliakovSerA@mpei.ru*

[1] *Национальный исследовательский университет «МЭИ», г. Москва, 111250, Россия*

В статье рассматриваются вопросы реализации модуля анализа данных на основе прецедентов (CBR, Case-Based Reasoning), позволяющего выполнять все этапы цикла обучения на основе прецедентов (CBR-цикла) для распределенных интеллектуальных систем. В настоящее время прослеживается устойчивая тенденция к широкому применению методов и средств интеллектуального анализа данных для решения различных прикладных задач. Все больше крупных компаний используют интеллектуальные системы и средства интеллектуального анализа данных для повышения эффективности своего бизнеса и сокращения расходов. В связи с развитием интернет-технологий и доступности облачных вычислений перспективным направлением в области искусственного интеллекта является создание распределенных интеллектуальных систем.

Распределенные интеллектуальные системы характеризуются распределением вычислительных и информационных ресурсов, что ведет к повышению адаптируемости, надежности, а также общего быстродействия системы ввиду возможности одновременно обрабатывать большие наборы данных. Распределенные интеллектуальные системы состоят из автономных узлов (агентов), которые могут действовать независимо друг от друга и асинхронно обмениваться информацией. Наличие в системе агентов, способных реализовать определенные интеллектуальные функции, характеризует тесную связь между распределенными интеллектуальными и многоагентными системами.

Главное внимание уделяется методу интеллектуального анализа данных, основанному на использовании накопленного ранее опыта в виде прецедентов. Указанный прецедентный метод (CBR-метод) позволяет решать новую задачу, используя (адаптируя) решение схожей уже известной задачи. Для решения задач интеллектуального анализа данных на основе прецедентов была разработана модульная прецедентная система (CBR-модуль), реализующая основные этапы CBR-цикла. Система позволяет работать с прецедентами, представленными в параметрическом и структурированном (на основе онтологий) видах. CBR-модуль представляет собой веб-приложение, реализованное на языке программирования Python 3.7.5 с использованием веб-фреймворка Flask и библиотеки Owlready2 для работы с онтологиями. CBR-модуль ориентирован на интеграцию в состав распределенных интеллектуальных систем для выполнения анализа данных на основе прецедентов.

Функционирование реализованного CBR-модуля было протестировано на примере задачи классификации на наборах данных, взятых из открытого репозитория Калифорнийского университета (UCI Machine Learning Repository).

Ключевые слова: *интеллектуальный анализ данных, прецедент, многоагентные системы, классификация данных.*

В области *искусственного интеллекта* (ИИ) актуальна проблема создания *распределенных интеллектуальных систем* (РИС), которые могут объединять (интегрировать) в себе различные методы ИИ, в частности, *интеллектуального анализа данных* (ИАД). РИС определяются тремя основными характеристиками – способами распределения задач между агентами, распределения полномочий и коммуникаций между агентами [1].

На сегодняшний день одним из перспективных подходов к ИАД является применение методов машинного обучения – класса методов, характерная черта которых – не прямое решение задачи, а обучение в процессе его поиска. В задачах машинного обучения широко используется поиск решений на основе прецедентов (Case-Based Reasoning – CBR).

В данной работе рассматриваются различные аспекты, касающиеся анализа современных подходов к созданию РИС, а также вопросы, связанные с разработкой программных средств для ИАД с использованием прецедентного подхода, ориентированных на их интегра-

цию в состав РИС в качестве одного из ее базовых компонентов.

Распределенные интеллектуальные системы

Распределенный ИИ является разделом ИИ, в основе которого лежат вопросы взаимодействия интеллектуальных агентов [2]. Распределенный ИИ тесно связан с теорией *многоагентных систем* (МАС) [1], поэтому будем рассматривать МАС как одну из разновидностей РИС.

Базовая концепция, лежащая в основе теории МАС, – понятие интеллектуального агента. В общем смысле агентом может быть любая сущность (чаще всего под агентом понимается некая компьютерная программа), способная воспринимать информацию и выполнять определенные действия. Интеллектуальным агентам присущи целеустремленность, обучаемость, социальность, независимость.

МАС определяется как сеть агентов, существующих в общей среде и взаимодействующих между собой для достижения тех или иных целей системы. Взаимодействие может осуществляться агентами либо прямым образом – путем обмена сообщениями, либо некоторым косвенным, когда одни агенты воспринимают присутствие других агентов через изменения во внешней среде, с которой они взаимодействуют. В МАС агенты имеют несколько важных характеристик [3]:

– автономность (агенты хотя бы частично независимы);

– ограниченность представления (ни у одного из агентов нет представления обо всей системе или система слишком сложна, чтобы знание о ней имело практическое применение для агента);

– децентрализация (нет агентов, управляющих всей системой).

МАС используются для решения проблем, которые сложно или невозможно решить с помощью одного агента или монолитной (полностью централизованной) системы.

Агенты могут обмениваться имеющейся информацией (полученными данными и знаниями), используя некоторый специальный язык (например, KQML и FIPA–ACL) и подчиняясь установленным правилам общения (протоколам) в системе [4].

Обычно МАС состоит из программных агентов и агентной платформы, которая поддерживает взаимодействие агентов. Широкое применение получили агентные платформы, построенные в соответствии с абстрактной архитектурой FIPA [5]. Примерами таких платформ, полностью поддерживающих архитектуру FIPA, являются JADE, JACK, Jadex и EMERALD, написанные на языке Java. Также существуют платформы на других языках программирования, поддерживающие стандарты FIPA. Примером такой платформы может служить написанная на Python агентная платформа PADE.

Агентная платформа является средой, населенной агентами; предоставляет агентам базовые сервисы, необходимые для их существования; реализует всю низкоуровневую инфраструктуру (не нужно писать весь код заново при создании очередной МАС); реализует определенные стандарты для обеспечения взаимодействия с другими платформами.

Однако возникают ситуации, когда использование существующих агентных платформ может быть неэффективным из-за достаточно сложного встраивания имеющейся платформы в архитектуру разрабатываемой МАС.

Пример архитектуры МАС для ИАД на основе прецедентов приведен в работе [6]. Блок управления в данной архитектуре представляет вариант агентной платформы, объединяющий компоненты, предназначенные для координации и поддержки работы основных компонентов МАС. Также имеется блок ИАД, который может включать в себя различные модули, в частности, CBR-модуль.

Рассуждения на основе прецедентов

Прецедент определяется как случай, имевший место ранее и служащий примером или оправданием для последующих случаев подобного рода [7]. Рассуждения на основе прецедентов (CBR, Case-Based Reasoning) – подход, позволяющий решить новую, неизвестную задачу, используя или адаптируя решение уже известной задачи, то есть ранее накопленный опыт решения подобных задач.

Для эффективного применения CBR-систем не требуется глубокий анализ предметной области, достаточно указать проблему и ее решение путем предоставления нескольких примеров аналогичных случаев и ссылок на некоторое сходство. Методы рассуждений на основе прецедентов активно применяются в таких областях, как юриспруденция, медицинская диа-

гностика, мониторинг и диагностика технических систем, банковское дело, бизнес, а также поиск решения в проблемных ситуациях и многих других. Моделирование рассуждений на основе прецедентов в ИС поддержки принятия решений реального времени для мониторинга и управления сложными объектами и процессами рассмотрено в работе [7].

Несмотря на множество различных реализаций CBR-систем, подход на основе прецедентов включает в себя базовый компонент – CBR-цикл, который обеспечивает:

– извлечение наиболее похожего прецедента (или прецедентов) для сложившейся ситуации из *базы прецедентов* (БП);

– повторное использование извлеченного прецедента для попытки решения текущей проблемы;

– пересмотр и адаптацию в случае необходимости полученного решения в соответствии с текущей проблемой;

– сохранение вновь принятого решения как части нового прецедента [7].

Способы представления прецедентов

Основные способы представления прецедентов можно разделить на следующие группы: параметрические, объектно-ориентированные, специальные (деревья, графы, логические формулы, онтологии и т.д.).

Для программной реализации CBR-модуля в качестве базовой используется параметрическая модель представления прецедентов, которая расширяется возможностью структурированного представления модели прецедентов с помощью онтологий.

Выбор параметрической модели в качестве базовой обусловлен тем, что большинство наборов данных из открытых репозиториев полностью или частично представлены в параметрическом виде, также параметрические БП характеризуются меньшими затратами на поддержание и сопровождение в отличие от БП на основе других методов представления прецедентов.

При использовании параметрической модели БП представляется в виде двух таблиц в БД. Первая таблица хранит прецеденты (набор параметров и решение), вторая содержит информацию обо всех параметрах прецедентов в БП, а именно: имя параметра, тип, область определения значений параметра, описание, единицы измерения, весовой коэффициент. Сведения о параметрах прецедентов позволяют получить более наглядное представление о значении параметров БП, задать степень важности параметра с помощью весовых коэффициентов, а также задать или получить области определения параметров (диапазоны). Области определения и веса в дальнейшем могут быть использованы в алгоритмах извлечения прецедентов и повышении эффективности и быстродействия БП.

Онтологическая модель БП определяется тройкой: $O = (X, R, Ф)$, где X – конечное множество концептов предметной области, которую представляет онтология; R – конечное множество отношений между концептами; Ф – конечное множество функций интерпретации, заданных на концептах и (или) отношениях.

Выбор онтологии для представления прецедентов обусловлен рядом важных достоинств, отличающих ее от других моделей представления знаний. Использование онтологии для представления прецедентов позволяет задать сложную структуру прецедента, включающую данные разных типов, и обеспечить естественность представления структурированных знаний и достаточно простое обновление их в относительно однородной среде.

В реализованном CBR-модуле модель БП представлена в виде иерархии концептов онтологии, а БП вместе с таблицей, описывающей параметры прецедентов из БП, хранятся в соответствующих таблицах БД.

Каждая онтология, представляющая модель БП, должна содержать концепт Case, описывающий прецедент и содержащий в себе концепты Situation и Solution. Situation содержит концепты, описывающие параметры прецедента. Данный концепт также должен содержаться в онтологии текущей ситуации при извлечении из БП. Solution соответствует классу, к которому относится прецедент из БП.

На рисунке 1 представлен пример модели структурированной БП в виде онтологии, построенной средствами Protégé (https://protege.stanford.edu/). В данном примере каждый прецедент в БП состоит из четырех параметров: *A*, *B*, *C*, *D*, причем параметр *D* является подклассом параметра *B*.

Способы извлечения прецедентов и повышения эффективности работы CBR-модуля

Для успешной реализации рассуждений на основе прецедентов необходимо обеспечить корректное извлечение прецедентов из БП,

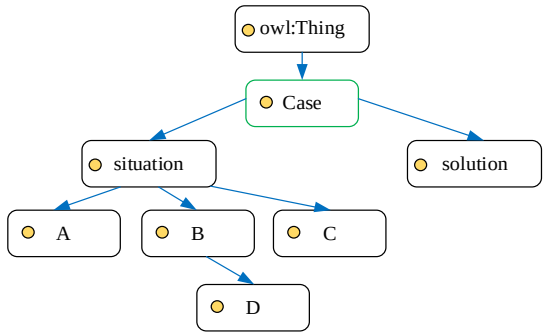

Рис. 1. Пример онтологической модели БП

Fig. 1. An example of a case base ontological model

то есть извлечение прецедентов, наиболее соответствующих сложившейся ситуации [8]. Выбор метода извлечения прецедентов напрямую связан со способом представления прецедентов и, соответственно, со способом организации БП.

В CBR-цикле перед сохранением при наличии тестовой выборки должна выполняться проверка корректности решения. Если решение прошло проверку и принято пользователем, оно сохраняется в БП как новый прецедент. Если проверка корректности решения на тестовых наборах завершается неудачно, прецедент сохраняется в *базе неудачных прецедентов* (БНП). Удачным называется прецедент, не ухудшающий качество работы CBR-модуля после его добавления в БП, а неудачным – прецедент, ухудшающий качество работы CBR-модуля после его добавления в БП. Таким образом, предлагается использовать тестовую (экспертную) выборку на этапе сохранения CBR-цикла для формирования БП и БНП.

Для параметрического представления прецедентов в CBR-модуле, как правило, используются метод *k*-ближайших соседей, а также его модификация при наличии библиотеки удачных и неудачных прецедентов [9].

Для структурированного представления прецедентов используется метод извлечения прецедентов на основе онтологии предметной области, базирующийся на теории структурного отображения SMT и методе *k*-ближайших соседей [9].

Основная идея метода *k*-ближайших соседей (*k-NN*) заключается в определении заданного числа *k*-ближайших соседей (прецедентов) в новой сложившейся ситуации в пространстве признаков (параметров). Число соседей может быть определено экспериментальным путем или же по критерию скользящего контроля (кросс-валидации). В случае решения задачи классификации определяется, к какому классу принадлежат большинство ближайших соседей и текущая проблемная ситуация.

Для определения ближайшего прецедента для текущей ситуации в CBR-модуле могут применяться различные метрики. Например, в качестве основных метрик могут выступать Евклидова метрика (расстояние), манхэттенское расстояние, расстояние Чебышева.

В процессе работы CBR-модуль накапливает прецеденты в БП, что способствует повышению качества решаемых задач, но при этом ведет к значительному увеличению временных затрат. В этом случае целесообразно оптимизировать работу с БП для повышения быстродействия CBR-модуля. Оптимизация БП может быть выполнена путем ее сокращения или обобщения накопленной информации (прецедентов). Возможно сокращение БП путем применения методов классификации и кластеризации прецедентов.

В разрабатываемом CBR-модуле реализованы методы оптимизации БП на основе кластеризации с использованием алгоритма *k*-средних и классификации с использованием алгоритма *k-NN* [10].

Реализация CBR-модуля для ИАД

Архитектура CBR-модуля состоит из следующих основных компонентов (рис. 2) [10]:

– пользовательский интерфейс для взаимодействия с экспертом или пользователем и отображения результатов работы;

– блок авторизации, содержащий ряд методов для регистрации новых пользователей в системе, аутентификации пользователей с помощью логина и пароля, а также методы для разграничения прав пользователей;

– блок извлечения прецедентов, содержащий ряд методов для извлечения прецедентов из параметрической и структурированной БП, а также извлечения из параметрической БП с использованием удачных (прецедент с положительным результатом применения) и неудачных (прецедент с отрицательным результатом применения) прецедентов;

– блок представления прецедентов, содержащий методы для создания, редактирования и

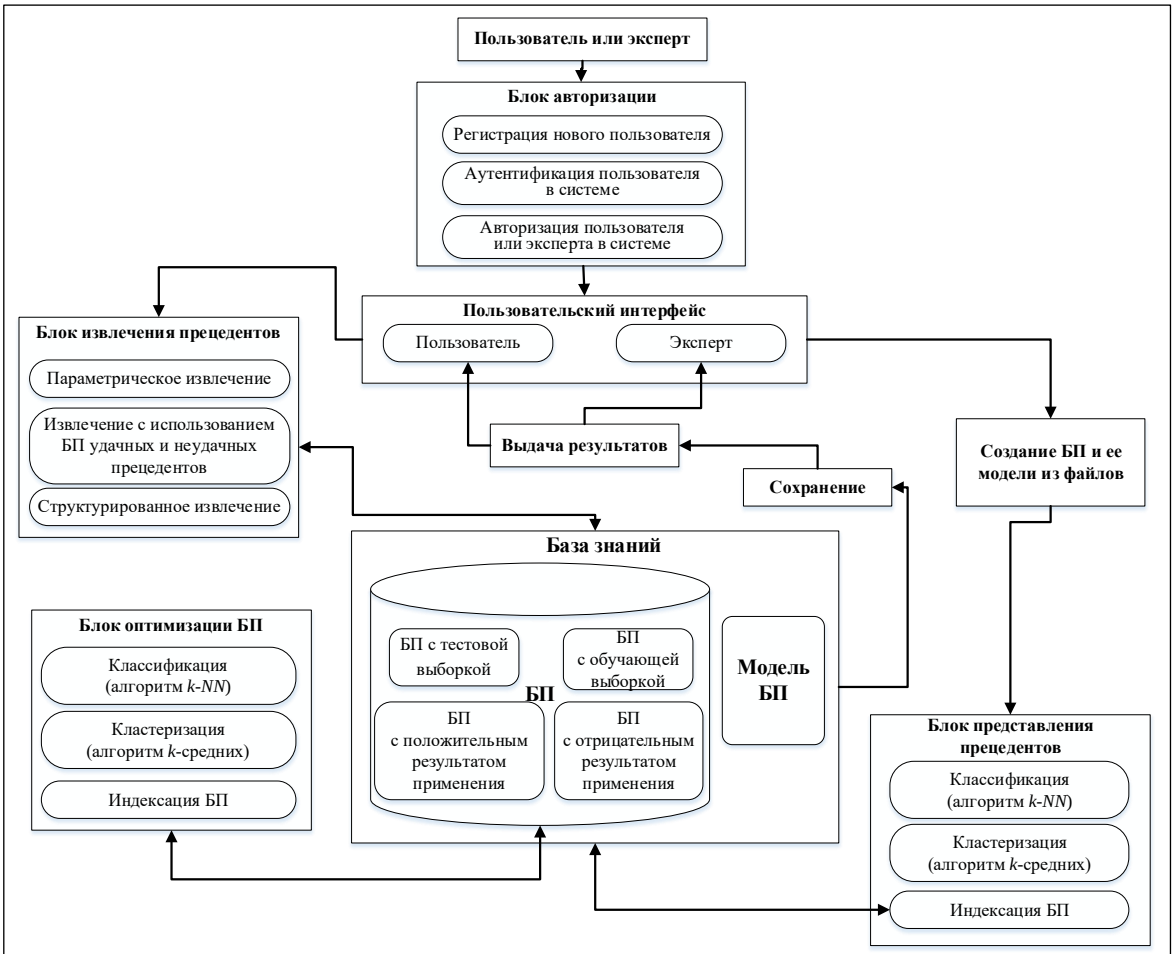

Рис. 2. Архитектура CBR-модуля для ИАД на основе прецедентного подхода

Fig. 2. The CBR-module architecture for data intelligent analysis based on the precedent approach

просмотра БП, создания модели БП с использованием онтологии для реализации структурированного представления, а также информации о параметрах прецедентов из БП (веса, диапазоны и т.д.);

– блок оптимизации БП, предназначенный для сокращения количества прецедентов в БП с использованием различных классификационных (*k-NN*) и кластерных алгоритмов (*k*-средних), а также повышения быстродействия работы системы с помощью индексации БП.

Разработанный CBR-модуль представляет собой веб-приложение, реализованное на языке Python 3.7.5 с использованием веб-фреймворка Flask [11]. Для реализации клиентской части приложения были использованы HTML, CSS, JavaScript, Jquery и фреймворк Bootstrap, а также библиотека vis.js для работы с онтологиями. Для хранения БП и администрирования пользователей используется СУБД MySQL.

Реализация CBR-модуля (прецедентной системы) в виде веб-приложения, состоящего из блоков отдельных подключаемых модулей, обеспечивает легкость в расширяемости CBR-модуля (добавления новых компонентов), а также возможность использования облачных технологий для увеличения производительности системы. Для работы с системой с помощью сети Интернет веб-приложение размещено на облачной платформе PythonAnywhere (https://www.pythonanywhere.com/), использующей облачные технологии Amazon.

Реализация CBR-модуля в виде веб-приложения дает возможность использования компонентов модуля как интеллектуального агента, реализующего ИАД на основе прецедентов при проектировании МАС.

Для работы приложения необходимо наличие БП. Для этого нужно создать структуру будущей БП и внести в нее прецеденты, а также загрузить данные из внешнего файла формата

Таблица 1

Созданная БП

Table 1

Created case base

id	Col0	Col1	Col2	Col3	Col4	Col5	Col6	Co67	Col8
1	2	2	4	2	1	3	3	10	A
2	2	2	4	2	3	3	1	10	A
3	2	2	4	2	1	3	3	10	A
4	3	1	3	2	3	3	1	15	S
5	1	3	3	3	1	3	2	15	A
6	2	2	3	2	1	3	2	10	S
7	3	2	4	2	1	1	1	10	A
8	2	3	3	2	1	3	1	15	A

Comma-Separated Values (.csv) или Text (.txt), содержащего набор данных. В этом случае программа автоматически сгенерирует структуру БП и заполнит ее данными из файла. В случае необходимости предусмотрена возможность редактирования БП.

Для создания структурированной БП на сервер загружается модель БП из внешнего файла, содержащего онтологию предметной области. При этом проверяется соответствие между параметрами прецедента в предметной области и параметрами прецедента в структуре таблицы, соответствующей параметрической БП. Модель БП представляется в виде иерархии концептов онтологии, а БП вместе с таблицей, описывающей параметры прецедентов из БП, хранятся в соответствующих таблицах в БД.

Рассмотрим работу реализованного модуля на примере решения задачи классификации данных, представленных в структурированном виде. Для этого воспользуемся набором данных Post-Operative Patient Data Set (набор данных о состоянии пациентов после операции), взятым из открытого репозитория UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Post-Operative+Patient), который был дополнен онтологией предметной области. Набор данных содержит 90 записей о состоянии пациентов после операции. Каждая из записей представлена 8 атрибутами, которые описывают состояние пациента: L_CORE – внутренняя температура пациента, L_SURF – температура тела пациента, L_O2 – насыщение кислородом, L_BP – последнее измерение артериального давления, SURF_STBL – стабильность температуры поверхности тела пациента, CORE_STBL – стабильность пациента, BP_STBL – стабильность артериального давления пациента, COMFORT – воспринимаемый пациентом комфорт при выписке. На основе этих данных необходимо принять одно из трех решений: I – пациент отправлен в отделение интенсивной терапии, A – пациент отправлен в больницу общего профиля, S – пациент готов к отправке домой. Для корректной работы алгоритма извлечения категориальные текстовые значения атрибутов набора были преобразованы в количественные. БП и ее модель после загрузки в систему представлены в таблице 1 и на рисунке 3 соответственно.

CBR-модуль позволяет вычислять оценки сходства текущей ситуации (рис. 4) и прецедентов из БП. Пользователь имеет возможность выбирать наиболее подходящий прецедент исходя из двух оценок сходства по струк-

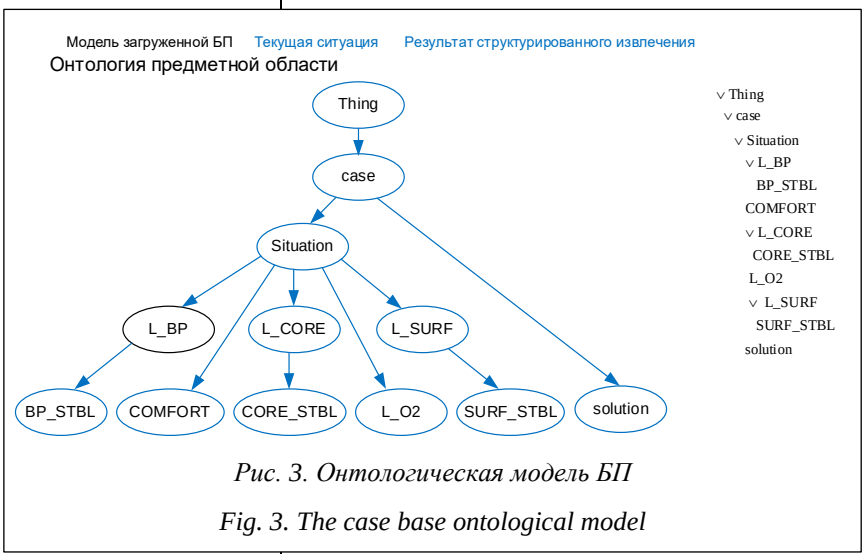

Рис. 3. Онтологическая модель БП

Fig. 3. The case base ontological model

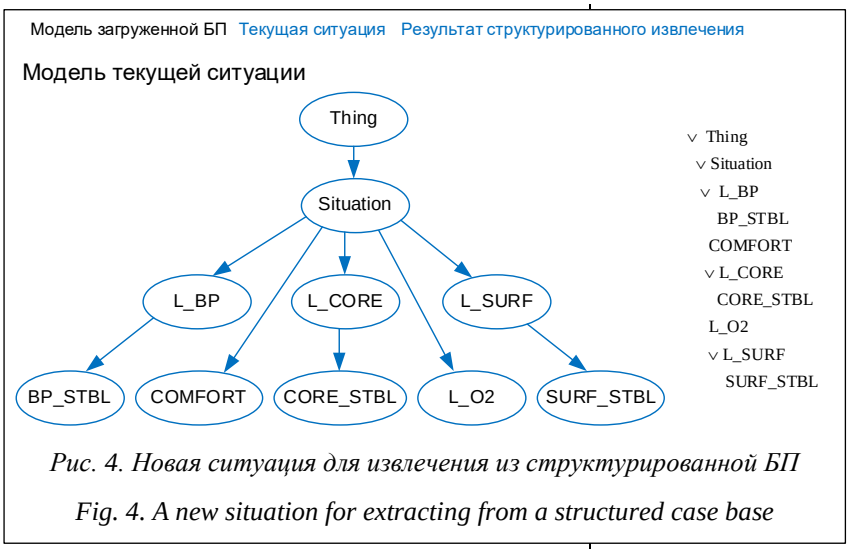

Рис. 4. Новая ситуация для извлечения из структурированной БП

Fig. 4. A new situation for extracting from a structured case base

туре (на основе онтологии предметной области и метода SMT) и по методу *k-NN*.

В результате структурированного извлечения для данной ситуации рекомендованным решением при различном числе соседей *k* и различных метриках является решение A – пациент отправлен в больницу общего профиля (табл. 2).

нове прецедентов, а также особенности конструирования РИС как МАС. Подробно описаны основные компоненты разработанного CBR-модуля (прецедентной системы), реализующего все этапы CBR-цикла, в виде веб-приложения на языке Python 3.7.5 с использованием веб-фреймворка Flask, состоящего из нескольких связанных компонентов (модулей) для обеспечения удобной интеграции в МАС.

Работа реализованного CBR-модуля протестирована на примере решения задачи классификации данных на основе прецедентов, представленных в структурированном виде с использованием набора данных Post-Operative Patient Data Set, взятого из открытого репозитория Калифорнийского университета (UCI Machine Learning Repository).

Таблица 2

Результат структурированного извлечения

Table 2

The structured extraction result

Номер парного соответствия	Оценка структурного соответствия	Оценка k-NN	Среднее арифметическое оценок	Решение
1	100.0	66.67	83.34	A
2	68.75	61.11	64.93	A
3	68.75	66.67	67.71	A
4	68.75	58.72	63.74	A
5	68.75	66.67	67.71	A
6	100.0	72.22	86.11	A

Заключение

В статье основное внимание уделено разработке и программной реализации CBR-модуля для ИАД, ориентированного на интеграцию в РИС (МАС). Рассмотрены возможности и особенности поиска решений (рассуждений) на основе прецедентов, а также особенности конструирования РИС как МАС. Подробно описаны основные компоненты разработанного CBR-модуля (прецедентной системы), реализующего все этапы CBR-цикла, в виде веб-приложения на языке Python 3.7.5 с использованием веб-фреймворка Flask, состоящего из нескольких связанных компонентов (модулей) для обеспечения удобной интеграции в МАС.

Работа реализованного CBR-модуля протестирована на примере решения задачи классификации данных на основе прецедентов, представленных в структурированном виде с использованием набора данных Post-Operative Patient Data Set, взятого из открытого репозитория Калифорнийского университета (UCI Machine Learning Repository).

В дальнейшем планируется использовать реализованный CBR-модуль как один из базовых компонентов РИС в составе интеллектуальной системы поддержки принятия решений реального времени для мониторинга и управления сложными техническими системами на примере объектов энергетики.

Работа выполнена при финансовой поддержке РФФИ, проекты №№ 18-29-03088, 20-07-00498, 20-57-00015.

Литература

1. Ponomarev S., Voronkov A.E. Multi-agent systems and decentralized artificial superintelligence. ArXiv. URL: https://arxiv.org/abs/1702.08529 (дата обращения: 12.05.2021).
2. Balaji P., Srinivasan D. An introduction to multi-agent systems. In: Innovations in Multi-Agent Systems and Applications-1, 2010, pp. 1–27. DOI: 10.1007/978-3-642-14435-6_1.

3. Andreadis G., Klazoglou P., Niotaki K., Bouzakis K.-D. Classification and review of multi-agents systems in the manufacturing section. Procedia Engineering, 2014, vol. 69, pp. 282–290. DOI: 10.1016/j.proeng.2014.02.233.

4. FIPA: The Foundation for Intelligent Physical Agents. Abstract Architecture Specification. URL: http://www.fipa.org/specs/fipa00001/index.html (дата обращения: 12.05.2021).

5. Городецкий В.И., Карсаев О.В., Самойлов В.В., Серебряков С.В. Прикладные многоагентные системы группового управления // Искусственный интеллект и принятие решений. 2009. № 2. С. 3–24.

6. Бредихин К.Н., Варшавский П.Р. Архитектура системы распределенного вывода на основе прецедентов для интеллектуальных систем // Программные продукты и системы. 2011. № 1. С. 50–53.

7. Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. № 2. С. 45–47.

8. Рассел С., Норвиг П. Искусственный интеллект: современный подход; [пер. с англ.]. М.: Вильямс, 2006. 1408 с.

9. Кхаинг З.Л., Мьо А.К., Варшавский П.Р., Алехин Р.В. Реализация прецедентного модуля для интеллектуальных систем // Программные продукты и системы. 2015. № 2. С. 26–31. DOI: 10.15827/0236-235X.110.026-031.

10. Varshavskii P., Alekhin R., Polyakov S., Blashonkov T., Mukhacheva I. Development of a modular case-based reasoning system for data analysis. Proc. Intern. Youth Conf. REEPE, 2020, pp. 458–461. DOI: 10.1109/REEPE49198.2020.9059242.

11. Гринберг М. Разработка веб-приложений с использованием Flask на языке Python; [пер. с англ.]. М.: ДМК-пресс, 2014. 272 с.

Software & Systems
DOI: 10.15827/0236-235X.135.381-389

Received 03.06.21
2021, vol. 34, no. 3, pp. 381–389

Software implementation of the data mining module based on case-based reasoning for distributed intelligent systems

A.P. Eremeev [1], *Dr.Sc. (Engineering), Professor, eremeev@appmat.ru*
P.R. Varshavskii [1], *Ph.D. (Engineering), Associate Professor, Head of Department, VarshavskyPR@mpei.ru*
S.A. Polyakov [1], *Postgraduate Student, PoliakovSerA@mpei.ru*

[1] *National Research University "Moscow Power Engineering Institute", Moscow, 111250, Russian Federation*

Abstract. The paper discusses the problems of implementing the Case-Based Reasoning (CBR) module that allows performing all stages of the case-based learning cycle (CBR-cycle) for distributed intelligent systems. Nowadays, there is a steady trend towards widespread using of methods and tools for data mining (DM) for solving various applied problems. More and more large companies are using intelligent systems and DM tools to improve their business efficiency and reduce costs. Due to developing Internet technologies and the availability of cloud computing, a promising direction in the field of artificial intelligence (AI) is the creation of distributed intelligent systems (DIS).

DIS systems are characterized by the distribution of computing and information resources, which increases adaptability, reliability, as well as the overall performance of the system due to the ability to simultaneously process large data sets. DIS systems consist of autonomous nodes (agents) that can act independently and exchange information with each other asynchronously. Since there are agents capable of implementing certain intelligent functions, this is an indication of the close connection between DIS systems and multi-agent systems (MAS).

The work focuses on one of the DM methods based on the use of previously accumulated experience (cases). The indicated case method (CBR-method) allows solving a new problem using (adopting) the solution of a similar and already known problem. To solve data analysis problems based on cases, there is a modular

case system (CBR-module) that implements the main stages of the CBR-cycle. The implemented system allows working with cases presented in a parametric and structured (based on ontologies) form. The CBR-module is a web application implemented in the Python 3.7.5 programming language using the Flask web framework and the Owlready2 library to work with ontologies. The developed CBR-module focuses on integrating into a DIS system to perform data mining based on cases.

The implemented CBR-module has been tested on the example of solving the classification task using data sets taken from the open repository of the University of California (UCI Machine Learning Repository).

Keywords: data mining, case, multi-agent systems, data classification.

Acknowledgements. *This work has been supported by RFBR, projects no. 18-29-03088, 20-07-00498, 20-57-00015.*

References

1. Ponomarev S., Voronkov A.E. Multi-agent systems and decentralized artificial superintelligence. *ArXiv*. Available at: https://arxiv.org/abs/1702.08529 (accessed May 12, 2021).
2. Balaji P., Srinivasan D. An introduction to multi-agent systems. In: *Innovations in Multi-Agent Systems and Applications-1*, 2010, pp. 1–27. DOI: 10.1007/978-3-642-14435-6_1.
3. Andreadis G., Klazoglou P., Niotaki K., Bouzakis K.-D. Classification and review of multi-agents systems in the manufacturing section. *Procedia Engineering*, 2014, vol. 69, pp. 282–290. DOI: 10.1016/j.proeng.2014.02.233.
4. *FIPA: The Foundation for Intelligent Physical Agents. Abstract Architecture Specification*. Available at: http://www.fipa.org/specs/fipa00001/index.html (accessed May 12, 2021).
5. Gorodetsky V.I., Karsaev O.V., Samoylov V.V., Serebryakov S.V. Applied multiagent systems of group control. *Scientific and Technical Information Processing*, 2010, vol. 37, no. 5, pp. 301–317. DOI: 10.3103/S0147688210050060 (in Russ.).
6. Bredikhin K.N., Varshavskii P.R. Distributed case-based reasoning system architecture for intelligent systems. *Software and Systems*, 2011, no. 1, pp. 50–53 (in Russ.).
7. Varshavskii P.R., Eremeev A.P. Modeling case-based reasoning in intelligent decision support systems. *Artificial Intelligence and Decision Making*, 2009, no. 2, pp. 45–47 (in Russ.).
8. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. Pearson publ., 1112 p. (Russ. ed.: Moscow, 2006, 1408 p.).
9. Khaing Z.L., Mio A.K., Varshavskii P.R., Alekhin R.V. Implementation of a case-based module for intelligent systems. *Software and Systems*, 2015, no. 2, pp. 26–31. DOI: 10.15827/0236-235X.110.026-031 (in Russ.).
10. Varshavskii P., Alekhin R., Polyakov S., Blashonkov T., Mukhacheva I. Development of a modular case-based reasoning system for data analysis. *Proc. Intern. Youth Conf. REEPE*, 2020, pp. 458–461. DOI: 10.1109/REEPE49198.2020.9059242.
11. Grinberg M. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media Publ., 2014, 314 p. (Russ. ed.: Moscow, 2014, 272 p.).

Для цитирования

Еремеев А.П., Варшавский П.Р., Поляков С.А. Программная реализация модуля анализа данных на основе прецедентов для распределенных интеллектуальных систем // Программные продукты и системы. 2021. Т. 34. № 3. С. 381–389. DOI: 10.15827/0236-235X.135.381-389.

For citation

Eremeev A.P., Varshavskii P.R., Polyakov S.A. Software implementation of the data mining module based on case-based reasoning for distributed intelligent systems. *Software & Systems*, 2021, vol. 34, no. 3, pp. 381–389 (in Russ.). DOI: 10.15827/0236-235X.135.381-389.