

Сафонов Дмитрий Сергеевич – старший инженер-программист, ООО «Компас Плюс», г. Магнитогорск, e-mail: dmitry.s.safonov@gmail.com.

Логунова Оксана Сергеевна – д-р техн. наук, профессор, заведующий кафедрой вычислительной техники и программирования ФГБОУ ВПО «Магнитогорский государственный технический университет им. Г.И. Носова». E-mail: logunova66@mail.ru.

Сафонов Д.С., Логунова О.С. Структура интерактивной системы автоматизированного проектирования конструкции секций вторичного охлаждения машины непрерывного литья заготовок // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2014. – №2. – С. 75-81.

Safonov, D.S. and Logunova, O.S., 2014. Structure of computer-aided design system for designing construction of the secondary cooling sections for continuous casting machine. Software of systems in the industrial and social fields, 2: 75-81.

УДК 004.42, 004.82

АНАЛИЗ ИЗБЫТОЧНОСТИ ВЕРСИОННОГО НАБОРА БИЗНЕС-ПРАВИЛ КОНТРОЛЯ ДОСТОВЕРНОСТИ ДАННЫХ

Фишбеин А.И., Шибанов С.В.

Аннотация. В статье рассматривается проблема анализа избыточности наборов бизнес-правил, предназначенных для контроля достоверности данных информационных систем. Приводятся требования к программным средствам, осуществляющим контроль достоверности данных. Выделяется необходимость функции программного анализа наборов бизнес-правил с целью их улучшения по различным критериям. На основании приведённых требований обосновывается выбор онтологической модели, базирующейся на дескрипционных логиках, для представления бизнес-правил. Формулируются общие принципы построения сигнатур и компонентов онтологий бизнес-правил. Приводится пример построения онтологии набора правил. В статье приводятся основные виды анализа наборов правил. Для анализа избыточности предлагается использование одной из базовых задач логического вывода - классификации онтологии. Рассматривается специфика применения этой процедуры для анализа наборов бизнес-правил. Предлагается и подробно рассматривается алгоритм анализа избыточности наборов бизнес-правил. Приводится пример операции анализа избыточности набора бизнес-правил. Приводится пример программной реализации анализа избыточности наборов бизнес-правил, представленных в формате OWL. Предложенный метод позволяет выполнять анализ избыточности наборов бизнес-правил для широкого класса предметных областей.

Ключевые слова. Бизнес-правила; контроль достоверности данных; информационные системы; анализ избыточности; онтология; дескрипционная логика; логический вывод.

REDUNDANCY ANALYSIS FOR VERSIONING SET OF DATA RELIABILITY CONTROL BUSINESS RULES

Fishbeyn A.I., Shibanov S.V.

Summary. In article the problem of redundancy analysis for business rules sets, designed to data reliability control of information systems is considered. Requirements to software which carry out data reliability control are provided. The need of program analysis function of business rules sets to improve them according to various criteria is highlighted. On the basis of provided requirements the choice of ontology model which is based on description logics is substantiated, to represent business rules. General principles of creation of signatures and components of business rules ontologies are formulated. The example of creation of ontology of rules set is given. In article main types of analysis of rules sets are given. For the redundancy analysis use of ontology classification, which is one of basic problems of logical inference, is offered. Specificity of application of this procedure for the analysis of sets of business rules is considered. The algorithm of redundancy analysis of business rules sets is proposed and considered in detail. The example of operation of redundancy analysis of business rules sets is given. The example of software implementation of redundancy analysis of business rules sets, which represented in the OWL format, is given. The proposed method allows to execute redundancy analysis of business rules sets for a wide class of application domains.

Keywords. Business rules; data reliability control; information systems; redundancy analysis; ontology; description logic; reasoning.

Проблема контроля достоверности данных информационных систем на основе бизнес-правил

Во всех информационных системах существуют правила, зависящие от предметной области, определяющие достоверность обрабатываемой информации. Достоверность данных в информационных системах – это соответствие данных бизнес-правилам, действующим в этой предметной области. Бизнес-правила (БП) в контексте контроля достоверности данных – это определяющие или ограничивающие утверждения, фиксирующие определённые закономерности в данных, которые можно использовать для проверки правильности и нахождения

ния возможных ошибок. Примеры БП: «количество студентов на курсе не может быть больше суммы количества бесплатных и платных мест», «размер выданной заработной платы должен равняться сумме оклада, надбавки за стаж и премии» [1].

Для организации бизнес-правил предлагается множество разных схем классификации. На рис. 1 приведена наиболее широко используемая классификация, применяемая в [15].



Рис. 1. Схема классификации бизнес-правил

Факты – это верные утверждения о бизнесе. Пример: «Со стоимости доставки налог с продаж не берётся».

Ограничения определяют, какие операции может выполнять система и её пользователи. Пример: «Постоянный посетитель библиотеки может отложить для себя до 10 книг».

Активаторы операций – это правила, при определённых условиях приводящие к выполнению каких-либо действий. Пример: «Если срок хранения контейнера с химикатом истёк, об этом необходимо уведомить лицо, у которого в данный момент находится контейнер».

Вычисления – это правила, определяющие вычисления, выполняемые с использованием математических формул и алгоритмов. Пример: «Цена единицы товара снижается на 10% при заказе от 6 до 10 единиц, на 20% - при заказе от 11 до 20 единиц и на 30% - при заказе свыше 20 единиц».

Выводами называют правила, устанавливающие новые реалии на основе достоверности определённых условий. Пример: «Если платёж не поступил в течение 30 календарных дней с момента отправки счёта, счёт считается просроченным» [15].

Как видно из этой классификации, бизнес-правила контроля достоверности данных являются подвидом ограничений.

Обычно в информационных системах (ИС) количество используемых бизнес-правил достаточно велико, и различным видам данных соответствуют разные группы правил. Зачастую бизнес-правила также имеют версии. Версия БП - это правило, связанное с другим, уже существующим, условием выбора между ними. То есть, выбор между тем, какое правило должно использоваться в некоторой ситуации, зависит от некоего условия, например, для проверки некоторых данных могут использоваться разные версии правила, в зависимости от текущей даты. Таким образом, ИС имеет дело с версионными наборами БП.

Каждое бизнес-правило контроля достоверности данных состоит из двух основных секций: условия достоверности и идентификатора правила. Для того, чтобы проверить корректность обрабатываемых данных, достаточно условий, при которых значение некоторых данных считается достоверным, и идентификатора, содержащего метаданные правила (сведения о наборе, о версии правила, и так далее). Таким образом, набор бизнес-правил контроля достоверности данных, в сущности, представляет собой набор помеченных идентификаторами условий корректности некоторых данных. Условия, в основном, представляют собой выражения, соединённые логическими и арифметическими операторами, в которых в качестве операндов участвуют разнообразные данные из предметной области, обрабатываемые ИС [1].

Программные средства контроля достоверности данных (ПСКДД) должны хранить описания БП и проверять данные на соответствие им, позволять редактировать наборы правил, а также поддерживать версионность правил [1].

Как правило, используемые бизнес-правила изменяются в ходе эксплуатации ИС. Это может быть связано с изменениями в законодательстве, расширением сферы деятельности организации, модификацией бизнес-процессов, и так далее. Важным требованием к ПСКДД является возможность изменять правила в процессе функционирования системы, не нарушая её работы.

Поскольку система бизнес-правил постоянно модифицируется, зачастую, особенно если изменений много или они вносятся в короткий срок, наборы бизнес-правил могут утратить согласованность – правила могут оказаться избыточны, дублироваться или противоречить друг другу. Также возможны случаи, когда в наборе отсутствуют правила, непосредственно следующие из уже имеющихся. Таким образом, весьма полезной является возможность программного анализа наборов БП на согласованность и непротиворечивость, позволяющая выделять правила, противоречащие прочим. Помимо этого, полезной является функция предложения новых правил на основе уже существующих. Однако, чаще требуется не расширять автоматизированно наборы правил, а, напротив, уменьшать наборы, сохраняя при этом их смысловое содержание. Таким образом, чрезвычайно желательной является возможность производить анализ избыточности наборов БП с целью нахождения дубликатов и перекрывающихся друг друга правил.

Для реализации этих требований необходимо, чтобы модель представления версионных наборов БП обладала гибкостью и богатыми выразительными возможностями, чтобы с её помощью могли быть описаны любые правила. Также требуется, чтобы модель позволяла формализованно описывать правила, при этом обладая хорошими вычислительными свойствами [2]. Это нужно для решения задач анализа правил.

Представление наборов бизнес-правил в виде онтологии на основе дескрипционной логики

С учётом вышесказанного, для представления бизнес-правил была выбрана онтологическая модель с базовым математическим формализмом – дескрипционными логиками (ДЛ). ДЛ – это семейство языков представления знаний, которые позволяют описывать понятия предметной области и связи между ними в формализованном виде. ДЛ сочетают в себе богатые выразительные возможности и хорошие вычислительные свойства, поэтому ДЛ предлагаются как формальный язык для представления знаний и логического вывода из них [3]. Таким образом, высокая выразительность этого подхода позволяет описывать сложные бизнес-правила, в то время как эффективные механизмы логического вывода обеспечивают проведение требуемых видов анализа наборов правил.

Типичная база знаний (онтология) ДЛ включает два компонента – *TBox* и *ABox* [3]. *TBox* содержит утверждения о концептах, в то время как *ABox* содержит утверждения об индивидах. Иногда утверждения для ролей выделяются в отдельный набор – иерархию ролей или *RBox*.

ДЛ – это не одна логика, а семейство логик, обладающих разными выразительными возможностями и разной вычислительной сложностью проблем вывода. Чем более выразителен язык, тем труднее логический вывод [3, 12]. Следует выбрать достаточно выразительную логику, которая при этом разрешима и имеет приемлемую вычислительную сложность основных задач вывода. Указанным требованиям удовлетворяет дескрипционная логика *SROIQ(D)*, поэтому именно она выбрана для решения поставленной задачи.

Онтология представления версионного набора бизнес-правил – это:
– множество концептов этой онтологии N_C . Концептами являются понятия предметной области, задействованные в правилах. Также концептами являются собственно правила – в виде утверждения правильных вариантов концепта, к которому относится правило. Помимо этого, N_C может включать так называемые концепты-номиналы (в данном случае – одноэлементные множества определённых типов данных, например, целых чисел);

- множество ролей этой онтологии N_R . В качестве ролей выступают отношения, в которых находятся понятия предметной области, задействованные в правилах. Также роли используются для моделирования свойств концептов;
- множество индивидов этой онтологии N_I . Индивиды в онтологии представления набора БП отсутствуют, соответственно, N_I – пустое множество;
- набор утверждений о концептах этой онтологии $TBox$. $TBox$ включает в себя определения концептов – понятий предметной области, а также определения концептов-правил. Для невырожденных онтологий правил $TBox$ всегда не пуст, так как в непустом наборе бизнес-правил есть как минимум одно правило, которое может быть представлено в виде аксиомы эквивалентности в $TBox$;
- набор утверждений об индивидах этой онтологии $ABox$. Поскольку индивиды отсутствуют, $ABox$ пуст;
- набор утверждений о ролях этой онтологии $RBox$. $RBox$ может содержать аксиомы при наличии сложных отношений между концептами в правилах, или же, в противном случае, может быть пуст.

Пример 1. Набор бизнес-правил, определяющих достоверность данных отчёта, представляющего собой таблицу. Таблица изображена на рис. 2.

- 1) "Значение в ячейке (1;2) должно быть меньше, чем в ячейке (2;2).";
- 2) "Значение в ячейке (2;2) должно быть меньше, чем в ячейке (3;2).";
- 3) "Значение в ячейке (1;2) должно быть меньше, чем в ячейке (2;2), и меньше, чем в ячейке (3;2).";
- 4) "Значение в ячейке (1;2) должно быть меньше, чем в ячейке (3;2).";
- 5) "Значение в ячейке (1;2) должно быть меньше, чем в ячейке (3;2).".

	Показатель 1	Показатель 2	Показатель 3	Показатель 4	Показатель 5
1					
2					
3					
4					
Итого	5				

Рис. 2. Отчёт, достоверность данных которого определяет набор БП

Пример 2. Онтология набора бизнес-правил из примера 1.

Сигнатура онтологии:

- множество концептов этой онтологии $N_C = \{Ячейка, Ячейка_{1,2}, Ячейка_{2,2}, Ячейка_{3,2}, Ячейка_{1,2,r}, Ячейка_{2,2,r}, Ячейка_{1,2,r,0}, Ячейка_{1,2,r,1}, Ячейка_{1,2,r,2}, \{1\}, \{2\}, \{3\}, T\}$;
- множество ролей $N_R = \{номер_строки, номер_столбца, содержит_значение, меньше_чем\}$;
- множество индивидов $N_I = \{\}$;
- $TBox$ содержит следующие аксиомы

$$Ячейка_{1,2} \equiv Ячейка \cap \exists номер_строки.\{1\} \cap \exists номер_столбца.\{2\} \tag{1}$$

$$Ячейка_{2,2} \equiv Ячейка \cap \exists номер_строки.\{2\} \cap \exists номер_столбца.\{2\} \tag{2}$$

$$Ячейка_{1,2,r} \equiv (Ячейка_{1,2} \cap \exists содержит_значение.T) \cap \exists меньше_чем.(Ячейка_{2,2} \cap \exists содержит_значение.T) \tag{3}$$

$$Ячейка_{3,2} \equiv Ячейка \cap \exists номер_строки.\{3\} \cap \exists номер_столбца.\{2\} \tag{4}$$

$$\begin{aligned} \text{Ячейка}_{2,2,r} \equiv & (\text{Ячейка}_{2,2} \cap \exists \text{содержит_значение.T}) \\ & \cap \exists \text{меньше_чем.}(\text{Ячейка}_{3,2} \\ & \cap \exists \text{содержит_значение.T}) \end{aligned} \quad (5)$$

$$\begin{aligned} \text{Ячейка}_{1,2,r,0} \equiv & (\text{Ячейка}_{1,2} \cap \exists \text{содержит_значение.T}) \\ & \cap \exists \text{меньше_чем.}(\text{Ячейка}_{2,2} \\ & \cap \exists \text{содержит_значение.T}) \\ & \cap \exists \text{меньше_чем.}(\text{Ячейка}_{3,2} \\ & \cap \exists \text{содержит_значение.T}) \end{aligned} \quad (6)$$

$$\begin{aligned} \text{Ячейка}_{1,2,r,1} \equiv & (\text{Ячейка}_{1,2} \cap \exists \text{содержит_значение.T}) \\ & \cap \exists \text{меньше_чем.}(\text{Ячейка}_{3,2} \\ & \cap \exists \text{содержит_значение.T}) \end{aligned} \quad (7)$$

$$\begin{aligned} \text{Ячейка}_{1,2,r,2} \equiv & (\text{Ячейка}_{1,2} \cap \exists \text{содержит_значение.T}) \\ & \cap \exists \text{меньше_чем.}(\text{Ячейка}_{3,2} \\ & \cap \exists \text{содержит_значение.T}) \end{aligned} \quad (8)$$

- $AVox$ пуст;
- $RVox$ пуст.

Аксиомы (3), (5), (6), (7) и (8), собственно, и являются бизнес-правилами контроля достоверности данных – они описывают условия, при соблюдении которых соответствующие данные считаются корректными. Фактически, эти аксиомы являются определениями правильных классов [4]. В качестве же идентификаторов правил могут выступать, например, имена определяемых этими аксиомами концептов вместе с именем онтологии.

Применение процедур логического вывода для анализа избыточности наборов бизнес-правил

Основными видами анализа набора БП являются анализ согласованности и непротиворечивости, анализ полноты и анализ избыточности набора. Для анализа наборов БП предлагается использование процедур логического вывода на знаниях о бизнес-правилах. Основные задачи логического вывода для ДЛ разрешимы и имеют относительно невысокую вычислительную сложность, и потому применимы на практике.

Для анализа избыточности набора бизнес-правил предлагается использование одной из базовых задач логического вывода – классификации (*classification*) базы знаний (онтологии). Классификация, в свою очередь, опирается в своём определении на другой базовый вывод – категоризацию концептов (вложенность концептов, включение концептов, *subsumption*).

Категоризация (обычно обозначается как $C \subseteq D$) – это проблема проверки, рассматривать ли концепт, помеченный D , как более общий, чем другой концепт, помеченный C . Другими словами, категоризация проверяет, обозначает ли всегда первый концепт подмножество множества, обозначенного вторым концептом [3].

Классификация – это задача построения категоризационной иерархии всех атомарных концептов. Классификация базы знаний нацелена на определение для каждого двух концептов A, B , является ли $A \subseteq B$ логическим следствием базы знаний. [5]

Обычно задачу классификации сводят к задаче проверки того, следует ли аксиома $A \subseteq B$ из онтологии, для каждой пары A, B концептов. Алгоритм проверки того, следует ли аксиома из онтологии (*axiom entailment* или *entailment checking*), в свою очередь, обычно сводится к другой базовой задаче логического вывода – к задаче проверки согласованности онтологии (*knowledge base consistency*). Количество проверок следования аксиомы из онтологии n вычисляется по формуле

$$n = k \cdot (k - 1), \tag{9}$$

где k – число концептов, для которых строится классификация [5].

В рассматриваемом случае нет необходимости строить полную классификацию всех концептов онтологии.

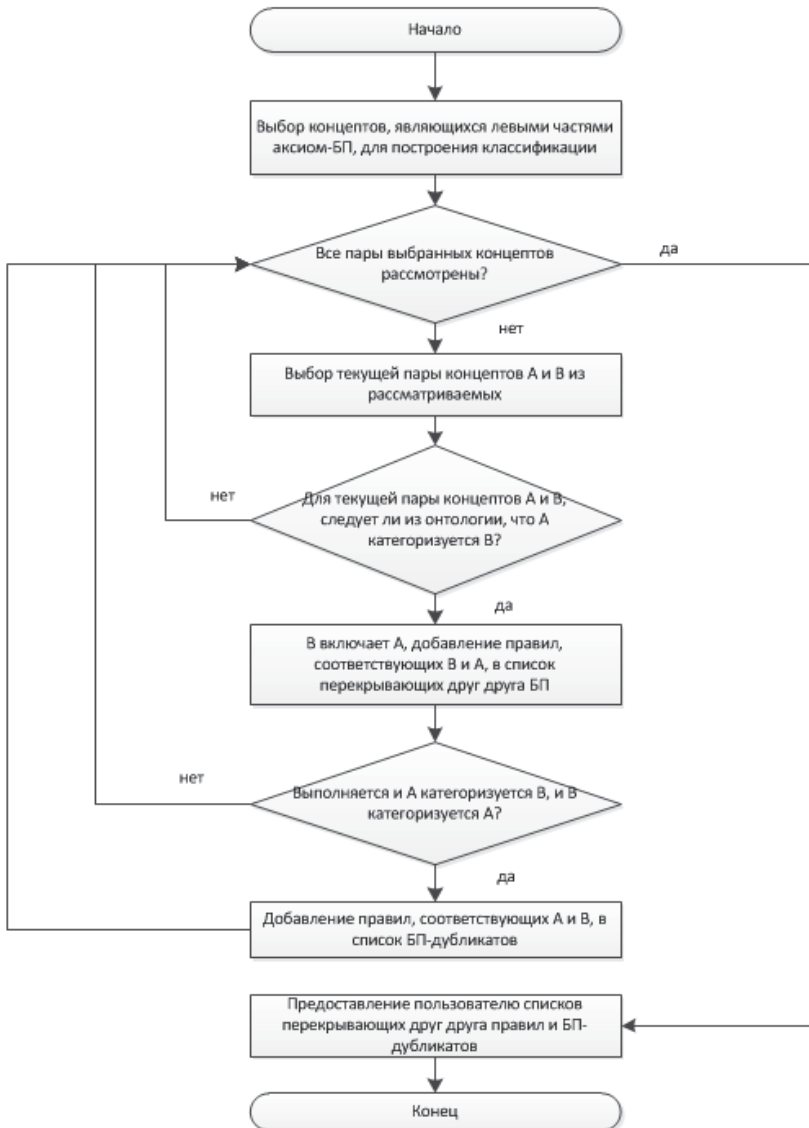


Рис. 3. Схема алгоритма анализа избыточности набора бизнес-правил

Достаточно построить иерархию только тех концептов, которые являются «правильными вариантами» других концептов, то есть, фактически, требуется иерархия концептов, являющихся левыми частями аксиом, определяющих бизнес-правила.

Следует отметить, что в некоторых случаях БП могут, например, перекрывать друг друга, при этом данное положение будет вполне корректным. К примеру, одна из версий некоторого правила может быть более общим вариантом другой версии, при этом наличие их обеих одновременно в онтологии вполне оправдано, поскольку, в зависимости от описанных условий, любая из них может быть применена. Таким образом, окончательное решение, удалять ли избыточные правила из онтологии, следует предоставлять пользователю.

На рис. 3 приведена схема алгоритма анализа избыточности набора БП.

Таким образом, общая последовательность действий для операции анализа избыточности набора следующая:

- 1) Из всех концептов онтологии выбирается множество концептов, классификацию которых нужно построить – это концепты, являющиеся левыми частями аксиом-БП;
- 2) Для каждой пары выбранных концептов A и B нужно проверить, следует ли аксиома $A \subseteq B$ из онтологии;
- 3) Если аксиома $A \subseteq B$ следует из онтологии, то B включает в себя A , и соответствующая пара правил добавляется в список перекрывающихся друг друга правил (причём, если выполняется и $A \subseteq B$, и $B \subseteq A$, то соответствующие A и B правила являются дубликатами);
- 4) Список перекрывающихся друг друга правил предоставляется пользователю.

Проверку следования аксиомы из онтологии сводим к проверке согласованности онтологии, используя доказательство от противного. Для этого предполагается, что аксиома не следует из онтологии, создаётся отрицание проверяемой аксиомы, точнее, аксиома, «противоположная» проверяемой. «Противоположная» аксиома добавляется к онтологии, после чего проверяется согласованность онтологии. Если в результате будет показана несогласован-

ность онтологии, то изначальное предположение о том, что аксиома не следует из онтологии, было неверным. Следовательно, проверяемая аксиома является логическим следствием онтологии. Данный способ решения этой задачи используется для различных ДЛ [5, 6]. Для проверки согласованности (выполнимости) онтологии можно использовать табличный алгоритм, описанный в [3, 5, 7 – 11].

И алгоритм, и примеры операции проверки следования аксиомы из онтологии рассмотрены во многих источниках [5, 6].

Пример 3. Операция анализа избыточности набора бизнес-правил.

Для онтологии из примера 2 требуется построить классификацию для следующих концептов: Ячейка_{1,2,r}, Ячейка_{2,2,r}, Ячейка_{1,2,r,0}, Ячейка_{1,2,r,1}, Ячейка_{1,2,r,2}. Таким образом, для каждой пары A и B из этих концептов нужно проверить, следует ли $A \subseteq B$ из онтологии. Согласно формуле (9), количество проверок следования аксиомы из онтологии

$$n = 5 \cdot (5 - 1) = 20.$$

Рассмотрим подробно проверку для следующей аксиомы:

$$\text{Ячейка}_{1,2,r,0} \subseteq \text{Ячейка}_{1,2,r}. \tag{10}$$

Требуется проверить, следует ли аксиома (10) из онтологии правил. Создадим аксиому, «противоположную» аксиоме (10). Согласно [5], её можно сформулировать как

$$(\text{Ячейка}_{1,2,r,0} \cap \neg \text{Ячейка}_{1,2,r})(c0). \tag{11}$$

То есть, аксиома (11) утверждает, что существует некоторый экземпляр $c0$, являющийся экземпляром концепта Ячейка_{1,2,r,0}, при этом являющийся экземпляром отрицания концепта Ячейка_{1,2,r}. Это невозможно, если аксиома (10) действительно следует из онтологии.

Добавляем аксиому (11) в онтологию. Теперь нужно проверить согласованность онтологии, то есть, попробовать построить модель, выполняющую аксиомы (1) – (8) и аксиому (11).

Инициализируем интерпретацию экземпляром $c0$ и, согласно (11), помечаем его принадлежностью к концепту $(\text{Ячейка}_{1,2,r,0} \cap \neg \text{Ячейка}_{1,2,r})$. Но можно заметить, что, согласно (3), (6) и (11), $c0$ должен одновременно являться экземпляром Ячейка_{1,2,r} и $\neg \text{Ячейка}_{1,2,r}$. Таким образом, мы сразу получаем противоречие, следовательно, модель построить невозможно, и онтология, включающая аксиомы (1) – (8) и аксиому (11), не согласована. Это означает, что изначальное предположение было неверным, то есть, аксиома (10) следует из онтологии правил примера 2.

Проверка прочих аксиом производится аналогичным способом. Результаты всех проверок приведены в табл. 1.

Таблица 1

Результаты проверок следования аксиом в ходе классификации онтологии-примера

Концепт	Ячейка _{1,2,r}	Ячейка _{2,2,r}	Ячейка _{1,2,r,0}	Ячейка _{1,2,r,1}	Ячейка _{1,2,r,2}
Ячейка _{1,2,r}	-	Ячейка _{1,2,r} \subseteq Ячейка _{2,2,r} – ложно	Ячейка _{1,2,r} \subseteq Ячейка _{1,2,r,0} – ложно	Ячейка _{1,2,r} \subseteq Ячейка _{1,2,r,1} – ложно	Ячейка _{1,2,r} \subseteq Ячейка _{1,2,r,2} – ложно
Ячейка _{2,2,r}	Ячейка _{2,2,r} \subseteq Ячейка _{1,2,r} – ложно	-	Ячейка _{2,2,r} \subseteq Ячейка _{1,2,r,0} – ложно	Ячейка _{2,2,r} \subseteq Ячейка _{1,2,r,1} – ложно	Ячейка _{2,2,r} \subseteq Ячейка _{1,2,r,2} – ложно
Ячейка _{1,2,r,0}	Ячейка _{1,2,r,0} \subseteq Ячейка _{1,2,r} – истинно	Ячейка _{1,2,r,0} \subseteq Ячейка _{2,2,r} – ложно	-	Ячейка _{1,2,r,0} \subseteq Ячейка _{1,2,r,1} – истинно	Ячейка _{1,2,r,0} \subseteq Ячейка _{1,2,r,2} – истинно
Ячейка _{1,2,r,1}	Ячейка _{1,2,r,1} \subseteq Ячейка _{1,2,r} – ложно	Ячейка _{1,2,r,1} \subseteq Ячейка _{2,2,r} – ложно	Ячейка _{1,2,r,1} \subseteq Ячейка _{1,2,r,0} – ложно	-	Ячейка _{1,2,r,1} \subseteq Ячейка _{1,2,r,2} – истинно
Ячейка _{1,2,r,2}	Ячейка _{1,2,r,2} \subseteq Ячейка _{1,2,r} – ложно	Ячейка _{1,2,r,2} \subseteq Ячейка _{2,2,r} – ложно	Ячейка _{1,2,r,2} \subseteq Ячейка _{1,2,r,0} – ложно	Ячейка _{1,2,r,2} \subseteq Ячейка _{1,2,r,1} – истинно	-

Таким образом, как видно из таблицы 1, правило для Ячейка_{1,2,r,0} перекрывает правила для Ячейка_{1,2,r}, Ячейка_{1,2,r,1} и Ячейка_{1,2,r,2}, то есть, правило для Ячейка_{1,2,r,0} является более детализированным вариантом трёх остальных правил. Одновременно с этим, правила для Ячейка_{1,2,r,1} и Ячейка_{1,2,r,2} являются дубликатами.

Пример программной реализации анализа избыточности наборов бизнес-правил

В настоящее время представление знаний в виде онтологий широко распространено, для работы с онтологиями существуют различные программы, такие, как редакторы онтологий, хранилища онтологий, и так далее. Наиболее распространённый сейчас язык описания онтологий – *OWL (Web Ontology Language)* [13]. Файлы описания онтологий на языке *OWL* представляют собой разновидность *XML*-файлов. Таким образом, программные средства контроля достоверности данных могут загружать наборы БП, описанные на языке *OWL*, и работать с таким представлением правил.

Помимо основного ПО, обеспечивающего создание, хранение и выполнение наборов бизнес-правил, ПСКДД также включают набор плагинов для различной работы с БП: тестирования наборов правил, выбора способа реализации правил, генерации исполняемого кода БП, анализа наборов БП. Этим плагинам также передаётся для обработки онтология правил в *OWL*-формате. ПС анализа наборов БП предоставляют возможность проведения анализа непротиворечивости, полноты и избыточности наборов правил. Все эти виды анализа производятся на основе различных процедур логического вывода в ДЛ.

Дескрипционные логики широко используются и поддерживаются многими современными программными средствами для работы с онтологиями. Существует множество машин логического вывода (*reasoner*), таких, как *FaCT++*, *Pellet*, *RacerPro*, *HermiT*, позволяющих производить логический анализ онтологий. Анализ избыточности набора бизнес-правил производится с помощью одного из таких программных средств. Ниже приведён пример программной реализации анализа на языке *C#* с использованием *HermiT*. Общая последовательность действий для анализа в этом случае будет слегка отличаться, поскольку широко применяемый интерфейс для работы с онтологиями *OWLAPI* [14] предусматривает проведение классификации всей онтологии целиком (а не только отдельных концептов) в процессе использования машины логического вывода.

```

...
OWLReasoner reasoner = reasonerFactory.createReasoner(o0); //создаём reasoner, с помощью HermiT
reasoner.precomputeInferences(); //запрашиваем reasoner произвести классификацию онтологии
java.util.Set cl_set = o0.getClassesInSignature(); //получаем список классов в онтологии
ArrayList messages = new ArrayList(); //создаём список для сообщений – результатов анализа избыточности
for (int i = 0; i < cl_set.size(); i++) {
    OWLClass clsB = (OWLClass)cl_set.toArray()[i];
    if (pm.getShortForm(clsB).IndexOf("_r") > 0)
        //если класс является определением бизнес-правила
        java.util.Set sub_classes = reasoner.getSubClasses(clsB, false).getFlattened(); //список его подклассов
    for (int j = 0; j < sub_classes.size(); j++) {
        OWLClass clsA = (OWLClass)sub_classes.toArray()[j];
        if (pm.getShortForm(clsA).IndexOf("_r") > 0)
            //если подкласс также является бизнес-правилом – добавляем сообщение
            string s = "Правило " + pm.getShortForm(clsA) + " является более конкретизированной версией правила " +
                pm.getShortForm(clsB) + ".";
            messages.Add(s);
    }
    Node sub_classes0 = reasoner.getEquivalentClasses((OWLClassExpression)clsB); //список классов, эквивалентных
    //текущему
    for (int k = 0; k < sub_classes0.getSize(); k++) {
        OWLClass clsA = (OWLClass)sub_classes0.getEntities().toArray()[k];
        if ((clsA.ToString().CompareTo(clsB.ToString()) != 0) && (pm.getShortForm(clsA).IndexOf("_r") > 0))
            //если эквивалентные классы не являются одним и тем же, и оба они – бизнес-правила
            string s = "Правила " + pm.getShortForm(clsA) + " и " + pm.getShortForm(clsB) + " являются дубликатами.";
            string s0 = "Правила " + pm.getShortForm(clsB) + " и " + pm.getShortForm(clsA) + " являются дубликатами.";
            bool isExists = false;
            for (int k0 = 0; k0 < messages.Count; k0++) {

```



```

if (s0.CompareTo((string)messages[k0]) == 0)
    {
        isExists = true; break;
    }
if (!isExists)
{
    messages.Add(s); //если эти классы ещё не рассматривались на эквивалентность – добавляем сообщение
}
}
}
}
}

```

Заключение

Как видно на основе представленной информации, наборы бизнес-правил контроля достоверности данных могут быть представлены в виде онтологии на основе дескрипционной логики. Такое представление позволяет выполнять различные виды анализа наборов бизнес-правил, в том числе анализ избыточности, с использованием процедур логического вывода. Предложенный метод анализа довольно легко реализуем на практике и позволяет выполнять анализ избыточности для различных наборов бизнес-правил, независимо от предметной области.

Список используемых источников

1. Фишбейн, А.И. Представление наборов бизнес-правил контроля достоверности данных в виде онтологии на основе дескрипционной логики // Новые информационные технологии и системы. Материалы конференции. Пенза, 2012. – С. 329-332.
2. Фишбейн, А.И. Выбор модели представления версионных наборов правил контроля достоверности данных // Теория и практика системного анализа. Сборник трудов Всероссийской молодежной конференции. Белгород, 2012. – С. 619-623.
3. The Description Logic Handbook. Theory, implementation and applications / F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider. New York: Cambridge University Press, 2003. – 574 p.
4. Дубинин В.Н. Использование дескриптивной логики и Web-онтологий в определении визуальных языков // Перспективные технологии искусственного интеллекта. Материалы Международной научно-практической конференции. Пенза, 2008.
5. Rudolph, S. Foundations of Description Logics // In Reasoning Web: Semantic Technologies for the Web of Data, 7th International Summer School, volume 6848 of Lecture Notes in Computer Science. Springer. 2011. – P. 76-136.
6. Horrocks, I. Reducing OWL entailment to description logic satisfiability / I. Horrocks, P. Patel-Schneider // Journal of Web Semantics. 2004. 1(4). P. 345-357.
7. Horrocks, I. A tableau decision procedure for SHOIQ / I. Horrocks, U. Sattler // Journal of Automated Reasoning. 2007. 39(3). P. 249-276.
8. Baader, F. An overview of tableau algorithms for description logics / F. Baader, U. Sattler // StudiaLogica. 2001. 69(1). P. 5-40.
9. The complexity of concept languages / F.M. Donini, M. Lenzerini, D. Nardi, W. Nutt // Information and Computation. 1997. 134. P. 1–58.
10. Horrocks, I. A Description Logic with transitive and inverse roles and role hierarchies / I. Horrocks, U. Sattler // Journal of Logic and Computation. 1999. 9(3). P. 385–410.
11. Haarslev, V. A Description Logic with concrete domains and role-forming predicates / V. Haarslev, C. Lutz, R. Moller // Journal of Logic and Computation. 1999. 9(3). P. 351–384.
12. Brachman, R.J. The tractability of subsumption in frame-based description languages / R.J. Brachman, H.J. Levesque // In Proc. of the 4th Nat. Conf. on Artificial Intelligence. 1984. P. 34-37.
13. OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition). URL: <http://www.w3.org/TR/owl2-syntax/> (дата обращения: 05.12.2013).
14. OWLAPI. URL: <http://owlapi.sourceforge.net/> (дата обращения: 05.12.2013).
15. Вигер, К.И. Разработка требований к программному обеспечению. М.: Русская Редакция, 2004. 576 с.

Фишбейн Анна Ильинична – аспирант кафедры «Математическое обеспечение и применение ЭВМ» ФГБОУ ВПО «Пензенский государственный университет». E-mail: vot49tak@yandex.ru.

Шибанов Сергей Владимирович – кандидат технических наук, доцент кафедры «Математическое обеспечение и применение ЭВМ» ФГБОУ ВПО «Пензенский государственный университет». E-mail: serega@pnzgu.ru.

Фишбейн А.И., Шибанов С.В. Анализ избыточности версионного набора бизнес-правил контроля достоверности данных // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2014. – №2. – С. 81-89.

Fishbeyn, A.I. and Shibanov, S.V., 2014. Redundancy analysis for versioning set of data reliability control business rules. Software of systems in the industrial and social fields, 2: 81-89.
