



УДК 004.891

### МОДИФИЦИРОВАННЫЙ МЕТОД ЛОГИЧЕСКОГО ВЫВОДА ЗНАНИЙ НА ОСНОВЕ НЕЧЕТКОЙ ОНТОЛОГИИ И БАЗЫ ПРЕЦЕДЕНТОВ

Мошкин В.С., Ярушкина Н.Г.

*Ульяновский государственный технический университет,  
г. Ульяновск, Российская Федерация*

**PostForVadim@yandex.ru**

**jng@ulstu.ru**

В данной работе представлены результаты применения методологии интеграции систем продукционных правил с нечеткой онтологией на примере решения задачи оценки состояния локальной вычислительной сети в процессе искусственного повышения трафика. Помимо этого, рассмотрена возможность модификация данного алгоритма за счет параллельного запуска механизма анализа прецедентов в процессе логического вывода знаний.

**Ключевые слова:** онтология, прецедент, логический вывод, нечеткие онтологии.

#### Введение

Для обеспечения высокого качества принимаемых решений в процессе проектирования сложных технических систем необходимы модели, ориентированные на описание объектных структур рассматриваемой предметной области, позволяющие идентифицировать, анализировать (включая наглядное представление) и манипулировать всем многообразием объектов и отношений, имеющихся в предметной области (ПрО).

Однако состояние дел в сфере информационно-аналитического обеспечения процесса проектирования сложных систем недостаточно отвечает потребностям управления в современных производственных организациях, и существует ряд научных проблем, требующих системного решения:

- необходимость разработки семантического базиса анализа хранимых знаний в рамках решения задач автоматизированного проектирования;
- отсутствие интегративных концептуальных моделей, использующих различные подходы хранения знаний о предметной области;
- необходимость универсализации процесса автоматизированной обработки хранимых знаний при решении проектных задач различного рода;
- необходимость одновременного использования разноаспектных описаний особенностей рассматриваемой предметной области;

- необходимость решения проблемы учета нечеткости в человеческих рассуждениях;
- необходимость учета прошлого опыта решения задач, подобных поставленной перед проектировщиком.

Таким образом, на сегодняшний день актуальной является задача интеграция различных форм и алгоритмов представления и логического вывода знаний с целью поддержки принятия решений в процессе проектирования сложных систем.

#### 1. Нечеткость в логическом выводе знаний

##### 1.1. Преимущества использования нечетких алгоритмов в ЭС

При попытке формализовать человеческие знания, необходимые в процессе логического вывода рекомендаций в ЭС, возникает проблема использования традиционного математического аппарата для их описания. Для решения данной проблемы были введены в процесс логического вывода элементы нечеткости как реализация аппарата нечеткой (fuzzy) алгебры, предложенной американским исследователем Лотфи Заде в начале 70-х годов XX века.

Быстрый рост областей применимости алгоритмов нечеткого логического вывода обусловлен главными преимуществами fuzzy-систем:

- возможность оперировать нечеткими входными данными;
- возможность нечеткой формализации критериев оценки и сравнения.;
- возможность проведения качественных оценок как входных данных, так и выходных результатов;
- возможность проведения быстрого моделирования сложных динамических систем и их сравнительный анализ с заданной степенью точности.

В таблице 1 представлен ряд систем, использующих один или несколько алгоритмов нечеткого логического вывода [Trausan-Matu, 2008], [Nunes et al, 2012], [Lei, 2012].

Таблица 1.1 Примеры экспертных систем на основе

Наименование	Применение	Базовый алгоритм
Risk Analysis Expert System(RA_X)	Экспертная система анализа рисков	Мамдани
Adaptive-Network-Based Fuzzy Inference System (ANFIS )	Система прогнозирования состояния морских волн	Суджено
Fuzzy inference system in river flow forecasting	Система прогнозирования течения реки	Мамдани, Такаги–Суджено–Канг
СУ БПЛА	Система управления беспилотными летательными аппаратами	Мамдани
FIS To Fault Diagnosis	Система тестирования подшипников в условиях повышения нагрузки	Суджено

Несмотря на то, что с каждым годом область применения систем, основанных на использовании алгоритмов нечеткого логического вывода, расширяется, каждый из используемых алгоритмов имеет значительные недостатки в случае рассмотрения поставленной задачи через призму сложной предметной области:

- не учитываются особенности взаимосвязей объектов рассматриваемой предметной области;
- все алгоритмы являются однотактными и не предполагают возможности нахождения выхода при изменении входных значений в случае итерационных алгоритмов;
- не учитывается возможная иерархия и синонимия объектов;
- алгоритмы не содержат элементов обучаемости, что значительно снижает потенциал их развития и повышает трудозатраты экспертов на полноценное описание особенностей поведения объектов рассматриваемой предметной области.

В связи с этим возникает задача разработки и реализации гибридных алгоритмов логического вывода с использованием нечеткости,

обеспечивающих исправление перечисленных недостатков. Одним из таких решений является применение класса нечетких онтологий.

## 1.2. Формальная модель нечеткой онтологии

Переход от обычных онтологий к нечетким выглядит вполне естественным, поскольку понятия и отношения естественного языка, представляющие собой исходный материал для построения онтологии неоднозначны, неточны и не имеют жестких границ. Поэтому адекватным средством формализации онтологий могут служить модели на базе лингвистических переменных, нечеткие множества, нечеткие отношения, нечеткие графы и нечеткие деревья, нечеткие ограничения, нечеткие реляционные и алгебраические системы [НГС, 2007].

В общем виде любая нечеткая FuzzyOWL-онтология может быть представлена следующим образом:

$$I = (I_f, C_f, P_f, D_f, Q_f, L_f, Mod_f). \quad (1)$$

где

- $I_f$  – множество объектов классов онтологии;
- $C_f$  – множество нечетких классов онтологии;

$$C_f = \{C_f^A, C_f^C\}, \quad (2)$$

где  $C_f^A$  – множество абстрактных классов,  $C_f^C$  – множество конкретных классов онтологии;

- $P_f$  – множество свойств объектов:

$$P_f = \{P_f^A, P_f^C\}, \quad (3)$$

где  $P_f^A$  – множество конкретных свойств, т.е. свойств объектов (Object Property),  $P_f^C$  – множество абстрактных свойств, т.е. свойств типа данных (Datatype Property);

- $D_f$  – множество аксиом онтологии:

$$D_f = \{A_f^{ABox}, A_f^{TBox}, A_f^{RBox}\}, \quad (4)$$

где  $A_f^{ABox}$  – множество утверждений об индивидах,  $A_f^{TBox}$  – множество терминологических аксиом,  $A_f^{RBox}$  – множество аксиом отношений (иерархия отношений). Часть аксиом может быть подклассом множества нечетких аксиом, которые предполагают истинность утверждения с определенной степенью.

- $O_f$  – множество значений степеней, которые могут быть добавлены к объекту нечеткой аксиомы:

$$O_f = \{LD_f, MD_f, ND_f, Var_f\}, \quad (5)$$

где  $LD_f$  – множество лингвистических переменных,  $MD_f$  – множество степеней модификации,  $ND_f$  – множество численных значений степеней,  $Var_f$  – множество переменных.

- $L_f$  – множество операторов нечетких логик соответствующих типов.

$$L_f = \{L_f^{Luk}, L_f^{Zad}, L_f^{Goed}, L_f^{Prod}\}, \quad (6)$$

где  $L_f^{Luk}$  – множество операторов логики Лукасевича,  $L_f^{Zad}$  – множество операторов логики Заде,  $L_f^{Goed}$  – множество операторов логики Геделя,  $L_f^{Prod}$  – множество операторов продукционной логики. Определяются с помощью встроенного отношения *hasSemantics*;

- $Mod_f$  – множество «нечетких модификаторов», т.е. функций модификации функций принадлежности, нечетких классов и нечетких отношений. Функции могут быть линейными или треугольными.

### 1.3. Применение алгоритма логического вывода на основе нечеткой онтологии

Взаимосвязь нечеткой онтологии предметной области и набора продукционных правил в процессе логического вывода осуществляется с помощью формирования запросов к онтологии, генерируемых системой анализа при выполнении набора правил [Ярушкина и др., 2015].

Базовый алгоритм формирования вывода на основе интеграции онтологии и систем продукции представлен на рис.1

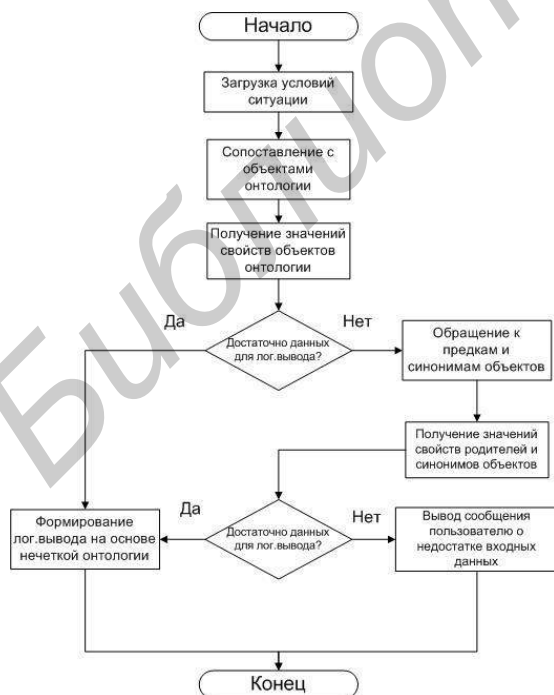


Рисунок 1 – Алгоритм вывода с использованием нечеткой онтологии

С целью сравнения методики взаимодействия онтологического анализа и механизмов логического вывода на основе четкой и нечеткой онтологии был проведен ряд экспериментов, в рамках которых были смоделированы возможные проблемные ситуации, возникающие в процессе работы локальной вычислительной сети при искусственном повышении загрузки каналов связи.

Объектом для проводимых экспериментов стала ЛВС центра разработки электронных мультимедиа технологий (ЦЭМТ) Ульяновского государственного технического университета. Сравнительная характеристика методик логического вывода в процессе моделирования проблемных ситуаций работы ЛВС представлена в таблице 2.

Таблица 2. Сравнение методик логического вывода на основе четких (OWL+SWRL) и нечетких (FuzzyOWL+SWRL) онтологий.

Рекомендации	OWL+SWRL	FuzzyOWL+SWRL
Нет решений	2	1
Несколько решений (в том числе верное или близкое к верному)	4	24
Несколько неверных решений	-	-
Одно верное (близкое к верному) решение	22	4
Одно неверное решение	2	1

Несмотря на то, что включение нечеткости в описание предметной области значительно снижает риски потери возможных результатов работы блока логического вывода и увеличивает гибкость процесса вывода, наличие большого количества вариантов решения задачи не дает пользователю в полной мере положиться на какой-либо из предложенных вариантов [Мошкин и др., 2015].

В этом случае необходима методика, позволяющая пользователю получать конкретное решение, основанное на практическом опыте предыдущих пользователей, решавших в прошлом аналогичную задачу.

## 2. Использование прецедентов в процессе логического вывода

### 2.1. Определение понятия рассуждения по прецедентам

При решении сложных задач в области автоматизированного проектирования пользователь должен получать всестороннюю поддержку со стороны системы, имея возможность выбора решения не только посредством использования экспертных знаний, хранящейся в базе системы, но и с учетом опыта решения данной задачи специалистами в прошлом. Такой опыт может быть формализован в виде набора прецедентов, которые должны обрабатываться и учитываться в рамках решения задачи параллельно основному алгоритму.

Согласно [Leake, 1996], **рассуждения по прецедентам** (case-based reasoning, CBR) — это метод формирования умозаключений, опирающийся не на логический вывод от исходных посылок (логические рассуждения), а на поиск и анализ случаев формирования подобных умозаключений в прошлом. Проверка корректности умозаключения может являться частью CBR-процесса.

С точки зрения решения задач, **рассуждения по прецедентам** — это метод получения решения путем поиска подобных проблемных ситуаций в памяти, хранящей прошлый опыт решения задач, и адаптации найденных решений к новым условиям. Применение CBR для решения задач оправдано в случае выполнения следующих условий, касающихся природы прикладной области:

1. Подобные задачи должны иметь подобные решения (*принцип регулярности*). В этом случае накопленный опыт решения задач может служить отправной точкой процесса поиска решения для новых подобных задач.

2. Виды задач, с которыми сталкивается решатель, должны иметь тенденцию к повторению. Это условие гарантирует, что для многих проблем в будущем будет существовать аналог в прошлом опыте [Pal, Shiu, 2004].

Процесс рассуждений по прецедентам может быть схематически представлен в виде цикла. В работе [Aamodt, 1994] приведено описание типового процесса рассуждений по прецедентам, который состоит из следующих этапов: формирование прецедента новой задачи (прецедента запроса); поиск наиболее похожих прецедентов и в БД прецедентов; повторное использование решений найденных прецедентов и их адаптация к новой задаче; проверка предлагаемого решения на адекватность и сохранение нового решения в БД прецедентов.

Существует аналогия между системами, основанными на правилах и прецедентах. И те, и другие необходимо каким-то образом индексировать, чтобы обеспечить эффективное извлечение. И те, и другие выбираются в результате сопоставления, причем выбор и ранжирование производятся на основе знаний, хранящихся в фоновых структурах, наиболее универсальными из которых являются онтологии.

Принципиальные различия между методиками логического вывода на основе систем продукций и прецедентов были в полной мере изложены в [Kolodner, 1993]. Базовые отличия данных методик:

1. Правила представляют собой шаблоны, т.е. содержат переменные и не описывают непосредственно решение, прецеденты же в свою очередь являются константами и оперируют конкретными объектами базы знаний.

2. Применение правил представляет собой итерационный цикл — последовательность шагов, приводящих к решению. Прецедент можно

рассматривать как приближенный вариант полного решения.

3. Правило выбирается на основе точного сопоставления антецедента и входных данных. Прецедент выбирается посредством частичного сопоставления, причем учитываются еще и знания о сущности характеристик, по которым выполняется сопоставление.

## 2.2. Формальная модель системы вывода, основанной на анализе прецедентов

К основным преимуществам рассуждений по прецедентам можно отнести отсутствие необходимости полного и углубленного рассмотрения знаний о конкретной предметной области; возможность напрямую использовать опыт, накопленный системой, без интенсивного привлечения эксперта в той или иной предметной области; возможность применения эвристик, повышающих эффективность решения задач [Дворянкин и др., 2008].

Формальная модель системы вывода, основанного на прецедентах, может быть представлена в виде упорядоченной тройки:

$$CBR = \{Cases, I, S_{CASE}\}, \quad (7)$$

где *Cases* — база прецедентов;

*I* — онтология предметной области (нечеткая);

*S<sub>CASE</sub>* — алгоритм поиска подходящего прецедента.

Структура прецедента может быть представлена в следующем виде:

$$Case = \{Index_{CASE}, D(Index_{CASE}), Eff(D(Index_{CASE}))\}, \quad (8)$$

где *Index<sub>CASE</sub>* — индекс прецедента, т.е. описание начальной ситуации;

*D(Index<sub>CASE</sub>)* — множество решений поставленной задачи;

*Eff(D(Index<sub>CASE</sub>))* — множество оценок эффективности принятого решения задачи.

Так как прецеденты, в отличие от правил, оперируют не переменными, а конкретными объектами классов, а также значениями свойств этих объектов, то, согласно (1), начальная ситуация описывается следующим множеством:

$$Index_{CASE} = \{I_f, P_f\}, \quad (9)$$

где *I<sub>f</sub>* — множество объектов классов онтологии;

*P<sub>f</sub>* — множество значений свойств соответствующих объектов.

Аналогична и модель решения задачи: решением является объект вспомогательного класса «Рекомендации» разработанной онтологии, который в качестве вывода в процессе работы алгоритма выдает лингвистическое значение свойства типа данных (DatatypeProperty) «имеетОписание» выбранного объекта класса «Рекомендации».

## 2.3. Модификация алгоритма логического вывода с использованием прецедентов

Для внесения в базу знаний начального набора прецедентов может использоваться универсальный редактор OWL-онтологии Protégé 4.x. Данная программная система предполагает возможность представления знаний в форме:

- нечетких онтологий (с использованием Fuzzy OWL Plugin);
- SWRL-правил;
- набора прецедентов в OWL-формате.

Для обработки прецедентов в OWL-формате в настоящее время используется java-фреймворк jColibri. Данный набор библиотек является бесплатным и свободно распространяемым. Корректный анализ прецедентов посредством использования фреймворка jColibri предполагает необходимость создания в OWL-онтологии трех базовых классов:

- CBR-CASE - класс, содержащий экземпляры прецедентов;
- CBR-DESCRIPTION - класс, описывающий ограничения, накладываемые на прецеденты;
- CBR-INDEX - хранит структуру прецедента, т.е. классы-потомки и их экземпляры, участвующие в процессе поиска подходящего прецедента.

Перечисленные вспомогательные классы онтологии напрямую согласуются с моделью вывода по прецедентам, представленной в (7) и (8).

Главное особенностью модифицированного алгоритма вывода рекомендаций (рис. 2) является проведение параллельного и независимого логического вывода результатов анализа на основании базы правил и базы прецедентов.

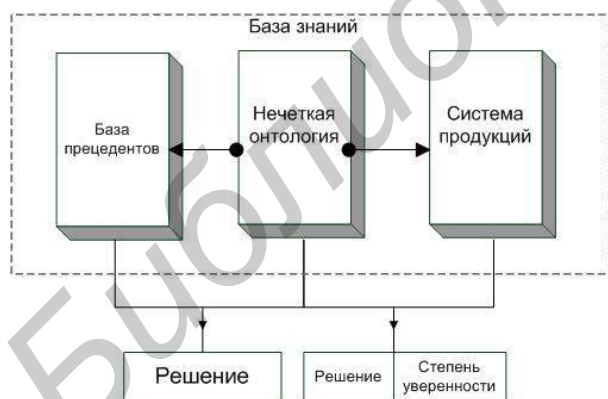


Рисунок 2 – Схема модифицированного алгоритма логического вывода с использованием базы прецедентов

Таким образом, пользователь получает возможность принятия решений как с учетом анализа закономерностей конкретной области, так и на основании опыта пользователей, уже сталкивавшихся с подобными задачами.

Помимо этого, наличие базы прецедентов позволяет внести элемент обучаемости данного алгоритма.



Рисунок 3 – Схема алгоритма формирования базы прецедентов

Главным преимуществом алгоритма формирования базы прецедентов, схема которого представлена на рисунке 3, является возможность пользователем выбора решения, который оказался верным при решении данной задачи.

Решение, полученное в результате логического вывода на основе базы продукционных правил и выбранное пользователем в качестве правильного, будет занесено в базу прецедентов с исходными условиями задачи. В результате данного действия, при решении аналогичной задачи в следующий раз пользователь получит эту рекомендацию в качестве априорной, т.е. полученной на основе анализа опыта решения подобной задачи.

## Заключение

Таким образом, предложенный в данной работе модифицированный алгоритм логического вывода с использованием в качестве базы знаний нечеткой FuzzyOWL-онтологии предметной области, набора продукционных правил, а также включающий в себя процесс анализа прецедентов, позволяет:

- снизить риски потери возможных результатов работы блока логического вывода, основанного на анализе предметной онтологии;
- увеличить гибкость процесса вывода рекомендации, предоставляя пользователю более широкий выбор вариантов, упорядоченных по степени их релевантности;
- приблизить формализованное представление знаний о предметной области к естественному для человека варианту их представления и восприятия;
- обеспечить получение пользователем возможности принятия решений как с учетом анализа закономерностей конкретной области, так и на основании опыта пользователей, уже сталкивавшихся с подобными задачами;
- обеспечить возможность обучаемости

системы за счет динамического формирования базы прецедентов.

## Библиографический список

[Дворянкин и др., 2008] Дворянкин А. М., Сипливая М. Б., Жукова И. Г. Интеграция рассуждений по прецедентам и онтологии в интеллектуальной системе поддержки инженерного анализа в области контактной механики // Известия Волгоградского государственного технического университета. Волгоград. – 2008. -№4. - Том 2.

[Мошкин и др., 2015] Мошкин В.С., Ярушкина Н.Г. Логический вывод на основе нечетких онтологий // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VIII-й Международной научно-практической конференции (Коломна, 18-20 мая 2015 г.). В 2-х томах. Т1. – М.: Физматлит, 2015. – С. 259–267.

[НГС, 2007] Нечеткие гибридные системы/ И.З.Батыршин, А.О.Недосекин, А.А.Стецко, В.Б.Тарасов, А.В.Язенин, Н.Г.Ярушкина. – М.: Физматлит, 2007.

[Ярушкина и др., 2015] Ярушкина Н.Г., Мошкин В.С. Применение алгоритма логического вывода на основе FuzzyOWL-онтологии // Радиотехника. – 2015. – № 6. – С. 68–72.

[Aamodt, 1994] Aamodt, A. Case-based reasoning: foundational issues, methodological variations, and system approaches/ A. Aamodt, E. Plaza// Communications. – 1994. – № 7. – С. 39–59.

[Kolodner, 1993] Kolodner J. L. (1993). Case-Based Reasoning. Los Altos, CA: Morgan Kaufmann.

[Leake, 1996] David B. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park, CA : AAAI Press/MIT Press, 1996, ISBN 0-262-62110-X.

[Pal, Shiu, 2004] Sankar K. Pal, Simon C. K. Shiu. Foundations of Soft Case-Based Reasoning. New Jersey: Wiley, 2004, ISBN: 978-0-471-64466-8.

[Trausan-Matu, 2008] Trausan-Matu, Stefan. A Framework for an Ontology-Based Information System for Competence Management. *Economy Informatics*, 1-4/2008, p.105.

[Nunes et al, 2012] Isabel L. Nunes and M'ario Simões-Marques Applications of Fuzzy Logic in Risk Assessment The RA\_X Case, *Fuzzy Inference System Theory and Applications*, Dr. Mohammad Fazle Azeem (Ed.), 2012. – pp. 21-40.

[Lei, 2012] Yaguo Lei. The Hybrid Intelligent Method Based on Fuzzy Inference System and Its Application to Fault Diagnosis, *Fuzzy Inference System Theory and Applications*, Dr. Mohammad Fazle Azeem (Ed.), 2012. – pp. 153-170.

## MODIFIED KNOWLEDGE INFERENCE METHODS BASED ON FUZZY ONTOLOGY AND SET OF USE CASES

Moshkin V.S., Yarushkina N.G.

*Ulyanovsk State Technical University, Russian  
Federation*

PostForVadim@yandex.ru

jng@ulstu.ru

This paper presents the results of applying the methodology of integration of production rules with fuzzy ontology as an example of solving the problem assessment of the local area network in an artificial rise in traffic. In addition, we consider the possibility of a modification of the algorithm by the parallel launch of the mechanism analysis of precedents in the process of inference of knowledge.

## Introduction

To ensure the high quality of decisions need to be assigned a model of object-oriented description of the structures of the subject area, allowing to identify,

analyze (including the visual representation) and manipulate all variety of objects and relations existing in the subject area (ABM).

Therefore, today is a very urgent task of integrating various forms of algorithms and representations and inference of knowledge to support decision-making in the design of complex systems.

## Main Part

Fuzzy ontology consisting of five levels reduce the subjective component evaluation data domain concepts of individual experts on the basis of its competence in a given subregion this area.

There are the most versatile approach to the construction of fuzzy ontologies - Methodology of Fuzzy OWL.

The experience can be formalized as a set of precedents that must be evaluated and taken into account within the framework of solving the problem of parallel to the main algorithm.

The arguments by precedents (case-based reasoning, CBR) - a method of forming conclusions, is not based on inference from the assumptions (logical reasoning), and on the search and analysis of cases, the formation of such conclusions in the past. Validation of conclusions can be part of CBR-process.

The presence of the base use case allows you to introduce an element of the learning algorithm.

## Conclusion

Thus, we proposed in this paper, a modified algorithm for inference using a knowledge base of fuzzy FuzzyOWL-domain ontology, a set of production rules, and including the process of analysis of precedents, you can:

- reduce the risks of loss of the possible outcomes of the unit of logical inference, based on the analysis of ontology;
- increase the flexibility of the process output recommendations, giving the user a wide range of options, ordered according to their relevance;
- provide users the opportunity to obtain decision-making both in terms of the analysis of patterns of a particular area, and based on the experience of users who are already faced with similar tasks;
- enable the learning system through the formation of dynamic database precedent.