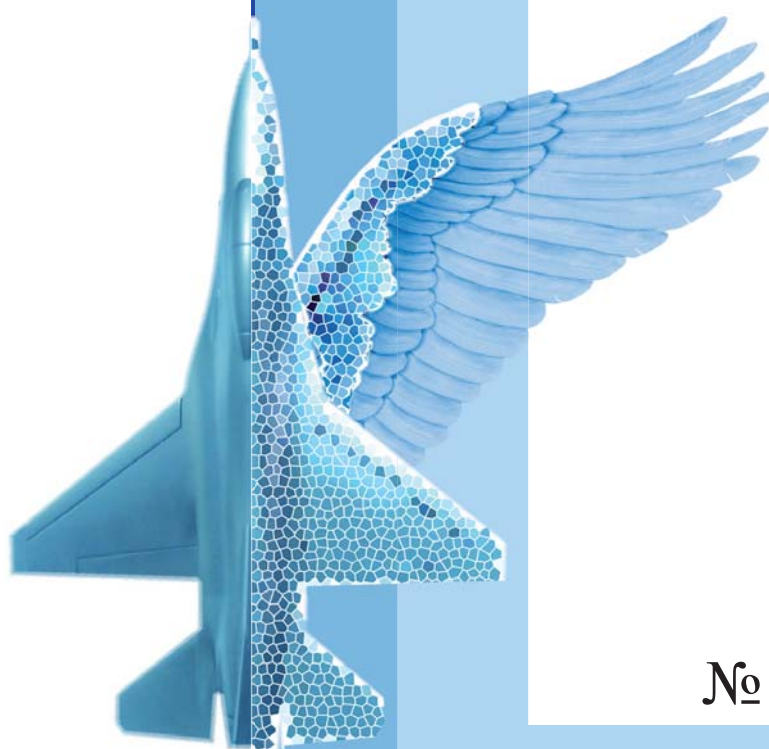


ОНТОЛОГИЯ ПРОЕКТИРОВАНИЯ



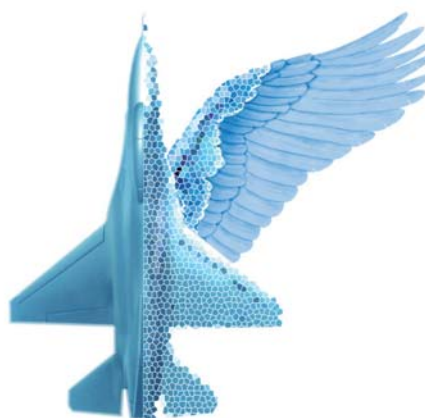
Том **8**
№ **3**(29)/2018

ОНТОЛОГИЯ
ПРОЕКТИРОВАНИЯ

Научный журнал

Том 8

№ 3(29)



EDITORIAL BOARD – РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Anatoly I. Belousov	Белоусов Анатолий Иванович, д.т.н., профессор, Самарский университет, Самара
Nikolay M. Borgest	Боргест Николай Михайлович*, к.т.н., профессор, Самарский университет, член ИАОА, Самара
Stanislav N. Vasiliev	Васильев Станислав Николаевич*, д.ф.-м.н., профессор, академик РАН, ИПУ РАН, Москва
Vladimir G. Gainutdinov	Гайнутдинов Владимир Григорьевич, д.т.н., профессор, КНИТУ-КАИ, Казань
Vladimir V. Golencov	Голенков Владимир Васильевич*, д.т.н., профессор, БГУИР, Минск
Vladimir I. Gorodetsky	Городецкий Владимир Иванович*, д.т.н., профессор, СПИИРАН, Санкт-Петербург
Yury A. Zagorulko	Загорюлько Юрий Алексеевич*, к.т.н., с.н.с., ИСИ СО РАН, Новосибирск
Anton V. Ivashchenko	Иващенко Антон Владимирович, д.т.н., профессор, Самарский университет, Самара
Alexander S. Kleshchev	Клещёв Александр Сергеевич*, д.ф.-м.н., профессор, ИАПУ ДВО РАН, Владивосток
Valery A. Komarov	Комаров Валерий Андреевич, д.т.н., профессор, Самарский университет, Самара
Sergey M. Krylov	Крылов Сергей Михайлович, д.т.н., профессор, СамГТУ, Самара
Victor M. Kureichik	Курейчик Виктор Михайлович*, д.т.н., профессор, Южный федеральный университет, Таганрог
Dmitry V. Lande	Ландэ Дмитрий Владимирович*, д.т.н., с.н.с., ИПРИ НАН Украины, Киев
Lyudmila V. Massel	Массель Людмила Васильевна*, д.т.н., профессор, ИСЭМ СО РАН, Иркутск
Aleksandr Yu. Nesterov	Нестеров Александр Юрьевич, д.филос.н., профессор, Самарский университет, Самара
Dmitry A. Novikov	Новиков Дмитрий Александрович, д.т.н., проф., член-корреспондент РАН, ИПУ РАН, Москва
Alexander V. Palagin	Палагин Александр Васильевич, д.т.н., проф., академик НАН Украины, Ин-т кибернетики, Киев
Semyon A. Piyavsky	Пиавский Семён Авраамович, д.т.н., профессор, ПГУТИ, Самара
Yury M. Reznik	Резник Юрий Михайлович, д.филос.н., профессор, Институт философии РАН, Москва
George Rzevski	Ржевский Георгий, профессор, Открытый университет, Лондон
Peter O. Skobelev	Скобелев Петр Олегович, д.т.н., НПК «Разумные решения», Самара
Sergey V. Smirnov	Смирнов Сергей Викторович*, д.т.н., ИПУСС РАН, Самара
Peter I. Sosnin	Соснин Петр Иванович*, д.т.н., профессор, УлГТУ, Ульяновск
Dzhavdet S. Suleymanov	Сулейманов Джавдет Шевкетович*, д.т.н., профессор, академик, вице-президент АН РТ, Казань
Boris E. Fedunov	Федунов Борис Евгеньевич*, д.т.н., профессор, ГосНИИ авиационных систем, Москва
Altynbek Sharipbay	Шарипбай Алтынбек*, д.т.н., профессор, Институт искусственного интеллекта, Астана
Boris Ya. Shvedin	Шведин Борис Яковлевич, к.психол.н., ООО «Дан Роуз», член ИАОА, Ростов-на-Дону

* - члены Российской ассоциации искусственного интеллекта - http://www.raai.org/about/about.shtml?raai_list

Executive Editorial Board - Исполнительная редакция

Chief Editor	Skobelev P.O.	Главный редактор	Скобелев П.О.	в.н.с. ИПУСС РАН
Deputy Chief Editor	Smirnov S.V.	Зам. главного редактора	Смирнов С.В.	зам. директора ИПУСС РАН
Executive Editor	Borgest N.M.	Выпускающий редактор	Боргест Н.М.	директор изд-ва «Новая техника»
Editor	Kozlov D.M.	Редактор	Козлов Д.М.	доцент Самарского университета
Technical Editor	Simonov A.U.	Технический редактор	Симонова А.Ю.	редактор изд-ва «Новая техника»
Translation Editor	Korovin M.D.	Редактор перевода	Коровин М.Д.	инженер Самарского университета

CONTACTS FOUNDERS – КОНТАКТЫ УЧРЕДИТЕЛЕЙ

ИПУСС РАН

443020, Самара, ул. Садовая, 61
тел./факс.: +7 (846) 333 27 70

Смирнов С.В.
smirnov@iccs.ru

Самарский университет

443086, Самара, Московское шоссе 34, корп. 10
тел.: +7 (846) 267 46 47, факс.: +7 (846) 267 46 46

Боргест Н.М.
borgest@yandex.ru

ООО «Новая техника» (издательство)

Адрес редакции: 443010, Самара, ул.Фрунзе, 145, тел.: +7 (846) 332 67 84, факс.: +7 (846) 332 67 81

The journal has entered into an electronic licensing relationship with EBSCO Publishing, the world's leading aggregator of full text journals, magazines and eBooks. The full text of JOURNAL can be found in the EBSCOhost™ databases.



Журнал размещен в коллекции «Издания по естественным наукам» на платформе EastView.

The journal has been successfully evaluated in the evaluation procedure for the ICI Journals Master List 2014, 2015 and journal received the ICV (Index Copernicus Value) of 67.46 points (2014), 67.64 (2015), 77.98 (2016).

Журнал включён в перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание учёной степени кандидата наук, на соискание учёной степени доктора наук (Письмо Департамента аттестации научных и научно-педагогических работников Минобрнауки РФ от 01.12.2015 № 13-6518) по группам научных специальностей 05.13.00 и 05.07.00.

Журнал включён в РИНЦ. Пятилетний импакт-фактор РИНЦ 1.000 (2013), 0.895 (2014), 1.305 (2015), 1.037 (2016), 0.913 (2017).

Журнал зарегистрирован в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство ПИ № ФС 77-70157 от 16.06.2017 г. (ранее выданное свидетельство ПИ № ФС 77-46447 от 07.09.2011 г.)

http://agora.guru.ru/scientific_journal/

© Все права принадлежат авторам публикуемых статей
© ООО «Новая техника» - «New Engineering» Ltd., 2011-2018
© Самарский университет - Samara University, 2015-2018
© ИПУСС РАН - ICCS RAS, 2015-2018



Отпечатано в ООО «Новая техника», г. Самара, пр.К.Маркса, 24-76.
Дата выхода 29.09.2018. Тираж 300 экз. Свободная цена. (6+)

СОДЕРЖАНИЕ

ОТ РЕДАКЦИИ

20 ЛЕТ «ПУМСС»: КРАТКИЕ ИТОГИ 321-322

ОБЩИЕ ВОПРОСЫ ФОРМАЛИЗАЦИИ ПРОЕКТИРОВАНИЯ: ОНТОЛОГИЧЕСКИЕ АСПЕКТЫ

С. Фогель, С. Рудольф 323-346
ПРОЕКТИРОВАНИЕ СЛОЖНЫХ СИСТЕМ ПРИ ПОМОЩИ ЯЗЫКОВ
ПРОЕКТИРОВАНИЯ: МЕТОДЫ, ПРИМЕНЕНИЕ И ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ

ПРИКЛАДНЫЕ ОНТОЛОГИИ ПРОЕКТИРОВАНИЯ

Д.А. Губанов, Д.А. Новиков 347-365
МЕТОДЫ ИЗВЛЕЧЕНИЯ И АНАЛИЗА ТЕРМИНОЛОГИЧЕСКИХ СТРУКТУР
СМЕЖНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ (НА ПРИМЕРЕ МЕТОДОЛОГИИ)

**В.А. Головкин, В.В. Голенков, В.П. Ивашенко, В.В. Таберко, Д.С. Иванюк,
А.А. Крощенко, М.В. Ковалёв** 366-386
ИНТЕГРАЦИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С БАЗАМИ ЗНАНИЙ

О.Ю. Марьясин 387-399
ПРОЕКТИРОВАНИЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ УПРАВЛЕНИЯ
ЗДАНИЕМ С ИСПОЛЬЗОВАНИЕМ ОНТОЛОГИЙ

Ю.К. Горелов, С.К. Киселев 400-411
ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЧЕСКОГО ПОДХОДА ПРИ ПРОЕКТИРОВАНИИ
МНОГОФУНКЦИОНАЛЬНОГО АВИАЦИОННОГО ИНДИКАТОРА

ИНЖИНИРИНГ ОНТОЛОГИЙ

Н.В. Максимов, О.Л. Голицына, М.Г. Ганченкова, Д.В. Санатов, А.В. Разумов 412-426
СЕМАНТИЧЕСКОЕ ЯДРО ЦИФРОВОЙ ПЛАТФОРМЫ

А.С. Клещев, В.А. Тимченко 427-448
РЕАЛИЗАЦИЯ ОБОЛОЧКИ И ПОРТАЛА ЗНАНИЙ ПО ВЕРИФИКАЦИИ
МАТЕМАТИЧЕСКИХ ДОКАЗАТЕЛЬСТВ НА ПЛАТФОРМЕ IASRAAS

МЕТОДЫ И ТЕХНОЛОГИИ ПРИНЯТИЯ РЕШЕНИЙ

С.А. Пиявский 449-468
МЕТОД УНИВЕРСАЛЬНЫХ КОЭФФИЦИЕНТОВ ПРИ ПРИНЯТИИ
МНОГОКРИТЕРИАЛЬНЫХ РЕШЕНИЙ

Ю.А. Скобцов, О.В. Ченгарь 469-479
ПАРЕТО-ОПТИМИЗАЦИЯ ПРОИЗВОДСТВЕННОГО РАСПИСАНИЯ
НА ОСНОВЕ МЕТОДА МУРАВЬИНЫХ КОЛОНИЙ

НАУЧНЫЕ КОНФЕРЕНЦИИ 480

CONTENT

FROM THE EDITORS	
20 YEARS OF «CSCMP»: BRIEF RESULTS	321-322
GENERAL ISSUES OF FORMALIZATION IN THE DESIGNING: ONTOLOGICAL ASPECTS	
S. Vogel, S. Rudolph COMPLEX SYSTEM DESIGN WITH DESIGN LANGUAGES: METHOD, APPLICATIONS AND DESIGN PRINCIPLE	323-346
APPLIED ONTOLOGY OF DESIGNING	
D.A. Gubanov, D.A. Novikov EXTRACTION AND ANALYSIS METHODS FOR THE TERMINOLOGICAL STRUCTURES OF RELATED SUBJECT AREAS	347-365
V.A. Golovko, V.V. Golenkov, V.P. Ivashenko, V.V. Taberko, D.S. Ivniuk, A.A. Kroshchanka, M. V. Kovalev INTEGRATION OF ARTIFICIAL NEURAL NETWORKS AND KNOWLEDGE BASES	366-386
O.Yu. Maryasin DESIGN OF MULTI-AGENT BUILDING MANAGEMENT SYSTEM WITH ONTOLOGIES	387-399
Y.K. Gorelov, S.K. Kiselev THE USE OF ONTOLOGY APPROACH FOR AIRCRAFT DISPLAY DESIGN	400-411
ONTOLOGY ENGINEERING	
N.V. Maksimov, O.L. Golitsina, M.G.Ganchenkova, D.V. Sanatov, A.V. Razumov SEMANTIC CORE OF DIGITAL PLATFORM	412-426
A.S. Kleshev, V.A. Timchenko IMPLEMENTATION OF THE SHELL AND KNOWLEDGE PORTAL FOR MATHEMATICAL PROOFS VERIFICATION ON THE IACPAAS PLATFORM	427-448
METHODS AND TECHNOLOGIES OF DECISION MAKING	
S.A. Piyavsky METHOD OF UNIVERSAL COEFFICIENTS FOR THE MULTI-CRITERIAL DECISION MAKING	449-468
Y.A. Skobtsov, O.V. Chengar PARETO-OPTIMIZATION OF THE PRODUCTION SCHEDULE BASED ON THE METHOD OF ANTS OF COLONIES	469-479
SCIENTIFIC CONFERENCES	480



20 ЛЕТ «ПУМСС»: КРАТКИЕ ИТОГИ 20 YEARS OF «CSCMP»: BRIEF RESULTS

Главное чудо – то, что мы живём.
Сергей Капица

**Дорогой наш читатель,
уважаемые авторы и члены редакционной коллегии!**

Институт проблем управления сложными системами Российской академии наук с партнёрами в период с 3 по 6 сентября 2018 года провёл в Самаре очередную XX Международную научную конференцию «Проблемы управления и моделирования в сложных системах» (ПУМСС)¹. Конференция прошла в формате, который в своё время предложил и долгие годы реализовывал ушедший от нас в прошлом году профессор *Виттик Владимир Андреевич*.

Конференция ПУМСС была задумана в конце 90-х годов прошлого столетия для того, чтобы предоставить академическому сообществу совместно с представителями органов государственной власти, проектных учреждений, промышленных предприятий и вузов площадку для дискуссий и обмена опытом в сфере управления сложными социотехническими системами. В то время подобные конференции, по крайней мере в России, были большой редкостью.

Первая конференция прошла в 1999 году и сразу же получила признание научной общест­венности, став впоследствии ежегодной, не потеряв своей актуальности до сегодняшнего дня. В работе конференции участвовали учёные и специалисты из России, Белоруссии, дру­гих стран СНГ, большинства стран Европейского союза, США, Турции, Швейцарии и Япо­нии. В разные годы активными участниками конференции были академики Николай Семиха­тов, Николай Кузнецов и Станислав Васильев, члены-корреспонденты РАН Дмитрий Нови­ков, Герман Себряков, Евгений Теряев, Рафаэль Юсупов, член-корреспондент РАН, летчик-космонавт СССР Владимир Соловьёв, начальник российского ЦУПа Максим Матюшин, профессор Сергей Капица и другие. В конференции принимали участие делегации органов государственной власти Самарской области, руководства города Самары. Своих представи­телей на ПУМСС направляли как крупные предприятия Самарской области - ОАО АВТОВАЗ, ОАО «Самаранефтегаз», ОАО «Самараэнерго» и другие, - так и созданные в Са­маре компании в сфере информационно-коммуникационных технологий «Разумные реше­ния», «Наукомп», российско-британская MagentA и др. Помощь предприятий позволила провести ряд конференций на теплоходах, следовавших по Волге и Каме от Самары до Яро­славля, Перми и Астрахани. Площадками для конференции были также Дом учёных Самар­ского научного центра РАН, конференц-залы Администрации города Самары. Финансовую поддержку проведению конференции многие годы оказывает РФФИ.

Тематика ПУМСС всегда отражала актуальные тренды мировой науки, ориентировала на лучшие практики применения научных достижений в различных областях. Так, уже на пер­вой конференции в 1999 г. в секции моделирования сложных систем были доклады, посвя­щённые концептуальному онтологическому моделированию предметных областей, и обсуж­дение этой проблематики стало отличительной чертой ПУМСС на все последующие годы. В 2001 году под влиянием активного участника всех конференций ПУМСС известного бри­танского профессора Георгия Ржевского началось обсуждение теории сложности и практики

¹ XX International Conference "Complex Systems: Control And Modeling Problems" (CSCMP), <http://www.iccs.ru/cscmp/cscmp.html>

её преодоления путём создания мультиагентных систем. Эти темы также определяют лицо ПУМСС.

За годы проведения ПУМСС на конференции работали секции информационного взаимодействия в сложных системах, управления и оптимизации в сложных технических системах, регионального управления, управления и моделирования в экологических системах, измерения, контроля и диагностики в экстремальных условиях и т.п. Новыми в последние годы стали темы цифрового сельского хозяйства и биоинформатики.

Традиционно тон каждой конференции задают специально подбираемые Программным комитетом пленарные доклады, посвящённые крупным актуальным проблемам науки и практики. Заметными на XX-й ПУМСС стали пленарные доклады профессора Сергея Кротова «К вопросу об интеллектуальной составляющей экономики России, переводимой на цифровые технологии», профессора Сергея Сергеева «К проблеме техномодификации и интеграции человека в техногенном мире», генерального директора АО «НИИ микроэлектронной аппаратуры "Прогресс"» Василия Шпака «Пути развития российского информационно-технологического комплекса "в обгон" мировых лидеров».

Ряд докладов XX-й Международной конференции ПУМСС, безусловно, будет интересен читателям нашего журнала. По рекомендации Программного комитета ПУМСС журнальные версии таких докладов будут готовиться к публикации. Уже в этом номере представлены статья немецких специалистов С. Фогеля и С. Рудольфа, существенно дополненная по сравнению с докладом работа учёного из Ярославля О.Ю. Марьясина, статья коллег из Санкт-Петербурга и Севастополя Ю.А. Скобцова и О.В. Ченгарь.

Мы поздравляем ИПУСС РАН и его партнеров в проведении Международной конференции ПУМСС с юбилейной XX-й сессией, желаем конференции новых достижений и надеемся на тесное сотрудничество в будущем.



Уважаемый автор!

Мы по-прежнему ждём *новых результатов* в области формализации знаний, практического использования таких моделей для создания лучшего будущего!

Ontologists and designers of all countries and subject areas, join us!

COMPLEX SYSTEM DESIGN WITH DESIGN LANGUAGES: METHOD, APPLICATIONS AND DESIGN PRINCIPLES¹

S. Vogel¹, S. Rudolph²

Institute for Aircraft Design, University of Stuttgart, Stuttgart, Germany

¹*Samuel.Peter.Vogel@gmail.com*, ²*Rudolph@ifb.uni-stuttgart.de*

Abstract

Graph-based design languages are presented as a method to encode and automate the complete design process and the final optimization of the product or complex system. The Unified Modeling Language (UML) is used to represent the design language which models the design process. A design language consists of a vocabulary (i.e. the digital building blocks) and a set of rules (i.e. the digital composition knowledge) along with an executable sequence of the rules (i.e. the incremental digital encoding of the design process). The rule-based mechanism instantiates a central and consistent global product data structure (the so-called design graph). Upon the incremental generation of the abstract central model, the domain-specific engineering models are automatically generated, remotely executed and their results are fed-back into the central design model for subsequent design decisions or optimizations. The design languages are manually modeled and automatically executed in a so-called design compiler. Up to now, a variety of product designs in the areas of aerospace (satellites, aircraft), automotive (space frame structures, automotive cockpits), machinery (robots, digital factory) and consumer products (coffeemakers, exhaust systems) have been successfully accelerated and automated using graph-based design languages. Different design strategies and mechanisms have been identified and applied in the automation of the design processes. Approaches ranging from the automated and declarative processing of constraints, through fractal nested design patterns, to mathematical dimension-based derivation of the sequence of design actions, are used. The existing knowledge for a design determines the global design strategy (i.e. top-down vs. bottom-up). Similarity-mechanics in the form of dimensionless invariants are used for evaluation to downsize the solution for an overall complexity reduction. Design patterns, design paradigms (i.e. form follows function, or function follows form) and design strategies (divide and conquer) from information science are heavily used to structure, manage and handle the design complexity.

Key words: *design languages, design automation, design method, design principles, design ontology.*

Citation: *Vogel S., Rudolph S. Complex System Design with Design Languages: Method, Applications and Design Principles. *Ontology of designing*. 2018; 8(3): 323-346. - DOI: 10.18287/2223-9537-2018-8-3-323-346.*

Introduction

The digitization of industrial processes, e.g. in the context of Industry 4.0, makes new design processes possible and necessary. The automation of the product development process promises a considerable increase in efficiency. Especially designs and decisions of the very early concept phase have a very large influence on the later life cycle costs of the product [1]. The development of modern and more competitive products requires to go even closer to the limits of what is physically feasible in order, for example, to squeeze the last bit of weight advantage or efficiency out of a product or system. Modern products are integrating typically multiple physical domains (mechanics, thermodynamics, electronics, logistics, ...) as well as a lot of system levels consisting of sub-systems or parts that mutually build on each other. The combination of both, multiple domains together with a number of system entities, results in a high level of design and process complexity that has to be handled. Digitized design processes can be used to cope with this complexity and to find more optimal product designs in even earlier project phases. This digitization mainly comprises the comput-

¹ The article is published in expanded content on the recommendation of the Program Committee of the XX International Conference "Complex Systems: Control and Modeling Problems" (CSCMP-2018). Samara, Russia. September 3-6, 2018.

er-aided synthesis of designs (CAD) including the automated generation of functional validation calculations and simulations (structural mechanics, fluid mechanics, controls...). In fact, a virtual product design shall be automatically generated and optimized based on given product requirements to optimally meet the performance targets.

In this paper graph-based design languages are presented as a method to implement such digital and re-executable representations of (conceptual) design actions. At first, the method of graph-based design languages itself is explained. The method is proven for more than fifteen years and has been mainly developed in the Similarity Mechanics Group of the Institute for Statics and Dynamics, which moved now to the Institute of Aircraft Design at the University of Stuttgart. Second, scientific applications are shown as well as an early industrial stage application. Finally, a collection of design principles to handle the complexity in product design is presented that has been identified in the scientific work with design languages over the past years.

1 Method

The method of graph-based design languages [2] is a further evolution step of generative, computer-based design synthesis methods [3]. These design synthesis methods can be divided in to string-based, shape-based and graph-based design representations. From the viewpoint of the authors, graph-based design languages belong to the most generic and abstract means of knowledge representation across different domains due to its graph representation. Alternative computer-based synthesis methods such as L-Systems or Shape Grammars [3] define a rule set on elementary shapes (vocabulary) which is recursively called in a production system to generate more complex shapes. Graph-based design languages expand this concept by generalizing the vocabulary to conceptual objects together with an adaptive procedural rule sequence, the so-called production sequence. Along with these approaches, there are other solutions in formalizing the process of designing and creating automatic design systems based on formal knowledge worth noting [4-9].

1.1 Philosophical Motivation

Rudolph gives a philosophical motivation for the design language concept in [10]. First it is observed that during the product design process different areas of concept, each with different levels of knowledge, are traversed. It is distinguished between the first area called 'believe' which covers uncertain design targets as *simplicity*, *aesthetics* or *adequacy*. The second concept is 'ability' which covers more concrete but still not exact formulated design aspects as *Design for Manufacturing*, *Design for Assembly*, *Design for Recycling*. The third concept covers exact aspects and is called 'knowledge'. It contains physical formulas and other reproducible, mathematically formalizable and provable laws and know-how. Figure 1 shows a schematic design process starting from an idea that is hosted in the concept of 'believe'. During the iterative design process different solution concepts are derived from the given idea and product embodiments are synthesized as product variants. These variants are validated towards specified requirements. The iterative procedure is conducted until a variant meets the requirements and becomes the final product, see Figure 1.

In order to formalize and digitalize this design process all three aspects have to be represented in a single, unified description. Rudolph proposes a language-based representation that is closely related to natural languages which are a convenient candidate as they are able to cover all three conceptual areas presented above. This language-based representation is called a *graph-based design language*. The linguistic aspects of natural languages are reinterpreted here in the engineering application in the following way [2].

- *Syntax of the design language:*

All designs that can be combinatorically represented by the classes in the class diagram.

- *Semantics of the design language:*
All designs that are technically or physically meaningful (e.g. it may depend of the application whether a car with six wheels makes sense, no collisions between parts and components, ...).
- *Pragmatics of the design language:*
The designs that are optimal with respect to given requirements and boundary conditions.

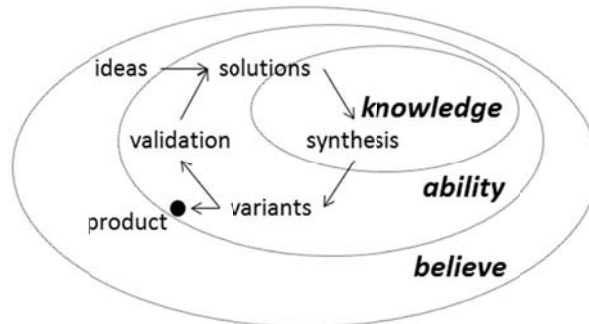


Figure 1 - Embedded areas of concept that are visited during an iterative product design process, starting from an idea.
Figure reproduced from [10]

The linguistic aspects are embedded within each other as the optimal designs (pragmatics) are a subset of the technically and physically meaningful designs (semantics). These meaningful designs are at the same time a subset of the designs that can be created or represented in the design language. But this is in fact the underlying research hypothesis of design languages that *design is computable* and that the optimal designs as a subset within the generated meaningful designs may be found via a *design evaluation method* [11].

1.2 Graph-Based Design Language

The following section explains and defines the method of graph-based design language. The figure 4 shows the main components of a design language in its graph-based representation on one page.

1.2.1 Unified Modeling Language

For the formal representation and the concrete modelling of the design process the Unified Modeling Language (UML) is used [12]. The UML's origin is in object-oriented software engineering to graphically model and specify object oriented software. Subsets of the UML are used in systems engineering as the modeling language SysML. The UML provides ready-to-use modeling diagrams and tools for the representation of (engineering) ontologies (class diagram and instance diagram) as well as for representing sequential and branched processes (activity diagram) that are used in graph-based design languages to create a graphical model of a product's design process. The original naming of the design languages building blocks was based on linguistics as stated above as the UML was later introduced as modeling language [13]. The following headers reflect the duality between the design languages components' UML nomenclature and the 'original' linguistic based nomenclature in parenthesis.

Using the UML for the representation of the product design process allows a natural integration of the software domain into the product design process, which is present in most modern products. Modern products are typically made up of mechanical and electrical parts with a corresponding control unit that executes at least a product-specific controller software. Therefore software engineering becomes an integrated and important part of the product development process.

1.2.2 Class Diagram (Vocabulary)

The class diagram represents an ontology of the product that is to be designed. The product is decomposed into its subsystems, components and even more granular entities that are assigned to classes. These classes are enriched with parameters that represent e.g. physical or cost variables. In this way, parametrized objects that are instantiated from classes form the vocabulary of a graph-based design language instead of words in natural languages. A class diagram is shown in figure 2 top for an exhaust aftertreatment system that reduces pollutants from a combustion engine's exhaust. The class *Catalyst* is a kind of *ExhaustSystem* through inheritance. Within the class diagram, associations can be drawn as links between classes to represent relationships between them. In the example in figure 2 top, the *ExhaustSystem* is connected to a *CombustionEngine* whose exhaust has to be cleaned and the *ExhaustSystem* itself is connected with the *Environment*, where the cleaned exhaust gas escapes to. These associations define which elements can be linked during an instantiation. Thereby the class diagram represents the (maximal combinatorial) template of the product that hosts all the information needed during the design process. Inheritance relationships can be defined between the classes, as is customary in object-oriented modeling. Abstract classes can be defined that cannot be instantiated. So components and entities can be mapped directly on classes on different levels of abstraction and detail.

Equations and constraints between the class parameters can be additionally modeled in the classes and are processed in an integrated solution path generator [14]. The equation and constraint network that is built on the instantiation of the classes is automatically solved in the solution path generator with an integrated computer algebra system. In the UML class definition physical dimensions can be assigned as data types to class parameters. This becomes especially important for the dimension analysis presented in the design principles section below.

1.2.3 Instance Diagram (Design Graph)

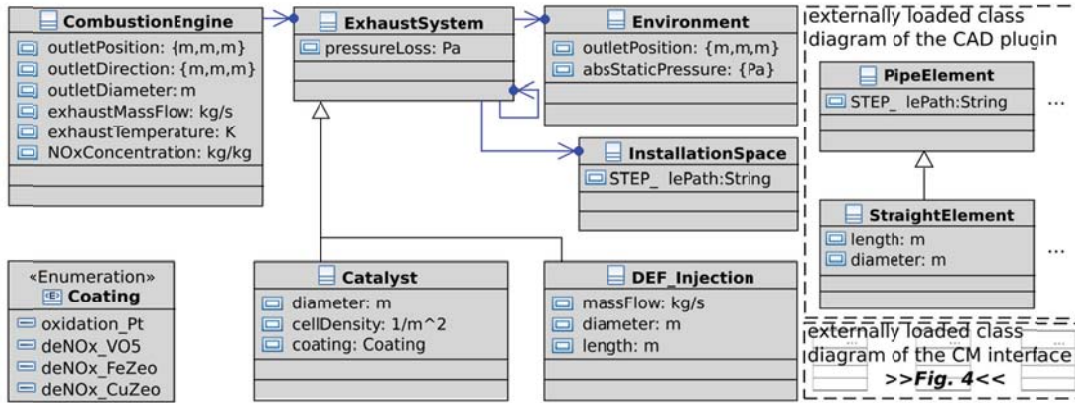
The classes from the class diagram can be instantiated into instances. The instantiated objects get an unambiguous name and the parameters defined in the class are provided with concrete values. The instances of associated classes can be linked with each other. The set of linked instances is called *design graph* in the context of graph-based design languages (figure 2 bottom). The instantiated objects form the nodes and the links form the edges of the graph. The specific values are stored in the parameters within each node. Thus, the topology of a product (an alternative name would be product architecture) can be mapped via the graph and the parametric of a product via the parameterization in the nodes. This design graph plays the role of the central data model in the virtual product design with design languages.

1.2.4 Rules (Grammar)

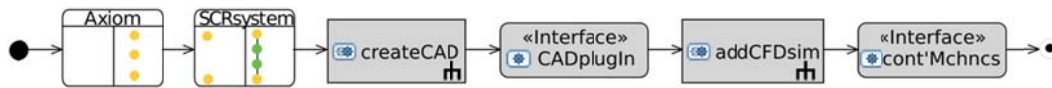
The engineering entities of the class diagram are rule-based instantiated into objects with specific parameter values. Graphical rules with a left-hand side (LHS) and a right-hand side (RHS) define the instantiation as manipulation on the instance diagram. Again, the design instances in the design graph are linked with each other according to the associations that are defined in the class diagram. These associations define the possible connection of instances, also called instance patterns, in the graphical rules. The instance pattern on the LHS of the graphical rule is looked-up in the design graph and replaced with the instance pattern on the RHS. The first rule is called 'axiom' and has an empty LHS as the design graph is empty in the beginning. The figure 2 center left shows the axiom rule that introduces the boundary conditions of an exhaust aftertreatment system, which comprises of a given combustion engine with its specific parameters (not all shown), the environment and the installation space as STEP geometry file that defines the available space for engineering the exhaust system. In the 'axiom' typically the requirements and given boundary conditions are

defined. The graphical rule in figure 2 center right shows the incremental design step that adds the main building blocks of an SCR system, *DEF_Injection* and *Catalyst*, to the initially created instance of the *CombustionEngine* class.

Class Diagramm

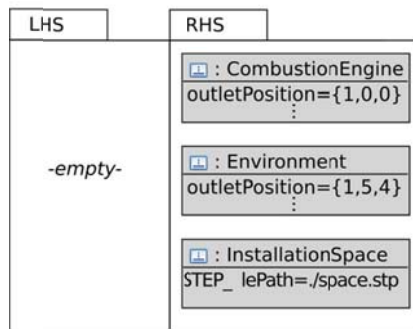


Production System

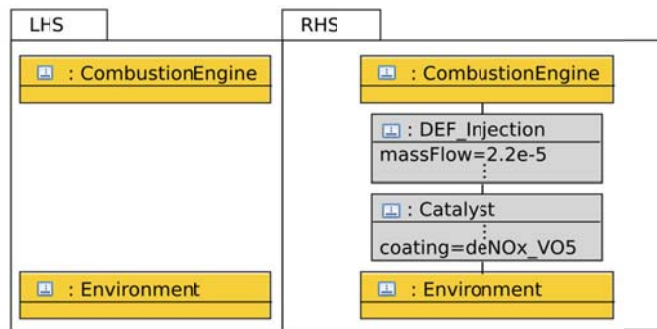


Graphical Rules (if-then scheme)

Rule "Axiom"



Rule "SCRsystem"



Design Graph = Central Data Model (UML instance diagram)

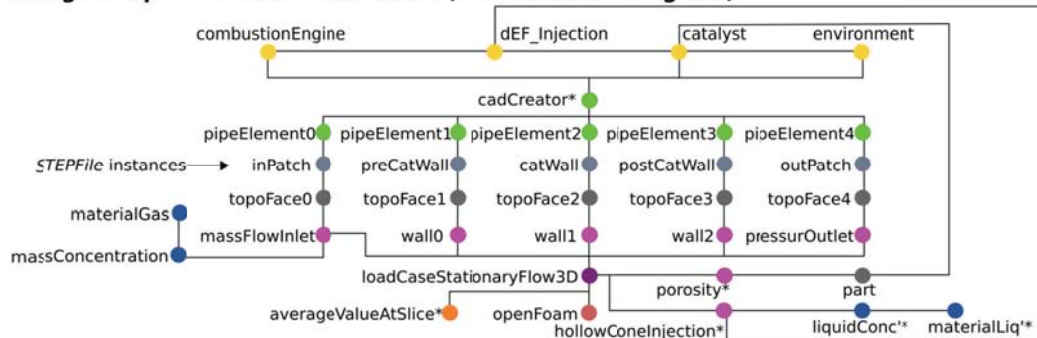


Figure 2 - Schematic graph-based grammar of an exhaust system and its main building blocks

1.2.5 Production System

The design process itself is then split into an incremental rule sequence. This predefined rule sequence is graphically modeled in an activity diagram and is called production system. Besides the graphical rules as building blocks an activity diagram can hierarchically host sub-activities which are activity diagrams themselves. As a third object, interface calls can be modeled that trigger the execution of engineering applications in so-called process chains that are described in the following section. The activity diagram in figure 2 top center shows from the left two calls of graphical rules followed by a two times alternating sub-activity and interface call. As a fourth element so-called decision nodes are available to branch the rule sequences in the production system in dependence of the state of the design graph (figure 3).

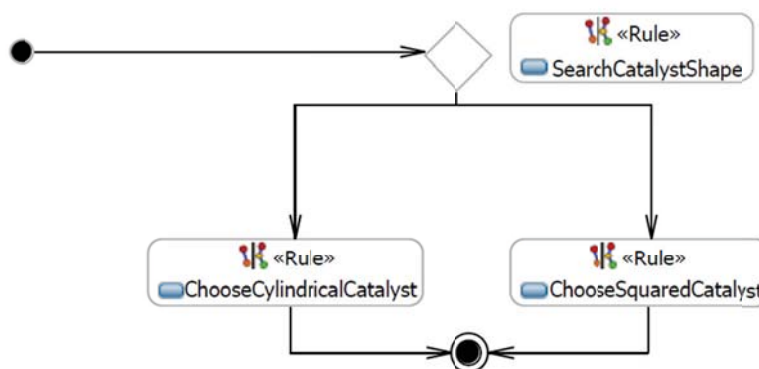


Figure 3 – Decision node (rhombus) in the activity diagram to trigger different rules based on the LHS pattern in rule *SearchCatalystShape*

With this feature a branching design process can be realized that adaptively and dynamically reacts on different model states and inputs as different rule sequences can be triggered. In figure 3 one of the two *Choose*Catalyst* rules is triggered, which differ in the geometric form of the catalyst that is added to the model. Which of the rules is chosen depends on a parameter in an instance on the LHS of the *SearchCatalystShape* rule on the top right in figure 3. In this way adaptive design sequences can be realized to create a wide variety of different product configurations.

1.2.6 Information Architecture

The figure 4 shows the information architecture that hosts the previous presented elements and shows their interaction. The class diagram (vocabulary), the rules, as well as the production system, represent a digital blueprint of the design process, wherein the design knowledge is encoded in the form of a design language (figure 4 left). The manual modeling of the class diagram, the rules and the production system is done in a so-called design compiler (further details in the following section). On the right side of figure 4 so-called process chains are shown. Process chains represent the automated creation of engineering models in the design language. This is realized with unidirectional model-to-text transformations between the abstract central model (design graph as UML instance diagram) and the product engineering models as CAD, simulation models, etc. The process chains extract the required information from the central design graph and create the CAD and simulation models automatically by executing the model-to-text transformations that are stored in the process chains. The results of the simulations runs are partially fed back to the design graph for influencing subsequent design rules and operations. Closed-loop optimization as well as adaptive and self-controlling design processes, using the decision nodes above, can be realized in this way.

This architecture with a central model (“*single source of truth*”) has multiple advantages. There are no longer different and/or outdated model versions since upon a model update in the central

model the process chains can be triggered to get an automatic update of all engineering models. There is no need of model-to-model transformations between each engineering model as all model updates are done or triggered in the central model with an subsequent update via calling the process chains. This reduces the number of model-to-model transformations drastically, that are necessary to have valid models, and drastically reduces the need for tracing and managing model changes.

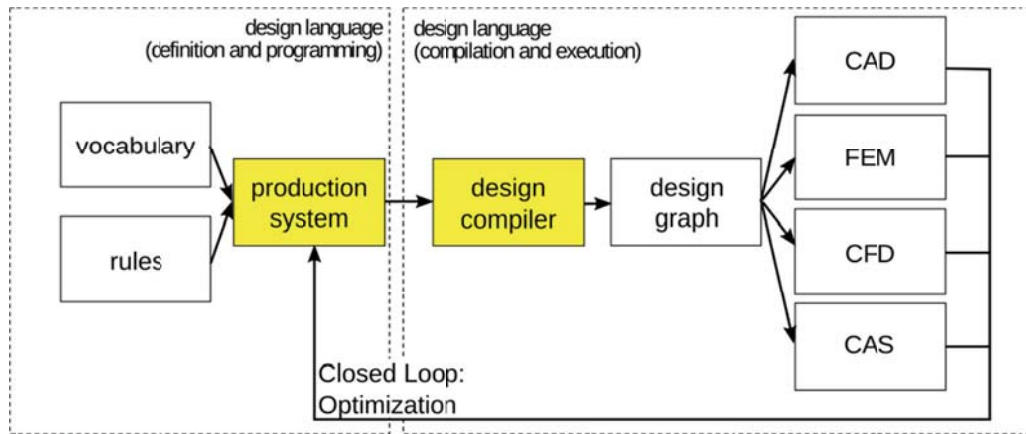


Figure 4 - Information architecture of graph-based design languages

1.2.7 Design Compiler

In analogy to programming languages a design compiler is used for modelling and execution of the graph-based design languages. The commercially available Design Compiler 43 (<https://www.iils.de/>) in its Version V2 was used in the applications projects presented below. The Design Compiler 43 is a standalone application based on the Eclipse IDE that provides the following functionalities and tools to model and execute graph-based design languages:

- Graphical editor to manually model the class diagram (figure 2 top).
- Process chains as engineering plugins in the form of class diagrams for modeling engineering tasks as CAD geometry creation, simulation model creation (CFD, FEM) and product integration problems (wiring, piping, packaging,...) that can be loaded into the central class diagram (figure 2 top right). In this way different ontologies from different engineering areas can be combined in the central class diagram.
- Graphical editor to manually model the production system including its sub elements as graphical rules, sub activities and interface calls (figure 2 center).
- Graphical view of the design graph with filter function to restrict the scope of view to sub patterns (figure 2 bottom).
- Design language debugging mode to execute the rules of the production system step by step with the possibility to stop during execution.
- *Java Rule* functionality to execute and use Java code on the design languages execution to manipulate the design graph code-based, especially useful for complex manipulations that contain difficult patterns and/or control structures as loops.

The solution path generator functionality is provided by the design compiler as plugin which solves, first, the equation system modeled in the class diagram in terms of a solution sequence and, second, passes the found solution sequence to a computer algebra system for solving the equation system symbolically or numerically if necessary. The bidirectional solution path generator writes the results back into the design graph as last execution step of the plugin.

2 Applications

A broad range of products has been designed using graph-based design languages. From applications in aerospace [15, 16] through consumer products [17] and off-road machinery components [18] up to automotive [19], the method has been successfully applied mainly in a scientific context. The scope of design languages has been prototypically extended up to downstream stages of the life-cycle by generating and designing the digital factory for a product in addition to the product itself [20]. Additional work has been done in the implementation of algorithms to automate engineering tasks as routing of cables and wires [16] and the automated creation of pipework in given installation spaces [21]. These intelligent wiring and piping algorithms become necessary as graph-based design languages fully automate the design process and therefore need to be able to create an intelligent integration and interaction of system components in given installation spaces for different product architectures.

2.1 Aeronautics: Air Cabin Design

Figure 5 shows results of a graph-based design language that automates the layout design of an aircraft cabin [22, 16]. Beginning with the requirements, the designer can (manually) define the seating requirements based on a number of ratios. A ratio might define how many passengers share the same lavatory in a certain class. The aircraft main dimensions are as well given as a requirement. The proposed seating configuration within the aircraft's hull can be additionally manually edited in a graphical editor that appears during the execution of the design language. The subsequent design process is schematically shown in figure 5. At first, an initial CAD model of the aircraft cabin is created (figure 5 top left) based on the previously found or manually edited seating configuration. From this CAD model the available routing space for cable routing is rule-based extracted. Due to the previous automated CAD model generation, the information of the position, size and shape of the area, which is accessible for routing, is explicitly available in the central data model (figure 5 top mid). In the following step the equipment boxes are positioned in the available routing space. This is done via a parametrization along the aircrafts main dimensions which can be later varied in an optimization run (figure 5 top right). Afterwards, the equipment boxes are subtracted from the routing space and the remaining space is meshed for conducting collision detections for the subsequent cable routing (figure 5 left bottom). Then the search algorithm, a modified A*-algorithm which is available as engineering plugin in the design compiler, is executed to identify the cable routes within the routing space (figure 5 mid bottom). Finally, collision-free CAD models of the cables are created which are used for an evaluation of the aircraft cabin configuration. Evaluation metrics comprise of cable length and weight as well as electromagnetic compatibility. Additional constraints in the aircraft cable design, as minimum distances between the cables of redundant systems, can also be taken into account by the design compiler's integrated routing functionality.

Using the graph-based design language reduces the time needed for an aircraft cabin layout from many weeks and months to a few hours. The complexity of the interaction of the coupled systems and components is handled through the interplay of the production system and the design rules together with the intelligent algorithms to solve the integration tasks of positioning and wiring the electrical components. Different aircraft cabin designs can thus be automatically evaluated and optimized in terms of total weight, cable length and wiring compatibility and validity [22]. Figure 6 shows a final cable routing together with the comparison of the results of two cabling variants with a different number of distribution boxes (SPDB). The diagrams on the right are showing the different resulting cable weights for the different networks and subsystems.

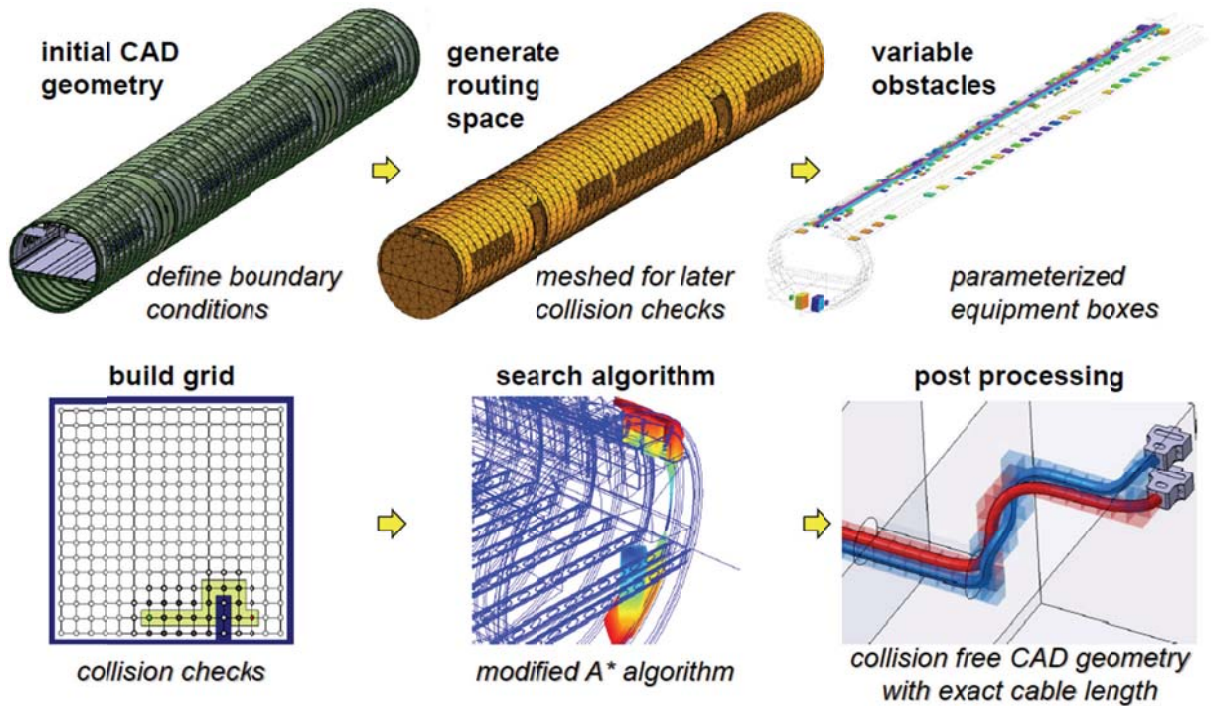


Figure 5 - Aircraft cabin design with automated generation of electric and climate system generated with a design grammar [22]

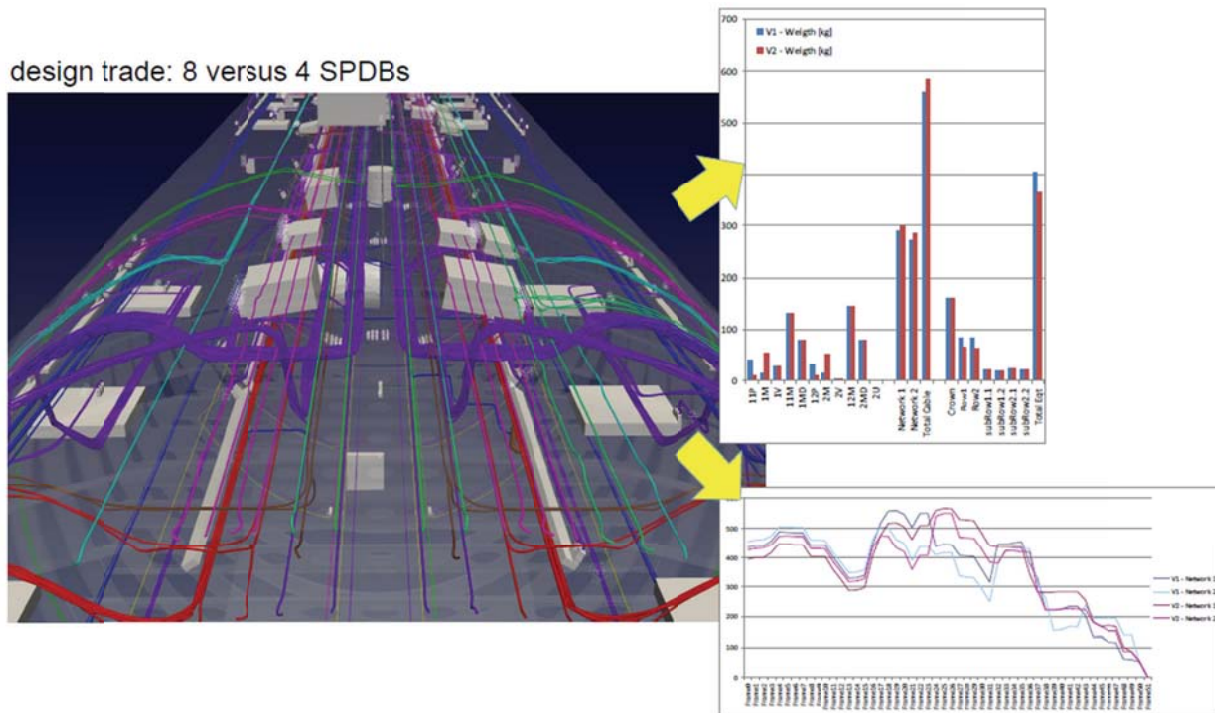


Figure 6 – Final aircraft cabin cable configuration in overhead area. Diagrams on the right show a schematic comparison of two cable variants (8 vs. 4 distribution boxes) in terms of cable weights [22]

2.2 Aerospace: Satellite Design

Figure 7 shows the result and intermediate steps of a graph-based design language for creating a small satellite [15]. The design language starts from a given mission that defines payload, target

orbit, energy and information demand of the payload as requirements. The design process of creating an optimal satellite for a given requirement set starts in [15] by solving the so called *enumeration problem*. In the enumeration problem different system topologies are synthesized and analyzed as different system topologies can match the requirements. The design for a given system topology is created rule-based in the design language presented in [15]. For each system topology the *configuration problem* has to be solved which comprises of integrating the subsystems and components, selected during solving the enumeration problem, in the available space (the so-called packaging). At last, the *integration problem* has to be solved, which means that the functionality of the whole systems has to be checked under all relevant loads and physical conditions. For conducting this integration test the corresponding simulation models, as well as the cables routes of the satellite, have to be generated. These steps have to be executed and repeated many times on different modeling levels of detail to properly resolve the couplings between the systems [15].

Figure 7 shows the synthesis of the FireSat satellite [15]. The rule set creates a complete virtual mockup based on the given requirements (figure 7 top row). The design language includes the automated creation and dimensioning of the control systems as well as the validation. Critical figures, such as the mass, energy and momentum balance, are calculated and balanced. A mission-related communication system is chosen based on the mission figures. All critical subsystems are selected and in the subsequent configuration step spatially arranged (i.e. packaged) and finally, with the previously mentioned routing algorithm, connected by wires (figure 7 bottom right). The integration of the components is validated for the defined orbits in terms of thermal loads arising from the incoming sun light as well as heat sources from components and systems (figure 7 bottom left).

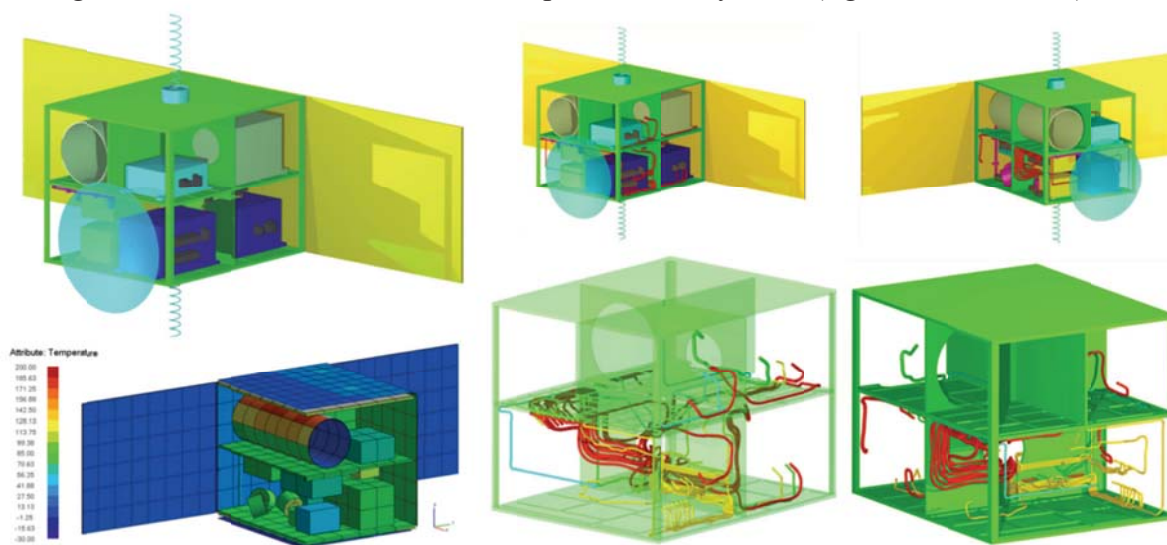


Figure 7 - Satellite design with graph-based design languages. Reproduced from [15]

Figure 8 shows the intermediate result of the FireSat design language for solving the enumeration problem of the communication system. For the given requirements transmission power and transmission data rate (speed) the best mass of each communication system topology can be seen.

Each combination of a transmitter and an antenna presents a unique topology and has its own mass to power to speed characteristic as shown in figure 8 in different colors in the central plot. During the execution of the design language this characteristic figures can be automatically created. Then the optimal communication system for a given requirements can be chosen based on this characteristic map. In cases where a chosen (sub-)system topology has still variable system parameters, a parametric heat map as in figure 9 can be used to find an optimal solution and to gain insight into

the system's behavior. The sensitivities of the calculated system characteristics values (vertical axis) are shown in dependence of given design parameters (horizontal axis).

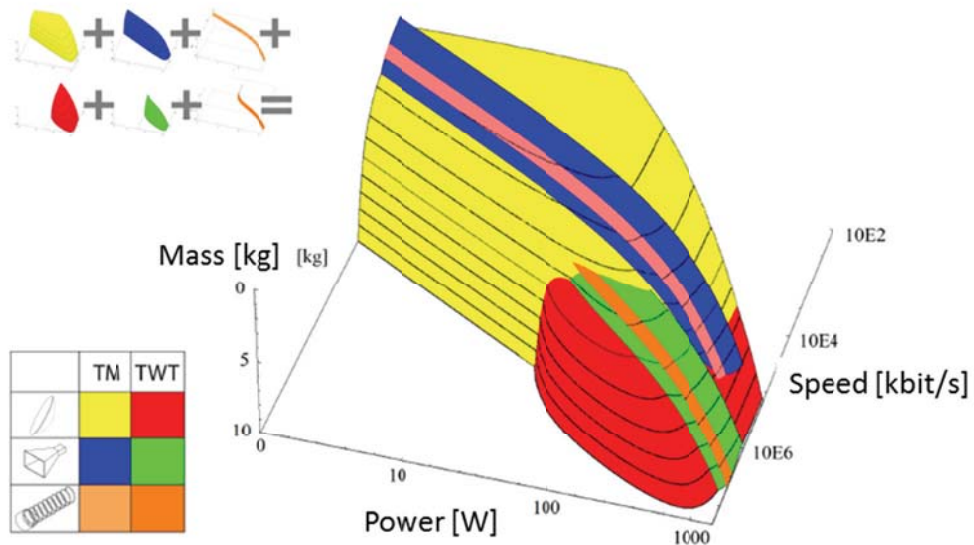


Figure 8 – Enumeration problem: The mass of different communication system topologies (colors) for required transmission power and speed. There are 3 antenna system variants (vertical) and 2 transmission amplifier variants (horizontal) [15]

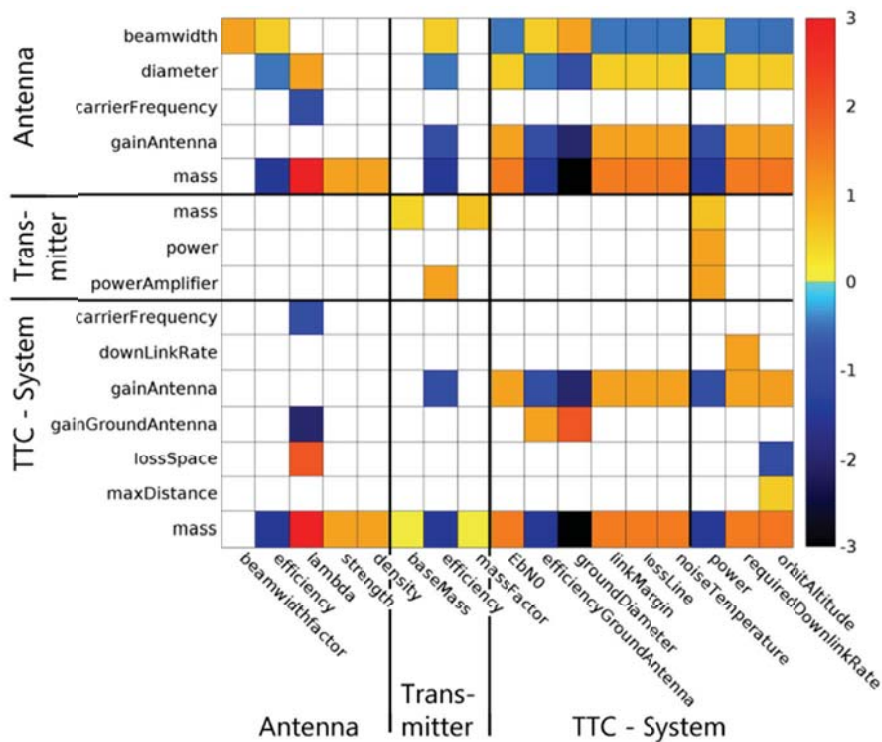


Figure 9 – Heat map showing the dependencies between the calculated communication system characteristics (vertical) and the given design parameters (horizontal) of a single communication system topology [15]

2.3 Manufacturing: From Design to Digital Factory

Graph-based design languages have also been proposed to bridge the gap between product design and manufacturing [20]. Design languages can support to close the current inconsistencies between the product design and the manufacturing engineering. Both disciplines are usually using different models, applications and data formats, even if both areas are closely coupled as changes in the product design can have a significant impact on the manufacturability in a given production infrastructure and vice versa. Arnold et al. implements a graph-based design language in [20] that automates the design of an aircraft panel. A plugin that automatically creates a digital factory for a given design is added. Using this approach in both, the design and the manufacturing planning, a central data model is shared in the form of the common design graph that extends over both interwoven domains.

Figure 10 shows four assembly configurations of an aircraft panel for finding the optimal configuration with the minimal turnaround time [20]. With this approach, the optimal digital factory setup can be found for a specifically designed aircraft panel to optimize the manufacturing process in the virtual reality before building the real factory. The design language in [20] provides a fully automated design stage of an aircraft panel based on given aircraft hull geometries and shapes. The simulation and optimization of the manufacturing process is realized with a corresponding design compiler plugin to the domain specific DELMIA digital factory software (www.3ds.com). The digital factory model is generated out of the central data model, the digital factory simulation is triggered and the turnaround times as well as information on the manufacturability of the panel are returned. Figure 11 shows the simulation of the coating process of the aircraft panel that is also created out of the design language's central data model. With this coating simulation critical coating parameters as local thickness and application efficiency can be optimized by a variation of the spray paths and spray configurations. It was thus possible to show in an exemplary manner that production processes and their process parameters can also be optimized with design languages [20].

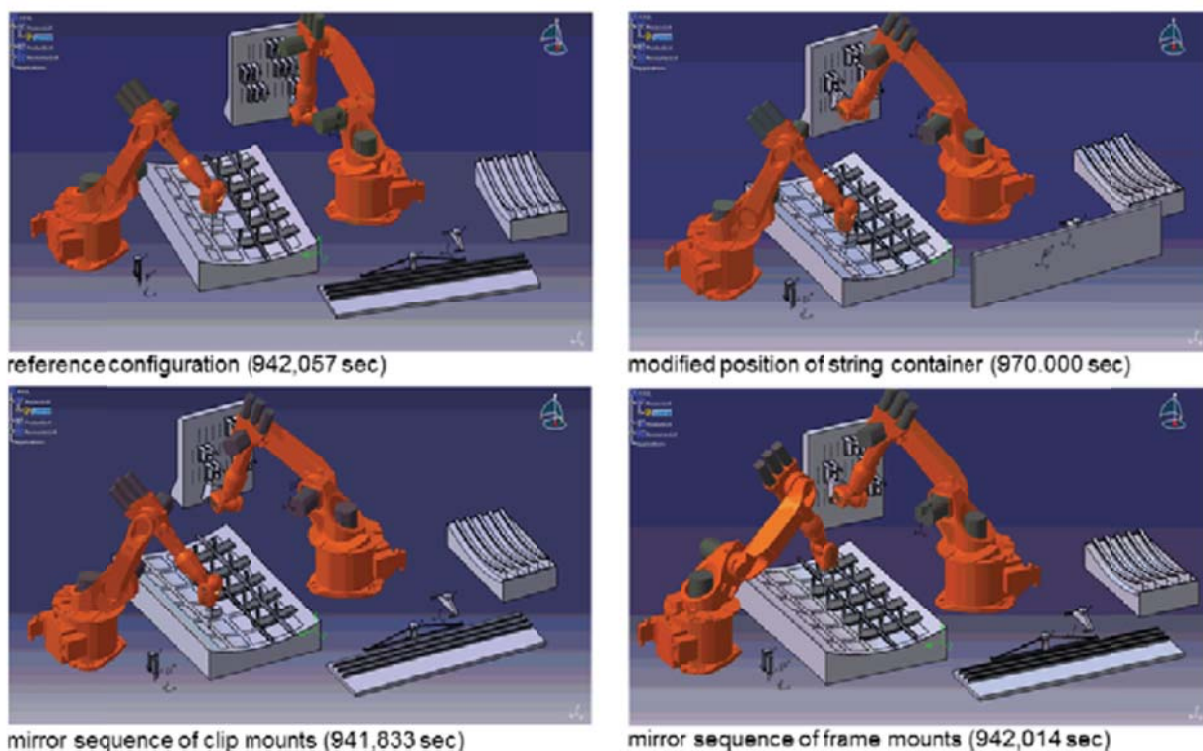


Figure 10 – Different digital factory assembly configurations with turnaround times [20]

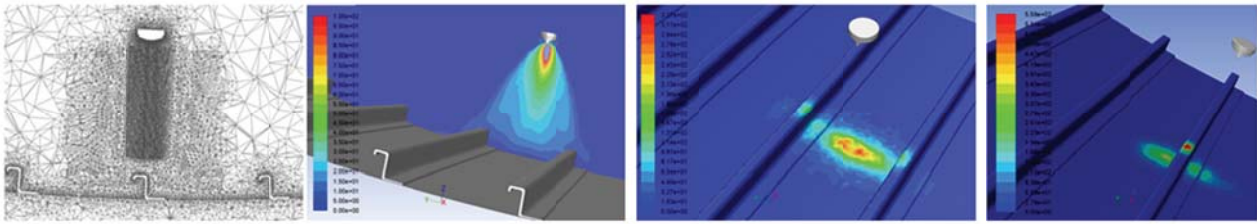


Figure 11 – Simulated coating process of the aircraft panel. Courtesy Fraunhofer IPA [20]

This approach is being further developed in a currently running project (“ZAFH Project Digital Product Lifecycle”). The project tries to expand the applicability of graph-based design grammars to the whole product life cycle. Among other things, production lines in automotive applications are created [23] especially with regard to costs, and their virtual commissioning is visualized within a virtual reality environment generated automatically in the design language [24]. The website of the project provides additional links to further publications (<https://dip.reutlingen-university.de/>).

2.4 Mechanical Engineering: Exhaust Aftertreatment Design

Finally, figure 12 shows results of a design language for the automated creation and functional optimization of SCR (selective catalytic reaction) exhaust aftertreatment systems that reduce the emissions of internal combustion engines [18, 21]. At the beginning, a combustion engine is given in terms of mass flow and exhaust temperature as well as raw emissions and emission target. An installation space is also given, where the exhaust aftertreatment has to fit in (blue box figure 12 top right) as 3D CAD STEP model. First, the catalyst is analytically dimensioned. Then a CAD model of the catalyst housing is created based on the previously calculated catalyst size. The catalyst box is automatically positioned in the installation space. This is done with a Dijkstra shortest path algorithm that identifies a path between the engine’s exhaust outlet and the exhaust target tail pipe position at the interface to the environment. This path is searched within the installation space under the constraint of having a maximum distance to the installation space’s walls. Installation positions for the catalyst box, that have a sufficient distance to the wall, are filtered from this path. In the next step a geometrical constraint pipework, made-up of only standardized pipe bends with predefined bend angles (eg. 45° and 90°), is created to connect the exhaust system components with the engine and the environment [21]. This pipework (figure 12 left) is identified within its own optimization runs that are nested within the production system of the overall design language that creates the whole exhaust system (figure 12 top right). The pipework can be created including CAD detailed mountings and connectors between the pipe elements.

Based on the pipework and the finally positioned catalyst box the overall CAD model is merged. This CAD model is automatically meshed and a fluid simulation is created to determine the emission reduction efficiency as well as the pressure loss of the system (figure 12 center bottom). A finite element simulation to evaluate the thermal expansion of the system is created and executed as well (figure 12 center top). The whole design language is integrated into a design of experiments (DoE) and optimization framework to conduct a design space exploration and to determine the Pareto-front as best possible trade-off between competing design targets (e.g. pressure loss and emission reduction efficiency). Figure 12 shows the results of a DoE run at bottom right. Each point in the plot represents one synthesized exhaust aftertreatment system with its characteristic evaluation figures pressure loss (horizontal axis) and outlet emissions (vertical axis) that both have to be minimized. The Pareto front is drawn as a line through those configurations for which there are no better configurations in both optimization targets. The system was proven to scale very well on high performance clusters. Thousands of variants could be created and evaluated within a few days up to

weeks. With this approach the physical limits of a given exhaust aftertreatment requirement set could be determined as Pareto front which is a very useful and valuable information, especially in an early project phase. The Pareto front gives an indication of the maximum system efficiency that can be achieved due to the physical limits for a given set of boundary and starting conditions (installation space, engine, ...).

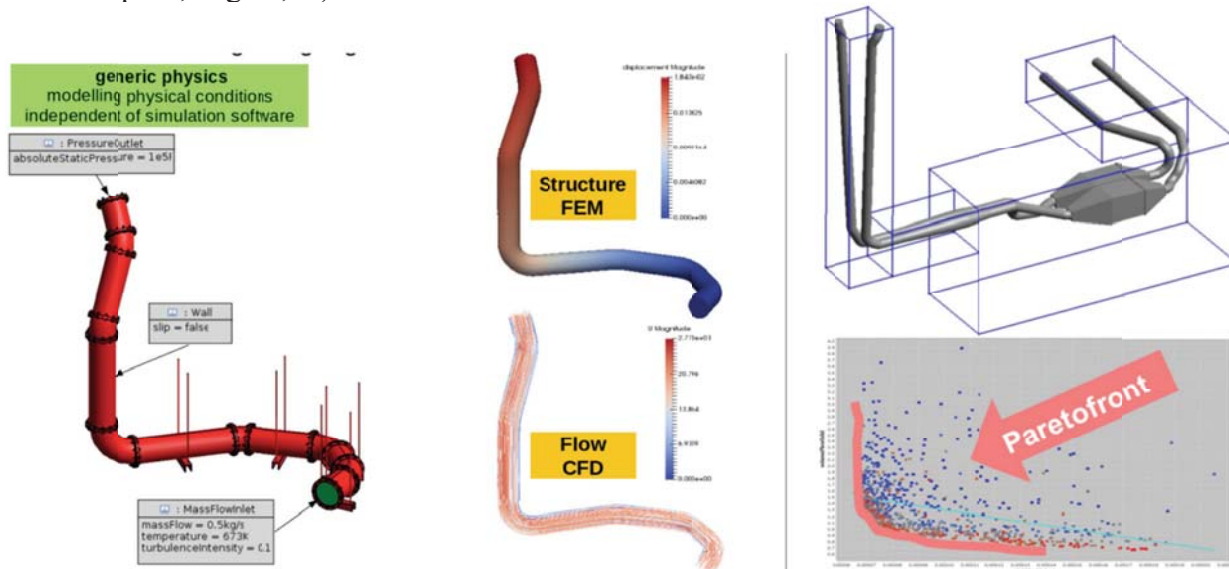


Figure 12 – Results of a graph-based design language for developing SCR exhaust aftertreatment systems in given installation spaces. Bottom right: results of DoE optimization runs with two conflicting optimization targets (horizontal: pressure loss, vertical: outlet emissions) with the resulting Pareto front. Reproduced from [18, 21].

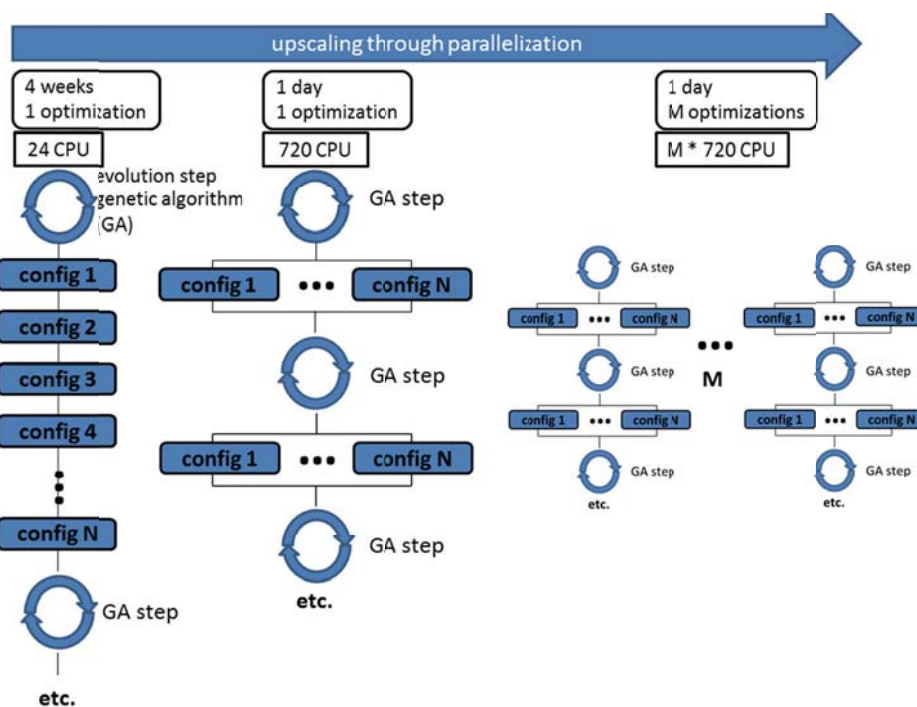


Figure 13 – Upscaling of design languages integrated into DoE/optimization runs.

The very good scalability of the design languages on HPC (high-performance computing) infrastructures could be demonstrated using the example of the design of exhaust aftertreatment systems. Figure 13 schematically shows the principle of upscaling the design process by executing the

genetic optimization(s) in parallel. The design language, which calculates a configuration based on a set of given requirements, is executed within a genetic optimizer. The configurations of the generation - one design language execution per configuration - can now run in parallel on an appropriate number of CPUs.

In the example in figure 13, this allows the optimization process to be reduced from several weeks to one day if the CPU number is correspondingly increased. There are no parallelization losses because each design language calculates an isolated configuration with corresponding input parameters, such as pipe diameter or dosing positions. This can be even further scaled up by a parallel execution of different optimization scenarios. The maximum runtime of the optimizations depends then only on the available computing power, since the entire design and validation process of the product is digitally mapped in the graph-based design language and can be re-executed as required without any manual intervention.

3 Design Principles

Different design strategies and mechanisms have been identified and applied in the automation of the design processes. Some principles have been identified by working on the explicit scientific problem of handling the complexity of engineering design and working on an engineering design theory. Others just emerged in application-centered design language projects as by-product.

3.1 Top-Down and Bottom-Up

The work in [25] proposes to distinguish between top-down and bottom-up designs. The definitions in [25] can be directly quoted: *“In the bottom-up approach it is attempted to achieve a higher level functionality by systematically combining basic building blocks into assemblies.”* A bottom-up design occurs especially in domains where only limited knowledge is available in advance. This phenomena often occurs in designs with non-linear physics, as fluid flows in the exhaust system example presented above, where small changes in geometry can produce significant changes in the overall physical behavior (e.g. flow separation). Figure 14 shows this approach schematically on the left side. The basic building blocks are combined to form assemblies that are able to fulfill higher-order functions and requirements. The systematical recombination is usually done in a kind of evolutionary approach, for example using a genetic optimization algorithm. This approach has a very high level of computational complexity, but it is able to produce new and creative solutions.

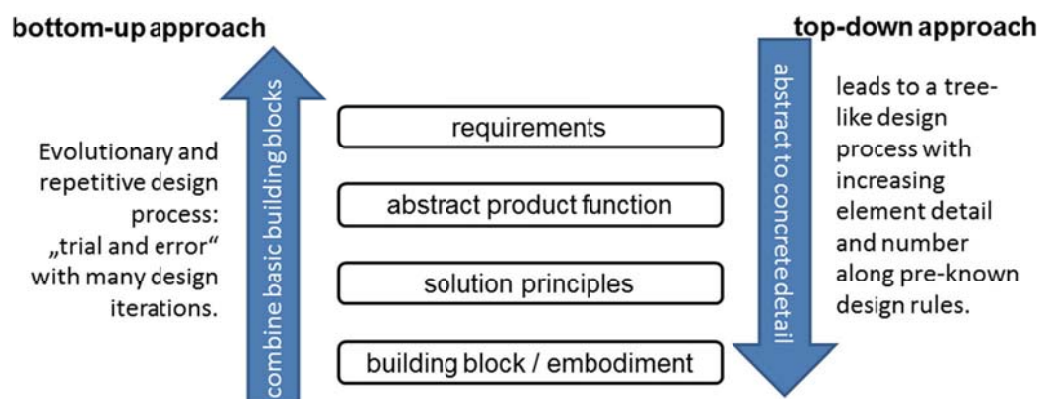


Figure 14 – Different design approaches according to [25]. The top-down approach as well as the design process steps in the center correspond to the design theory presented in [1]

For the counterpart of the bottom-up design approach the following is stated in [25]: “*In the top-down approach design synthesis ... is beginning with the requirements definition, the evolution of the design object is constituted out of subsequent decompositions from abstract conceptual descriptions into more detailed functional representations which finally find an embodiment into material components.*” The presented satellite example is an example for a top-down design where a priori knowledge is used to synthesize designs directly from the requirements. The right side of figure 14 shows this approach where the abstract requirements are stepwise decomposed into a final concrete embodiment. This approach follows the popular design theory that was presented by Pahl and Beitz in [1]. In a practical and realistic design process both approaches have to be combined as usually not all engineering problems can be solved using a-priori knowledge. At least the interplay between the enumeration, integration and configuration problem, as shown in the satellite example above, requires a bottom-up approach for non-trivial design tasks. This is true even when the system selection in the enumeration problem could be done in a top-down manner based on available knowledge. Graph-based design languages can implement both: bottom-up and top-down approaches.

3.2 Dimensionless Evaluation

Rudolph identifies the evaluation of engineering objects as crucial challenge in the (automated) design of products [11] as the chosen evaluation method directly influences and determines the outcome of the product optimization process. Using the Pi-theorem [26] with its dimensionless invariants addresses the three main problems of evaluation: “How can the *evaluation* of parts and components be found and represented? How are partial results *aggregated* into a single evaluation? How are the goal criteria *structured*, arranged and are they complete? [11]. In this sense, a complete description of a product entity in terms of design parameters combined to dimensionless invariants forms a valid evaluation: “*Any minimal description in the sense of the Pi-theorem is an evaluation*” [11]. This leads to the evaluation pattern shown in figure 15. The description of the engineering problem (physical equations) has a physical dimension and forms the parameter set (x_1, \dots, x_n) . The physical equations form a description function $X = X(x_1, \dots, x_n)$ that is defined on these dimensional parameters and describe the behavior of the system. Each component and subsystem contributes to this equation system which is assembled during the design process as components and subsystems are put together with an increasing level of detail to form the final product. From the fact that fully similar designs must get the same evaluation, it is concluded that the evaluation has to take place in dimensionless invariants (π_1, \dots, π_m) . This transition into dimensionless space, according to the Pi-theorem, ensures that fully similar designs fall within the same evaluation point. The description function is transformed to the evaluation function Π in the same manner. Since all information and variables, including physical dimensions, are stored in the design languages in the central data model, dimensionless evaluation parameters can in principle be obtained automatically.

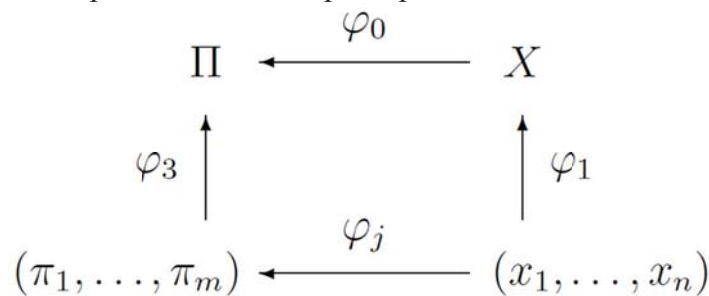


Figure 15 – Evaluation pattern in engineering based on PI-theorem [26]: The product parameters (x_i) are (physically) coupled through the description function X . In the equivalence of description and evaluation, the dimensionless evaluation parameters π_i derived from the x_i are forming the evaluation function Π [10]. Taken from [11]

3.3 Declarative Constraint Processing

In the process of designing complex products, consisting of many components, systems and spanning over multiple physical domains, large equation systems arise from the aggregation of analytical models that are part of each subsystem and component. Within the classes of the design languages equations between the class' parameters as well as between associated classes can be defined. On the execution of the production system the classes are instantiated and the class parameter values are set during the rule executions. So the equation system is created automatically alongside and it is stored in the design graph. For getting a fully automated design process the solution of this equation system needs to be automated. A solution using a so called solution path generator was proposed in [27]. The solution path generator identifies a sequence along which the equations of the equation system can be solved. This leads to a declarative processing of the equation system as only the problem has to be stated in form of the equation system and the solution sequence is internally determined. As an initial requirement, the given variables must be marked as constant in the design graph. This is also done in the production system. So the declarative processing of the equation system in the solution path generator frees the designer from providing a solution sequence. Different design scenarios with different set of given variables ("Who determines what?") can be realized in this way.

The resulting ordered equation system is automatically solved in a computer algebra system that is integrated into the design compiler (see figure 16). In addition, sensitivity analysis can be automatically executed on the equation system to determine the critical design drivers and main dependencies within a product design [14]. This design compiler feature was used in figure 9 in the satellite example presented above.

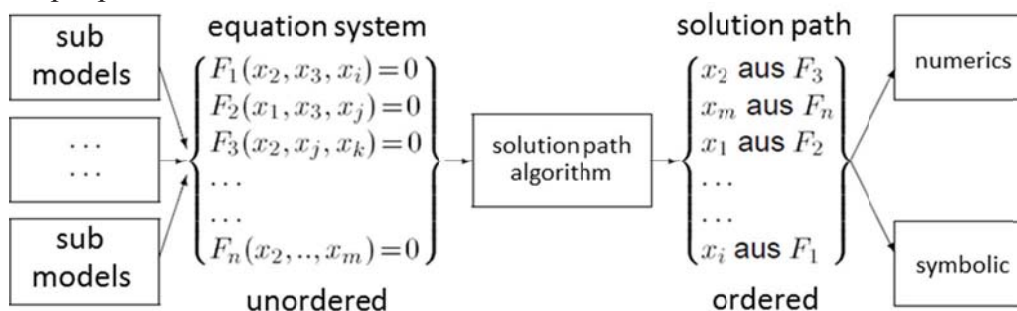


Figure 16 – The algebraic equations from sub models aggregate to a equation system. The solution path generator determines the sequence of equation solution algorithmically. The ordered equation sequence is solved numerically or symbolically in a computer algebra system. Reproduced from [10]

3.4 Self-Similar, Nested Design Patterns

Kormeier identifies in [28] a generic design pattern as observation from recurring design patterns in the application of design languages. In a first step a design process is identified that is an "...iterative sequence of design synthesis, analysis and evaluation..." [28]. This reflects the fact that during the design process usually a design of a component or a subsystem is created, analyzed using simulations or calculations and finally evaluated based on this results. Then it is checked whether the requirements, broken down to the component, are fulfilled. If not, then the cycle is repeated and, if necessary, changes must be made in an external cycle enclosing the loop under consideration in order to arrive at a solution. Thus, these design cycles are nested within each other and can be found at the most diverse abstraction and different levels of detail of the design. In this sense the pattern can be regarded as being self-similar [28]. The generic pattern is called integrated design pattern and refines the step of design synthesis of the process explained above. It is defined in the following way: "...The design synthesis itself is subdivided into the definition of requirements, the

decomposition, the formulation of functional solutions and the combinatorial exploration process.”

Figure 17 shows the integrated design pattern graphically. It unifies the two design approaches presented above, the top-down and bottom-up approaches, into one pattern. The red line divides the step of combination (finding solutions) into a bottom-up approach like part and a top-down approach like part. The red dividing line now shifts in one direction or the other depending on which design type prevails. The pattern thus provides a template that can be used in the implementation of design languages as a kind of ideal guideline for structuring the design process.

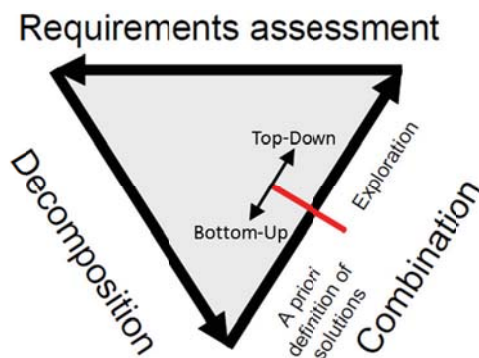


Figure 17 – Integrated design paradigm. Design synthesis as iterative pattern: “...definition of requirements, the decomposition, the formulation of functional solutions and the combinatorial exploration process...” [28].

Taken from [28] and extended.

3.5 Dimension-Based Design Sequence

The paper [18] deals with the problem of splitting the design process into a sequential step-by-step process. It is shown that in order to find an optimal design the dimensioning and integration of sub-systems and components has to occur at once within one design step. But this implies that the design parameters of the components and subsystems that are to be integrated have to be determined at once. This leads to a very high-dimensional problem which practically can't be solved in reasonable time. In general, the parameters have a non-linear behavior and subsets of parameters are strongly coupled with each other. Usually subsystem dimensioning has to be at least partially conducted in a bottom-up approach as the optimal parameter values cannot be determined in advance. Therefore it is necessary to split the design process into sequential steps to find a good design in reasonable time.

Since there are many sequences into which the design process can be divided, a preferred order of the sequential design tasks is proposed based on the mathematical dimension of the involved subsystems [18]. When two systems are sequentially integrated, the (common) design parameters of the lower dimensional system have to be fixed before fixing the design parameters of the higher dimensional system (“Begin with the system that has less degrees of freedom”). In figure 18 the concept of design parameters having different degrees of freedom associated with a different mathematical dimension is exemplarily illustrated. To explain the concept, the simplified problem of the installation or integration of the red square box in an installation space with an existing component (grey square box) is used. The left column a) just shows the two dimensional parameter space of the installation positions of the red box in the plane. The red shaded area in column b) shows the possible parameter values without caring for intersections of the two boxes. If the possible parameters are restricted to the technical meaningful installation positions where the boxes touch each other one gets the parameter set in c) which has now a lower mathematical dimension. Another step down to a point-shaped parameter set is done in column d) where the set is restricted to the installation position where the boxes are aligned which is regarded as being particularly advantageous in this

example. This example shall visualize that in the product design parameter sets with different mathematical dimensions occur, depending on their number of parameters as well as on restrictions due to the context of the design (technical meaningful, ...).

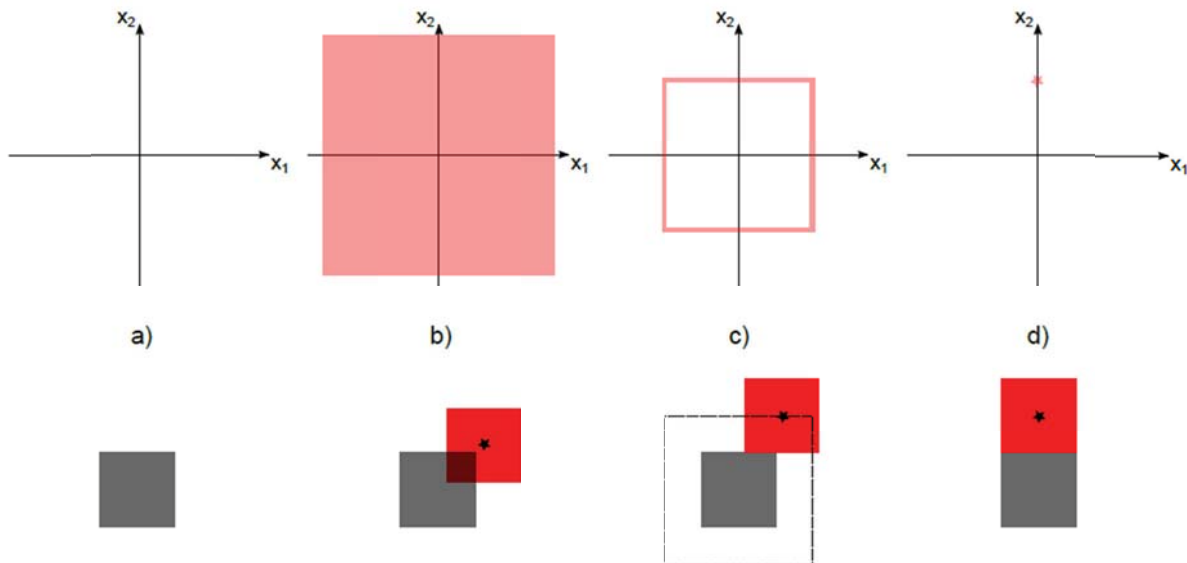


Figure 18 – Lower half: Assembly of two parts (red square and fixed grey square). Upper half: Installation positions (light red) of the red square as x_1, x_2 -parameter set for different levels of integration: a) x_1, x_2 -parameter space, b) combinatorial possible installation positions, c) technical meaningful installation positions (no intersection), d) optimal installation position. The different parameter sets have different mathematical dimensions [18].

It has been shown in [18] that in a sequential engineering process, where components and subsystems are integrated stepwise, the preferred order is given by the mathematical dimensions of the parameter sets that have to be combined. These parameter sets are not restricted to geometrical parameters as in figure 18. The concept also applies to the non-geometric parameters of the components and systems which must be matched and integrated with one another. This can be applied to the example in figure 18. The grey box has a fixed position and the mathematical dimension of its installation parameter set is therefore zero. The red box, on the other hand, initially has a two-dimensional parameter set. If the installation of the boxes should now happen sequentially, it makes no sense to start with the installation of the red box and to position it somewhere within its parameter set, since the probability of having it installed in exactly such a preferably way at the fixed position as in column d), goes towards zero. However, if the grey box, with the lower dimensional parameter set, is first installed, the degrees of freedom of the red box can be used to find a meaningful installation position in relation to the grey box.

3.6 Design Patterns and Paradigms

In the creation of a graph-based design languages it is possible to adopt design patterns [29] from object-oriented software engineering [29]. The use of object-oriented design patterns from software engineering in design languages is made possible by the extension of the design language classes with methods and interfaces proposed in [30]. With this extension the design languages are adopting central elements of object-oriented software engineering. The design patterns are solutions for recurring kind of problems in programming as for object creation, object composition (structural) and for object interaction (communication). They are typically made of two or more classes that are associated with each other and/or inherit behavior from each other, together with abstract methods in the classes and/or interfaces and a defined interplay and usage of these methods to solve a specific problem.

Figure 19 shows two patterns that are used in design grammars. The builder pattern on the top can be used within the design compilers plugin mechanism. The abstract information to trigger a process chain is extracted from the central model by a translator class. An engineering simulation application is set by calling the corresponding method of the translator class to select the application which shall be executed in the plugin. For example, it is possible to switch between different simulation programs, using the builder pattern shown, without having to change anything in the design language. The information needed to run a simulation is stored abstractly in the central data model [30, 31]. The lower part of figure 19 shows the composite pattern which is used to model a systems-of-systems relation that is the basic structure of all systems engineering activities where a system itself is made up of further systems or of components on the lowest level. The pattern provides methods to add and remove subsystems and components as well as to call operations or actions on all system elements in a recursive manner.

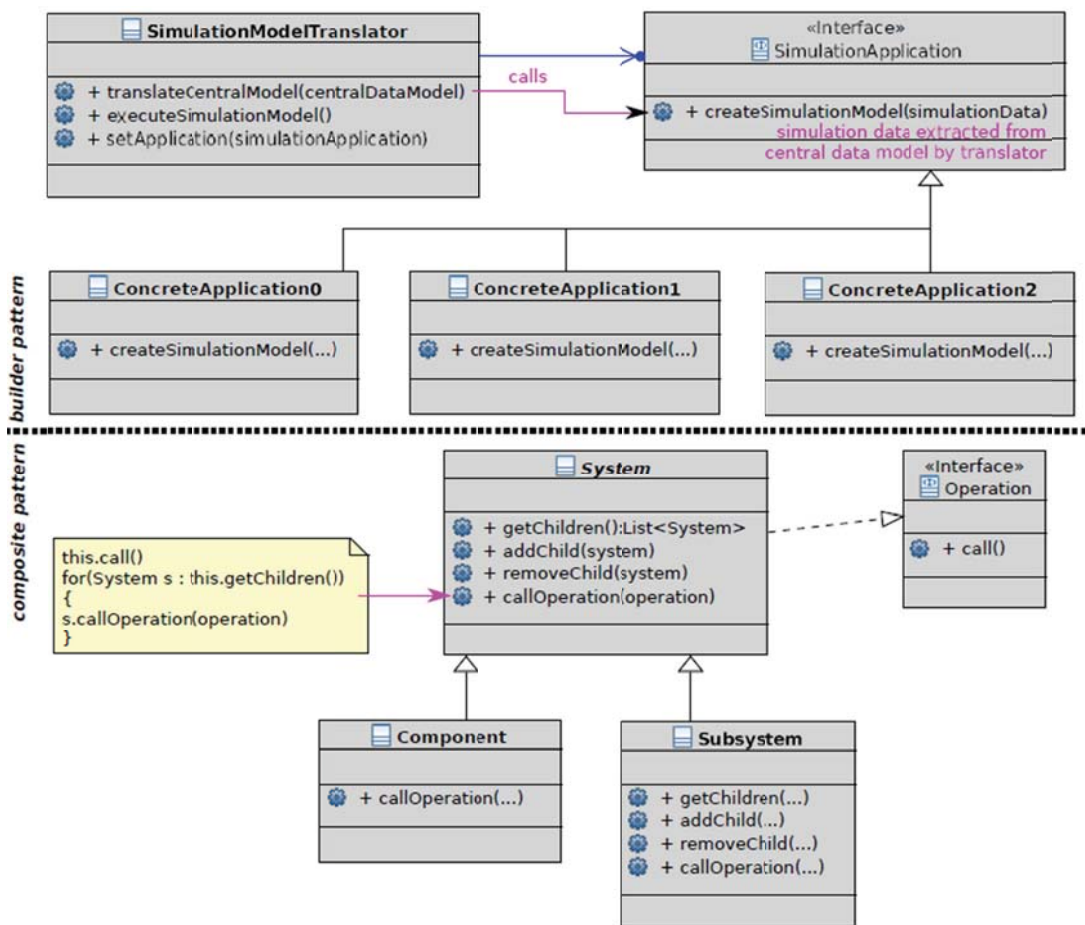


Figure 19 – Two design patterns from object-oriented software engineering [29]. Top: Builder pattern for calling different applications in an engineering plugin that shares a common central model. Bottom: Composite pattern for modeling a system-of-systems relation in systems engineering [30]

The ‘Design for X’ paradigms [10], mentioned above in the philosophical motivation, can also be seen as a kind of design pattern. As these paradigms are less specific than the previously presented design patterns from object-oriented software engineering they are part of the ‘ability’ concept area. This paradigms help the designer to structure the design process, at least mentally, when designing a graph-based design language. Nevertheless, it is often possible to transform the less specific paradigms in to rule-based heuristics when implementing a design language.

4 Discussion

In this paper an overview on graph-based design languages was given. The method allows a graphical programming of design processes that helps to manage the complexity in modern product design. The data model is separated from the operational procedural aspects in an object-oriented way. The object-oriented UML modeling language is used for an open, re-useable, compact and non-proprietary knowledge representation of the product entities and design process. Modeling redundancy is avoided through inheritance between classes and decomposition into classes, modules and plugins. Storing engineering knowledge in incremental rules and adaptive rule sequences allows a hierarchical decomposition of the engineering process itself into smaller chunks that can be more easily captured and overviewed by the engineer even for complex products and systems. This approach allows an easy reuse of knowledge in further design languages as external classes and packages can be easily loaded. The design languages are preferably used to automate recurring design tasks as the implementation of a design language in the design compiler takes some upfront effort.

The presented example applications show that the method of graph-based design languages is able to solve substantial real-world engineering problems in a fraction of the time that would be necessary in manual engineering. In fact, the design time collapses to the addition of the (potentially concurrent) run-times of the algorithms which is close to the lower theoretical limit. Furthermore, it is our observation that a graph-based design language, when embedded in an optimization framework, is often able to find more optimal engineering solutions as can be found in a conventional manual engineering process. The presented language is able to capture engineering knowledge digitally which leads to a highly scalable and re-executable digital blueprint of recurring design tasks. When executing design languages in optimization or DoE runs on HPC or cloud infrastructure, the available hardware power becomes the only limiting factor in accelerating the design tasks. The practical execution of optimizations with design languages on HPC environments shows that new challenges and questions arise from the resulting flood of results: How are result outliers to be treated and how are they caused? How can structures in the results be interpreted? Are these structures caused by properties of the design language or by the physics of the problem? Can generally valid technological laws be derived from this? Nevertheless, a significant upfront invest is necessary as implementing a product's generic design process is apparently more expensive than creating a few product designs by hand. The wide acceptance of this kind of modeling method in industry is at the moment therefore still often hindered by the traditional 'silo mentality' present in today's companies which look on the short-term profitability of the individual business unit instead on the mid- or long-term profitability of the company overall.

The presented design principles provide various patterns, templates and aids to handle the occurring complexity in product design. They can be used on different levels and stages during the implementation of engineering processes. Some of the shown principles as the design approaches, generic design patterns and the dimension-based design sequence help to understand the process of designing itself to simplify the implementation of design languages. Others, as the declarative constraint processing and the dimensionless evaluation scheme support an unambiguous evaluation of products which is a prerequisite for an unbiased optimization of the product itself. At last, design patterns from software engineering can be reused in design languages due to the close relation of the design languages and the design compiler with tools from object-oriented software engineering. All the shown design principles help to handle the complexity in product design. This is an important task as the complexity of the products themselves is steadily increasing and even more domains and disciplines have to be considered to get a holistic evaluation, validation and optimization of a product's life cycle from cradle to grave. In this sense, graph-based design languages have

shown to be a successful method to push the limits of the still controllable complexity in industrial engineering further away.

References

- [1] **Pahl G, Beitz W, Feldhusen J, Grote KH.** Engineering Design: A Systematic Approach. Series: Solid mechanics and its applications, Springer London, 2006.
- [2] **Kröplin B, Rudolph S.** Entwurfsgrammatiken - Ein Paradigmenwechsel? Der Prüflingenieur, 26:34-43, 2005.
- [3] **Antonsson E, Cagan J.** Formal Engineering Design Synthesis. Cambridge University Press, 2001.
- [4] **Borgest NM, Vlasov SA, Gromov AIA, Gromov Ana, Korovin MD, Shustova DV.** Robot-designer: on the road to reality. *Ontology of Designing*. – 2015; 5(4): 429-449. - DOI: 10.18287/2223-9537-2015-5-4-429-449.
- [5] **Borgest N, Gromov An, Gromov AI, Korovin M.** The concept of automation in conventional systems creation applied to the preliminary aircraft design. In: Wiliński A., et al. (eds.) *Soft Computing in Computer and Information Science*, Part II. AISC, Springer, 2015; vol. 342: 155–165. – DOI: 10.1007/978-3-319-15147-2_13.
- [6] **Borgest N, Korovin M.** Ontological approach towards semantic data filtering in the interface design applied to the interface design and dialogue creation for the “robot-aircraft designer” informational system. *Advances in Intelligent Systems and Computing*. Springer, 2017; vol. 534: 93-101. – DOI: 10.1007/978-3-319-48429-7_9.
- [7] **Jaroslav Sobieszczanski-Sobieski, A J Morris, M J L van Tooren.** Multidisciplinary Design Optimization Supported by Knowledge Based Engineering. - Chichester, UK : John Wiley and Sons, 2015.
- [8] **Muenzer C.** Constraint-Based Methods for Automated Computational Design Synthesis of Solution Spaces, ETH Zuerich, Diss., 2015. - 119 p. - <https://www.research-collection.ethz.ch/handle/20.500.11850/113711>.
- [9] **La Rocca, Gianfranco.** Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. *Advanced Engineering Informatics* 26 (2012), P.159–179. - DOI: 10.1016/j.aei.2012.02.002.
- [10] **Rudolph S.** Übertragung von Ähnlichkeitsbegriffen. Habilitationsschrift, Universität Stuttgart, 2002.
- [11] **Rudolph S.** Eine Methodik zur systematischen Bewertung von Konstruktionen. PhD thesis, Universität Stuttgart, 1995.
- [12] **Object Management Group, OMG** Unified Modeling Language. www.uml.org. Version: 2014.
- [13] **Reichwein A.** Application-specific UML Profiles for Multidisciplinary Product Data Integration. PhD thesis, Universität Stuttgart, 2011.
- [14] **Bölling M.** Lösungspfadbasierte Analysen im Entwurf komplexer Systeme. PhD thesis, Universität Stuttgart, 2013.
- [15] **Gross J.** Aufbau und Einsatz von Entwurfssprachen zur Auslegung von Satelliten. PhD thesis, Universität Stuttgart, 2014.
- [16] **Rudolph S, Beichter J, Eheim M, Hess S, Motzer M, Weil R.** On multi-disciplinary architectural synthesis and analysis of complex systems with graph-based design languages. In 62. Deutscher Luft- und Raumfahrtkongress (DGLR 2013), Stuttgart, September 10-12, 2013.
- [17] **Tonhäuser C, Rudolph S.** Individual Coffee Maker Design Using Graph-Based Design Languages. In: Gero J. (eds.) *Design Computing and Cognition '16*, pp 513-533, 2016. Springer, Cham. – DOI: 10.1007/978-3-319-44989-0_28.
- [18] **Vogel S.** Über Ordnungsmechanismen im wissensbasierten Entwurf von SCR-Systemen. PhD thesis, Universität Stuttgart, 2016.
- [19] **Haq M, Rudolph S.** A design language for generic space-frame structure design. In: *Int. J. Comput. Appl. Technol.*, Band 30(1/2): S. 77-87, 2007. - DOI: 10.1504/IJCAT.2007.015699.
- [20] **Arnold P, Rudolph S.** Bridging the gap between product design and product manufacturing by means of graph-based design languages. In: TMCE, 2012.
- [21] **Vogel S, Rudolph S.** Automated Piping with Standardized Bends in Complex Systems Design. In: *Proceedings of the Seventh International Conference on Complex Systems Design & Management, CSD&M Paris*, 2016.
- [22] **Rudolph S, Arnold P, Eheim M, Hess S, Motzer M, Riestenpatt M, Schmidt J, Weil R.** Design languages for multi-disciplinary architectural synthesis and analysis of complex systems in the context of an aircraft cabin, CE-AS Conference, Toulouse, November 25-27, 2014.
- [23] **Breckle T, Kiefer J, Rudolph S, Manns M.** Engineering of assembly systems using graph -based design languages, 21st International Conference on Engineering Design (ICED 17), Vancouver, Canada, 21-25.08.2017.
- [24] **Kiesel M, Klimant P, Beisheim N, Rudolph S, Putz M.** Using Graph-based Design Languages to Enhance the Creation of Virtual Commissioning Models, 27th CIRP Design Conference on Complex Systems Engineering and Development, 2017. - DOI: 10.1016/j.procir.2017.01.047.

- [25] **Alber R, Rudolph S.** A Generic Approach for Engineering Design Grammars. In: AAAI Spring Symposium Technical Report SS-03-02, March 24-26, 2003.
- [26] **Buckingham E.** On Physically Similar Systems: Illustration of the Use of Dimensional Equations. Phys. Review 4, 345-376, 1914. - DOI: 10.1103/PhysRev.4.345.
- [27] **Bölling M, Rudolph S.** Multi-disciplinary airship design using a graph-based design language. In: DGLR Jahrestagung, Band I, 2005.
- [28] **Kormeier T, Rudolph S.** On self-similarity as a design paradigm. In: Proceedings of IDETC/CIE, International Design Engineering Technical Conference and Computers and Information in Engineering Conference, September 24-28, 2005.
- [29] **Gamma E.** Design Patterns: Elements of Reusable Object-oriented Software. Pearson Education, 2004 (Addison-Wesley Professional Computing Series).
- [30] **Vogel S, Arnold P.** Object-orientation in graph-based design grammars. Computing Research Repository, abs/1712.07204, 2017.
- [31] **Vogel S.** An application-independent continuum mechanics interface for virtual engineering, Engineering with Computers, 2018.

ПРОЕКТИРОВАНИЕ СЛОЖНЫХ СИСТЕМ ПРИ ПОМОЩИ ЯЗЫКОВ ПРОЕКТИРОВАНИЯ: МЕТОДЫ, ПРИМЕНЕНИЕ И ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ

С. Фогель¹, С. Рудольф²

Институт проектирования самолетов Штутгартского университета, Штутгарт, Германия
¹Samuel.Peter.Vogel@gmail.com, ²Rudolph@ifb.uni-stuttgart.de

Аннотация

Языки проектирования, основанные на графах, представлены как способ преобразования информации, автоматизации процессов проектирования и оптимизации продукта или сложной системы. Унифицированный язык моделирования (UML) используется для создания языка, моделирующего процесс проектирования. Язык проектирования состоит из терминологии («цифровых строительных блоков») и набора правил («знаний цифровой композиции») выполнения последовательности действий (т.е. последовательного преобразования в цифровой вид процесса проектирования). С использованием основанного на правилах метода создаётся обобщённая центральная согласованная схема данных об объекте проектирования (так называемый граф проектирования). После генерации абстрактной центральной модели автоматически генерируются инженерные модели, отражающие специфику конкретной предметной области, и после удалённого выполнения их результаты вносятся в центральную модель проектирования для принятия последующих проектных решений или оптимизаций. Языки проектирования моделируются вручную и автоматически выполняются в так называемом компиляторе проектирования. Языки проектирования, основанные на графах, успешно применяются при создании разнообразных изделий аэрокосмической (космические аппараты, самолёты), автомобильной (пространственные конструкции, кабины автомобилей), машиностроительной (роботы, цифровые производства) отраслей и потребительских товаров (кофе-машины, вытяжные системы) для повышения эффективности процесса проектирования и степени его автоматизации. Рассмотрены различные стратегии и механизмы проектирования с целью применения их к автоматизации процесса проектирования. Используются подходы, начиная с автоматизированной и декларативной обработки ограничений, фрактальных вложенных шаблонов проектирования до математического определения последовательности действий проектирования. Имеющиеся знания определяют общую стратегию проектированию (т.е. нисходящее или восходящее проектирование). С целью снижения размерности и общей сложности задачи используется построение безразмерных инвариантов на основе теории подобия. Шаблоны проектирования, парадигмы проектирования (т.е. форма следует за функцией или функция следует за формой) и стратегии проектирования (разделяй и властвуй) из информатики широко используются для структурирования и управления сложностью проекта.

Ключевые слова: языки проектирования, автоматизация проектирования, метод проектирования, принципы проектирования, онтология проектирования.

Сведения об авторах



Dr. Samuel Vogel (b. 1982) is currently external habilitation candidate in the group of similarity mechanics at the Institute of Aircraft Design at the University of Stuttgart. He is currently working at MTU Friedrichshafen GmbH (subsidiary of Rolls Royce Holding plc) in the research and technology development. Samuel Vogel studied physics at the University of Stuttgart and finished in 2007. After that he worked for five years at a development service provider as team lead and head of the department "Predevelopment, Methods and Analytics". In 2013, he moved to his current position at MTU. Between 2009 and 2015, Samuel Vogel did in parallel his doctorate on the automated design of exhaust systems with design languages in Stephan Rudolph's group. From 2016 to 2018 he also did research in the same research group as a postdoctoral student in part time. His research interests include the interplay between the (automated) design process and complex systems science as well as design optimization with design languages

especially in conjunction with computational fluid dynamics. Automated 3D CAD model and geometry creation for autonomous optimization and validation workflows are also an important research interest of Samuel Vogel.

Д-р Самюэль Фогель (1982 г.р.) в настоящее время является кандидатом на внешнюю хабилизацию в группе механики подобия в Институте проектирования самолетов Штутгартского университета и работает в MTU Friedrichshafen GmbH (дочерняя компания Rolls Royce Holding plc). Самюэль Фогель окончил Штутгартский университет в 2007 году, после этого пять лет работал в качестве руководителя команды и начальника отдела «Предварительная разработка, методы и аналитика». В 2013 году он перешёл на свою нынешнюю должность в MTU. В период с 2009 по 2015 годы Самуэль Фогель параллельно выполнил свою докторскую работу по автоматизированному проектированию выхлопных систем с использованием языков проектирования в группе Стефана Рудольфа. С 2016 по 2018 годы он проводил исследования в той же исследовательской группе в качестве постдокторанта. Его исследовательские интересы включают взаимодействие между (автоматизированным) процессом проектирования и теорией сложных систем, а также оптимизацией проектирования с использованием языков проектирования. Автоматизированное 3D-моделирование в САПР и создание геометрии для автономных процессов оптимизации и валидации являются важным исследовательским интересом Самуэля Фогеля.



PD Dr. Stephan Rudolph (b. 1961) is currently head of similarity mechanics group at the Institute of Aircraft Design at the University of Stuttgart. Stephan Rudolph studied, earned his doctorate and habilitated at the Faculty of Aerospace Engineering and Geodesy at the University of Stuttgart. Additional one-year study visits at a French Grande Ecole for Aerospace (ENSICA) in Toulouse, France, and at the Massachusetts Institute of Technology (MIT) in Cambridge, USA. After obtaining his PhD in 1995 from the University of Stuttgart, he spent six months as a PostDoc in the Systems and Design Group at the MIT. Habilitation and Lecturer in the field of "Design Methodology" at the Faculty of Aerospace Engineering and Geodesy at the University of Stuttgart in 2002. Head of the research group "Similarity Mechanics" since 1996.

Dr. Rudolph's research interests include formal methods of Model-Based System Engineering (MBSE) and formal design synthesis with graph-based design languages, automatic model generation and design evaluation methods as well as applications of similarity mechanics in engineering and artificial intelligence. Within the focus of graph-based design languages, the theoretical focus is on questions of uniqueness, consistency, validation and verification of design languages, as well as the practical focus on the language development of graph-based design languages for automatic product design of satellites, aircraft, vehicle structures and their digital factories.

Д-р Стефан Рудольф (1961 г.р.) возглавляет группу механики подобия в Институте проектирования самолетов Штутгартского университета. Стефан Рудольф учился и получил докторскую степень на факультете аэрокосмической техники и геодезии Штутгартского университета (1995). Получив степень PhD, прошёл стажировки в Высшей школе аэрокосмической техники (ENSICA) в Тулузе (Франция) и в течение шести месяцев в качестве постдоктора в группе «Системы и проектирование» Массачусетского технологического института (MIT) в Кембридже (США). В 2002 году получил хабилизацию как доктор и лектор в области методологии проектирования на факультете аэрокосмической техники и геодезии в Штутгартского университета. Руководитель исследовательской группы механики подобия с 1996 года. Научные интересы доктора Рудольфа включают формальные методы системного инжиниринга на основе моделей (MBSE) и формальное проектирование с использованием языков, основанных на графах, автоматическое создание и оценки проектных моделей с применением механики подобия в проектировании и искусственного интеллекта. В языках проектирования, основанных на графах, наибольший интерес для доктора Рудольфа в теоретическом аспекте представляют вопросы уникальности, согласованности, валидации и проверки языков проектирования. Наибольший практический интерес в этой области представляет разработка языков проектирования, основанных на графах, которые бы позволили автоматизировать проектирование космических аппаратов, самолётов, конструкций машин и их цифровых производств.

УДК 51-7

МЕТОДЫ ИЗВЛЕЧЕНИЯ И АНАЛИЗА ТЕРМИНОЛОГИЧЕСКИХ СТРУКТУР СМЕЖНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ (НА ПРИМЕРЕ МЕТОДОЛОГИИ)

Д.А. Губанов¹, Д.А. Новиков²

Институт проблем управления имени В.А. Трапезникова РАН, Москва, Россия

¹dmitry.a.g@gmail.com, ²novikov@ipu.ru

Аннотация

В работе предложен автоматизированный экспертный подход к извлечению терминологических структур из монографий. Постулируется, что содержание предметной области целиком отражается в некоторой монографии, это позволяет извлекать и анализировать терминологические структуры без использования заранее подготовленного корпуса специализированных текстов предметной области. В рамках подхода рассмотрена процедура итеративного извлечения и анализа терминологических структур, регулирующая работу эксперта с монографией и позволяющая ему контролировать процесс. Предложены методы сравнительного анализа терминологических структур двух смежных предметных областей. Возможности разработанного подхода продемонстрированы на примере общей методологии и методологии комплексной деятельности.

Ключевые слова: терминологическая структура, теория графов, общая методология, методология комплексной деятельности.

Цитирование: Губанов, Д.А. Методы извлечения и анализа терминологических структур смежных предметных областей (на примере методологии) / Д.А. Губанов, Д.А. Новиков // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.347-365. – DOI: 10.18287/2223-9537-2018-8-3-347-365.

Введение

В соответствии с определением, приведенным в [1], *теория* – форма достоверного научного знания о некоторой совокупности объектов, представляющая собой систему взаимосвязанных утверждений и доказательств и содержащая методы объяснения и предсказания явлений и процессов некоторой *предметной области (ПрО)*, то есть всех явлений и процессов, описываемых данной теорией.

Изложение научных результатов, полученных в той или иной ПрО, ведется, как правило, на соответствующем профессиональном языке, использующем, помимо общеупотребительной, свою специальную терминологию. То есть в каждой ПрО можно условно поставить в соответствие множество терминов, характеризующих эту ПрО и используемых в ней [2]. Совокупность этих терминов и связей между ними будем называть *терминологической структурой* ПрО. Постулируем, что содержание ПрО целиком отражается в некоторой *монографии*¹, поэтому исходными данными для анализа терминологической структуры является текст этой монографии.

В настоящей работе рассматривается задача автоматизированного извлечения терминологической структуры из текста соответствующей монографии, а также формулируется задача сравнительного анализа терминологических структур различных ПрО.

¹ Согласно ГОСТ 7.60-2003, монография - «Научное или научно-популярное издание, содержащее полное и всестороннее исследование одной проблемы или темы ...».

В разделе 1 рассматриваются методы извлечения терминологической структуры ПрО, в разделе 2 – методы сравнительного анализа терминологических структур смежных ПрО, в разделе 3 приводится сценарий организации процесса извлечения терминов из текста. В разделе 4 приводятся примеры анализа терминологических структур таких ПрО как *общая методология* (ОМ) – см. монографию [1] – и *методология комплексной деятельности* (МКД) – см. монографию [3], а также приводится их сравнительный анализ.

Методология - учение об организации деятельности. В [1] изложена ОМ (основания ОМ, характеристики деятельности, ее логическая и временная структуры), а также рассмотрены методология научной, практической, художественной, учебной и игровой деятельности. Монография [3] посвящена изложению МКД, развивающей ОМ на случай любой сложной человеческой деятельности (см. [4]). Значительное внимание в МКД уделяется организации и управлению, неопределённости, а также жизненным циклам (ЖЦ) деятельности, её субъектов, предметов, ресурсов, знаний и технологий.

1 Методы извлечения терминологической структуры ПрО

Используемый подход предполагает последовательное решение следующих взаимосвязанных задач: извлечение терминов и извлечение связей между терминами.

1.1 Извлечение терминов

Пусть имеется *монография* (один документ), результатом должен быть ранжированный список терминов. Общая последовательность этапов имеет следующий вид (см., например, подобные примеры последовательности в [5, 6]):

- 1) предварительная обработка текста,
- 2) выявление терминов-кандидатов (или кандидатов в термины),
- 3) вычисление признаков терминов-кандидатов,
- 4) отбор кандидатов.

Рассмотрим подробнее эти этапы.

Этап 1. *Предварительная обработка текста* может быть представлена как следующая последовательность действий.

- Преобразование исходного файла в текстовый формат, фильтрация артефактов преобразования и неинформативных элементов.
- Токенизация полученного текста (разбиение текста на слова).
- Лемматизация слов или стемминг.

Этап 2. *Выявление терминов-кандидатов.* На этом этапе формируется множество кандидатов в термины (потенциальных терминов).

- *Выявление n-грамм* (терминов-кандидатов).
- *Фильтрация терминов-кандидатов.* Применяются экспертные знания о том, какие кандидаты в термины заведомо не являются терминами. На практике полезными являются следующие виды фильтров: лингвистическая фильтрация (например, рассматриваются только кандидаты-словосочетания с существительным в роли главного слова); фильтрация кандидатов по частоте встречаемости; фильтрация по содержанию в составе кандидата стоп-слов (по стоп-слову из экспертного списка, по стоп-слову в начале или в конце кандидата стоп-слова (например, «и»), по символу пунктуации (например, «,»)); фильтрация по длине кандидата (по числу слов в последовательности); фильтрация на основе регулярных выражений и др.

Этап 3. *Расчет значений признаков терминов-кандидатов.* На этом этапе формируются и рассчитываются значения признаков-кандидатов, определяющих их вероятность быть термином.

- *Формирование множества признаков терминов-кандидатов.* Для терминов-кандидатов определяется список основных и вспомогательных признаков (таким образом каждый кандидат может быть представлен в виде вектора признаков). Основные признаки характеризуют терминологичность кандидата, т.е. силу связи кандидата с ПрО. В массе своей это признаки, которые основываются на статистиках вхождений слова в сегменты документа (или документ, или корпус документов) и используют разного рода сопутствующие эвристики («навешивающие» вознаграждения и штрафы на те или иные статистики). Примером является абсолютная и относительная частота слова в документе. Вспомогательные же признаки косвенно определяют значимость кандидата. Например, значимость кандидата выше, если он находится в аннотации документа или в секции ключевых слов (таким образом можно определить множество структурных признаков, характеризующих встречаемость слова в различных логических разделах документа). Также можно определить признаки, характеризующие вхождение кандидата во внешнюю энциклопедию (напр., Википедию, см. [7]), словарь, тезаурус или онтологию; использовать признаки, характеризующие форматирование слова в исходном документе (курсив, жирный шрифт и т.д.); использовать признаки, характеризующие наличие тех или иных частей речи в кандидате.
- *Расчёт значений признаков.* Производится расчет признаков кандидата. Предлагается использовать расчет основных признаков с использованием статистик вхождений кандидата в текст (см. [8, 9]).
- *Расчёт интегрального показателя.* Далее на основе совокупности признаков кандидатов оценивается общая значимость каждого кандидата (оценка того, насколько он «является термином»). Такая оценка может быть вычислена при помощи различных методов: линейной комбинации признаков, алгоритмов голосования или машинного обучения.

Этап 4. Отбор кандидатов. Кандидаты ранжируются и отбирается топ (например, первые 300 кандидатов). Затем ранжированный список кандидатов оценивается и фильтруется экспертом в соответствующей ПрО.

Приведём примеры типовых подходов, в которых реализуются те или иные шаги приведённого алгоритма-шаблона.

Пример 1. Подход на основе статистик вхождения:

- 1) формирование списка 1 и 2-грамм, приведённых в нормальную форму;
- 2) фильтрация n -грамм, содержащих слова из стоп-листа или имеющих размер (в символах) меньше некоторого порога;
- 3) расчёт частоты n -грамм (и мер сопряжённости для биграмм);
- 4) ранжирование n -грамм по убыванию значения статистики и отбор топ- K терминов (где $K \approx 10^2$).

Пример 2. Пример подхода на основе морфологических шаблонов (например, <существительное> + <существительное в родительном падеже>, см., например, [10]):

- 1) разработка морфологических шаблонов;
- 2) определение кандидатов при помощи шаблонов;
- 3) расчёт частоты встречаемости кандидатов;
- 4) ранжирование кандидатов и отбор.

1.2 Извлечение связей между терминами

Для построения терминологической структуры ПрО необходимо установить связи между входящими в неё терминами.

Возможны следующие подходы к извлечению связей между терминами.

На основе лексико-синтаксических шаблонов. Извлечение из корпуса текстов родовидовых отношений с использованием лексико-синтаксических шаблонов (см., например, [12]).

На основе статистик совместной встречаемости. Если термины встречаются совместно, то между ними может существовать связь. Самый простой вариант статистики: чем чаще термины встречаются вместе, тем сильнее между ними связь.

Более сложные варианты основываются на векторном представлении документов / слов. Традиционно используемый для выявления связей между словами метод основан на формировании матрицы совместной встречаемости слов в текстах коллекции документов ПрО (на основе матрицы «слова-на-документы»), оценке сходства рассматриваемых терминов и – в конечном итоге – формировании связи между терминами. Среди недостатков этого подхода (разреженность матрицы совместной встречаемости слов, большой размер матрицы и ее зашумленность) наиболее критичным в нашем случае является необходимость работы с коллекцией документов.

Один из современных подходов к векторному представлению слов – «word embeddings» [13] (при этом не требуется использовать большую матрицу «слова-на-документы», в процессе работы используется ближайшая окрестность слов). В рамках этого подхода алгоритмы (например, нейронные сети) прогнозируют появление слова в заданном контексте, и в процессе обучения алгоритма формируется матрица параметров (в том числе векторное представление слов небольшой размерности). Полученное векторное представление слов можно использовать для оценки сходства терминов (см. пример извлечения сети семантических связей из Википедии [14]).

В данной работе сила связи между терминами оценивается исходя из их совместной встречаемости в предложениях исследуемого документа (соответствующей монографии).

2 Методы сравнительного анализа терминологических структур смежных ПрО

Установление соответствий и различий между двумя ПрО и их терминологическими структурами может быть полезно для решения следующих задач:

- анализ содержательного соотношения ПрО;
- согласование терминологических структур;
- обогащение одной терминологической структуры за счёт отсутствующих в ней элементов из другой терминологической структуры;
- интеграция терминологических структур, в процессе которой происходит создание новой терминологической структуры, и т.д.

Для проведения сравнительного анализа терминологических структур предлагаются следующие подходы.

2.1 Статистический подход

Статистический подход основывается на сравнении встречаемости терминов в текстах двух ПрО (здесь и далее предполагается, что для сравнительного анализа терминологических структур важна значимость термина и контекст появления термина).

Статистики отдельного термина:

- сравнение частот вхождений термина в тексты разных ПрО;
- сравнение контекстов использования термина в текстах разных ПрО, расчёт близости/схожести использования термина по контекстам встречаемости.

Статистики терминологической структуры:

- сравнение ранжирования терминов в двух ПрО, получение оценки сходства ранжирований (например, при помощи ранговой корреляции).

- сравнение вероятностных распределений терминов в двух ПрО, получение оценки сходства вероятностных распределений (например, при помощи расстояния Кульбака-Лейблера).

2.2 Структурно-статистический подход

Структурно-статистический подход основывается на сравнении статистик, связанных с отдельными вершинами графа терминологической структуры и/или статистик, характеризующих весь граф терминологической структуры.

Сравнение терминов (вершин графа).

- Сравнение значимости термина в разных терминологических сетях (показатели структурной центральности, показатели влияния, см. [15]). Для проведения сравнительного анализа в работе используются следующие виды структурных центральностей: центральность вершины по степени (число инцидентных вершине ребер в графе/сети), центральность вершины по близости (величина обратная сумме расстояний от вершины до всех остальных вершин сети), центральность вершины по посредничеству (оценивает то, как часто вершина появляется на кратчайших путях между парами вершин сети), центральность вершины по PageRank (является результатом случайного блуждания по вершинам сети), а также влияние вершины, основанная на акциональной модели [16]. Следует отметить, что рассчитать такую влияние для узлов статичных неориентированных сетей можно при помощи имитационного моделирования распространения действий (в данном случае в качестве модели распространения действий выбрана модель независимых каскадов [17]).

- Оценка сходства терминов. Термины (вершины) считаются похожими, если они прямо или косвенно связаны с похожими терминами. Например, можно сравнивать использование термина, рассчитывая расстояние между эго-сетями [16] термина в разных терминологических структурах.

Сравнение терминологических структур (графов).

- Сравнение ранжирований терминов в разных терминологических структурах по убыванию значимости (структурной центральности).
- Сравнение графов и оценка их структурного сходства. Сходство двух графов (близость) можно определить, например, с использованием максимально общего изоморфного подграфа. В частности, если обозначить граф g_1 , граф g_2 и максимально общий изоморфный подграф $mcs(g_1, g_2)$, то сходство g_1 и g_2 оценивается следующим образом: $\frac{|mcs(g_1, g_2)|}{\max\{|g_1|, |g_2|\}}$.

Кроме того, расстояние между графами можно рассчитать при помощи минимального количества операций редактирования (вставки, удаления и замены вершин или ребер графа), необходимых для превращения одного графа в другой граф (Graph Edit Distance).

Можно использовать для сравнения и оценки сходства векторное представление графов. Примерами признаков могут быть типовые сетевые структуры, в частности, множество эго-сетей заданных экспертом понятий или множество всех путей заданной длины, содержащихся в графах (или наиболее вероятные / значимые маршруты). Далее рассчитываются значения признаков графов, и вычисляется расстояние между векторами графов.

3 Процесс извлечения терминов

Извлечение терминов ПрО из монографии производится в полуавтоматическом режиме, в этот процесс вовлечены эксперт в рассматриваемой ПрО и разработчик (или соответствующая программа, обладающая развитым пользовательским интерфейсом). Сценарий взаимодействия эксперта и разработчика представлен на рисунке 1.

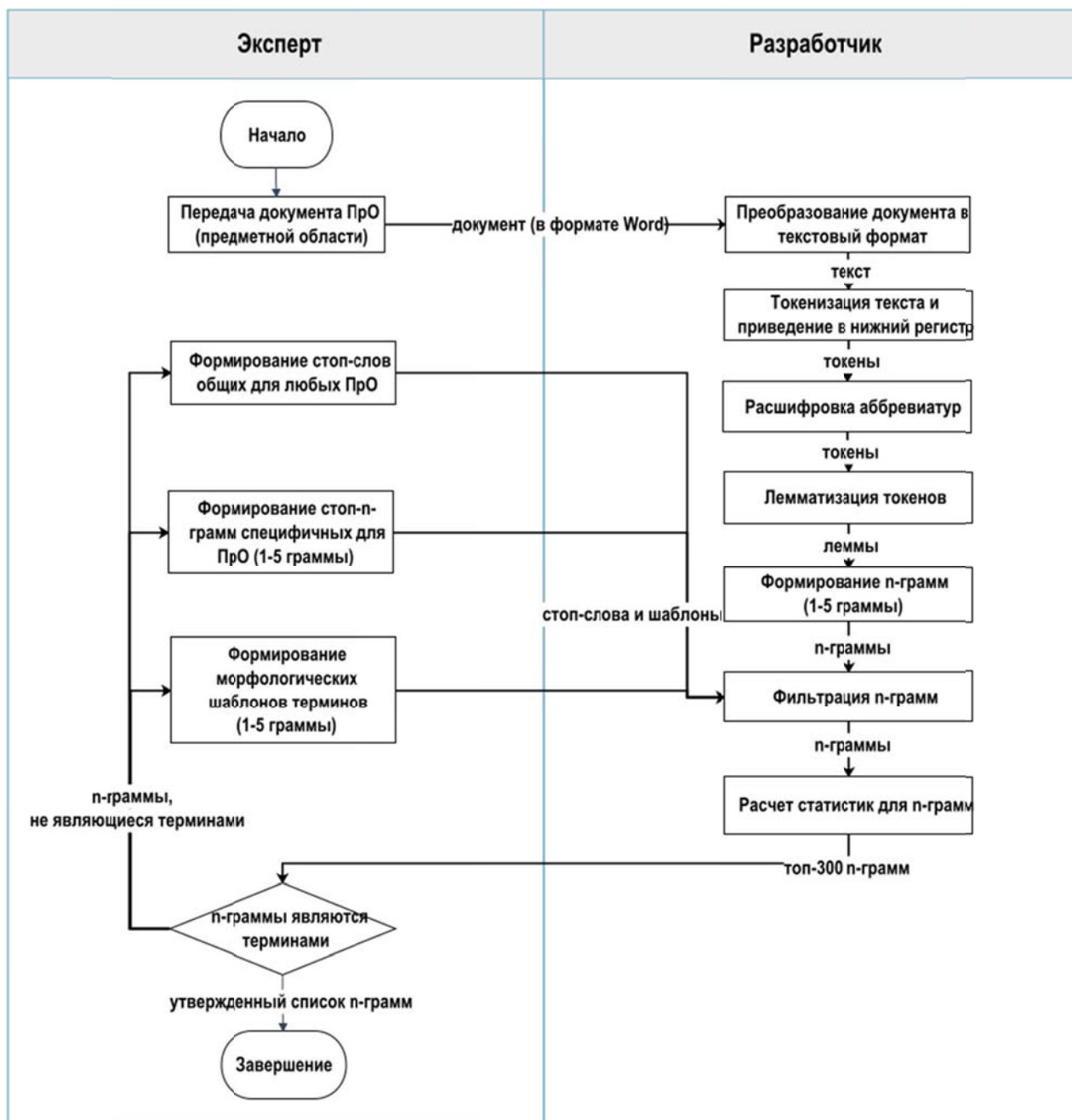


Рисунок 1 – Сценарий взаимодействия эксперта и разработчика

Приведём некоторые пояснения к рисунку 1.

1. *Расшифровка аббревиатур.* Каждый токен в последовательности, принадлежащий экспертному списку аббревиатур, заменяется на соответствующие токены из расшифровки. Например, аббревиатура «СЭД» заменяется на «структурный», «элемент», «деятельности».

2. *Лемматизация токенов.* Производится при помощи размеченного корпуса русского языка OpenCorpora (opencorpora.org). Часть речи слова и его начальная форма определяются исходя из частоты встречаемости (слово + часть речи) в корпусе OpenCorpora.

3. *Формирование n-грамм.* Из ранее полученной последовательности лемм текста формируются все допустимые последовательности из n лемм.

4. *Фильтрация n-грамм.* Используемые правила фильтрации:

- частота встречаемости n -граммы в тексте не менее 5;
- длина первого и последнего слова в n -грамме больше 1;
- n -граммы не должны содержать символы пунктуации;
- n -граммы не должны содержать числа;
- n -граммы не должны содержать в себе общеупотребительные слова (возможно дополненных экспертом), а также не должны заканчиваться или начинаться предлогами или союзами;
- n -граммы не должны входить в составленный экспертом список стоп- n -грамм (эти n -граммы не являются терминами ПрО, их список может быть сформирован заранее, а затем дополнен экспертом при просмотре результатов извлечения терминов из текста);
- n -граммы должны соответствовать определённому морфологическому шаблону, например:

	1 слово	2 слово	3 слово	4 слово	5 слово
1-грамма	Сущ/Прил/Прич/?				
2-грамма	Сущ/Прил/Прич/?	Сущ/?			
3-грамма	Сущ/Прил/Прич/?		Сущ/?		
4-грамма	Сущ/Прил/Прич/?			Сущ/?	
5-грамма	Сущ/Прил/Прич/?				Сущ/?

4 Анализ терминологической структуры

4.1 Анализ терминологической структуры МКД

В качестве исходного корпуса текста используется монография [3].

4.1.1 Выделение терминов–униграмм

Выделение терминов из текста проводилось в рамках предложенного подхода (раздел 1) согласно рассмотренному в разделе 3 сценарию. Всего получено 134 термина, наиболее часто в тексте встречаются следующие термины (топ-30 в порядке убывания частоты):

деятельность, элемент, комплексный, технология, модель, система, субъект, ресурс, результат, неопределённость, управление, структура, цикл, жизненный, операция, предмет, организация, цель, событие, элементарный, спрос, причинно-следственный, вид, логический, методология, потребность, характеристика, процесс, работа, связь. Распределение терминов МКД представлено на рисунке 2.

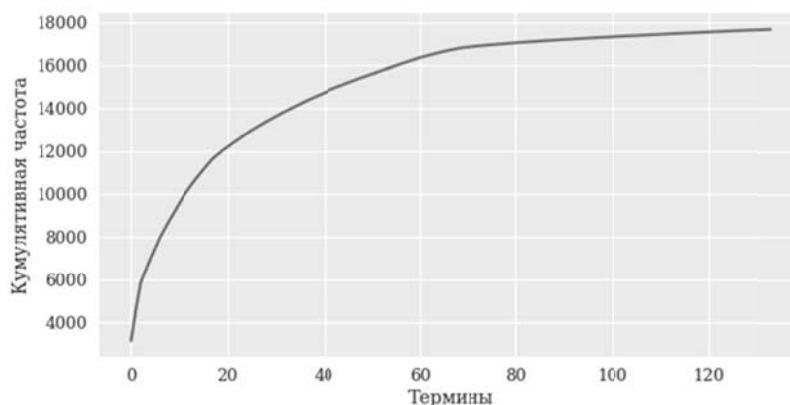


Рисунок 2 – Распределение терминов МКД по убыванию частоты

Основные термины МКД в виде облака тегов представлены на рисунке 3.

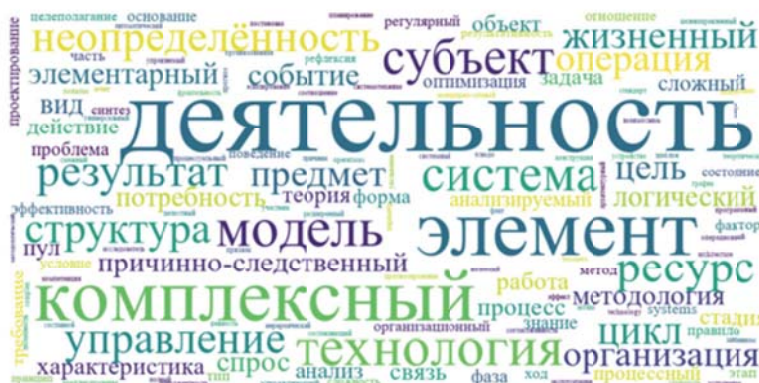


Рисунок 3 – Облако терминов МКД

4.1.2 Выделение терминов – n -грамм ($n > 1$)

Ранжирование n -грамм производилось при помощи различных методов: частота (прямой подсчет количества n -грамм), поточечная взаимная информация, t -тест, χ^2 -тест, отношение функций правдоподобия (см. [8]). Оказалось, что согласно экспертной оценке самый простой способ (ранжирование по частоте) – наилучший (этот вывод подтверждается результатами другого экспериментального исследования [18]).

Биграммы МКД (топ-30):

комплексный деятельность, элемент деятельность, структурный элемент, ЖЦ, организационно-технический система, элементарный операция, пул ресурс, информационный модель, логический структура, процессный модель, причинно-следственный структура, новый технология, причинно-следственный модель, внешний среда, событие неопределённость, результат деятельность, создание технология, нижестоящий операция, нижестоящий элемент, предмет деятельность, субъект деятельность, формирование спрос, цикл ресурс, регулярный деятельность, технология деятельность, systems engineering, достижение цель, получение результат, истинный неопределённость, репликативный деятельность.

3-граммы МКД (топ-30):

структурный элемент деятельность, элемент комплексный деятельность, нижестоящий структурный элемент, анализируемый структурный элемент, методология комплексный деятельность, организация и управление, вышестоящий структурный элемент, модель комплексный деятельность, структура комплексный деятельность, создание новый технология, цикл комплексный деятельность, реализация ЖЦ, выполнение комплексный деятельность, стадия ЖЦ, ЖЦ ресурс, результативность и эффективность, технология комплексный деятельность, реализация комплексный деятельность, модель структурный элемент, субъект комплексный деятельность, реакция на неопределённость, результат комплексный деятельность, целеполагание и создание, управление комплексный деятельность, ЖЦ пул, процедура формирование спрос, неопределённость комплексный деятельность, предмет комплексный деятельность, деятельность в целом, цикл пул ресурс.

4-граммы МКД (топ-30):

нижестоящий структурный элемент деятельность, анализируемый структурный элемент деятельность, вышестоящий структурный элемент деятельность, ЖЦ комплексный деятельность, субъект анализируемый структурный элемент, модель структурный элемент деятельность, логический структура комплексный деятельность, реализация ЖЦ ресурс, субъект нижестоящий структурный элемент, субъект вышестоящий структурный элемент, ЖЦ пул ресурс, деятельность и элементарный операция, логический и причинно-следственный структура, структура структурный элемент деятельность, выполнение структурный элемент деятельность, порождение элемент комплексный деятельность, процессный модель структурный элемент, целеполагание и создание технология, креативный структурный элемент деятельность, модель элемент комплексный деятельность, ЖЦ структурный элемент, комплексный деятельность в целом, спрос и осознание потребность, фиксация спрос и осознание, цикл структурный элемент деятельность, цикл элемент комплексный деятельность, действие и получение результат, оптимизация выполнение комплексный деятельность, субъект структурный элемент деятельность, цепочка реакция на неопределённость.

5-граммы МКД (топ-30):

субъект анализируемый структурный элемент деятельность; субъект нижестоящий структурный элемент деятельность; субъект вышестоящий структурный элемент деятельность; элемент деятельность и элементарный операция; процессный модель структурный элемент деятельность; результативность и эффективность комплексный деятельность; фиксация спрос и осознание потребность; ЖЦ элемент комплексный деятельность; ЖЦ структурный элемент деятельность; требование к методология комплексный деятельность; организация и управление комплексный деятельность; комплексный деятельность с известной технология; комплексный деятельность и организационно-технической система; выполнение действие и получение результат; логический структура структурный элемент деятельность; элемент деятельность и нижестоящий операция; запрос, получение и организация; фаза, стадия и этап; целеполагание и создание новой технология; деятельность, структурный элемент деятельность; задача оптимизация выполнение комплексный деятельность; внутренний упорядоченность, согласованность взаимодействие; стадия ЖЦ комплексный деятельность; порождение новой элемент комплексный деятельность; текущий содержание конкретный агрегат оборудование; структура анализируемый структурный элемент деятельность; модель анализируемый структурный элемент деятельность; анализ, синтез, конкретизация; синтез, конкретизация, регулирование; утверждение о состав комплексный деятельность.

Дальнейший анализ проводится только для терминов, являющихся униграммами.

4.1.3 Построение сети терминов МКД

На основе совместной встречаемости терминов (в пределах предложений текста) построен взвешенный граф терминов, в котором вершинами являются термины, а ребрами – связи между ними, вес ребра равен числу совместных появлений соответствующей пары терминов.

На рисунке 4 показана наибольшая компонента связности графа терминов МКД с ребрами, вес которых больше 50. На дугах – частота совместного появления терминов в предложениях текста.

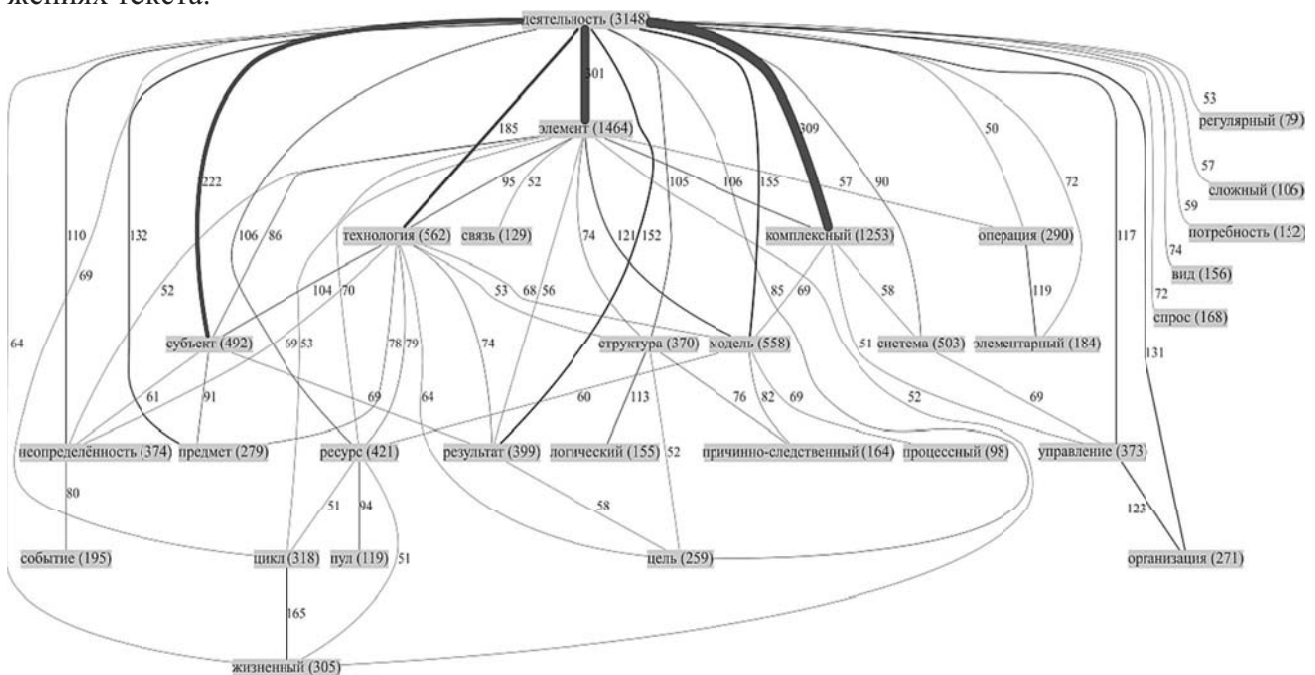


Рисунок 4 – Граф терминов МКД

4.1.4 Анализ структурных центральных терминов

Имея сеть терминов, можно для каждого термина рассчитать его структурную значимость и таким образом определить наиболее важные термины в интерпретации, зависящей от используемого показателя (см. таблицу 1). Для оценки структурной значимости терминов используются следующие показатели: центральность по степени, центральность по PageRank, центральность по близости, центральность по посредничеству, а также влиятельность (влиятельность по акциональной модели).

Таблица 1 – Топ-10 терминов МКД по каждому из показателей

№	Частота	Степень	Близость	Посредничество	Pagerank	Влиятельность
1	деятельность	деятельность	деятельность	деятельность	деятельность	деятельность
2	элемент	элемент	элемент	система	элемент	элемент
3	комплексный	система	комплексный	элемент	технология	технология
4	технология	субъект	субъект	технология	модель	модель
5	модель	структура	технология	модель	субъект	комплексный
6	система	модель	модель	результат	комплексный	субъект
7	субъект	технология	результат	управление	результат	результат
8	ресурс	ресурс	организация	неопределённость	ресурс	ресурс
9	результат	комплексный	управление	ресурс	структура	система
10	неопределённость	вид	предмет	структура	система	структура

На рисунке 5 представлены значения коэффициентов корреляции Спирмена для значений показателей терминов МКД. Высокие коэффициенты ранговой корреляции Спирмена ($> 0,9$) имеют место между пятью переменными – частота, степень, близость, PageRank и влиятельность. Исходя из этого, можно считать, что для оценки структурной значимости терминов достаточно использовать частоту встречаемости терминов и центральность по посредничеству.

4.1.5 Выделение кластеров терминов МКД в сети

Для выделения кластеров терминов в взвешенном графе использованы методы выявления сообществ теории социально- сетевого анализа. Качество кластеризации обычно оценивается при помощи показателя модулярности, принимающего значения от -1 до $+1$ (больше – лучше). Наилучшие результаты показал метод спинового стекла [19], качество кластеризации составило $0,124$.

При помощи метода спинового стекла получено четыре кластера терминов:

- 1) ресурс, цикл, жизненный, пул, стадия, форма, фаза, проектирование, этап, рефлексия, эксплуатация, планирование, операционный, график, взаимосвязь, мера, календарно-сетевой, стоимость, увольнение, компетенция;
- 2) элемент, модель, структура, операция, элементарный, причинно-следственный, вид, логический, связь, анализируемый, процессный, организационный, тип, принцип, управляющий, агент, механизм, иерархический, комплексирование, организованная, полный, фрактальность, конструкция, исследователь;
- 3) технология, субъект, результат, неопределённость, цель, событие, спрос, потребность, характеристика, действии, задача, требование, оптимизация, эффективность, условие, регулярный, правило, отношение, фактор, ход, синтез, целеполагание, результативность, эффект, параметр, полезность, прогноз, устранение, эскалирование, участник, повторение, блюдо, факт, постановка, желаемый, соотношение, прогнозирование;
- 4) деятельность, комплексный, система, управление, предмет, организация, методология, процесс, работа, анализ, объект, теория, сложный, знание, проблема, основание, часть, systems, метод, состояние, поведение, сложность, процессуальный, согласованность, причина, управляемый, признак, расширенный, системный, системотехника, стандарт, complex, архитектурный, целенаправленный, программный, ценность, логика, теоретический, смежный, эмерджентность, универсальный, шаблон, architecture, operations, business, составляющий, устройство, составной, попытка, автоматический, целостный, методологический.

Такие кластеры имеют хорошую содержательную интерпретацию и соотносятся со структурой МКД: кластер 1 включает, в основном, термины, связанные с причинно-следственной (временной) структурой комплексной деятельности (КД) и с жц её элементов; кластер 2 – с логической структурой КД, кластер 3 – с процессной структурой и технологией

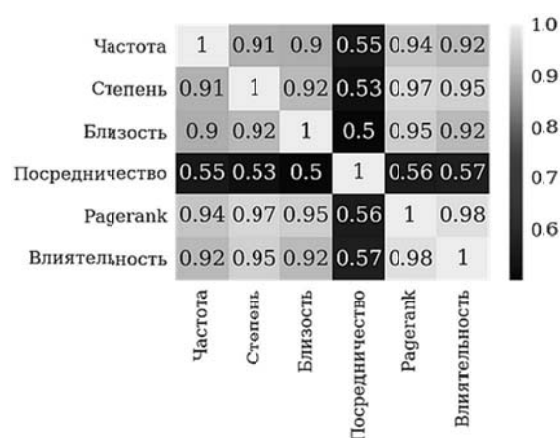


Рисунок 5 – Коэффициенты корреляции Спирмена для МКД

КД, кластер 4 – с организацией и управлением КД и с организационно-техническими системами.

4.2 Анализ терминологической структуры ОМ

В качестве исходного корпуса текста ОМ используется монография [1].

4.2.1 Выделение терминов

Выделение терминов (только униграмм) из текста проводилось согласно сценарию, рассмотренному в разделе 3. Всего получено 150 терминов, наиболее часто в тексте встречаются следующие термины (топ-30 в порядке убывания частоты):

деятельность, научный, система, метод, исследование, процесс, проект, результат, знание, игра, учебный, человек, наука, работа, цель, обучение, художественный, проблема, теория, модель, методология, организация, форма, объект, средство, задача, вид, решение, общий, развитие.

Распределение терминов ОМ показано на рисунке 6, а основные термины ОМ в виде облака тегов представлены на рисунке 7.

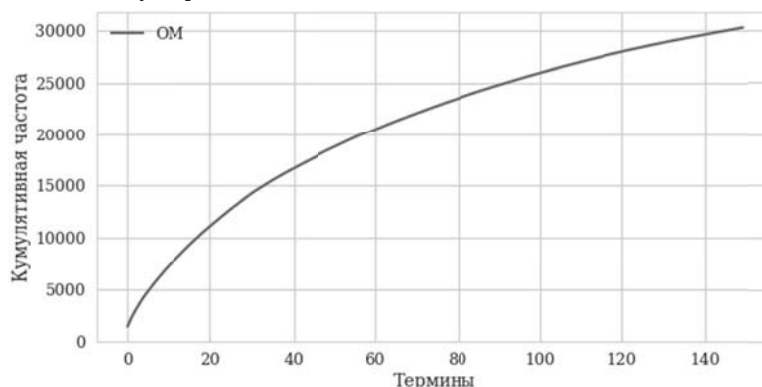


Рисунок 6 – Распределение терминов ОМ по убыванию частоты

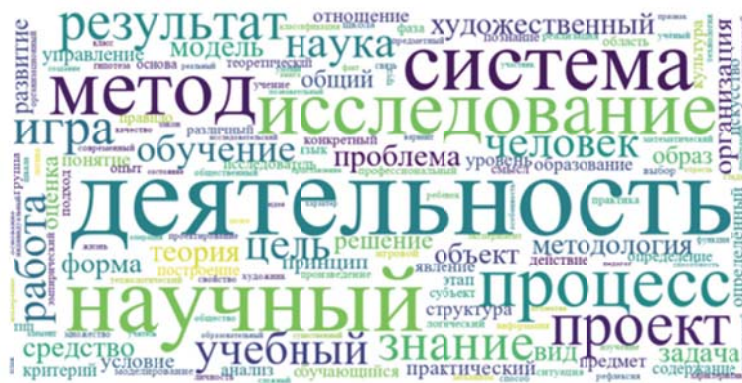


Рисунок 7 – Облако терминов ОМ

4.2.2 Построение сети терминов ОМ

На основе совместной встречаемости терминов ОМ в предложениях текста [1] построен взвешенный граф терминов, в котором вершинами являются термины, а рёбрами – связи между ними, вес ребра равен числу совместных появлений соответствующей пары терминов. На рисунке 8 показана наибольшая компонента связности графа терминов ОМ с рёбрами, вес которых больше или равен 50. На дугах – частота совместного появления терминов в предложениях текста.

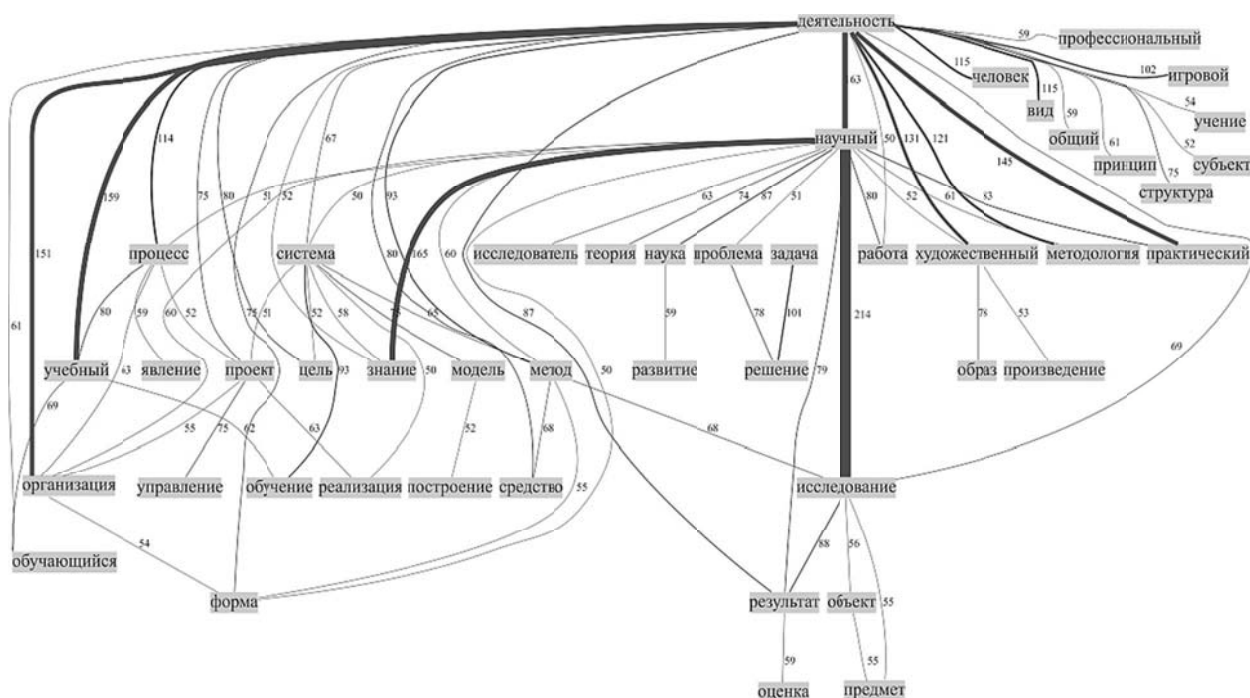


Рисунок 8 – Граф терминов ОМ

4.2.3 Анализ структурных центральностей терминов

Построим для ОМ аналогичные таблице 1 и рисунку 5 таблицу 2 (топ-10 терминов ОМ по каждому из показателей) и рисунок 9 (коэффициенты корреляции для показателей для ОМ). Имея сеть терминов, можно для каждого термина рассчитать его структурную значимость. Для оценки структурной значимости терминов используются следующие показатели: центральность по степени, центральность по PageRank, центральность по близости, центральность по посредничеству, а также влияние по акциональной модели.

Таблица 2 – Топ-10 терминов ОМ по каждому из показателей

№	Частота	Степень	Близость	Посредничество	Pagerank	Влиятельность
1	деятельность	деятельность	деятельность	деятельность	деятельность	деятельность
2	научный	процесс	научный	научный	научный	научный
3	система	цель	учебный	система	система	система
4	метод	форма	исследование	исследование	процесс	процесс
5	исследование	система	организация	учебный	исследование	исследование
6	процесс	результат	практический	художественный	метод	метод
7	проект	вид	художественный	знание	знание	знание
8	результат	общий	процесс	проект	результат	результат
9	знание	метод	знание	метод	учебный	форма
10	игра	научный	методология	объект	цель	учебный

На рисунке 9 представлены значения коэффициентов корреляции Спирмена для значенных показателей терминов ОМ.

Как и для МКД, для случая ОМ высокие коэффициенты ранговой корреляции Спирмена ($> 0,8$) имеют место между частотой и такими переменными как степень, близость, PageRank и влияние. Поэтому можно подтвердить сделанный выше вывод, что для оценки структурной значимости терминов достаточно использовать частоту встречаемости терминов и центральность по посредничеству.

4.2.4 Выделение кластеров терминов ОМ в сети

Для выделения кластеров терминов в взвешенном графе использованы методы выявления сообществ теории социально- сетевого анализа. Наилучшие результаты для ОМ, как и для МКД, показал метод спинового стекла. При помощи метода спинового стекла получено четыре кластера терминов ОМ:

1) деятельность, процесс, учебный, обучение, методология, организация, форма, средство, вид, принцип, практический, структура, образование, обучающийся, культура, содержание, подход, тип, профессиональный, учение, игровой, логический, школа, классификация, организационный, основание, способ, современный, логика, человеческий, характеристика, образовательный, учитель, особенность, педагог, книга;

2) научный, метод, исследование, результат, знание, наука, работа, теория, объект, развитие, понятие, предмет, явление, исследователь, опыт, теоретический, область, различный, основа, познание, эмпирический, практика, гипотеза, эксперимент, связь, учёный, изучение, предметный, свойство, измерение, закон, существенный, представление, признак, исследовательский, факт, отрасль, коллектив, идея;

3) система, проект, цель, проблема, модель, задача, решение, общий, оценка, условие, управление, критерий, анализ, этап, построение, определение, действие, конкретный, фаза, моделирование, реализация, выбор, стадия, проектирование, рефлексия, технология, шкала, информация, множество, программа, технологический, математический, механизм, элемент, функция, вариант, создание, сложный, план, состояние, операция, участник;

4) игра, человек, художественный, образ, отношение, уровень, определённый, искусство, правило, язык, смысл, субъект, группа, ситуация, эстетический, произведение, художник, ребёнок, качество, общество, жизнь, общественный, личность, класс, умение, труд, реальный, целое, индивидуальный, характер, коллективный, способность, познавательный.

Такие кластеры имеют хорошую содержательную интерпретацию и соответствуют структуре ОМ: кластер 1 – термины, характерные, в основном, для методологии практической и образовательной деятельности; кластер 2 – термины, характерные, в основном, для методологии научной деятельности; кластер 3 – «общеметодологические» термины; кластер 4 – термины, характерные, в основном, для методологии игровой и художественной деятельности.

4.3 Сравнение терминологических структур МКД и ОМ

Проведем сравнение между терминологическими структурами МКД и ОМ (см. методы сравнительного анализа в разделе 2), ограничимся при этом первыми 134 топ-терминами.

Рассмотрим распределение терминов по убыванию относительной частоты встречаемости в МКД и ОМ (частота встречаемости термина в соответствующем тексте нормируется на общее число слов в этом тексте, см. рисунок 10).

Как видно из рисунка 10, кривая кумулятивной частоты для МКД круче соответствующей кривой для ОМ и переходит в «плато» быстрее её. Это можно объяснить тем, что Про МКД более чёткая, в то время как в рамках ОМ рассматриваются разные виды деятельности, существенно отличающиеся друг от друга.

Будут ли часто встречающиеся в МКД термины столь же популярными в ОМ, и наоборот? Для этого упорядочим термины по убыванию частоты встречаемости и введём функцию сходства двух ранжирований S (по частоте встречаемости в МКД) и T (по частоте встречаемости в ОМ):

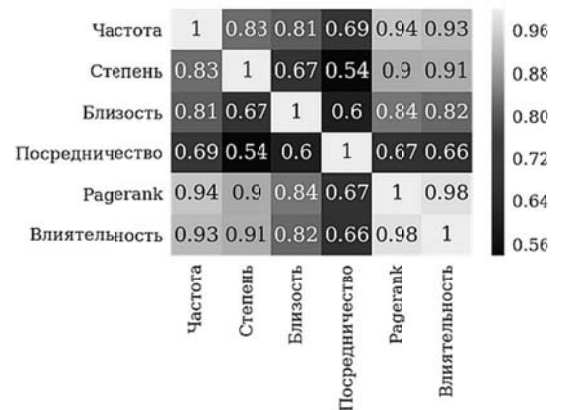


Рисунок 9 – Коэффициенты корреляций Спирмена для ОМ

$$Sim(S, T, k) = \frac{|S_{1:k} \cap T_{1:k}|}{|S_{1:k} \cup T_{1:k}|}$$

где k – число топ-терминов, $S_{1:k}$ – множество первых k элементов ранжирования S , $T_{1:k}$ – множество первых k элементов ранжирования T . Зависимость сходства двух ранжирований от величины топа приведена на рисунке 11.

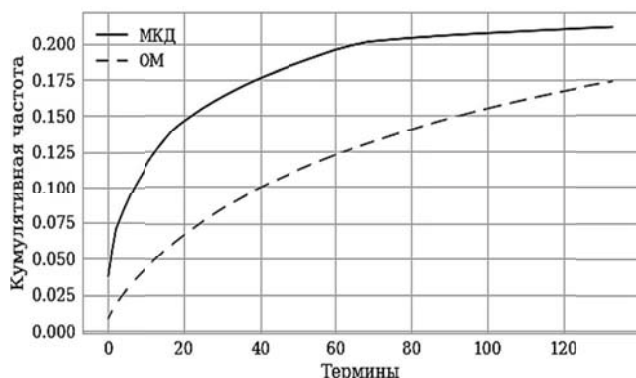


Рисунок 10 – Распределение терминов МКД и ОМ

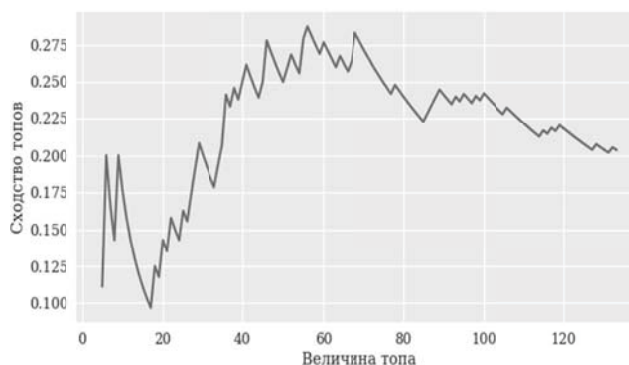


Рисунок 11 – Сходство топов МКД и ОМ

Из рисунка 11 видно, что при малых значениях параметра k множества часто встречающихся терминов МКД и ОМ сильно различаются (таким образом, наиболее популярные термины не являются одинаковыми в МКД и ОМ), затем достигается максимум сходства множеств (в этот момент в этих множествах появляются «популярные» и общие и для МКД, и для ОМ термины), а далее эти множества начинают все больше различаться (в них появляются узкоспециализированные термины).

В целом соотношения между множествами терминов МКД и ОМ показаны в таблице 3 (в скобках для каждого термина указана встречаемость в соответствующем тексте).

Таблица 3 – Ранжированные списки терминов

МКД\ОМ	ОМ\МКД	ОМ ∩ МКД
комплексный (1253)	научный (905)	деятельность (3148, 1440)
ресурс (421)	исследование (619)	элемент (1464, 95)
неопределённость (374)	проект (520)	система (503, 724)
цикл (318)	игра (470)	модель (558, 351)
жизненный (305)	учебный (443)	результат (399, 495)
операция (290)	человек (423)	метод (74, 652)
событие (195)	наука (419)	технология (562, 119)
элементарный (184)	обучение (404)	цель (259, 406)
спрос (168)	художественный (377)	процесс (135, 528)
причинно-следственный (164)	средство (325)	субъект (492, 155)
потребность (152)	решение (307)	управление (373, 242)
пул (119)	общий (296)	организация (271, 340)
анализируемый (102)	развитие (289)	структура (370, 232)
процессный (98)	образ (288)	знание (87, 472)
требование (94)	оценка (270)	работа (132, 406)
оптимизация (90)	практический (251)	предмет (279, 216)
эффективность (86)	понятие (226)	методология (153, 340)
регулярный (79)	уровень (214)	теория (114, 367)
ход (77)	определённый (211)	вид (156, 312)
фактор (77)	образование (210)	проблема (86, 372)
часть (76)	критерий (210)	объект (115, 332)
systems (75)	обучающийся (208)	задача (113, 321)
синтез (62)	культура (203)	форма (98, 335)
целесолагание (60)	явление (198)	условие (80, 249)
состояние (60)	искусство (196)	анализ (122, 206)
поведение (58)	определение (190)	отношение (78, 233)
результативность (54)	построение (190)	действие (124, 185)

МКДОМ	ОММКД	ОМ ∩ МКД
сложность (46)	содержание (190)	принцип (47, 252)
управляющий (36)	опыт (176)	логический (155, 138)
эффект (22)	подход (174)	этап (81, 192)
агент (22)	конкретный (171)	правило (78, 185)
процессуальный (21)	область (170)	связь (129, 115)
эксплуатация (20)	различный (165)	фаза (89, 153)
параметр (20)	основа (164)	стадия (112, 127)
согласованность (19)	язык (161)	тип (74, 162)
планирование (19)	смысл (160)	характеристика (137, 96)
полезность (19)	группа (152)	основание (86, 127)
причина (18)	познание (151)	организационный (81, 128)
иерархический (18)	профессиональный (150)	проектирование (82, 126)
расширенный (16)	моделирование (150)	исследователь (10, 185)
признак (16)	реализация (150)	сложный (106, 81)
управляемый (16)	учение (147)	теоретический (12, 171)
complex (15)	ситуация (145)	рефлексия (48, 120)
архитектурный (15)	эстетический (143)	механизм (21, 97)
системотехника (15)	игровой (143)	логика (12, 105)
системный (15)	выбор (138)	
стандарт (15)	эмпирический (137)	
целенаправленный (14)	школа (137)	
программный (14)	произведение (136)	
прогноз (13)	практика (133)	
комплексирование (13)	классификация (132)	
операционный (13)	художник (132)	
ценность (13)	ребёнок (124)	
устранение (13)	способ (122)	
график (13)	качество (120)	
универсальный (12)	гипотеза (119)	
организованная (12)	современный (118)	
фрактальность (12)	эксперимент (118)	
смежный (12)	шкала (114)	
эскалирование (12)	общество (111)	
мера (12)	учёный (110)	
полный (12)	информация (110)	
эмерджентность (12)	жизнь (110)	
взаимосвязь (12)	множество (109)	
календарно-сетевой (11)	программа (107)	
operations (11)	человеческий (105)	
повторение (11)	математический (104)	
блюдо (11)	технологический (104)	
шаблон (11)	личность (103)	
стоимость (11)	изучение (103)	
business (11)	общественный (103)	
компетенция (11)	предметный (100)	
факт (11)	свойство (100)	
architecture (11)	измерение (97)	
увольнение (11)	класс (95)	
участник (11)	умение (93)	
конструкция (11)	функция (92)	
соотношение (10)	закон (92)	
составляющий (10)	труд (89)	
попытка (10)	вариант (89)	
technology (10)	образовательный (88)	
желаемый (10)	реальный (87)	
автоматический (10)	целое (86)	
прогнозирование (10)	индивидуальный (86)	
постановка (10)	создание (84)	
составной (10)	существенный (83)	
устройство (10)	характер (83)	
целостный (9)	учитель (80)	
методологический (7)	план (80)	

Из таблицы 3 видно (см. также рисунок 12), что общими для двух ПрО являются общие категории, в то время как для МКД специфичны «комплексность», «неопределённость», «технологии» и тесно связанные с ними термины; а для ОМ специфичны термины, отража-

ющие особенности различных видов деятельности (научной, образовательной и др.). На рисунке 12 приводится граф терминов, являющийся объединением графа терминов МКД (вес рёбер в таком графе ≥ 50) и графа терминов ОМ (вес рёбер в таком графе ≥ 50). Метки вершин включают в себя термин, частоту термина в МКД и частоту термина в ОМ. Красным цветом помечены вершины, относящиеся только к МКД, зелёным – только к ОМ, цвет остальных вершин «смешивается» из красного и зелёного в зависимости от соотношения частот терминов в МКД и ОМ. Толщина ребра графа зависит от суммы весов соответствующих рёбер в исходных графах, цвет рёбер, также как и цвет вершин, определяется соотношением этих весов.

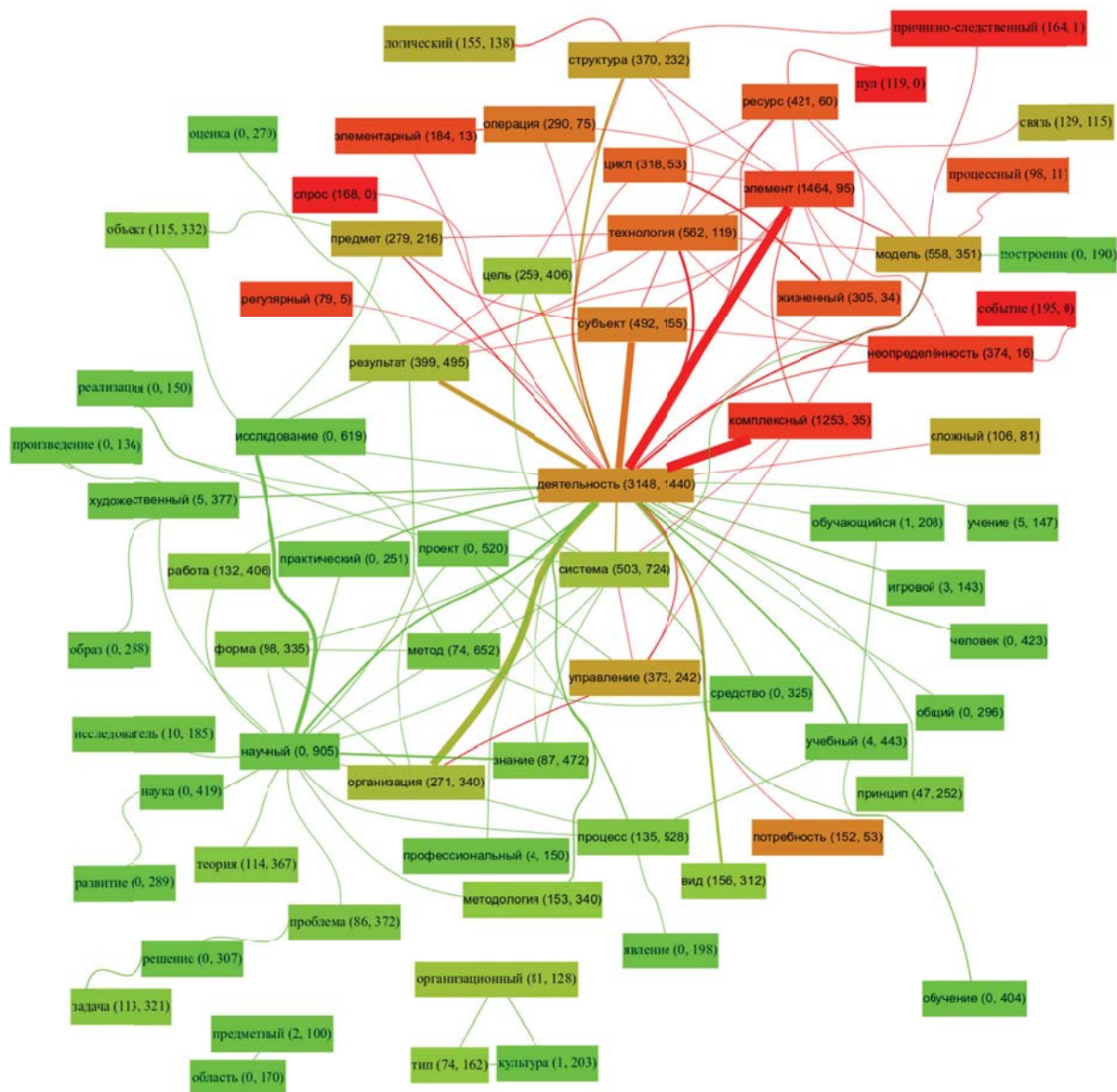


Рисунок 12 – Граф терминов МКД и ОМ

Заключение

Предложенные и апробированные на примере таких Про как МКД и ОМ методы извлечения и анализа терминологических структур позволяют:

- интерактивно (во взаимодействии с экспертом ПрО) извлекать термины ПрО из релевантной ей монографии (без дополнительного корпуса текстов);
- извлекать связи между терминами и формировать терминологическую структуру ПрО;
- оценивать структурную значимость терминов в терминологической структуре и определять наиболее подходящие показатели структурной значимости;
- выявлять содержательно интерпретируемые кластеры терминов терминологической структуры;
- проводить сравнительный анализ терминологических структур различных ПрО.

Список источников

- [1] *Новиков, А.М.* Методология. 2-е изд. / А.М. Новиков, Д.А. Новиков – М.: УРСС, 2013. – 663 с.
- [2] *Губанов, Д.А.* Методы анализа терминологической структуры предметной области (на примере методологии) / Д.А. Губанов, А.В. Макаренко, Д.А. Новиков // Управление большими системами. - 2013. - № 43. - С.5–33. - DOI: 10.1134/S0005117914120133.
- [3] *Белов М.В., Новиков Д.А.* Методология комплексной деятельности. – М.: Ленанд, 2018. – 320 с.
- [4] *Белов, М.В.* Структура методологии комплексной деятельности / М.В. Белов, Д.А. Новиков // Онтология проектирования. – 2017. – Т.7, №4(26). - С.366-387. – DOI: 10.18287/2223-9537-2017-7-4-366-387.
- [5] *Astrakhantsev, N.A.* Methods for Automatic Term Recognition in Domain-Specific Text Collections: A Survey / N.A. Astrakhantsev, D.G. Fedorenko, D.Yu. Turdakov // Programming and Computer Software, 2015. - Vol. 41. - No. 6. – P. 336–349. - DOI: 10.1134/S036176881506002X.
- [6] *Verberne, S.* Evaluation and analysis of term scoring methods for term extraction / S. Verberne, M. Sappelli, D. Hiemstra et al. // Information Retrieval Journal, 2016. - Vol. 19. - Issue 5. - P.510-545. - DOI: 10.1007/s10791-016-9286-2.
- [7] *Астраханцев, Н.А.* Автоматическое извлечение терминов из коллекции текстов предметной области с помощью Википедии / Н.А. Астраханцев // Труды ИСП РАН. - 2014. - Том 26. - Выпуск 4. - С.7–20. - DOI: 10.15514/ISPRAS-2014-26(4)-1.
- [8] *Manning, C.* Foundations of Statistical Natural Language Processing / C. Manning, H. Schütze. – MIT press, 1999. - 620 p.
- [9] *Frantzi, K.* Automatic recognition of multi-word terms: The C-value/NC-value method / K. Frantzi, S. Ananiadou, H. Mima // International Journal on Digital Libraries. - 2000. - Vol.3. – Issue 2. - P.115–130. - DOI: 10.1007/s007999900023.
- [10] *Браславский, П.И.* Сравнение четырех методов автоматического извлечения двухсловных терминов из текста / П.И. Браславский, Е.А. Соколов // Компьютерная лингвистика и интеллектуальные технологии. Сборник трудов международной конференции Диалог 2006. Москва, 2006. - С.88–94.
- [11] *Justeson, J.S.* Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering / J.S. Justeson, S.M. Katz. - 1995. Vol.1. Issue 1. - P.9–27. - DOI: 10.1017/S1351324900000048.
- [12] *Hearst, M.A.* Automatic acquisition of hyponyms from large text corpora // Proceedings of the 14th conference on Computational linguistics. - Vol. 2. – Association for Computational Linguistics, 1992. – P.539-545. DOI: 10.3115/992133.992154.
- [13] *Mikolov T. et al.* Distributed Representations of Words and Phrases and their Compositionality // Advances in Neural Information Processing Systems. 2013. – P.3111-3119.
- [14] *Pelevina, M.* Making Sense of Word Embeddings / M. Pelevina, N. Arefyev, C. Biemann, A. Panchenko // Proceedings of the 1st Workshop on Representation Learning for NLP. Berlin, Germany, August 11th, 2016. - P.174–183. - DOI: 10.18653/v1/W16-1620.
- [15] *Губанов, Д.А.* Акциональная модель влиятельности пользователей социальной сети / Д.А. Губанов, А.Г. Чхартишвили // Проблемы управления. – 2014. - № 4. С.20–25. - DOI: 10.1134/S0005117915070139.
- [16] *Wasserman, S.* Social network analysis: Methods and applications / S. Wasserman, K. Faust. – Cambridge university press, 1994. - 857 p. - DOI: 10.1017/CBO9780511815478.
- [17] *Kempe, D.* Maximizing the Spread of Influence through a Social Network / D. Kempe, J. Kleinberg, E. Tardos // Theory of Computing. - 2015. Vol. 11. No. 4. - P.105–147. - DOI: 10.4086/toc.2015.v011a004.
- [18] *Wermter, J.* You can't beat frequency (unless you use linguistic knowledge): a qualitative evaluation of association measures for collocation and term extraction / J. Wermter, U. Hahn // Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics / Association for Computational Linguistics. - 2006. - P.785-792. - DOI: 10.3115/1220175.1220274.

- [19] **Reichardt, J.** Statistical Mechanics of Community Detection / J. Reichardt, S. Bornholdt // Physical Review E. – 2006. – V. 74. – No. 1. – P.16110. - DOI: 10.1103/PhysRevE.74.016110.

EXTRACTION AND ANALYSIS METHODS FOR THE TERMINOLOGICAL STRUCTURES OF RELATED SUBJECT AREAS

D.A. Gubanov¹, D.A. Novikov²

Trapeznikov Institute of Control Sciences Russian Academy of Sciences, Moscow, Russia

¹dmitry.a.g@gmail.com, ²novikov@ipu.ru

Abstract

We suggest an automated expert approach to extract terminological structures from monographs. It is postulated that the content of scientific subject area is fully reflected in some monograph, it allows to extract and analyze terminological structures without using a pre-prepared corpus of specialized domain documents. Within the framework of the approach, we consider the procedure of iterative extraction and analysis of terminological structures that regulates the work of an expert with a monograph and allows him to control the process. Comparative analysis methods for the terminology structures of two related subject areas are proposed. The possibilities of the developed approach are demonstrated on the example of the general methodology and methodology of complex activities.

Key words: *subject area, terminology structure, graph theory, general methodology, methodology of complex activity.*

Citation: *Gubanov DA, Novikov DA. Extraction and analysis methods for the terminological structures of related subject areas [In Russian]. Ontology of designing. 2018; 8(3): 347-365. - DOI: 10.18287/2223-9537-2018-8-3-347-365.*

References

- [1] **Novikov AM, Novikov DA.** Methodology [In Russian]. – Moscow.: URSS, 2013. – 663 p.
- [2] **Gubanov DA, Makarenko AV, Novikov DA.** Analysis methods for the terminological structure of a subject area // Automation and Remote Control. 2014; 75(12): 2231-2247. DOI: 10.1134/S0005117914120133.
- [3] **Belov MV, Novikov DA.** Methodology of complex activity [In Russian]. – Moscow: Lenand; 2018.
- [4] **Belov MV, Novikov DA.** Structure of methodology of complex activity [In Russian]. Ontology of designing. 2017; 7(4): 366-387. DOI: 10.18287/2223-9537-2017-7-4-366-387.
- [5] **Astrakhantsev NA, Fedorenko DG, Turdakov DYU.** Methods for Automatic Term Recognition in Domain-Specific Text Collections: A Survey // Programming and Computer Software, 2015; 41(6): 336–349. DOI: 10.1134/S036176881506002X.
- [6] **Verberne S., Sappelli M, Hiemstra D. et al.** Evaluation and analysis of term scoring methods for term extraction // Information Retrieval Journal, 2016; 19(5): 510-545. DOI: 10.1007/s10791-016-9286-2.
- [7] **Astrakhantsev NA.** Automatic term recognition in a domain-specific text collection using Wikipedia [In Russian]. Proceedings of ISP RAS, 2014; 26(4): 7–20. - DOI: 10.15514/ISPRAS-2014-26(4)-1.
- [8] **Manning C, Schütze H.** Foundations of Statistical Natural Language Processing. – MIT press, 1999. – 620 p.
- [9] **Frantzi K, Ananiadou S, Mima H.** Automatic recognition of multi-word terms: The C-value/NC-value method // International Journal on Digital Libraries. – 2000; 3(2): 115–130. DOI: 10.1007/s007999900023.
- [10] **Braslavskii PI, Sokolov EA.** Comparison of four methods for automatic recognition of two-word terms in text [In Russian]. Computational Linguistics and Intellectual Technologies, 2006. - P.88–94.
- [11] **Justeson JS, Katz SM.** Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering. 1995; 1(1): 9–27. DOI: 10.1017/S1351324900000048.
- [12] **Hearst MA.** Automatic acquisition of hyponyms from large text corpora // Proceedings of the 14th conference on Computational linguistics. - Vol. 2. – Association for Computational Linguistics, 1992. – P.539-545. DOI: 10.3115/992133.992154.
- [13] **Mikolov T. et al.** Distributed Representations of Words and Phrases and their Compositionality // Advances in Neural Information Processing Systems. 2013. – P.3111-3119.

- [14] *Pelevina M, Arefyev N, Biemann C, Panchenko A*. Making Sense of Word Embeddings / Proceedings of the 1st Workshop on Representation Learning for NLP. Berlin, Germany, August 11th, 2016. P. 174–183. DOI: 10.18653/v1/W16-1620.
- [15] *Gubanov DA, Chkharitshvili AG*. An actional model of user influence levels in a social network // Automation and Remote Control. 2015; 76(7): 1282–1290. DOI: 10.1134/S0005117915070139.
- [16] *Wasserman S, Faust K*. Social network analysis: Methods and applications. – Cambridge university press, 1994. – 857 p. DOI: 10.1017/CBO9780511815478.
- [17] *Kempe D, Kleinberg J, Tardos E*. Maximizing the Spread of Influence through a Social Network // Theory of Computing. 2015; 11(4): 105–147. DOI: 10.4086/toc.2015.v011a004.
- [18] *Wermter J, Hahn U*. You can't beat frequency (unless you use linguistic knowledge): a qualitative evaluation of association measures for collocation and term extraction // Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics / Association for Computational Linguistics. 2006. - P.785-792. DOI: 10.3115/1220175.1220274.
- [19] *Reichardt J, Bornholdt S*. Statistical Mechanics of Community Detection // Physical Review E. 2006; 74(1): 16110. DOI: 10.1103/PhysRevE.74.016110.

Сведения об авторах



Губанов Дмитрий Алексеевич, 1984 г. рождения, кандидат технических наук, старший научный сотрудник Института проблем управления Российской академии наук, автор более 60 научных работ. Область научных интересов – информационное управление в социальных сетях, социально-сетевой анализ, теория управления социально-экономическими системами.

Dmitry Alexeyevich Gubanov was born in 1984. He received Candidate of Science degree in engineering (Ph.D. degree) from the Institute of Control Sciences of the Russian Academy of Sciences, Moscow, in 2009. At present, he is a Senior Researcher with the Institute of Control Sciences of the Russian Academy of Sciences, Moscow, Russia. He is the author of more than 60 publications. His research interests include social network analysis, big data analytics, decision-making and mechanisms of control of social and economic systems.



Новиков Дмитрий Александрович, 1970 г. рождения, доктор технических наук, профессор, член-корреспондент РАН, директор Института проблем управления Российской академии наук, заведующий кафедрой интегрированных киберсистем Московского физико-технического института. Автор более 500 научных работ по теории управления системами междисциплинарной природы, в том числе – по системному анализу, теории игр, принятию решений, управлению проектами и математическим моделям механизмов управления социально-экономическими системами.

Dmitry Aleksandrovich Novikov was born in 1970. He is Doctor of Science in engineering. At present, he is Director of the Institute of Control Sciences of the Russian Academy of Sciences, professor, Corresponding Member of Russian Academy of Sciences. He is the author of more than 500 publications in the field of control of systems of interdisciplinary nature, including

system analysis, game theory, project management, decision-making and mechanisms of control of social and economic systems.

УДК 004.032.26, 004.822.2

ИНТЕГРАЦИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С БАЗАМИ ЗНАНИЙ

В.А. Головки^{1, a}, В.В. Голенков^{2, b}, В.П. Ивашенко^{2, c}, В.В. Таберко^{3, d},
Д.С. Иванюк^{3, e}, А.А. Крощенко^{1, f}, М.В. Ковалёв^{2, g}

¹ Брестский государственный технический университет, Брест, Республика Беларусь

^agva@bstu.by, ^fkroschenko@gmail.com

² Белорусский государственный университет информатики и радиоэлектроник, Минск, Республика Беларусь

^bgolen@bsuir.by, ^civashenko@bsuir.by, ^gmichail.kovalev7@gmail.com

³ ОАО «Савушкин продукт», Брест, Республика Беларусь

^dtab@pda.savushkin.by, ^eid@pda.savushkin.by

Аннотация

Статья посвящена вопросам и направлениям интеграции искусственных нейронных сетей с базами знаний. Рассмотрены два направления интеграции: коммуникация через входы и выходы искусственной нейронной сети с целью использования интеграции баз знаний и искусственных нейронных сетей для решения прикладных задач; через представление искусственных нейронных сетей с помощью онтологических структур и их интерпретацию средствами представления знаний в базе знаний с целью создания интеллектуальной среды по разработке, обучению и интеграции искусственных нейронных сетей, совместимых с базами знаний. Базы знаний, с которыми интегрируются искусственные нейронные сети, построены на основе однородных семантических сетей, а обработка знаний в них осуществляется с помощью многоагентного подхода. Предложена онтологическая модель представления искусственных нейронных сетей и их спецификаций в рамках модели унифицированного семантического представления знаний, отличающаяся возможностью представления в базах знаний искусственных нейронных сетей, их динамики и знаний других видов, включая спецификации искусственных нейронных сетей в виде текстов одного языка представления знаний с общей теоретико-модельной семантикой. Предложена многоагентная модель решения задач с использованием искусственных нейронных сетей и знаний других видов, отличающаяся взаимодействием агентов в соответствии с заданной темпоральной моделью через общую память, хранящую знания, интегрированные в единую базу знаний.

Ключевые слова: искусственные нейронные сети, база знаний, интеграция, интеллектуальная среда разработки.

Цитирование: Головки, В.А. Интеграция искусственных нейронных сетей с базами знаний / В.А. Головки, В.В. Голенков, В.П. Ивашенко, В.В. Таберко, Д.С. Иванюк, А.А. Крощенко, М.В. Ковалёв // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.366-386. – DOI: 10.18287/2223-9537-2018-8-3-366-386.

Введение

В задачах, решаемых современными интеллектуальными системами (ИС), используют различных модели представления знаний. Интеграция моделей такого рода в рамках одной системы зачастую сопряжена со значительными трудностями.

К одному из наиболее активно развиваемых направлений в области искусственного интеллекта относится направление, связанное с решением задач на основе методов машинного обучения. Популярность методов решения задач на основе машинного обучения в значительной степени вызвана развитием теоретических моделей искусственных нейронных сетей (ИНС) и производительных аппаратных платформ для их реализации. Постоянно увеличивается многообразие архитектур, методов, направлений и способов применения ИНС.

Становится актуальной разработка подходов по интеграции ИНС и баз знаний (БЗ), использующих онтологии. Такие интегрированные системы способны сочетать возможность семантической интерпретации обрабатываемых ИНС данных, используя представление прикладных задач, решаемых ИНС, спецификацию её входных и выходных данных с представлением самой структуры ИНС, описанием её свойств и состояний, позволяющим упростить понимание её работы.

1 Искусственные нейронные сети и знания

Появление ИНС связывают с работами У. Мак-Каллока и У. Питтса [1], в которых, однако, предпочитается термин *сеть нейронов*. Термин «нейронная сеть» («нейронная сетка») закрепился, начиная с работ К. Фукусимы [2].

Особенный интерес имеет применение глубоких нейронных сетей для формирования абстракций. Эти сети обладают преимуществом, отличающим их от поверхностных моделей – возможность формирования сложной иерархии признаков [3]. Впервые эффективный алгоритм обучения глубоких нейронных сетей был предложен Дж. Хинтоном в 2006 году [4]. Начиная с этого момента, появилось много работ, посвящённых обучению такого типа сетей (например, [5, 6]). Были предложены и принципиально иные подходы (см., например, [7, 8]).

В данной работе рассматриваются ИНС, построенные на основе понятия формального нейрона [9]. Формальный нейрон задаётся композицией двух функций:

- функция синаптического преобразования;
- функция активации нейрона.

ИНС на основе формальных нейронов задаётся четвёркой компонентов:

- множество вершин нейронной сети (V);
- множество рёбер нейронной сети (E ($E \subseteq V \times V$));
- множество формальных нейронов, их свойств и параметров нейронной сети (N);
- множество соответствий между объединением множеств вершин и рёбер нейронной сети и множеством формальных нейронов, их свойств и параметров нейронной сети S .

$$(1) \quad S \subseteq 2^{(V \cup E) \times N}$$

ИНС можно соотносить с понятием знания в соответствии с разными подходами к пониманию знаний. Соотнесение на основе признаков [10]:

- связность (ИНС имеют связный ориентированный граф, но не всегда сильносвязный);
- сложная структура (существуют ИНС со сложной структурой, состоящей из множества слоёв, вершин нейронной сети и обратных связей);
- интерпретируемость (программные реализации ИНС позволяют их интерпретировать);
- активность (программные реализации ИНС обладают активностью);
- семантическая метрика (семантическая метрика в общем виде и случае для (вершин) ИНС отсутствует).

В соответствии с IDEF-5 [11] знания выражаются в виде отношений, свойств, видов и атрибутов. В соответствии с этим у ИНС есть только числовые атрибуты и возможно, иногда, виды и отношения, заданные с помощью характеристических функций (предикатов). Однако, в общем случае характеристические функции не определены и не известны.

С точки зрения структурного подхода ИНС – математическая структура (модель), соответствующая принципам коннекционизма, которая соответствует информационному процессу, протекающему в технической (кибернетической) системе, выражаемому в виде соответствия или функции.

С прагматико-математической точки зрения данные, обрабатываемые ИНС, есть элементы некоторого множества, зачастую интерпретируемые как признаковый набор (образ, атри-

буты некоторой сущности), а само множество – как признаковое (векторное) пространство. Под признаком понимается числовой, количественный атрибут объекта, сущности, который может быть выражен, задан с помощью соответствия или функции. Различают бинарные, номинальные, порядковые и количественные признаки.

2 Классификация задач, решаемых ИНС

Задачу можно задать отношением между описанием исходной ситуации и описанием множества целевых ситуаций. Классифицировать задачи можно в соответствии с разными признаками. Исходя из общеметодологических принципов, можно выделить признаки классов задач, которые удобно решать с помощью ИНС:

- трудноформализуемые задачи, решение которых обладает естественным (векторным) параллелизмом или параллелизмом данных;
- задачи, решение которых устойчиво к наличию НЕ-факторов в данных и знаниях.

Перечислим возможные признаки классов задач, которые невыгодно решать только с помощью ИНС:

- сложное концептуальное описание в задаче;
- наличие НЕ-факторов и неустойчивость решения к ним;
- преобладание сложно описываемых последовательных процессов в решении задачи;
- необходимость интроспективного анализа и объяснения полученных результатов в сложных ИНС.

Накопленный опыт показывает, что ИНС успешно применяются для решения задач следующих видов (см. таблицу 1).

Таблица 1 – Классификация видов задач, решаемых с применением ИНС

Концептуально-прагматическая классификация	Математическая классификация
	аппроксимация:
управление процессами	- оптимизация
прогнозирование	- экстраполяция
генерация образов	
сопоставление образов	
выявление аномальных явлений	
логический вывод	логический вывод:
трансляция образов:	
- кластеризация образов	- векторное квантование
- сжатие образов	- понижение размерности
- линейное разделение образов	- повышение размерности
- ассоциирование образов:	
- классификация образов	
- распознавание образов	

Среди классов задач можно выделить классы задач, условно являющиеся специализированно-прикладными, и классы задач, условно являющиеся абстрактно-фундаментальными [12].

3 Интеграция ИНС с базами знаний при решении прикладных задач

ИС с БЗ при решении прикладных задач с помощью нейросетевых алгоритмов могут использоваться как для внутренних задач, таких как обучение нейронной сети или оптимизация её работы, так и для решения целевой задачи путём обработки входных и выходных пара-

метров ИНС. Такие ИС могут использовать нейросетевые методы наравне с другими имеющимися в системе методами для решения только одной, выделенной системой из подзадач, при решении какой-либо прикладной задачи.

Далее будут рассмотрены примеры применения интеграции ИНС с БЗ при решении прикладных задач.

3.1 Интеграция нейронных сетей с базами знаний в системе распознавания маркировки продукции в ОАО «Савушкин продукт»

Одной из производственных задач в ОАО «Савушкин продукт» является задача распознавания маркировки продукции [13]. На базе ИНС [14, 15] разрабатывается модуль, который анализирует изображение с камеры, находящейся над конвейером (рисунок 1), которая распознаёт нанесённую на крышку маркировку - строку с датой изготовления и сроком годности (рисунок 2).



Рисунок 1 – Конвейер с блоком камеры



Рисунок 2 – Бутылка с йогуртом с маркированными крышками

Простейший способ интеграции данного модуля в систему розлива предполагает приостановку конвейера и уведомление об этом инженера КИПиА (Контрольно Измерительные Приборы и Автоматика) посредством аварийной звуковой и световой сигнализации, если модулем будет обнаружено n (например, 5) подряд идущих бутылок с ошибкой распознавания маркировки. Он должен определить причину неисправности и её устранить. В этом случае данный модуль распознаёт строку и передаёт результат в БЗ, в которой действие осуществляется согласно следующим правилам:

- если распознанная строка не совпадает с эталонной и это единичный случай, то надо просто эту бутылку отбраковать (при наличии модуля отбраковки);
- если распознанная строка пустая, и это повторяется для n подряд идущих бутылок, то надо остановить конвейер и уведомить инженера КИПиА о необходимости заправить принтер чернилами, которые закончились;
- если распознанная строка не пустая, но не совпадает с эталонной, и это повторяется для n , то надо остановить конвейер и уведомить инженера КИПиА о необходимости очистить печатающий блок принтера, в котором забились сопла.

Таких правил может быть много. Кроме того, необходимо иметь возможность их просматривать и редактировать (возможно, даже инженером КИПиА).

Так как постоянно меняется потребительская упаковка (и маркировка), возникает вопрос: кто и как будет производить настройку данной системы на новый вид маркировки? Предположим, сегодня печатаются арабские цифры, а завтра маркировать продукцию нужно будет римскими цифрами. Что необходимо для этого сделать - сформировать обучающую выборку, обучить новую ИНС, запустить её. Такую работу не может выполнить инженер КИПиА (недостаточно знаний в данной области). Но если будет существовать некоторый

посредник между ними в виде ИС, то тогда все может выглядеть так: ИС получает от нейронной сети информацию о том, что не удаётся распознать надписи, а от пользователя (инженер КИПиА) информацию о том, что всё в порядке и просто запущен новый вид маркировки, тогда она самостоятельно может выполнить перечисленные ранее задачи для адаптации модуля распознавания к новой упаковке.

3.2 Применение ИНС для поддержки и актуализации базы знаний

Сегодня исследователями в области ИНС предлагаются архитектуры, состоящие из сетей различных типов. Один из путей интеграции ИНС и БЗ лежит в плоскости использования таких архитектур-гибридов, формирующих систему обработки потоков знаний для решения сложных задач. Примером такой структуры является гибридная сеть, использующая свёрточные нейронные сети (CNN – convolutional neural network) и рекуррентные LSTM-сети (long short-term memory) [16]. Схема подобной нейросетевой модели LRCN (Long-term Recurrent Convolutional Network), представлена на рисунке 3.

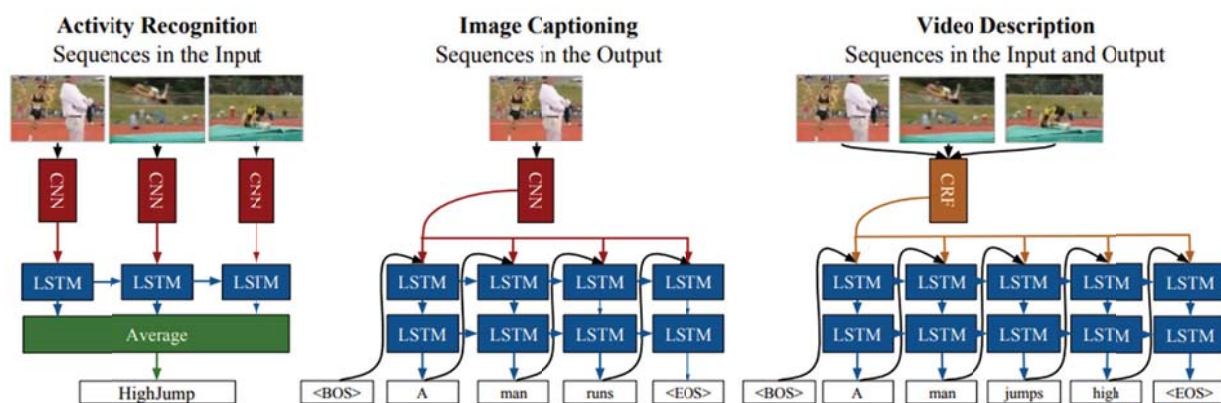


Рисунок 3 – Долгосрочная рекуррентная свёрточная сеть [16]

Здесь фактически представлено два различных варианта такой архитектуры, каждый из которых решает свою задачу. Первый вариант предполагает усреднение результатов работы LSTM-сети и получение компактного описания производимого действия. Такой результат можно рассматривать как распознавание действия, представленного последовательностью изображений. Во втором варианте осуществляется последовательное применение LSTM-сети для формирования текстового описания к изображению. В обоих вариантах задействовано два типа нейросетевой архитектуры. Первый тип – CNN, которая обрабатывает исходное подаваемое на вход изображение и формирует признаки. Второй тип – LSTM-сеть, принимающая на вход признаки и генерирующая текстовое описание. В соответствии с классификацией, приведённой в разделе 2, такая задача относится к задачам распознавания образов (1-й вариант модели) и задачам генерации образов (2-й вариант модели).

Итоговое текстовое описание, полученное с помощью LRCN, может быть проанализировано, сформулировано в виде правила и интегрировано в семантическую сеть. Семантическая сеть осуществляет накопление полученных знаний в виде конкретных объектов и связей между ними и способна в дальнейшем выполнять логические заключения. Помимо анализа изображений, похожая архитектура применяется для формирования текстовых описаний видеопотока.

Другой вариант использования базируется на применении в качестве базового подхода метода word2vec [17]. Этот метод позволяет осуществлять семантический анализ текста с выделением наиболее близких по смыслу слов. В основе метода лежит использование про-

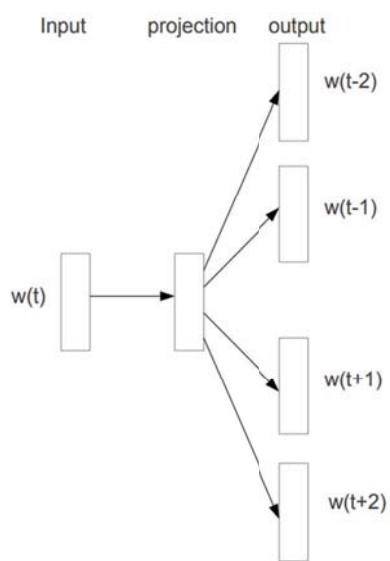


Рисунок 4 – Вариант skip-gram

стой по своей структуре многослойной нейронной сети, которая осуществляет формирование контекста на основе одного целевого слова (вариант skip-grams – рисунок 4 [17]), либо формирование целевого слова на основе передаваемого контекста (вариант CBOW – Continuous Bag of Words).

Под контекстом понимается совокупность слов (слева и справа), окружающая целевое слово, взятая в пределах определённого окна. В результате обученная ИНС осуществляет отображение слова, записанного в виде унитарного кода (и, соответственно, принадлежащего словарю) в пространство меньшей размерности, которое впоследствии используется для оценки семантической близости слов. Полученное погружение может быть использовано для предсказания семантических отношений. Например, *король для королевы* то же что *отец для ? (king is to queen as father is to ?)*.

Приведём пример визуализации, полученной для выборки из 100000 англоязычных документов Википедии и общего размера словаря 50000 слов. В эксперименте использовалась упрощённая архитектура skip-grams, включающая 50.000 входных нейронов, соответствующих целевому слову, 300 скрытых и 50000 выходных нейронов, соответствующих контекстному слову. При обучении были сформированы пары слов (*целевое слово, контекстное слово*), которые подавались на нейронную сеть мини-батчами по 128 пар в каждом. После обучения к редуцированным кодам слов был применен алгоритм t-SNE для уменьшения размерности данных. Полученная двумерная карта семантического сходства изображена на рисунке 5.



Рисунок 5 – Карта семантического сходства

Метод word2vec может использоваться как для формирования БЗ в определённой ПрО (например, [18]), так и извлечения семантических реляций в общем [19].

По классификации, приведённой в разделе 2, эта задача относится к задачам сжатия (понижения размерности).

3.3 Пример использования базы знаний на этапе обучения ИНС

Вариантом интеграции нейронных сетей с БЗ является использование БЗ при обучении ИНС. При таком подходе БЗ содержит в себе описание ПрО, в рамках которых решается задача, и с их помощью можно определить входные и выходные значения ИНС. С БЗ работает машина обработки знаний, которая сможет интерпретировать хранящиеся правила *валидации* (проверка на соответствие погружаемого значения некоторым условиям), проверки на *непротиворечивость* (проверка того, что новые знания не нарушают логические правила, хранящиеся в БЗ) и *корректировки* (замена одних знаний другими) знаний.

Такая интеграция может быть использована для выполнения следующих задач.

- Валидация и корректировка обучающих, тестовых и экзаменационных выборок. Все элементы выборок погружаются в БЗ, что подразумевает проверку на непротиворечивость. В системе также может храниться набор правил по валидации (тестированию) и корректировке значений признаков входных элементов.
- Проверка непротиворечивости и корректности выхода ИНС на этапе обучения и осуществление соответствующей корректировки обучения. Обучение ИНС корректируется в зависимости от промежуточных результатов, которые с помощью обращения к БЗ проверяются на валидность и противоречивость. Результат может быть скорректирован, для чего в БЗ должны присутствовать соответствующие правила. В зависимости от результата обращения к БЗ может изменяться алгоритм обучения за счёт прямого учёта семантики.
- Валидация и корректировка результатов работы обученной ИНС. Результат работы ИНС может проверяться на валидность и противоречивость, а также корректироваться как с помощью правил, использовавшихся во время обучения, так и с помощью другого специализированного набора правил.

Для выполнения этих задач может использоваться как уже готовая БЗ некоторой ИС, так и специально разработанная для обучения конкретной ИНС БЗ.

Рассмотрим предлагаемый метод на примере задачи о классификации. Постановка задачи классификации: имеется группа объектов, у которых выделяются n признаков, необходимо отнести каждый объект к одному из p заданных классов. ИНС обучается на некоторой выборке объектов с известными признаками и в зависимости от типа обучения (с учителем или без) с известной принадлежностью к классу.

Процесс обучения можно улучшить, если описать классы объектов и их признаки в БЗ. К примеру, на этапе обучения без учителя ИНС отнесла объект к некоторому классу j . Так как в БЗ хранится описание этого класса, можно будет проверить, не противоречит ли значение признаков этого объекта описанию класса, в который его отнесли. И уже зная это, можно будет делать вывод о необходимости и способе корректировки весов ИНС, таким образом расширяя алгоритм обучения.

Возьмём условную задачу определения стиля текста. Имеется текст и автор. Из текста извлекаются такие условные признаки, как средняя длина предложений, средняя длина абзацев, наличие прямой речи, частота использования глаголов и существительных. Необходимо определить, какому из пяти традиционных стилей – разговорный, научный, официально-деловой, публицистический, художественный – принадлежит текст. На рисунке 6 представлено схематическое изображение входов и выходов ИНС, решающей эту задачу.

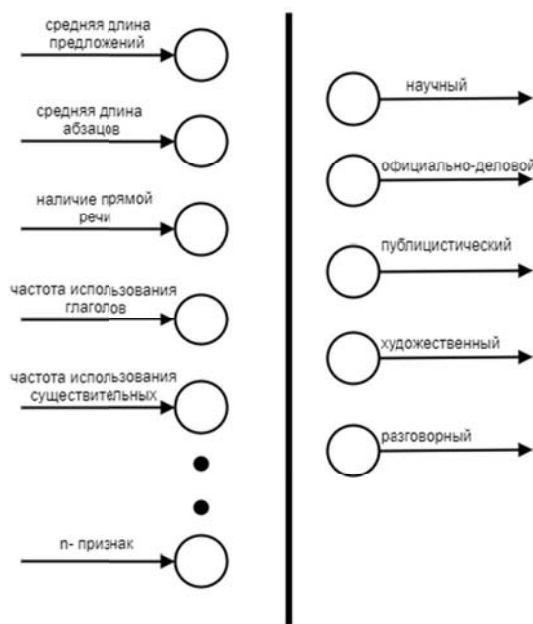


Рисунок 6 – Схематическое изображение входов и выходов искусственной нейронной сети для решения задачи определения стиля текста

Каждый элемент обучающей выборки погружается в БЗ. Семантическая окрестность погружённых элементов расширяется уже имеющимися в системе знаниями. На рисунке 7 представлен фрагмент БЗ, описывающий погружённый элемент выборки текстов для определения их стиля. Его семантическая окрестность расширяется уже имеющимися в системе знаниями об авторе текста и стилях, которые он использует в своих текстах.

Эта информация может быть использована для валидации и корректировки результатов ИНС во время обучения. К примеру, во время обучения ИНС определила стиль текста как разговорный. В БЗ имеется правило валидации, при котором тексту можно определить только такой стиль, который уже использовал ранее его автор. Также в системе имеется правило корректировки результата, при котором если определен стиль текста, который автор ранее не использовал, то можно заменить его

на схожий, который автор использовал. Из фрагмента БЗ, представленного на рисунке 7, система сможет применить описанные правила валидации и корректировки, и заменить разговорный стиль на наиболее близкий ему художественный. Причём в зависимости от того, были ли применены правила валидации и корректировки и от результатов применения этих правил, может меняться алгоритм пересчета весов в ИНС.

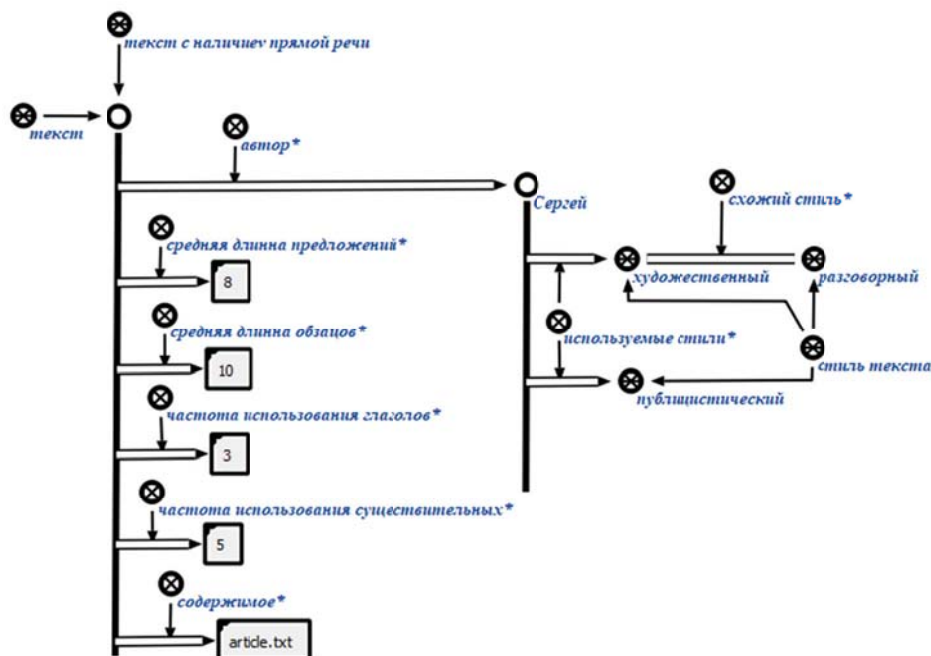


Рисунок 7 – Фрагмент БЗ, описывающий погружённый элемент выборки текстов для определения их стиля

Описанный подход поможет увеличить степень учёта семантики во время обучения, отбросить или скорректировать заведомо неудачные элементы выборок, а также проверить результат работы ИНС на правдоподобность.

4 Интеллектуальная среда

Интеллектуальная среда – это набор инструментов, способный осуществлять автоматический подбор параметров используемой модели в зависимости от решаемой задачи с минимальным участием конечного пользователя. Такая система способна помочь людям, не являющимся специалистами в области машинного обучения и нейронных сетей, успешно применять последние разработки в своей сфере деятельности.

Успехи в разработке высокоуровневых фреймворков последних лет связаны с возможностями использования в качестве аппаратной платформы для расчётов видеокарт (GPU), а также со способностью выполнять параллелизацию вычислений на разные устройства как одного компьютера, так и целого кластера компьютеров [20].

Известно большое количество различных библиотек, позволяющих успешно внедрять решения на основе интеллектуальных алгоритмов. Однако для этого необходимо иметь набор знаний и навыков, позволяющий модифицировать и улучшать стандартные решения. Выделим часто используемые фреймворки [21].

TensorFlow – одна из самых популярных на настоящий момент библиотек. Была разработана подразделением Google (2015). Позволяет запускать модели на нескольких CPU- и GPU-устройствах (хорошо масштабируется). Доступна для разных платформ, поддерживает разные языки программирования (C++, R, Python).

Caffe/Caffe2 – одна из первых библиотек глубокого обучения. Написана на C++, имеет интерфейс на Python. В основном ориентирована на обучение и использование моделей, построенных на свёрточных и многослойных сетях. Создано большое количество предобученных архитектур для Caffe. Facebook в 2017 году предложил новую версию Caffe2, которая предлагает большую гибкость в построении высокопроизводительных глубоких моделей.

Theano – одна из первых библиотек, реализующих алгоритмы глубокого обучения. Пользуется устойчивой популярностью у специалистов в области глубокого обучения. Часто используется как нижележащий фреймворк для многих высокоуровневых библиотек, которые предоставляют API-обертки (к примеру, Keras).

Keras – предоставляет упрощённый интерфейс для работы с Theano, Tensotflow или CNTK. Очень простой в изучении и использовании. Хорошо документирован. Позволяет описать создание и обучение ИНС в нескольких строчках кода. Написан на Python.

Torch – фреймворк, базирующийся на Lua. Возможно использование библиотек C/C++ и CUDA. Очень простая библиотека, процесс построения моделей максимально упрощен. Есть более современная реализация на Python, называемая PyTorch.

Отметим также другие фреймворки, которые активно используются. Это Microsoft CNTK, MXNet и базирующийся на нем Gluon, ONNX, выпущенный в сентябре 2017 года и представляющий собой открытый формат для представления глубоких нейронных моделей.

Основной недостаток разрабатываемых фреймворков — это требование к наличию знаний о структурах моделей, которые должны использоваться для каждой конкретной задачи.

Известно, что выбор нейронной сети для решения задачи — процесс во многом творческий и в большей степени эмпирический, поскольку предполагает получение результатов путём подбора и оценки эффективности различных архитектур сетей. Однако уже сейчас успешно разрабатываются подходы к автоматической генерации моделей нейронных сетей [22], которые показывают преимущество перед ручным подбором параметров.

5 Предметная область ИНС и их денотационная и операционная семантика

Основной частью системы, разрабатываемой с помощью технологии OSTIS [13, 23, 24] является онтологическая модель (sc-модель БЗ), построение которой осуществляется на основе онтологического подхода. Данный подход подразумевает построение онтологий как систем абсолютных и относительных понятий, описывающих ту или иную ПрО. В рамках технологии OSTIS уточняется понятие онтологии как спецификации ПрО [25, 26], выделяется их типология.

5.1 Представление нейронных сетей в базе знаний

Предлагаемый подход основан на использовании БЗ, соответствующих модели унифицированного семантического представления знаний. В рамках этой модели используются однородные семантические сети – семантические сети, в основе которых лежит базовая теоретико-множественная семантическая интерпретация, построенная на отношении (ситуативной) принадлежности (непринадлежности) элемента множеству (единственный базовый тип отношения) [25], связки которого обозначаются sc-дугами. Языки, входящие в модель унифицированного семантического представления знаний, называются sc-языками, тексты которых состоят из sc-элементов, а формируемые из них (ситуативные) множества, структуры и онтологические представления, онтологические модели называются соответственно sc-множествами, sc-структурами и sc-моделями. Семантика языков модели унифицированного семантического представления знаний соответствует модели ситуативных (событийных) множеств. Ситуативные множества представляют собой более гибкий и адекватный аппарат для представления знаний, чем классические множества, позволяя учитывать НЕ-факторы знаний и адаптироваться к представляемой ПрО, сохраняя онтологическую модель и её семантику. Это достигается не только за счёт учёта временных свойств, что позволяет интерпретировать систему ситуативных множеств как развитие систем L-нечётких множеств, но и за счёт того, что в рамках модели унифицированного семантического представления знаний процесс употребления аппарата ситуативных множеств способен иметь динамический характер, поддерживаемый динамикой алфавитных меток.

Поэтому представляется целесообразным употребление ситуативных множеств и их достоинств для представления ИНС средствами модели унифицированного семантического представления знаний. Семантика текстов sc-языков в рамках модели унифицированного семантического представления знаний является модельной, однако возможно её описание в денотационном виде, в этом случае можно говорить о денотационной семантике sc-языков на базе ситуативных множеств.

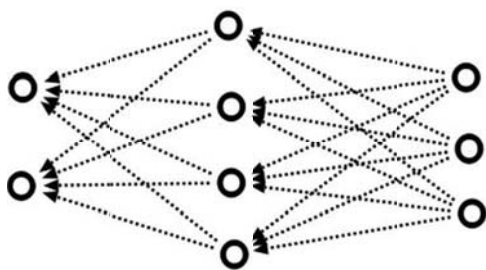


Рисунок 8 – Представление многослойной ИНС

Таким образом, при представлении ИНС можно трактовать каждую её вершину (узел), не являющуюся рецепторной, как обозначение ситуативного множества (sc-множества) всех вершин, из которых сигнал приходит в этот узел (рисунок 8). Каждая рецепторная вершина – это обозначение ситуативного множества признаков сущностей, которые она выражает.

ИНС представляется как ситуативное множество sc-элементов, обозначающих вершины, связи, их параметры, функции, свойства, связки и соответ-

ствия между ними (рисунок 9).

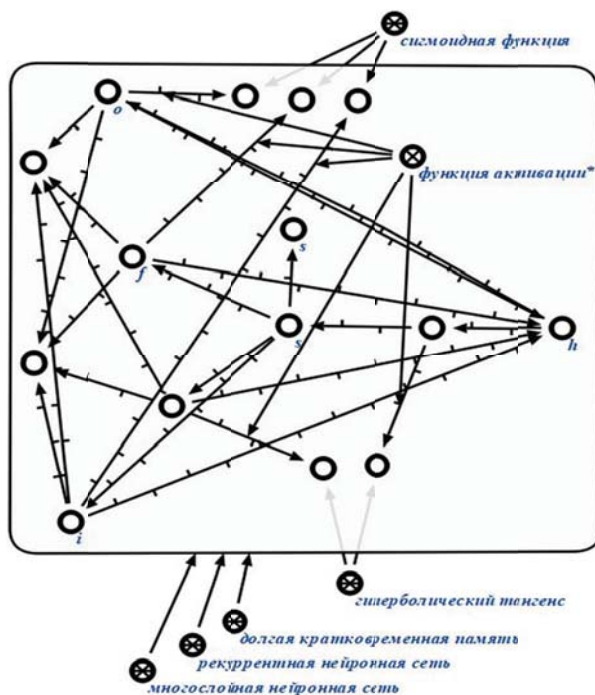


Рисунок 9 – Представление ИНС как ситуативное множество sc элементов

Значения весовых коэффициентов для синаптических связей и выходных сигналов нейронных элементов представляются как ситуативное значение некоторой величины, полученное измерением соответствующих элементов ИНС (рисунок 10).

Для указания функций синаптического преобразования и активации (рисунок 11), значений весовых коэффициентов (рисунок 10) и выходных сигналов нейронных элементов используются ключевые элементы: *функция синаптического преобразования**, *функция активации**, *синапс**, *нейронный узел**, обозначающие ситуативные отношения.

Между этими значениями могут быть установлены математические взаимосвязи, выражаемые с помощью математических операций и отношений, таких как сумма и произведение (рисунок 12). Эти математические отношения могут быть заданы в общем виде высказыванием или программой (рисунок 13).

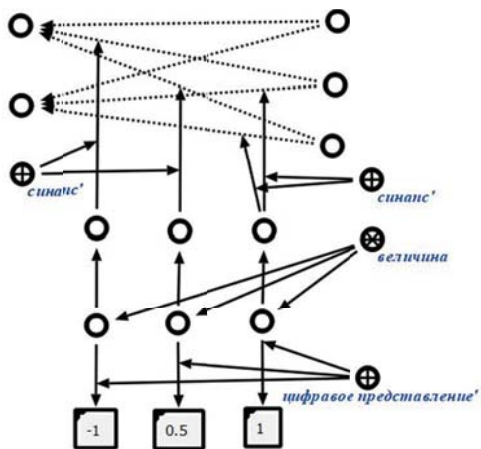


Рисунок 10 – Представление весовых коэффициентов синаптических связей ИНС

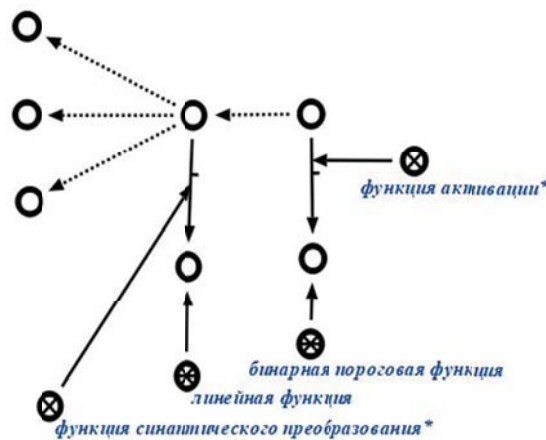


Рисунок 11 – Представление функций синаптического преобразования и активации узлов ИНС

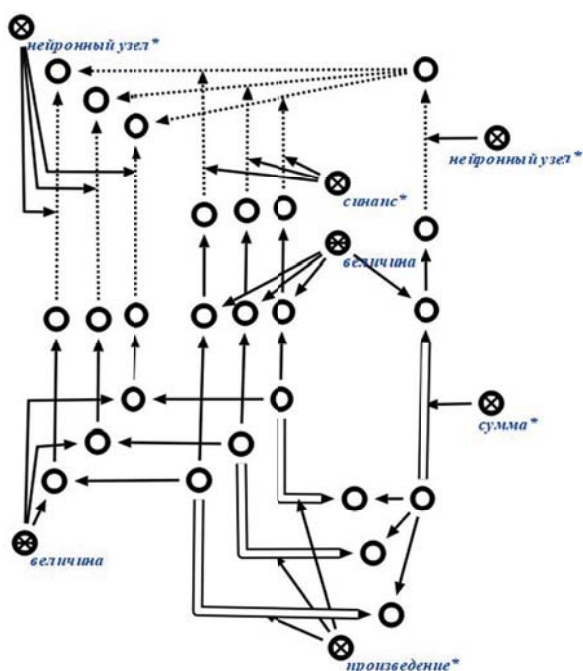


Рисунок 12 – Представление выходных значений элементов ИНС и их математических отношений

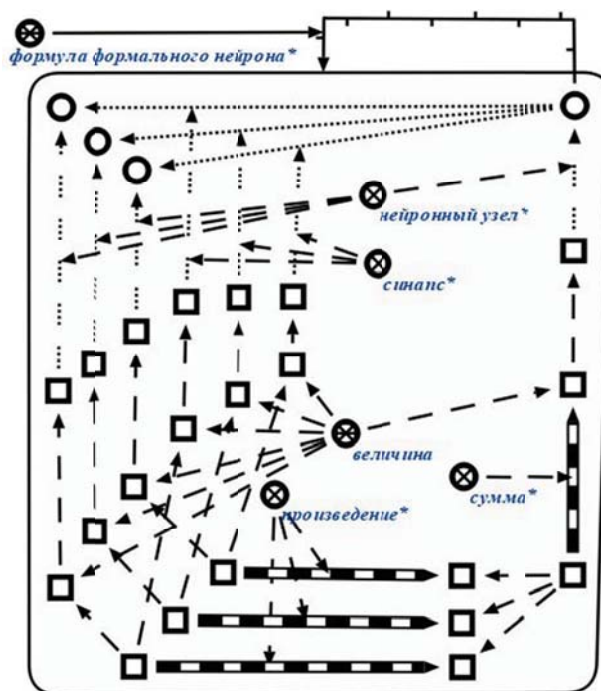


Рисунок 13 – Представление формального описания зависимости выходных значений элементов ИНС

Представление ИНС со сложной структурой позволяет представлять их отдельные части, такие как скрытые слои в многослойных нейронных сетях (рисунок 14), встречные связи в ИНС встречного распространения (рисунок 15), обратные связи в рекуррентных ИНС (рисунок 10) и т.п.

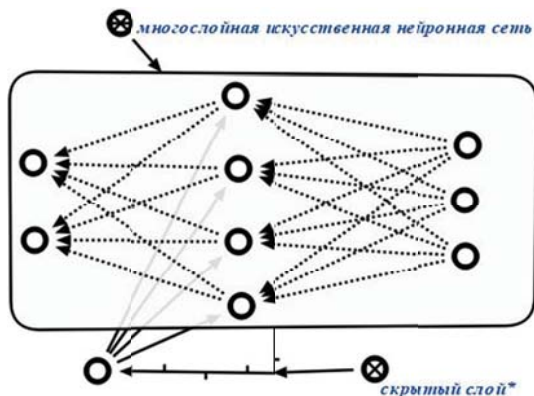


Рисунок 14 – Представление многослойной ИНС со скрытым слоем

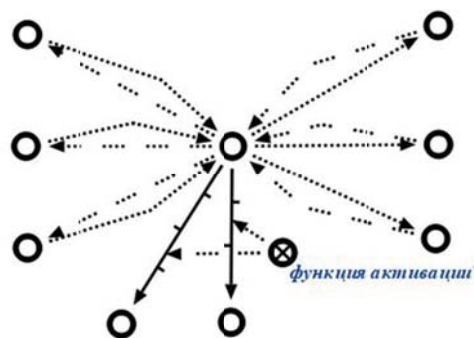


Рисунок 15 – Представление ИНС встречного распространения

Каждая ИНС задаёт соответствие между своими входами и выходами (рисунок 16, рисунок 17), описание этого факта представляется с помощью ключевых элементов *соответствие**, *область определения соответствия**, *область определения'*, *соответствие области определения'*, *область значений соответствия**, *область значений'*, *соответствие области значений'*. Для уточнения значений входных нейронных элементов ИНС используются ключевые элементы *домен'*, *домен**, *отношение'*.

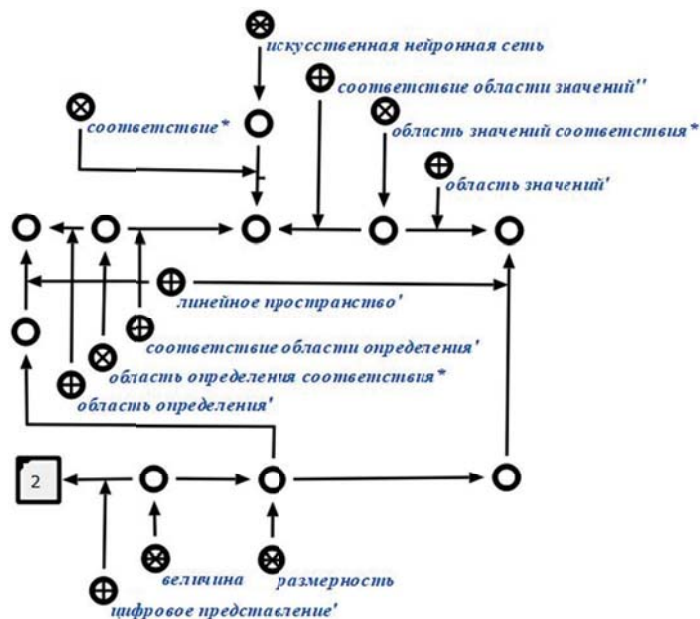


Рисунок 16 - Описание входов и выходов сети

Перечисленные ключевые элементы наравне с другими используются для спецификации признаков (рисунок 18), значения которых являются значениями входных и выходных нейронных элементов (рецепторов и эффекторов) ИНС.

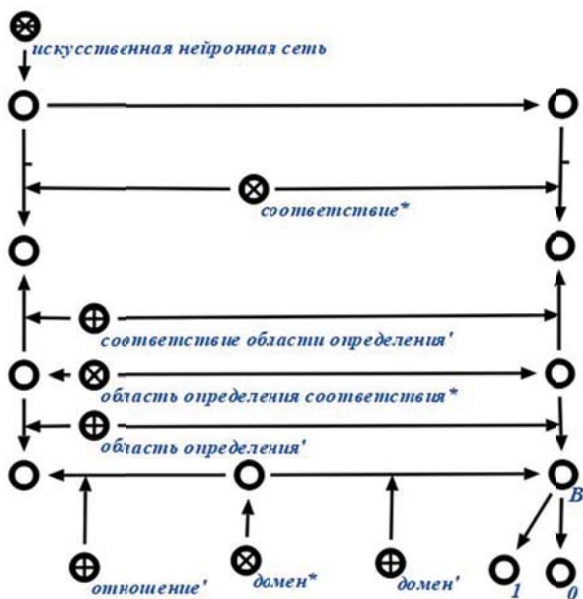


Рисунок 17 – Описание входного нейронного элемента сети



Рисунок 18 – Описание бинарного признака

5.2 Структура предметной области

ПрО – структура, которая задана на множестве sc-элементов.

На ПрО определены те же отношения, что и на sc-структурах, которые могут быть связаны отношениями включения (ситуативного подмножества), изоморфного вложения и т.д.

Эти отношения являются аналогом отношений модели спецификации знаний [27]. Следует отличать эти отношения от ситуативных отношений временного включения, гипотетического включения, временного изоморфного вложения и гипотетического изоморфного вложения, которые обуславливаются присутствием таких НЕ-факторов, как неполнота и неопределённость.

ПрО A является частной ПрО B , если A включается в B и её (максимальное) исследуемое множество включается в (максимальное) исследуемое множество B .

Важнейшим этапом разработки БЗ по технологии OSTIS является формирование иерархии ПрО (на основе понятия частной ПрО), определяющей структуру БЗ, и их спецификация [26]. Рассмотрение БЗ с позиции соотношения с ПрО позволяет рассматривать исследуемые объекты на различных уровнях детализации, которые отражаются в различных типах онтологий, описывающих определенное направление описания свойства объекта в рамках описываемой ПрО. К таким онтологиям относятся: структурная спецификация ПрО, логическая онтология, логическая система понятий и их определений, теоретико-множественная онтология и другие.

Рассмотрим классификацию ИНС, представленную в рамках теоретико-множественной онтологии соответствующей ПрО.

- Конечнзначные нейронные сети;
- Комплексно-численные нейронные сети:
 - Вещественно-численные нейронные сети:
 - Рационально-численные нейронные сети:
 - Целочисленные нейронные сети.
- Рациональные нейронные сети;
- Повышающие размерность нейронные сети;
- Понижающие размерность нейронные сети;
- Прерывные нейронные сети;
- Непрерывные нейронные сети;
- Дифференцируемые нейронные сети;
- Недифференцируемые нейронные сети;
- Гомогенные нейронные сети;
- Гетерогенные нейронные сети;
- Нейронные сети без обратных связей;
- Нейронные сети с обратными связями;
- Нейронные сети без скрытых нейронов;
- Нейронные сети со скрытыми нейронами;
- Многослойные нейронные сети;
- Однослойные нейронные сети;
- Детерминированные нейронные сети;
- Релаксационные нейронные сети.

Кроме указанных классов ИНС и соответствующих им ПрО можно выделить задачи, классы нейронных сетей и соответствующие ПрО, связанные с обучением ИНС.

Можно выделить два основных класса:

- Обучение ИНС на основе эталонных значений выходных элементов ИНС;
- Обучение ИНС без использования эталонных значений выходных элементов ИНС.

К первому классу можно отнести подклассы:

- Обучение ИНС на основе дельта-правила;
- Обучение ИНС на основе градиентных методов:
 - Обучение ИНС методом обратного распространения ошибки;

- Обучение ИНС методом сопряжённых градиентов;
- Обучение ИНС методом переменной метрики.

К частным классам второго класса относятся:

- Обучение ИНС без использования эталонных значений на основе правила Хебба [28];
- Обучение ИНС без использования эталонных значений на основе правила ВТА [29].

6 Агенты интеллектуальной среды обработки нейронных сетей

Операционная семантика БЗ, построенных в соответствии с моделью унифицированного семантического представления знаний, выражается в командах sc-машин обработки знаний [30]. Каждая sc-машина соответствует формальной модели обработки информации, языком которой является некоторый sc-язык. Также каждая sc-машина обладает начальной информационной конструкцией и множеством операций, которые она реализует, которые могут быть запрограммированы в виде процедур.

В соответствии с многоагентным подходом каждая sc-машина может быть реализована в виде совокупности агентов (sc-агентов). Операционная семантика ИНС в БЗ сводится к операционной семантике реализующих её sc-агентов, операционная семантика которых в свою очередь сводится к операционной семантике программ (команд) их операций.

Все операции sc-агентов выполняются асинхронно. Все sc-агенты взаимодействуют через общую память, передавая друг другу данные в виде конструкций семантической сети (текстов sc-языка, состоящих из sc-элементов). Условием запуска операции (инициирования sc-агента) является некоторое событие в общей (графовой) памяти. Такими событиями являются изменения временной непринадлежности на временную (актуальную) принадлежность элемента ситуативному множеству, которое трактуется как множество команд инициирования sc-агентов. Каждая команда представляет собой данные, которые будут обработаны агентом. Такими данными могут быть отдельный sc-элемент и его семантическая окрестность, доступная в общей памяти, или некоторая структура (sc-структура), обозначенная таким sc-элементом (рисунок 19).

После того как операция запущена и выполнена, временная принадлежность заменяется на временную (актуальную) непринадлежность. Однако, могут появиться новые временные (актуальные) принадлежности, которые будут инициировать работу других sc-агентов.

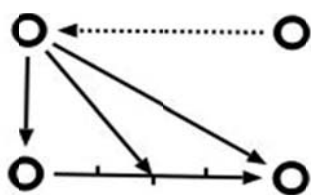


Рисунок 19 – Передача данных в виде структуры

Таким образом работа ИНС и процессы обработки знаний в БЗ сводятся к некоторому порядку изменений временной принадлежности на непринадлежность и наоборот. Агенты, которые обеспечивают семантическое протоколирование [31] работы ИНС формируют метазнание в виде структур специального вида, обеспечивающих возможность поиска и анализа порядка работы ИНС, сводящемуся к взаимодействию sc-агентов в общей графовой памяти.

Для реализации ИНС можно выделить агенты следующих классов.

- Агенты по интерпретации и обработке входов и выходов ИНС;
- Агенты по обучению ИНС:
 - Агенты управления обучением ИНС;
 - Агенты семантического протоколирования процессов обучения.
- Агенты по интеграции различных ИНС:
 - Агенты клонирования (агентов) ИНС;
 - Агенты поиска (агентов) рецепторов ИНС;

- Агенты поиска (агентов) эффекторов ИНС;
- Агенты синтеза (агентов) ИНС:
 - Агенты синтеза (агентов) слоя ИНС.
- Агенты синтеза агентов множества нейронов, синаптически связанного с множеством нейронов;
- Агенты синтеза агентов дуальной сети обратного распространения (ошибок);
- Агенты удаления (агентов) синаптических связей;
- Агенты удаления (агентов) ИНС;
- Агенты трансляции онтологического представления ИНС на языки программирования;
- Агенты семантического протоколирования интеграции ИНС.

Заключение

Рассмотренные направления применения интеграций ИНС с БЗ ориентированы на учёт семантики решаемых задач, что, с одной стороны, позволит сделать решение этих задач более структурированным и прозрачным для пользователя, с другой стороны, позволит вносить дополнительные корректировки в процесс обучения нейронной сети и решения задачи. Реализация описанной ИС, а также соответствующей интеллектуальной среды, позволит уменьшить порог вхождения разработчиков в область решения задач с помощью ИНС. Предоставляемые интеллектуальной средой возможности интроспекции ИНС и сохранения состояний ИНС во время обучения позволяют обеспечить более глубокий анализ их работы.

Список источников

- [1] *McCulloch, W.* A Logical Calculus of Ideas Immanent in Nervous Activity / W. McCulloch, W. Pitts // Bulletin of Mathematical Biophysics, 1943, p.115-133.
- [2] *Fukushima, K.* Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position / K. Fukushima // Biological Cybernetics, 1980, p.193-202.
- [3] *Bengio, Y.* Learning deep architectures for AI. Foundations and Trends in Machine Learning / Y. Bengio // Foundations and Trends® in Machine Learning, 2009, Vol. 2: No. 1, p.1-127
- [4] *Hinton, G.E.* A fast learning algorithm for deep belief nets / G.E. Hinton, S. Osindero, Y. Teh // Neural Computation, 2006, p.1527-1554.
- [5] *Glorot, X.* Deep sparse rectifier neural networks / X. Glorot, A. Bordes, Y. Bengio // 14th International Conference on Artificial Intelligence and Statistics, 2011, vol. 15, p.315-323.
- [6] *Golovko, V.A.* Learning Technique for Deep Belief Neural Networks / V.A. Golovko, A.A. Kroshchanka, U. Rubanau, S. Jankowski // Communication in Computer and Information Science, 2014. vol. 440. - p.136-146.
- [7] *Golovko, V.A.* Deep learning: an overview and main paradigms / V.A. Golovko // Optical Memory and Neural Networks (Information Optics), 2017, Vol. 26, Issue 1, - p.1-17.
- [8] *Golovko, V.* The nature of unsupervised learning in deep neural networks: A new understanding and novel approach / V. Golovko, A. Kroschanka // Optical memory and neural networks. 2016. No 3. - p.127-141.
- [9] *Rosenblatt F.* Principles of neurodynamics; perceptrons and the theory of brain mechanisms / F. Rosenblatt // In: Palm G., Aertsen A. (eds) Brain Theory. Springer, Berlin, Heidelberg. 1986. – DOI: https://doi.org/10.1007/978-3-642-70911-1_20.
- [10] *Поспелов, Д.А.* Искусственный интеллект: В 3 книгах. Книга 2. Модели и методы: Справочник, Москва, Радио и связь, 1990, - 304 с.
- [11] Information Integration for Concurrent Engineering (IICE), IDEF5 Method Report, Texas, USA, Knowledge Based Systems, Inc., 1994.
- [12] *Graves, A.* Hybrid computing using a neural network with dynamic external memory / A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S.G. Colmenarejo, E. Grefenstette, T. Ramalho // Nature, 2016, - p.471-476. - DOI:10.1038/nature20101.
- [13] *Голенков, В.В.* Проектирование предприятий рецептурного производства на основе онтологий / В.В. Таберко, Д.С. Иванюк, В.В. Голенков, К.В. Русецкий, Д.В. Шункевич, И.Т. Давыденко, В.В. Захаров,

- В.П. Ивашенко, Д.Н. Корончик // *Онтология проектирования*. – 2017. – Т. 7, №2(24). – С.123-144. – DOI: 10.18287/2223-9537-2017-7-2-123-144..
- [14] **Diederik P. Kingma**. Adam: A Method for Stochastic Optimization / Diederik P. Kingma, Jimmy Lei Ba. - arXiv:1412.6980 [cs.LG], December 2014.
- [15] **Klambauer G**. Self-normalizing neural networks / G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter // In *Advances in Neural Information Processing Systems (NIPS)*. arXiv:1706.02515, 7 Sep 2017.
- [16] **Donahue J**. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, vol. 39, no. 4, p.677-691.
- [17] **Mikolov T**. Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado, J. Dean // *ICLR conference*, 2013, arXiv preprint arXiv:1301.3781.
- [18] **Xiong S**. Deep Knowledge Representation based on Compositional Semantics for Chinese Geography / S. Xiong, X. Wang, P. Duan, Z. Yu, A. Dahou // In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, 2017, p.17-23.
- [19] **Pelevina M**. Making Sense of Word Embeddings / M. Pelevina, N. Arefyev, C. Biemann, A. Panchenko // 2017, arXiv preprint arXiv:1708.03390.
- [20] **Goldsborough P**. A Tour of TensorFlow / P. Goldsborough // *ICLR conference*, 2016, arXiv preprint arXiv:1610.01178.
- [21] Battle of the Deep Learning frameworks - Part I: 2017, even more frameworks and interfaces. - <https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750>.
- [22] **Liu H**. Hierarchical Representations for Efficient Architecture Search / H. Liu, K. Simonyan, O. Vinyals, C. Fernando, K. Kavukcuoglu // *ICLR conference*, 2017, arXiv preprint arXiv:1711.00436.
- [23] **Голенков, В.В.** Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания / В.В. Голенков, Н.А. Гулякина // *Онтология проектирования*, 2014, № 1(11), с.42–64.
- [24] **Голенков, В.В.** Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 2: Унифицированные модели проектирования / В.В. Голенков, Н.А. Гулякина // *Онтология проектирования*, 2014, № 4(14), с.34–53.
- [25] **Хорошевский, Б.Ф.** Базы знаний интеллектуальных систем / Б.Ф. Хорошевский, Т.А. Гаврилова // Санкт-Петербург, Питер, 2001, 384 с.
- [26] **Давыденко, И.Т.** Технология компонентного проектирования баз знаний на основе унифицированных семантических сетей / И.Т. Давыденко // *Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013)*, 2013. – с.185–190.
- [27] **Ивашенко, В.П.** Модели и алгоритмы интеграции знаний на основании однородных семантических сетей / В.П. Ивашенко // *Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015)*: 2015, с.111-132.
- [28] **Hebb, D.O.** *The Organization of Behavior* / D.O. Hebb // New York, Wiley & Sons, 1949, 383p
- [29] **Grossberg, S.** Contour enhancement, short-term memory, and constancies in reverberating neural networks / S. Grossberg // *Studies in Applied Mathematics*, 52:213, 1973
- [30] **Шункевич, Д.В.** Модели и принципы проектирования компонентов машины обработки знаний на базе семантических сетей / Д.В. Шункевич // *Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013)*: 2013, с.269-280
- [31] **Ивашенко, В.П.** Семантическое протоколирование процессов обработки знаний / В.П. Ивашенко // *Информационные технологии и системы 2017 (ИТС 2017) = Information Technologies and Systems 2017 (ITS 2017)* : материалы междунар. науч. конф. (Республика Беларусь, Минск, 25 октября 2017 года) / редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2017, с.110-111.
-

INTEGRATION OF ARTIFICIAL NEURAL NETWORKS AND KNOWLEDGE BASES

V.A. Golovko^{1,a}, V.V. Golenkov^{2,b}, V.P. Ivashenko^{2,c}, V.V. Taberko^{3,d}, D. S. Ivniuk^{3,e},
A.A. Kroshchanka^{1,f}, M. V. Kovalev^{2,g}

¹ Brest State Technical University, Brest, Belarus

^agva@bstu.by, ^fkroschenko@gmail.com

² Belarussian State University of Informatics and Radioelectronics, Minsk, Belarus

^bgolen@bsuir.by, ^civashenko@bsuir.by, ^gmichail.kovalev7@gmail.com

³ JSC «Savushkin product», Brest, Belarus

^dtab@pda.savushkin.by, ^eid@pda.savushkin.by

Abstract

This article reviews the questions and directions of integration of artificial neural networks with knowledge bases. Two main directions of integration are considered: the inputs and outputs of artificial neural network to use integration of knowledge bases and artificial neural networks for solutions of application problems; by artificial neural network representation on the basis of ontological structures and its interpretation by means of knowledge processing in the knowledge base providing an intelligent environment for the development, training and integration of different artificial neural networks compatible with knowledge bases. The knowledge bases that are integrated with artificial neural networks are built on the basis of homogeneous semantic networks and multiagent approach to represent and process knowledge. An ontological model for representing artificial neural networks and their specifications within the framework of the model of unified semantic representation of knowledge is proposed. It is distinguished by the ability to represent artificial neural networks, its dynamics and other types of knowledge, including the specifications of artificial neural networks, as the common language for the representation of knowledge with a common theoretical-model semantics. A multiagent model for solving problems using artificial neural networks and other types of knowledge is proposed. It is distinguished by the interaction of agents in accordance with a given temporal model through a common memory that stores knowledge integrated into a single knowledge base.

Key words: ANN, knowledge base, integration, frameworks.

Citation: Golovko VA, Golenkov VV, Ivashenko VP, Taberko VV, Ivaniuk, DS, Kroshchanka AA, Kovalev MV. Integration of artificial neural networks and knowledge bases [In Russian]. *Ontology of designing*. 2018; 8(3): 366-386. - DOI: 10.18287/2223-9537-2018-8-3-366-386.

References

- [1] McCulloch W, Pitts W. A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, 1943, p.115-133.
- [2] Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, 1980, p.193-202.
- [3] Bengio Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning, Foundations and Trends® in Machine Learning*, 2009; 2(1): 1-127.
- [4] Hinton GE, Osindero S, Teh Y. A fast learning algorithm for deep belief nets, *Neural Computation*, 2006, p.1527-1554.
- [5] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks, *14th International Conference on Artificial Intelligence and Statistics*, 2011; 1: 315-323.
- [6] Golovko VA, Kroshchanka AA, Rubanau U, Jankowski S. Learning Technique for Deep Belief Neural Networks, *Communication in Computer and Information Science*, 2014; 440: 136-146.
- [7] Golovko VA. Deep learning: an overview and main paradigms // *Optical Memory and Neural Networks (Information Optics)*, 2017; 26(1): 1-17.
- [8] Golovko V, Kroshchanka A. The nature of unsupervised learning in deep neural networks: A new understanding and novel approach // *Optical memory and neural networks*. 2016; 3: 127-141.
- [9] Rosenblatt F. Principles of neurodynamics; perceptrons and the theory of brain mechanisms. In: Palm G., Aertsen A. (eds) *Brain Theory*. Springer, Berlin, Heidelberg. 1986. – DOI: https://doi.org/10.1007/978-3-642-70911-1_20.

- [10] **Pospelov DA**. Искусственный интеллект: В 3 книгах. Книга 2. Модели и методы: Справочник (Artificial Intelligence: In 3 books. Book. 2. Models and methods: Handbook [In Russian]), Moscow, Radio i svyaz', 1990, - 304 p.
- [11] Information Integration for Concurrent Engineering (IICE), IDEF5 Method Report, Texas, USA, Knowledge Based Systems, Inc., 1994.
- [12] **Graves A, Wayne G, Reynolds M, Harley T, Danihelka I, Grabska-Barwińska A, Colmenarejo SG, Grefenstette E, Ramalho T**. Hybrid computing using a neural network with dynamic external memory, *Nature*, 2016, p.471-476. - DOI:10.1038/nature20101.
- [13] **Golenkov VV, Taberko VV, Ivanyuk DS, Rusetski KV, Shunkevich DV, Davydenko IT, Zakharov VV, Ivashenko VP, Koronchik DN**. Designing Batch Manufacturing Enterprises Using Ontologies [In Russian]. *Ontology of designing*. 2017; 24(2): 123-144. – DOI: 10.18287/2223-9537-2017-7-2-123-144.
- [14] **Diederik P. Kingma, Jimmy Lei Ba**. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG], December 2014.
- [15] **Klambauer G, Unterthiner T, Mayr A, Hochreiter S**. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. arXiv:1706.02515, 7 Sep 2017.
- [16] **Donahue J**. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017; 39(4): 677-691.
- [17] **Mikolov T, Chen K, Corrado G, Dean J**. Efficient estimation of word representations in vector space, 2013, ICLR conference, arXiv preprint arXiv:1301.3781.
- [18] **Xiong S, Wang X, Duan P, Yu Z. and Dahou A**. Deep Knowledge Representation based on Compositional Semantics for Chinese Geography , In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, 2017*, p.17-23.
- [19] **Pelevina M, Arefyev N, Biemann C, Panchenko A**. Making Sense of Word Embeddings, 2017, arXiv preprint arXiv:1708.03390.
- [20] **Goldsborough P**. A Tour of TensorFlow, ICLR conference, 2016, arXiv preprint arXiv:1610.01178.
- [21] Battle of the Deep Learning frameworks - Part I: 2017, even more frameworks and interfaces. - <https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750>.
- [22] **Liu H, Simonyan K, Vinyals O, Fernando C, Kavukcuoglu K**. Hierarchical Representations for Efficient Architecture Search, ICLR conference, 2017, arXiv preprint arXiv:1711.00436.
- [23] **Golenkov VV, Guliakina NA**. Project of Open Semantic Technology of the Componental Design of Intelligent Systems. Part 1: The Principles of Creation [In Russian]. *Ontology of designing*, 2014; 1 (11): 42-64.
- [24] **Golenkov VV, Guliakina NA**. Project of Open Semantic Technology of the Componental Design of Intelligent Systems. Part 2: Unified Model Design [In Russian]. *Ontology of designing*, 2014; 4(14): 34-53.
- [25] **Horoshevskij BF, Gavrilova TA**. Bazy znaniy intellektual'nyh system (Intelligent Systems Knowledge Base) [In Russian], St.Petersburg, Piter, 2001, 384 p.
- [26] **Davydenko IT**. Tekhnologiya komponentnogo proektirovaniya baz znaniy na osnove unificirovannyh semanticheskikh setej (The technology of component design of knowledge bases on the basis of unified semantic networks) [In Russian], *Open Semantic Technologies for Intelligent Systems*, 2013, p.185-190.
- [27] **Ivashenko VP**. Modeli i algoritmy integracii znaniy na osnove odnorodnyh semanticheskikh setej (Models and algorithms of knowledge integration based on homogeneous semantic networks) [In Russian], *Open Semantic Technologies for Intelligent Systems*, 2015, p.111-132.
- [28] **Hebb DO**. The Organization of Behavior, New York, Wiley & Sons, 1949, 383 p.
- [29] **Grossberg S**. Contour enhancement, short-term memory, and constancies in reverberating neural networks, *Studies in Applied Mathematics*, 52:213, 1973.
- [30] **Shunkevich DV**. Models and Means of Component Design of Knowledge-Processing Machines on the Basis of Semantic Networks [In Russian]. *Open semantic technologies for designing intelligent systems*, 2013, p.269-280.
- [31] **Ivashenko, VP**. Semanticheskoe protokolirovanie processov obrabotki znaniy (Semantic logging of knowledge processing processes) [In Russian], *Information Technologies and Systems 2017 (ITS 2017) = Information Technologies and Systems 2017 (ITS 2017): materials of the international. sci. Conf. (Republic of Belarus, Minsk, 25 October 2017) / Rare. : L. Yu. Shilin [and others]. - Minsk: BSUIR, 2017, p.110-111.*
-

Сведения об авторах



Головки Владимир Адамович 1960 г. рождения. В 1984 году окончил Московское высшее техническое училище им. Н.Э. Баумана по специальности «Автоматические информационные устройства». В 1986–1989 обучался в аспирантуре Института технической кибернетики АН БССР (г. Минск). Д.т.н. (2003). Профессор (2004). Создал научную школу «Нейроинтеллектуальные технологии обработки информации». Главный редактор журнала «Вестник Брестского технического университета» по серии «Физика, математика и информатика», член редколлегии международного журнала «Компьютинг», русского журнала «Нейроинформатика», американского журнала «Recent Patents on Signal Processing». Организатор международной научной конференции «Нейронные сети и искусственный интеллект» (ICNNAI). В списке научных трудов более 100 работ в области

нейросетевых технологий.

Vladimir Adamovich Golovko (b. 1960). In 1984 he graduated from the Moscow Higher Technical School Named after Bauman on a specialty "Automatic information devices". In 1986-1989 he studied in graduate school of the Institute of Technical Cybernetics of the Academy of Sciences of the BSSR (Minsk), specializing in "Management in technical systems". D.t.sc. (2003). Professor (2004). He created a scientific school "Neuro Intelligence Information Processing Technologies". He is the chief editor of the journal "Bulletin of the Brest Technical University" in the series "Physics, Mathematics and Computer Science", a member of the editorial board of the international journal "Computing", the Russian journal Neuroinformatics, the American journal Recent Patents on Signal Processing. Organizer of the international scientific conference "Neural Networks and Artificial Intelligence" (ICNNAI). In the list of scientific works more than 100 works in the field of neural network technologies.

Голенков Владимир Васильевич, 1949 г. рождения. В 1971 г. с отличием окончил физический факультет Белорусского государственного университета, д.т.н. (1996), профессор, заведующий кафедрой интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, инициатор ежегодной международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем», член Российской ассоциации искусственного интеллекта, член редколлегии журналов «Программные продукты и системы», «Онтология проектирования», «Электроника-ИНФО», «Речевые технологии». В списке научных трудов более 100 работ в области семантических технологий.



Vladimir Vasilievich Golenkov (b. 1949). In 1971 he graduated from the physics department of Belarusian State University with honors diploma, Dr. of science (1996), professor. The head of the department of Intelligent information technologies of Belarusian State University of Informatics and Radioelectronics, initiator of the annual international science and technical conference «Open Semantic technologies for Intelligent Systems» (OSTIS), member of Russian association of artificial intelligence, editorial board member of such journals as «Program products and systems», «Ontology of designing», «Electronics-INFO», «Speech technologies». The list of his published works consists of more than 100 articles on the various aspects of semantic approach in the intelligent systems design.



Иващенко Валерьян Петрович, 1978 г. рождения. Окончил с отличием Белорусский государственный университет информатики и радиоэлектроники по специальности «Искусственный интеллект» в 2000 г., к.т.н. (2015). Доцент кафедры интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники. Имеет более 50 печатных работ в области семантических технологий, информатики и ИИ.

Valerian Petrovich Ivashenko (b. 1978) graduated from the Belarusian State University of Informatics and Radioelectronics in 2000 (Artificial Intelligence), PhD (2015). He is Assistant professor at Belarusian State University of Informatics and Radioelectronics (Department of Intellectual Information Technologies). He is co-author over 50 published works in the field of semantic technologies, CS and AI.

Таберко Валерий Васильевич, 1963 г. рождения. В 1985 г. окончил факультет прикладной математики Белорусского государственного университета. С 1985 г. до 1993 г. работал на Брестском электромеханическом заводе, в 1993 г. перешел на предприятие «Брестский молочный комбинат» начальником информационно-вычислительного центра, в настоящее время зам. Ген. директора по логистике на ОАО «Савушкин продукт».

Valery Vasilievich Taberko (b. 1963). In 1985 he graduated from the faculty of Applied Mathematics of Belarusian State University. Since 1985 till 1993 worked at Brest electromechanical plant, in 1993 moved to a new job at Brest milk plant as Head of Information and Computing Center, now works as Deputy Director for Logistics at JSC «Savushkin product».





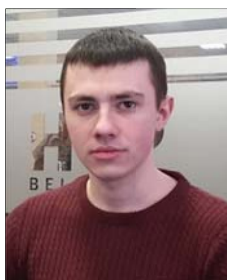
Иванюк Дмитрий Сергеевич, 1982 г. рождения. В 2005 г. с отличием окончил факультет электронно-информационных систем Брестского государственного технического университета, исследователь технических наук, ведущий инженер-программист отдела программирования ОАО «Савушкин продукт», старший преподаватель кафедры интеллектуальных информационных технологий Брестского государственного технического университета. Имеет 10 работ в области нейронных сетей и семантических технологий.

Dzmitry Sergeevich Ivanyuk (b. 1982). In 2005 he graduated from the intelligent information technologies department of Brest State Technical University with honors diploma, Researcher of technical sciences. Lead software development engineer at JSC «Savushkin product», senior lecturer at department of Intelligent Information Technologies in Brest State Technical University. He has 10 published works on the subject of semantic technologies and artificial neural

networks.

Крощенко Александр Александрович. 1985 г. рождения. Окончил математический факультет Брестского государственного университета имени А.С. Пушкина по специальности «Прикладная математика (научно-производственная деятельность)». Магистр физико-математических наук (2009). Работал преподавателем кафедры информатики и прикладной математики с 2009 года. Обучался в аспирантуре Брестского государственного технического университета с 2013 по 2016 гг. С 2016 г. – преподаватель кафедры прикладной математики и информатики. Имеет свыше 30 публикаций.

Alexander Alexandrovich Kroshchenko (b. 1985). Graduated from the Faculty of Mathematics of the Brest State University named after A.S. Pushkin specialty "Applied mathematics (research and production activities)." Master of Physical and Mathematical Sciences (2009). Worked as a teacher of the Department of Informatics and Applied Mathematics since 2009. Studied at the post-graduate course of the Brest State Technical University from 2013 to 2016. (the supervisor of studies is VA Golovko). Since 2016 - lecturer of the Department of Applied Mathematics and Informatics. He has over 30 publications.



Ковалёв Михаил Владимирович. 1995 г. рождения. Окончил Белорусский государственный университет информатики и радиоэлектроники по специальности «Искусственный интеллект» в 2017 г., магистрант кафедры интеллектуальных информационных технологий БГУИР с 2017 г. Имеет 4 печатные работы в области нейронных сетей и семантических технологий.

Mikhail Vladimirovich Kovalev (b. 1995). Graduated from the Belarusian State University of Informatics and Radioelectronics, specializing in Artificial Intelligence in 2017, master student of the Department of Intelligent Information Technologies of the Belarusian State University of Information Technologies (BSUIR) since 2017. He has 4 published works on the subject of semantic technologies and artificial neural networks.

УДК 519.711.3

ПРОЕКТИРОВАНИЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЗДАНИЕМ С ИСПОЛЬЗОВАНИЕМ ОНТОЛОГИЙ¹

О.Ю. Марьясин

Ярославский государственный технический университет, Ярославль, Россия
maryasin2003@mail.ru

Аннотация

В статье рассматриваются вопросы проектирования системы управления зданием. Одной из проблем, возникающих при проектировании, является проблема организации слаженной работы традиционных и новейших систем управления инженерным оборудованием. Другой проблемой является выявление предпочтений и учёт деятельности людей. Для решения указанных проблем автор предлагает использовать мультиагентный подход. На основе анализа опубликованных архитектур мультиагентных систем управления зданием проведена классификация агентов, в результате которой сформирован перечень агентов и определены классы агентов и их характеристики. Показан пример построения архитектуры мультиагентной системы управления зданием и приведено описание функций отдельных агентов. Произведена проверка проектных решений путём компьютерного моделирования мультиагентной системы в среде AnyLogic. Впервые предложены классы агентов, являющиеся основой для построения архитектуры мультиагентной системы управления зданием.

Ключевые слова: микроклимат, энергосбережение, мультиагентная система, AnyLogic.

Цитирование: Марьясин, О.Ю. Проектирование мультиагентной системы управления зданием с использованием онтологий / О.Ю. Марьясин // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.387-399. – DOI: 10.18287/2223-9537-2018-8-3-387-399.

Введение

Современные городские жители более 80% своей жизни проводят в закрытых помещениях. Микроклимат помещений существенно влияет на физическое и эмоциональное состояние людей, а также на их работоспособность [1]. Поэтому поддержание параметров микроклимата, определённых известными санитарными нормами и комфортных для проживания людей, является главной задачей климатического оборудования, установленного в зданиях. С другой стороны, необходимость энергосбережения и снижения коммунальных платежей требует минимизации затрат энергоресурсов на поддержание микроклимата. Обеспечить выполнение этих часто противоречивых требований должны комплексные системы управления микроклиматом и энергопотреблением (Energy and Comfort Management System – ECMS) зданий, включающие оборудование систем отопления, вентиляции и кондиционирования (ОВК или HVAC - акроним от англ. Heating, Ventilation, & Air Conditioning - так их называют за рубежом) и, в свою очередь, являющиеся частью автоматизированных систем управления зданием (АСУЗ или Building Management System – BMS).

АСУЗ, кроме ОВК системы и системы энергоснабжения (к которой обычно относят и систему освещения), может включать также и другие инженерные системы, например систему безопасности (охранно-пожарной сигнализации, контроля доступа и видеонаблюдения). Од-

¹ Статья в расширенном варианте печатается по рекомендации Программного комитета XX международной научной конференции «Проблемы управления и моделирования в сложных системах» (ПУМСС-2018). 3-6 сентября 2018, Самара, Россия

нако в данной работе АСУЗ ограничивается только функциями по управлению микроклиматом и энергопотреблением.

Современные тенденции снижения стоимости потребительской электроники и микропроцессорной техники и повышения уровня автоматизации и интеллектуализации инженерных систем зданий дают возможность совершенствования систем управления инженерным оборудованием зданий. В частности, вместо традиционных пропорционально-интегрально-дифференцирующих (ПИД) регуляторов с изменением задания по температурным графикам, для управления тепловым режимом помещений могут использоваться более сложные алгоритмы автоматического и оптимального управления. Известны работы, в которых для управления микроклиматом и энергопотреблением зданий предлагаются системы адаптивного и оптимального регулирования, системы на базе нечёткой логики и с использованием нейросетевых и генетических алгоритмов. Обзор таких работ можно найти, например в [2].

Вместе с тем, традиционные, хорошо зарекомендовавшие себя системы, работающие по принципу включено/выключено, а также различные системы регулирования на базе двухпозиционных, трёхпозиционных и ПИД-регуляторов продолжают и будут эксплуатироваться ещё долгое время. Поэтому возникает проблема организации слаженной работы разнородных систем управления инженерным оборудованием здания, без решения которой АСУЗ нельзя будет достигнуть требуемых показателей комфорта и энергоэффективности.

В большинстве работ, посвящённых управлению микроклиматом и энергопотреблением, люди рассматриваются как один из внешних факторов, то есть человеку там отводится лишь пассивная роль. Однако люди могут активно влиять на энергопотребление и микроклимат помещений, включать и выключать электроприборы и оборудование, изменять режимы и даже противодействовать работе автоматических систем. Определение наличия людей в помещениях, выявление их предпочтений и правильная реакция на их действия - вот задачи, которые необходимо решить АСУЗ.

Таким образом, с одной стороны, системы управления инженерным оборудованием зданий становятся всё более интеллектуальными. Они могут длительное время функционировать в автоматическом режиме. С другой стороны, человек, действуя на основе своих предпочтений, может оказывать активное влияние на работу инженерных систем.

Кроме того, при проектировании АСУЗ должен учитываться принцип эргономичности, согласно которому такие характеристики среды обитания человека как температурный режим, состав воздуха, свет, звук, информация и другие параметры, на основе которых формируется архитектура и дизайн объекта, должны подчиняться законам эргономики [3]. Это отвечает концепции «умного здания», которое представляет сложную интегрированную экологическую систему, повышает комфорт жилой зоны и экономит затраты и энергоресурсы.

1 Анализ архитектур мультиагентных систем управления зданием

Путём к решению названных проблем является рассмотрение АСУЗ как мультиагентной системы (МАС), в которой существует множество агентов, имеющих индивидуальное поведение и характеристики [4]. Человека в данном случае также можно рассматривать как одного из агентов. Автономность агентов, их целеустремленность, возможность наличия интеллекта и способность работать в распределённой среде делают МАС удобным и гибким инструментом для реализации АСУЗ. Развёрнутые обзоры по МАС можно найти в работах [5, 6]. Применению мультиагентного подхода для управления микроклиматом и энергосбережением зданий посвящены работы [7-9]. В статье [10] рассматривается возможность реализации мультиагентного подхода на базе известных инструментов моделирования МАС.

Можно отметить отсутствие единого взгляда на архитектуру МАС управления микроклиматом и энергопотреблением зданий. Так, в работе [9] МАС включает персональных агентов, локальных агентов, центральных агентов и агентов контроля и управления. Персональные агенты связаны с пользователями и предоставляют такую информацию о пользователях, как их предпочтения (предпочтительная температура и влажность в помещении) и местоположение. Каждой зоне здания назначается свой локальный агент. Локальные агенты действуют как посредники и поставщики информации для центральных агентов. Для каждого параметра микроклимата назначается агент контроля и управления. Задачей центрального агента является управление всей системой и координация инженерных систем здания.

Подобная система, основанная на персональных агентах, локальных агентах и центральных агентах, рассматривалась и в работе [11]. Здесь персональные агенты способны прогнозировать предпочтения пользователей на основе изучения их поведения. Локальные агенты решают задачи управления отдельными инженерными системами, а центральные агенты обеспечивают координацию между локальными агентами. В работе [12] используются только центральные и локальные агенты. Центральные агенты являются интеллектуальными и вычисляют задания по температуре, освещённости и влажности на основе предпочтений пользователя и метеоданных, используя метод роя частиц. Локальные агенты используют нечёткое регулирование для поддержания заданных значений параметров микроклимата. В работе [13] рассмотрена МАС, включающая персональных агентов комфорта, агентов отдельных помещений, агентов параметров микроклимата и агентов, отслеживающих местоположение людей в здании.

В работе [14] представлена МАС, включающая агентов-поставщиков и агентов-потребителей. Агенты-потребители определяют требования к параметрам микроклимата помещений и передают их агентам-поставщикам. Агенты-поставщики оценивают расходы, необходимые для обеспечения требований агентов-потребителей. Управление системой осуществляется путём переговоров между этими агентами. В работе [15] дополнительно к агентам-поставщикам, которые производят энергию, и агентам-потребителям, которые потребляют необходимое для поддержания комфорта количество энергии, вводятся агенты-дистрибьютеры, занимающиеся транспортировкой и распределением энергии, и агенты-среды, предоставляющие необходимую информацию о параметрах микроклимата.

МАС, предложенная в работе [16], содержит различные типы агентов: агент-помещение (агент-потребитель), агент-поставщик, агент-посредник и агент-брокер. Агенты-помещений являются потребителями тепла или холода. Их задача - оценить потребность в тепле или холоде и сделать соответствующий запрос агенту-брокеру. В агентах-помещениях используются ПИД-регуляторы, и на основе этого определяется мощность, необходимая для поддержания заданных параметров микроклимата. Запросы мощности могут быть сделаны от каждого отдельного агента-помещения, что позволяет пользователям корректировать параметры микроклимата в соответствии с их потребностями. Агенты-поставщики представляют собой объекты в системе, которые поставляют тепло или холод, например, котлы, тепловые насосы. Их задача – продать брокеру тепло или холод и предоставить необходимую мощность при успешной сделке. Агенты-посредники могут одновременно выступать и как агенты-поставщики, и как агенты-потребители. С помощью данных агентов в МАС можно строить каскадные структуры.

2 Онтологический анализ агентов

Онтологический подход к проектированию предполагает проведение классификации объектов предметной области. Это позволяет представить предметную область в виде сети

связанных объектов [17]. В случае МАС основным объектом предметной области является агент. Поэтому проведение анализа агентов и определение классов агентов является первоочередной задачей при проектировании МАС.

На основе проведённого анализа был составлен следующий перечень агентов МАС управления микроклиматом и энергопотреблением зданий:

- персональный агент (агент, связанный с человеком);
- локальный агент;
- агент-помещение;
- зональный агент;
- центральный агент;
- агент-оборудование;
- агент контроля и управления;
- агент-поставщик;
- агент-брокер;
- агент-дистрибьютер;
- агент внутренней среды (внутренний агент);
- агент внешней среды (внешний агент).

Персональные агенты связаны с пользователями и должны предоставлять информацию о предпочтениях пользователя и о его местоположении в здании. Локальный агент выполняет локальную задачу по управлению микроклиматом и энергосбережением в здании. Он может отвечать за поддержание микроклимата в отдельном помещении. В этом случае его можно считать агентом-помещений. Локальный агент может также выполнять функции по регулированию отдельных параметров микроклимата. В этом случае он может выступать как агент-оборудование или агент контроля и управления.

Агенты-помещений связаны с отдельными помещениями зданий. В отличие от них, зональные агенты предназначены для управления группой помещений или целой зоной здания. Это может быть секция или этаж здания. Центральные агенты предназначены для управления всем зданием. Они могут выполнять функции координации локальных или зональных агентов.

Агент-оборудование управляет конкретным инженерным оборудованием. Например, приточно-вытяжным агрегатом системы вентиляции, бойлером, солнечным коллектором и т.д. Агент-оборудование может использоваться для управления сложным климатическим оборудованием. Для простого оборудования его функции могут выполнять агенты контроля и управления. Агенты контроля и управления предназначены для управления параметрами внутренней среды с помощью законов управления. Это может быть двухпозиционное, трёхпозиционное или ПИД-регулирование.

Агенты-поставщики, как правило, выступают в роли поставщиков энергоресурсов для агентов-помещений, локальных, зональных или центральных агентов. Если в системе используется сложный протокол согласования цен за ресурсы или если необходимо организовать между локальными или зональными агентами рыночные отношения, то в такую систему вводятся агенты-брокеры. Агенты-поставщики могут быть как внешними, так и внутренними. В последнем случае в качестве агентов-поставщиков могут выступать агенты-оборудования. Агенты-дистрибьютеры могут потребоваться там, где возникают сложности с транспортировкой и распределением энергоресурсов.

Агенты внутренней среды, как правило, связаны с внутренними параметрами микроклимата здания, а агенты внешней среды - с внешними метеорологическими параметрами. Введение этих агентов может потребоваться, например, чтобы учесть влияние особых климатических условий или микроклимат специальных помещений.

С учётом возможности замещения функций отдельными агентами был предложен следующий перечень агентов MAC:

- агент, связанный с человеком (HumanAgent);
- локальный агент (LocalAgent);
- зональный агент (ZonalAgent);
- центральный агент (ZentralAgent);
- внешний агент-поставщик (ProviderAgent);
- агент внутренней среды (IndoorAgent);
- агент внешней среды (OutdoorAgent).

Диаграмма классов агентов, соответствующая данному перечню, показана на рисунке 1.

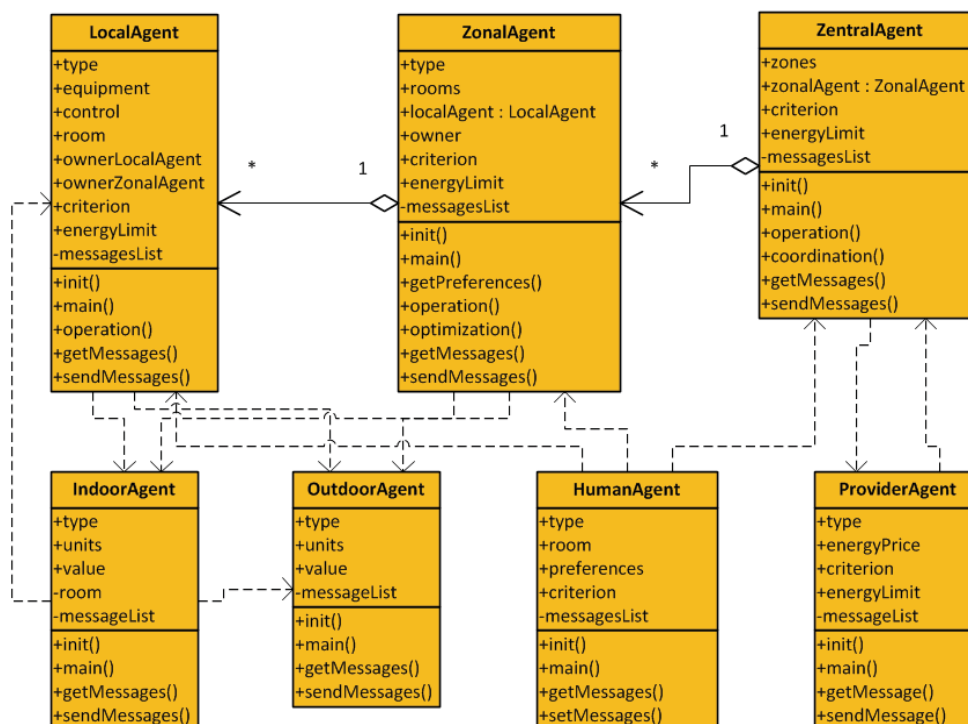


Рисунок 1 – Диаграмма классов агентов MAC

Ключевым элементом MAC является локальный агент. Локальный агент может быть связан с конкретным оборудованием, например, термостатом и локальной системой регулирования, реализующей, например, ПИД-закон регулирования температурой. Локальный агент всегда привязан к конкретному помещению. Кроме того, локальный агент привязан к конкретным видам энергоресурсов и по каждому из них имеет определённый лимит. Владелец локального агента может быть другим локальным или зональным агентом.

Для выполнения своих операций локальный агент может получать информацию (сообщения) от внутреннего и внешнего агентов. Локальный агент также может получать команды (сообщения) от агента-человека, от зонального агента и от другого локального агента. У каждого агента есть своя функция полезности, которую агент стремится максимизировать. Например, в случае термостата целью локального агента является поддержание температуры в помещении на заданном уровне. При достижении цели функция полезности локального агента будет максимальной.

Зональные агенты также привязаны к одному или нескольким помещениям. Особенностью зонального агента является наличие процедур выявления предпочтений человека и оптимизации микроклимата и энергосбережения. Зональный агент может получать информа-

цию от локальных агентов, от внутреннего и внешнего агентов, от агентов-людей, от агентов-поставщиков, от других зональных агентов. Он также может получать команды от людей и от своего центрального агента. Особенностью центрального агента является наличие процедуры координации зональных агентов, а также функций по взаимодействию с агентами поставщиками. Центральный агент может получать информацию от зональных агентов, от агентов-людей, от агентов-поставщиков, от других центральных агентов.

Приведённый перечень агентов не является окончательным. При расширении системы, например при добавлении в систему управления зданием системы безопасности, приведённый список необходимо дополнить агентом-безопасности.

3 Предлагаемая архитектура мультиагентной системы управления зданием

Проведённый анализ агентов может служить основой для построения структуры МАС управления микроклиматом и энергосбережением зданий. Конкретная структура получается путём связывания классов агентов с конкретным зданием, инженерным оборудованием здания и другими данными задачи проектирования АСУЗ, полученными, например, из информационной модели здания (ИМЗ или Building Information Modeling – BIM). ИМЗ даёт возможность использовать в процессе проектирования всю архитектурно-конструкторскую, технологическую, экономическую и иную информацию о здании со всеми её взаимосвязями и зависимостями, когда здание и всё, что имеет к нему отношение, рассматриваются как единый объект [18]. В результате определяются экземпляры агентов, их типы и число, производится детализация связей между ними. Один из вариантов полученной архитектуры МАС приведён на рисунке 2. Данная архитектура подходит для большинства жилых, общественных и административных зданий в России.

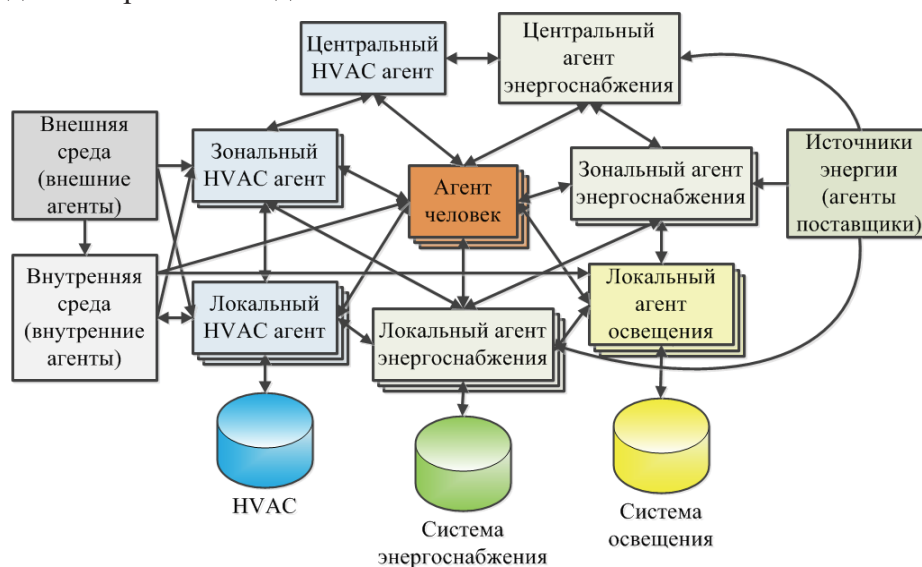


Рисунок 2 – Архитектура МАС управления зданием

На рисунке 2 управление ОВК оборудованием, электроприборами и приборами освещения осуществляется локальными ОВК-агентами, локальными агентами энергоснабжения и локальными агентами-освещения. Локальные агенты представляют собой традиционное инженерное оборудование (термостаты, вентиляторы, приборы освещения и другие) и локальные системы управления данным оборудованием, реализующие несложные сценарии или законы управления. Пример диаграммы состояния локального ОВК-агента типа термостат приведён на рисунке 3.

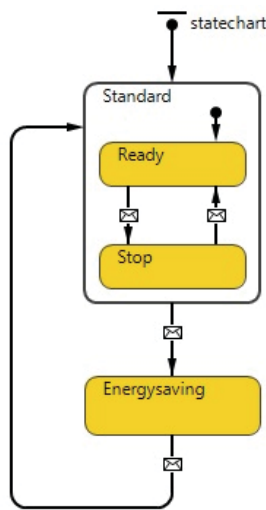


Рисунок 3 – Диаграмма состояний локального ОВК-агента

Локальный ОВК-агент после включения переходит в стандартный режим работы (Standard) и в состояние готовности (Ready), где он выполняет свою основную функцию. Для термостата - это функция поддержания температуры в помещении на заданном уровне, которую он может реализовывать, используя, например, ПИД-закон регулирования температурой. По командам человека или зонального ОВК-агента локальный ОВК-агент может быть выключен (перейти в состояние Stop) или переведён в энергосберегающий режим (перейти в состояние Energysaving).

Локальные ОВК-агенты и локальные агенты-освещения получают информацию о параметрах внешней и внутренней среды от соответствующих агентов, а локальные агенты энергоснабжения - о текущей потребляемой и максимально допустимой мощности энергоносителей. Команды управления локальные ОВК-агенты и локальные агенты освещения могут получать от зональных ОВК-агентов, локальных агентов энергоснабжения и непосредственно от агентов-людей.

Зональные ОВК-агенты и зональные агенты-энергоснабжения представляют собой интеллектуальные системы управления. Они получают информацию о параметрах внешней и внутренней среды от соответствующих агентов, информацию от локальных агентов, информацию о предпочтениях людей и их наличии в помещениях. На основе полученной информации они формируют сигналы-задания для локальных агентов с использованием методов оптимизации, оптимального или интеллектуального управления.

Если зональные ОВК-агенты решают задачи оптимального или оптимального прогнозирующего управления микроклиматом [19], то задача, решаемая i -ым зональным ОВК-агентом, может быть сформулирована следующим образом. Необходимо обеспечить минимум критерия оптимальности

$$(1) \quad J_i(t) = J_i(x_i(t), q_{ik}(t)) \rightarrow \min_{q_{ik}},$$

при условиях

$$(2) \quad \dot{x}_i(t) = f_i(x_i(t), q_i(t), w_i(t)),$$

$$(3) \quad x_{imin}(t) \leq x_i(t) \leq x_{imax}(t),$$

$$(4) \quad q_i(t) = \sum_{k=1}^{K_i} d_k q_{ik}(t),$$

$$(5) \quad q_{ikmin}(t) \leq q_{ik}(t) \leq q_{ikmax}(t), \quad k = 1, 2, \dots, K_i,$$

где t – время, $q_{ik}(t)$ – расход k -го вида энергоресурсов i -ой зоны, K_i – число видов энергоресурсов i -ой зоны, $x_i(t)$ – вектор параметров микроклимата, $w_i(t)$ – вектор возмущающих параметров, $x_{imin}(t)$, $x_{imax}(t)$ – минимальное и максимальное значения параметров микроклимата i -ой зоны, $q_i(t)$ – тепловая энергия, полученная i -ой зоной, d_k – масштабные коэффициенты, $q_{ikmin}(t)$, $q_{ikmax}(t)$ – минимальное и максимальное количество k -го вида энергоресурса на энергосбережение i -ой зоны.

Выражение для критерия оптимальности (1) должно соответствовать цели обеспечения максимальной комфортности для людей при минимуме энергозатрат для i -ой зоны. Его чаще всего записывают в линейном или квадратичном виде. В основе выражения (2) лежат урав-

нения математической модели микроклимата здания, составленные на базе уравнений теплового баланса здания и материального баланса по переносу влаги и газовых примесей. Выражение (3) задаёт ограничения на параметры микроклимата, определяемые санитарными нормами или предпочтениями пользователей; выражение (4) связывает тепловую энергию, полученную i -ой зоной с расходом k -го вида энергоресурсов, а выражение (5) задаёт ограничения на потребляемые энергоресурсы.

Центральные агенты согласовывают работу зональных агентов в плане достижения глобальной цели обеспечения максимальной комфортности для людей при минимуме энергозатрат для всего здания. Они производят обработку и накопление полученной информации, а также контролируют соблюдение глобальных ограничений на энергоресурсы для всей системы.

Для задачи (1) - (5) задача координации, решаемая центральным агентом, может иметь вид

$$(6) \quad J(t) = \sum_{i=1}^N J_i(t) \rightarrow \min_{q_i},$$

$$(7) \quad \sum_{i=1}^N q_{ik}(t) \leq q_{kmax}, k = 1, 2, \dots, K,$$

где N – общее число зон, а K – общее число энергоресурсов. Здесь выражение (6) задаёт глобальный критерий оптимальности, а выражение (7) глобальные ограничения на энергоресурсы для всей системы. Значения q_{kmax} могут быть установлены по согласованию с агентами-поставщиками.

4 Мультиагентная модель в системе AnyLogic

Для проверки проектных решений была разработана компьютерная модель в системе AnyLogic [20]. Система поддерживает технологию агентного моделирования, а также включает пешеходную библиотеку для моделирования движения пешеходов (людей) в физическом пространстве. В моделях, созданных с помощью пешеходной библиотеки, пешеходы движутся в пространстве, реагируя на различные виды препятствий в виде стен и других пешеходов. Пешеходная библиотека AnyLogic позволяет визуализировать моделируемый процесс с помощью анимации, исследовать функционирование системы при различном уровне человекопотока и определять время пребывания людей в определённых участках модели.

В основе разработанной компьютерной модели лежит поэтажный план реального здания. Фрагмент модели в процессе имитации в системе AnyLogic 8 показан на рисунке 4. На плане положение пешеходов (людей) отмечено кружками. Люди могут передвигаться по коридору и заходить в помещения. Люди, попавшие в помещения, могут задерживаться там на различное время. Внутри помещения работники могут включать различные электроприборы, управлять микроклиматом помещений. В нижней части рисунка 4 показаны блоки пешеходной библиотеки, реализующие логику перемещения пешеходов внутри области разметки.

При создании новой модели с использованием пешеходной библиотеки автоматически будут добавлены типы агентов *основной* (класс, моделирующий среду обитания для других агентов) и *пешеход* (класс, моделирующий пешеходов). Остальные типы агентов необходимо добавлять вручную. Состав агентов модели показан на рисунке 5. Здесь представлены локальные ОВК-агенты (localHVACAgents), локальные агенты-освещения (localLightAgents), локальные агенты-энергоснабжения (localEnergyAgents), зональные ОВК-агенты (zoneHVACAgents) и центральный ОВК-агент (centralHVACAgent). Все агенты кроме сен-

traHVACAgent являются популяциями агентов. Кроме этого, для описания поведения людей в модели был создан тип агента-человека HumanAgent, расширяющий класс *пешеход* пешеходной библиотеки. Класс HumanAgent хранит массив характеристик человека, в соответствии с которым определяется алгоритм его поведения, задающий как человек будет перемещаться по зданию и как долго он будет находиться в здании; как он будет взаимодействовать с другими агентами для того чтобы добиться соответствия параметров окружающей среды своим предпочтениям.

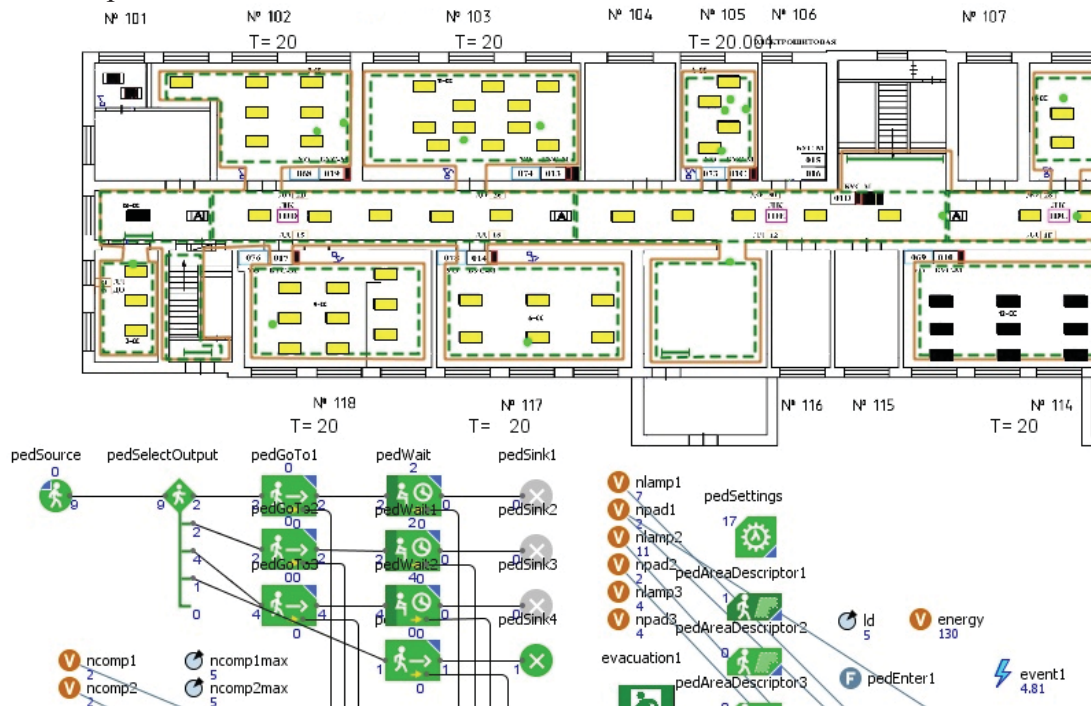


Рисунок 4 – Фрагмент модели в процессе имитации в системе AnyLogic

Все агенты, за исключением HumanAgent, не визуализируются на плане здания, но могут изменять значения связанных с ними переменных модели, отображаемых на плане. Для редактирования свойств и поведения агентов в графическом редакторе системы AnyLogic используются отдельные закладки, называемые диаграммами агентов. Все параметры, переменные, функции и диаграммы, относящиеся к конкретному типу агента, расположены на соответствующей диаграмме агента. Фрагмент диаграммы для зонального ОВК-агента приведён на рисунке 6. На данном фрагменте показана часть глобальных переменных и функций агента, а также его главная диаграмма действий. Диаграмма действий включает опрос и получение информации от локальных ОВК-агентов, опрос и получение информации о предпочтениях от агентов-людей, выполнение функций оптимального управления и выдачу команд управления в адрес локальных ОВК-агентов. Для взаимодействия между агентами модели используется механизм сообщений.

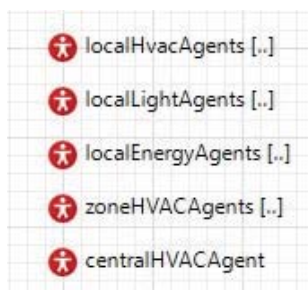


Рисунок 5 – Состав агентов модели

Работа локальных агентов электроснабжения и освещения здания моделировалась с помощью средств пешеходной библиотеки, диаграмм действий и состояний, программирования на языке Java. Моделирование микроклимата в помещениях проводилось с помощью компонентов «Системная динамика» системы AnyLogic. Динамика микроклимата помещения описывалась системой дифференциальных уравнений, рассмотренной в работе [21].

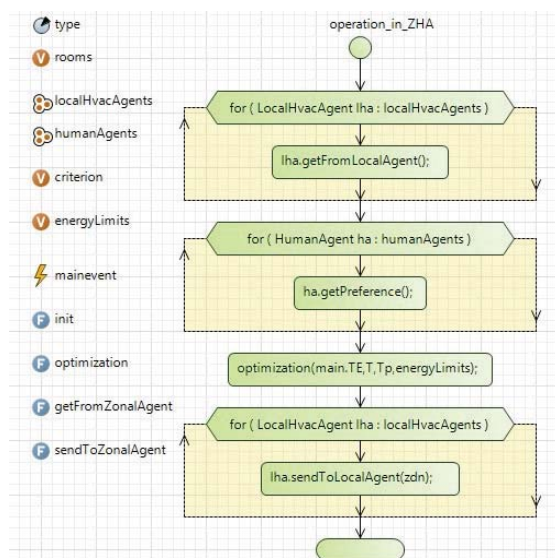


Рисунок 6 – Фрагмент диаграммы для зонального ОВК-агента

ОВК-система должна регулировать расход тепла от отопительных приборов (радиаторов, конвекторов) и расход свежего воздуха так, чтобы обеспечить требуемый микроклимат в помещениях. При этом необходимо добиваться экономии тепла и электроэнергии, потребляемой при работе оборудования ОВК-системы. Это возможно за счёт использования зональными ОВК-агентами специальных алгоритмов оптимального управления микроклиматом, реализующих решение задачи (1) - (5) и периодического (когда это возможно) отключения или снижения мощности климатического оборудования (насосов, вентиляторов).

Система AnyLogic не имеет встроенных возможностей по выполнению функций оптимизации и оптимального управления в процессе моделирования. Однако она позволяет обращаться к внешним библиотекам на языке Java. Поэтому для выполнения функций оптимального управления микроклиматом и энергосбережением в здании

была написана внешняя Java библиотека, использующая библиотеку оптимизации JOM [22]. Данная библиотека содержит популярные решатели задач оптимизации, в том числе, решатель нелинейных задач оптимизации большой размерности IPOPT [23].

На разработанной компьютерной модели были проведены численные эксперименты для различных режимов и значений параметров модели. Полученные в результате выполнения одного из экспериментов графики расхода электроэнергии и тепла в здании показаны на рисунке 7.

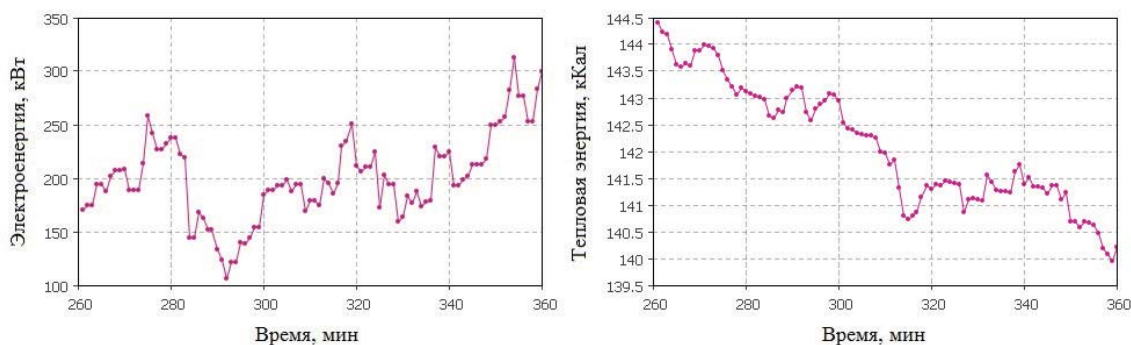


Рисунок 7 – Графики потребления электрической и тепловой энергии

Графики соответствуют дневному времени суток весеннего периода года, когда под влиянием повышения температуры наружного воздуха, солнечной радиации и бытовых тепловыделений потребление тепловой энергии в результате работы зональных ОВК-агентов и центрального ОВК-агента постепенно снижается.

Модель позволяет отработать вопросы взаимодействия между агентами, исследовать зависимость энергопотребления от интенсивности людского потока в коридорах и на лестничных клетках, числа людей в помещениях, количества включённых светильников и электроприборов, внешних метеорологических факторов, применяемых алгоритмов автоматического управления и т.д. Использование мультиагентного подхода позволяет полнее учесть эффекты взаимодействия между отдельными элементами системы, а также влияние, оказываемое людьми на работу инженерного оборудования.

Заключение

Рассмотрены вопросы проектирования МАС управления зданием. На основе анализа известных архитектур МАС управления зданием проведена классификация агентов, сформирован перечень агентов, определены классы агентов и их характеристики. Показан пример построения архитектуры МАС управления зданием, приведено описание функций отдельных агентов МАС. Осуществлено компьютерного моделирования МАС в системе AnyLogic.

Проведённые исследования показали перспективность применения мультиагентного подхода при создании АСУЗ. Разработанная модель АСУЗ, позволяет учесть большое количество факторов, оказывающих влияние на функционирование здания, а также рассмотреть различные варианты реализации алгоритмов управления инженерным оборудованием зданий.

Список источников

- [1] **Кувшинов, Ю.Я.** Основы обеспечения микроклимата зданий: Учеб. для вузов / Ю.Я. Кувшинов, О.Д. Самарин. – М.: Издательство Ассоциации строительных вузов, 2012. – 200 с.
- [2] **Shaikh, P.H.** A review on optimized control systems for building energy and comfort management of smart sustainable buildings / Shaikh P.H., Nor N.M., Nallagowngan P., Elamvazuthi I., Ibrahim T. // *Renewable and Sustainable Energy Reviews*, 34, 2014. - P.409–429. – DOI: 10.1016/j.rser.2014.03.027.
- [3] **Бударин, Е.Л.** Особенности принципа эргономичности в архитектуре и дизайне современного жилища / Е.Л. Бударин, Н.А. Сапрыкина // *Онтология проектирования*. – 2016, № 2(20). – С.205-215. – DOI: 10.18287/2223-9537-2016-6-2-205-215.
- [4] *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 2002, - 585 p.
- [5] **Городецкий, В.И.** Многоагентные системы (Обзор) / В.И. Городецкий, М.С. Грушинский, А.В. Хабалов // *Новости искусственного интеллекта*. 2008. № 2. - С.64–116.
- [6] **Городецкий, В.И.** Промышленные применения многоагентных систем: прогнозы и реалии / В.И. Городецкий, П.О. Скобелев, О.Л. Бухвалов, И.В. Майоров // В сборнике: *Проблемы управления и моделирования в сложных системах Труды XVIII Международной конференции*. Под редакцией: Е.А. Федосова, Н.А. Кузнецова, В.А. Виттиха. - Самара, 2016. - С.137-162.
- [7] **Abras, S.** A multi-agent home automation system for power management / Abras S., Ploix S., Pesty S., Jacomino M. // *Third International Conference in Control, Automation and Robotics*, 2006. - P.59–68. DOI: 10.1007/978-3-540-79142-3_6.
- [8] **Klein, L.** Coordinating occupant behavior for building energy and comfort management using multiagent systems / Klein L., Kwak J., Kavulya G., Jazizadeh F., Becerik-Gerber B., Varakantham P., Tambe M. // *Automation in Construction*, 22, 2012. - P.525–536. – DOI: 10.1016/j.autcon.2011.11.012.
- [9] **Booy, D.** A semiotic multi-agent system for intelligent building control / Booy D., Liu K., Qiao B., Guy C. // In *Proceedings of the 1st International Conference on Ambient Media and Systems*, 2008. – DOI: 10.4108/icst.ambisys2008.2881.
- [10] **Kollmann, S.** Towards a Cognitive Multi-agent System for Building Control / Kollmann S., Siafara L.C., Schaat S., Wendt A. // *Procedia Computer Science*, Volume 88, 2016. - P.191–197. – DOI: 10.1016/j.procs.2016.07.424.
- [11] **Yang, R.** Development of multi-agent system for building energy and comfort management based on occupant behaviors / Yang R., Wang L. // *Energy and Buildings*, 2013. - P.1–7. – DOI: 10.1016/j.enbuild.2012.10.025.
- [12] **Wang, Z.** Multi-agent control system with information fusion based comfort model for smart buildings / Wang Z., Wang L., Dounis A.I., Yang R. // *Applied Energy*, 99, 2012. - P.247–254. – DOI: 10.1016/j.apenergy.2012.05.020.
- [13] **Davidsson, P.** Saving energy and providing value added services in intelligent buildings: A MAS approach / Davidsson P. Boman M. // In *Agent Systems, Mobile Agents, and Applications*, 2000. - P.166–177. – DOI: 10.1007/978-3-540-45347-5_14.
- [14] **Huber, M.** Purely agent based control of building energy supply systems / Huber M., Brust S., Schütz T., Constantin A., Streblov R., Müller D. // In *ECOS - International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems*, 2015.
- [15] **Lacroix, B.** Multi-Agent Control of Thermal Systems in Buildings / Lacroix B., Paulus C., Mercier D. // *Conference Agent Technologies in Energy Systems*, 2012.
- [16] **Sangi, R.** A platform for the agent-based control of HVAC systems / Sangi R., Bünning F., Fütterer J., Müller D. // *Proceedings of the 12th International Modelica Conference*, 2017. - P.799-808.

- [17] **Смирнов, С.В.** Онтологии как смысловые модели / С.В. Смирнов // Онтология проектирования. – 2013, № 2 – С.12-19.
- [18] **Капустин, П.В.** Онтологические вопросы в кастомизированном архитектурном онлайн проектировании персонализированных жилых домов / П.В. Капустин, Д.М. Канин, И.Л. Чураков // Онтология проектирования. – 2015, № 3(17). – С.256-277. - DOI: 10.18287/2223-9537-2015-5-3-256-277.
- [19] **Марьясин, О.Ю.** Методы оптимального управления теплоснабжением зданий с использованием прогнозирующих моделей / О.Ю. Марьясин // В сб.: Проблемы управления и моделирования в сложных системах. Труды XVIII Междунар. конф. Под ред.: Е.А. Федосова, Н.А. Кузнецова, В.А. Виттиха. – Самара, 2017. - С.137-162.
- [20] AnyLogic – инструмент многоподходного имитационного моделирования. - <http://www.anylogic.ru/>.
- [21] **Колодкина, А.С.** Комплексное моделирование переноса тепла, влаги и газовых примесей для управления микроклиматом в здании / А.С. Колодкина, А.А. Огарков, О.Ю. Марьясин // Математика и естественные науки. Теория и практика: Межвуз. сб. науч. тр. вып. 10. - Ярославль: Издат. дом ЯГТУ, 2015. – С.145-154.
- [22] JOM (Java Optimization Modeler). - <http://www.net2plan.com/jom/index.php>.
- [23] Welcome to the Ipopt home page. - <https://projects.coin-or.org/Ipopt>.
-

DESIGN OF MULTI-AGENT BUILDING MANAGEMENT SYSTEM WITH ONTOLOGIES

O.Yu. Maryasin

*Yaroslavl State Technical University, Yaroslavl, Russia
maryasin2003@mail.ru*

Abstract

This article describes the design of a building management system. One of the problems that arise in the design is the task of organizing the harmonious work of traditional and advanced control systems for engineering equipment. Another problem is the identification of preferences and of people's activities. To solve these problems, the author suggests using a multi-agent approach. Based on the analysis of the architectures of multi-agent building management systems from literary and Internet sources, the classification of agents was conducted, as a result of which a list of agents was created and agent classes and their characteristics were determined. An example of constructing the architecture of a multi-agent building management system is shown, a description of the functions of individual agents is given. The design solutions were tested by computer simulation of the multi-agent system in the AnyLogic environment. For the first time classes of agents are proposed, which are the basis for building the architecture of a multi-agent building management system.

Key words: *indoor environment, energy saving, multi-agent system, AnyLogic.*

Citation: *Maryasin OYu.* Design of multi-agent building management system with ontologies [In Russian]. *Ontology of designing.* 2018; 8(3): 387-399. - DOI: 10.18287/2223-9537-2018-8-3-387-399.

References

- [1] **Kuvshinov YuYa, Samarin OD.** Fundamentals of microclimate buildings [In Russian]: Proc. for universities. – Moscow: Publishing house of the Association of Construction Universities, 2012. - 200 p.
- [2] **Shaikh PH, Nor NM, Nallagowngen P, Elamvazuthi I, Ibrahim T.** A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews*, 2014; 34: 409–429. – DOI: 10.1016/j.rser.2014.03.027.
- [3] **Budarin EL, Saprykina NA.** Features of the principle of ergonomics in architecture and design of the modern housing [In Russian]. *Ontology of Designing.* – 2016; 6(2): 205-215. – DOI: 10.18287/2223-9537-2016-6-2-205-215.
- [4] *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* The MIT Press, 2002, 585 p.
- [5] **Gorodetsky VI, Grushinsky MS, Khabalov AV.** Multi-agent systems (Overview) [In Russian]. *News of art intelligence.* 2008; 2: 64-116.

- [6] **Gorodetsky VI, Skobelev PO, Bukhvalov OL, Mayorov IV.** Industrial applications of multi-agent systems: forecasts and realities [In Russian]. In: Problems of control and modeling in complex systems of the XVIII International Conference. Edited by: E.A. Fedosov, N.A. Kuznetsov, V.A. Wittich. - Samara, 2016. - P.137-162.
- [7] **Abras S, Ploix S, Pesty S, Jacomino M.** A multi-agent home automation system for power management. In: Cetto J.A., Ferrier J.L., Costa dias Pereira J., Filipe J. (eds) Informatics in Control Automation and Robotics. Lecture Notes Electrical Engineering, vol 15. Springer, Berlin, Heidelberg. 2008. - DOI: 10.1007/978-3-540-79142-3_6.
- [8] **Klein L, Kwak J, Kavulya G, Jazizadeh F, Becerik-Gerber B, Varakantham P, Tambe M.** Coordinating occupant behavior for building energy and comfort management using multi-agent systems. *Automation in Construction*, 2012; 22: 525–536. – DOI: 10.1016/j.autcon.2011.11.012.
- [9] **Booy D, Liu K, Qiao B, Guy C.** A semiotic multi-agent system for intelligent building control. In Proceedings of the 1st International Conference on Ambient Media and Systems, 2008. – DOI: 10.4108/icst.ambisys2008.2881.
- [10] **Kollmann S, Siafara LC, Schaaf S, Wendt A.** Towards a Cognitive Multi-agent System for Building Control, *Procedia Computer Science*, 2016; 88: 191–197. – DOI: 10.1016/j.procs.2016.07.424.
- [11] **Yang R, Wang L.** Development of multi-agent system for building energy and comfort management based on occupant behaviors. *Energy and Buildings*, 2013. - P.1–7. – DOI: 10.1016/j.enbuild.2012.10.025.
- [12] **Wang Z, Wang L, Dounis AI, Yang R.** Multi-agent control system with information fusion based comfort model for smart buildings. *Applied Energy*, 99, 2012. - P.247–254. – DOI: 10.1016/j.apenergy.2012.05.020.
- [13] **Davidsson P, Boman M.** Saving energy and providing value added services in intelligent buildings: A MAS approach. In *Agent Systems, Mobile Agents, and Applications*, 2000. - P.166–177. – DOI: 10.1007/978-3-540-45347-5_14.
- [14] **Huber M, Brust S, Schütz T, Constantin A, Streblow R, Müller D.** Purely agent based control of building energy supply systems. In *ECOS - International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems*, 2015.
- [15] **Lacroix B, Paulus C, Mercier D.** Multi-Agent Control of Thermal Systems in Buildings, *Conference Agent Technologies in Energy Systems*, 2012.
- [16] **Sangi R, Bünnig F, Fütterer J, Müller D.** A platform for the agent-based control of HVAC systems. *Proceedings of the 12th International Modelica Conference*, 2017. – P.799-808.
- [17] **Smirnov SV.** Ontologies as semantic models [In Russian]. *Ontology of Designing*. - 2013, No.2. - P.12-19.
- [18] **Kapustin PV, Kanin DM, Churakov IL.** The ontological questions of personalize homes customized architectural online designing [In Russian]. *Ontology of Designing*. – 2015; 5(3): 256-277. - DOI: 10.18287/2223-9537-2015-5-3-256-277.
- [19] **Maryasin OYu.** Methods for the optimal control of the heat supply of buildings using predictive models [In Russian]. In: Problems of control and modeling in complex systems of the XVIII International Conference. Edited by: E.A. Fedosov, N.A. Kuznetsov, V.A. Wittich. - Samara, 2017. - P.137-162.
- [20] AnyLogic is a tool for multi-approach simulation modeling. - <http://www.anylogic.ru/>.
- [21] **Kolodkina AS, Ogarkov AA, Maryasin OYu.** Complex modeling of the transfer of heat, moisture and gas impurities to control the microclimate in the building [In Russian]. In: "Mathematics and natural sciences. Theory and Practice", No. 10. - Yaroslavl: Publication. house of the YSTU, 2015. - P.145-154.
- [22] JOM (Java Optimization Modeler). - <http://www.net2plan.com/jom/index.php>.
- [23] Welcome to the Ipopt home page. - <https://projects.coin-or.org/Ipopt>.

Сведения об авторе



Марьясин Олег Юрьевич, 1964 г. рождения. Окончил Ярославский политехнический институт в 1986 г., к.т.н. (1992). Доцент кафедры «Кибернетика» Ярославского государственного технического университета. Член Российской ассоциации инженеров по отоплению, вентиляции, кондиционированию воздуха, теплоснабжению и строительной теплофизике (АВОК). В списке научных трудов более 50 работ в области автоматизации технических систем, автоматизации инженерных систем зданий, компьютерного моделирования и оптимизации сложных систем.

Oleg Yurevich Maryasin (b. 1964) graduated from the Yaroslavl Polytechnic Institute (Yaroslavl-city) in 1986, PhD (1992). He is Associate Professor at Yaroslavl State Technical University (Department of Cybernetics). He is member of Russian Association of engineers for heating, ventilation, air-conditioning, heat supply and building thermal physics. He is co-author of more than 50 publications in the field of automation of technical systems, automation of building engineering systems, complex systems simulation and optimization.

УДК 004.822

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЧЕСКОГО ПОДХОДА ПРИ ПРОЕКТИРОВАНИИ МНОГОФУНКЦИОНАЛЬНОГО АВИАЦИОННОГО ИНДИКАТОРА

Ю.К. Горелов¹, С.К. Киселев²

Ульяновский государственный технический университет, Ульяновск, Россия

¹ gur-ok@yandex.ru, ² ksk@ulstu.ru

Аннотация

В статье предлагается использовать онтологический подход для создания экспертной системы поддержки принятия решений в задачах концептуального и эскизного проектирования авиационного индикатора. Для создания онтологии использовался редактор Protégé. Показана возможность внедрения аппарата нечёткой логики в онтологию при помощи языка семантических данных SWRL. Схематично изображена структура отношений нечёткой онтологии авиационного индикатора. Для формирования запросов к онтологии и выдачи результатов использовался язык SQWRL. В качестве примера рассмотрен процесс выдачи рекомендации типа подсвета дисплея в зависимости от размера диагонали экрана и требуемой яркости при помощи нечёткой онтологии. Фаззификация и задание функции принадлежности нечёткого множества реализованы средствами редактора Protégé через свойства объектов онтологии. Для получения чёткого значения рекомендации проведён процесс дефаззификации, сформирован набор нечётких правил вывода экспертной системы, приведена методика расчёта коэффициентов рекомендации. В процесс вычислений применялись встроенные в SWRL функции. Показана возможная реализация рекомендаций экспертной системы в проектах индикаторов, в частности представлены типовые конструкции модуля подсвета авиационного ЖК-дисплея. Новым является применение онтологий с элементами нечёткой логики в решении задач проектирования авиационного индикатора.

Ключевые слова: онтология, база знаний, авиационный индикатор, экспертная система, нечёткая логика.

Цитирование: Горелов, Ю.К. Использование онтологического подхода при проектировании многофункционального авиационного индикатора / Ю.К. Горелов, С.К. Киселев // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.400-411. DOI: 10.18287/2223-9537-2018-8-3-400-411.

Введение

При создании нового сложного технического устройства, такого как многофункциональный авиационный индикатор, инженер-конструктор должен работать с большим объёмом информации. К современным индикаторам предъявляется большое количество требований в зависимости от области применения, назначения и т.д. Конструктивные недоработки при разработке устройства могут привести к ухудшению технических характеристик конечного изделия. Частичным решением этой проблемы является создание базы знаний (БЗ) по предметной области (ПрО) многофункционального авиационного индикатора. Такая база должна содержать информацию о всех модулях, конструктивных особенностях, материалах и решениях, применяемых в конструировании бортовых систем отображения информации. БЗ позволит инженеру-конструктору направленно генерировать новые технические решения для создания новых индикаторов или улучшения эксплуатационных характеристик уже имеющихся. Формируемую информационную базу необходимо пополнять, своевременно актуализировать и корректно использовать в процессе проектирования нового или усовершенствования существующего устройства. Выполнение перечисленных задач, а также поиск и выяв-

ление оптимальной компоновки индикатора является трудоёмкой задачей, так как подразумевает обработку больших объёмов информации и её последующий экспертный анализ. Для решения задачи концептуального и эскизного проектирования авиационного индикатора предлагается использовать онтологический подход. Использование онтологии в качестве БЗ обосновано возможностью наиболее цельно представить сведения о ПрО, включая понятия, их свойства и отношения. Кроме того, имея онтологию, можно получить логические следствия, т.е. получить факты, которые не представлены в онтологии буквально, но следуют из её семантики. Наиболее важным является получение из онтологии данных, подлежащих дальнейшей математической обработке, что позволяет разработать методику оптимального проектирования авиационного индикатора в условиях многокритериальности.

На рисунке 1 изображена условная схема работы пользователя с онтологией.

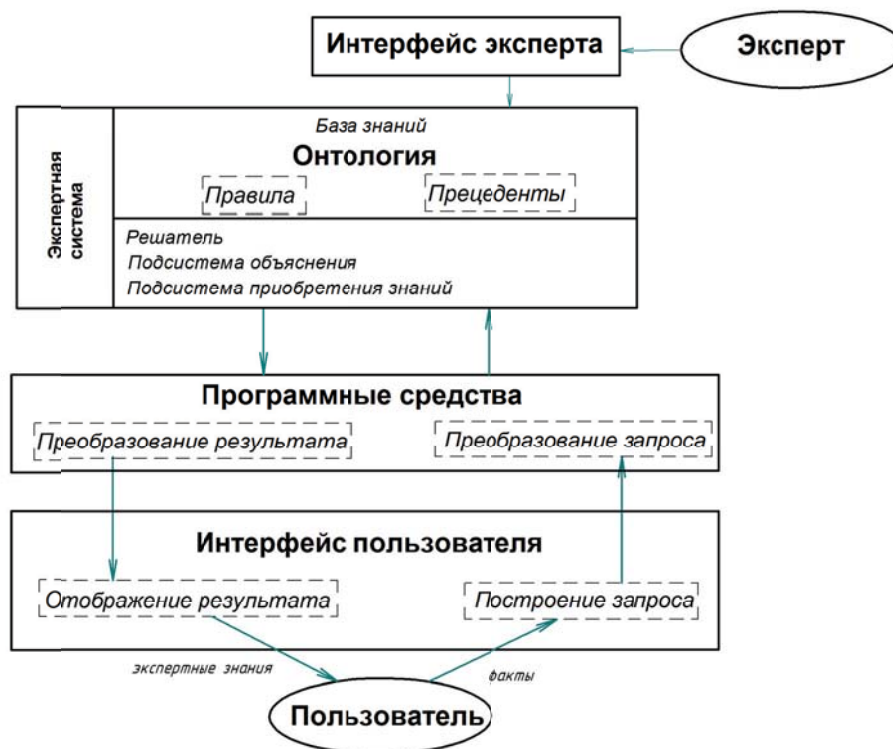


Рисунок 1– Схема работы пользователя с онтологией

В роли пользователя выступает инженер-разработчик индикатора, делающий запрос к онтологии посредством специального интерфейса. Программная система преобразовывает запрос пользователя, забирает и обрабатывает данные из онтологии, выдаёт ответ пользователю в виде готового проектного решения.

1 Создание онтологии предметной области авиационного индикатора

Для создания семантического графа онтологии авиационного индикатора использовался Protégé – свободный, открытый редактор онтологий и фреймворк для построения БЗ. Онтологии, построенные в Protégé, могут быть экспортированы во множество форматов, включая RDF (RDF Schema), OWL и XML Schema [1].

Точная спецификация ПрО – основная задача онтологии. Онтология определяет общий словарь для специалистов, которым нужно совместно использовать информацию в ПрО. Онтология авиационного индикатора содержит основные понятия ПрО, такие как: характери-

стики индикатора (масса, габариты, надёжность, стоимость, энергопотребление, вибромеханическая прочность, ремонтпригодность, оптические показатели и т.д.), модульный состав (модуль питания, модуль дисплейный, вычислительный модуль и т.д.), теплоотведение (пассивное, принудительное), электромагнитная совместимость, конструкционные материалы (сталь, алюминий, оргстекло, рассеивающие и светоусиливающие плёнки, клеи, герметики и т.д.), документация, программное обеспечение и т.д. Точное представление связей между зафиксированными концептами впоследствии кодируется на формальном языке. В процессе кодирования эти концепты используются для создания классов, подклассов, связей и ограничений отношений между ними. Учёт взаимосвязь объектов онтологии - важная часть разрабатываемой БЗ. Пример взаимосвязи понятий из области проектирования авиационного индикатора - когда значения таких параметров, как масса, габариты, стоимость, механическая прочность зависят от выбора конструкционных материалов изделия. Или, например, светотехнические параметры зависят от конструктивных особенностей модуля подсвета, что в свою очередь также связано с энергопотреблением и проблемой теплоотведения. Кроме того, в индикаторах могут использоваться различные сенсоры и преобразователи, поэтому надёжность, точность измерений, быстродействие, технологичность зависят от физических принципов датчиков.

Схема назначения отношений онтологии представлена на рисунке 2, где изображены два класса (множества) индивидов: «Индикаторы» и «Модули». Индивиды класса «Индикаторы» связаны с индивидами множества «Модули» свойством объектов «hasModule», которое означает наличие в составе индикатора какого-либо модуля. Кроме свойств объектов, концепты в Protégé могут иметь свойства данных. Например, индивиды класса «Индикаторы» имеют свойство данных «hasMass» типа integer, обозначающее расчётную массу конкретного индикатора. На таких триплетах строится вся онтология.

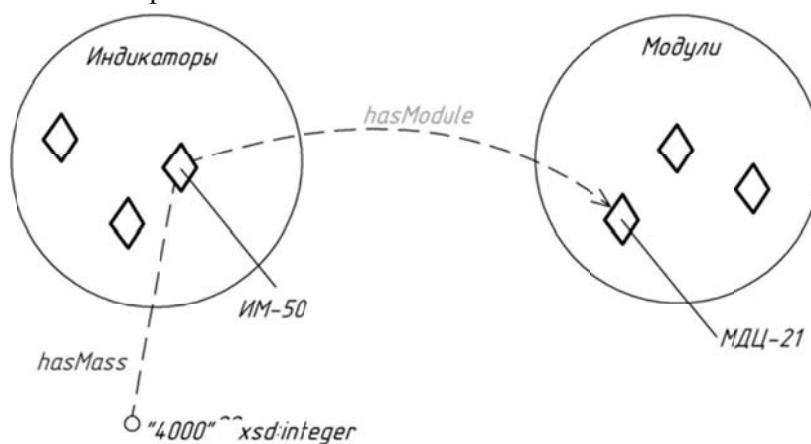


Рисунок 2 – Свойства объектов и свойства данных в онтологии

Редактор онтологий также позволяет накладывать дополнительные ограничения на свойства объектов, например, в виде создания иерархии свойств, по аналогии с иерархией классов и подклассов. На рисунке 3 изображена связь индивида множества «Индикаторы» с индивидом из множества «Модули_дисплейные» через подсвойство «hasDisplayModule» свойства «hasModule». Подсвойство «hasDisplayModule» является определяющим для всего класса «Индикаторы», т.к. все индикаторы должны иметь в своём составе дисплейные модули. Также в качестве дополнительного ограничения, например, свойство объекта можно сделать функциональным, т.е. применимым к объекту только единожды. Свойство «producedBy» в созданной онтологии обозначает предприятие, разработавшее индикатор, и, следовательно, должно быть функциональным. Все описания классов и заданные отношения

между ними требуются для создания более широкой и глубокой семантической паутины с целью обращения к ней с информационными запросами.

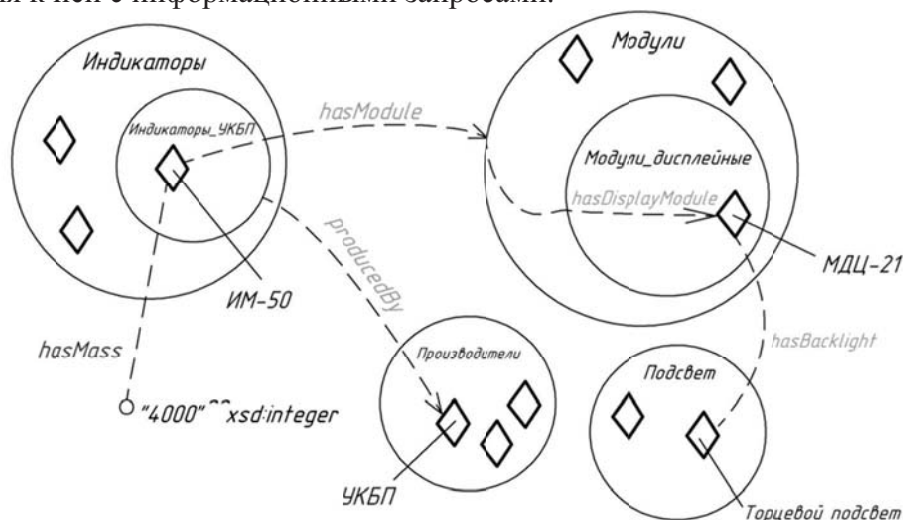


Рисунок 3 – Отношения объектов в онтологии

Созданная онтология визуализируется в виде графа [2]. На рисунке 4 показан пример структуры и отношений онтологии авиационного индикатора, построенной с помощью редактора Protégé.

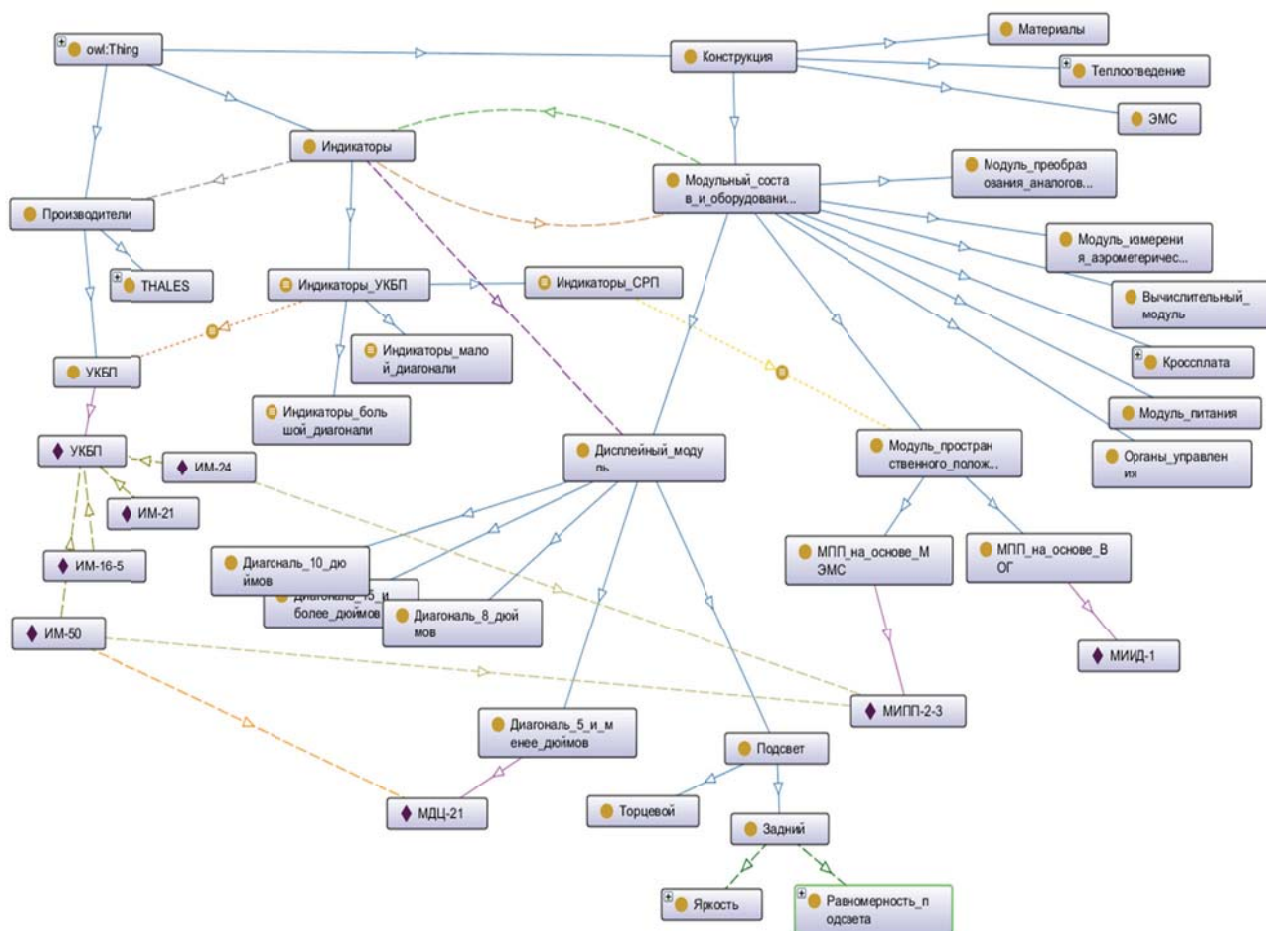


Рисунок 4– Структура и отношения ПрО в виде графа

2 Экспертная система на основе онтологии с нечёткой логикой

Целью разработки онтологии авиационного индикатора является создание на её основе экспертной системы (ЭС) помощи при проектировании. Входными данными такой системы являются требования технического задания на разработку нового изделия, а на выходе система должна выдавать рекомендации инженеру-проектировщику по конструкции и компоновке. Для создания аппарата, способного моделировать человеческие рассуждения и объяснять человеческие приёмы принятия решений в ходе решения задачи проектирования, предлагается использовать нечёткую логику. Ввод нечёткой логики в онтологию направлен на расширение возможности классической логики и позволяет применять концепцию неопределённости в логических выводах [3, 4].

Нечёткая онтология, которую описал Silvia Calegari [5], - это онтология, дополненная нечёткими величинами. Задать нечёткую онтологию можно двумя функциями:

$$f: (\text{Классы} \cup \text{Индивиды}) \times \text{Свойства} \rightarrow \text{Величина_свойствах} [0,1],$$

например, $f: (\text{Модуль}A, \text{вес}) \rightarrow (\text{тяжелый}, 0.8)$;

$$g: (\text{Классы} \cup \text{Индивиды}) \times (\text{Свойства} \cup \text{Величина_свойства}) \rightarrow [0,1],$$

например, $g: (\text{Модуль}A, \text{дешевый}) \rightarrow 0.4$.

Нечёткая онтология содержит комбинации функций f и g при помощи логических операторов И и ИЛИ.

Для этого некоторые концепты онтологии получают степень принадлежности (membership degree) $(md) \rightarrow [0,1]$. Эта степень определяется явным образом в виде функциональной зависимости либо дискретно – путём задания конечной последовательности значений [6]. Фрагмент структуры нечёткой онтологии в рассмотренной ПрО показан на рисунке 5.

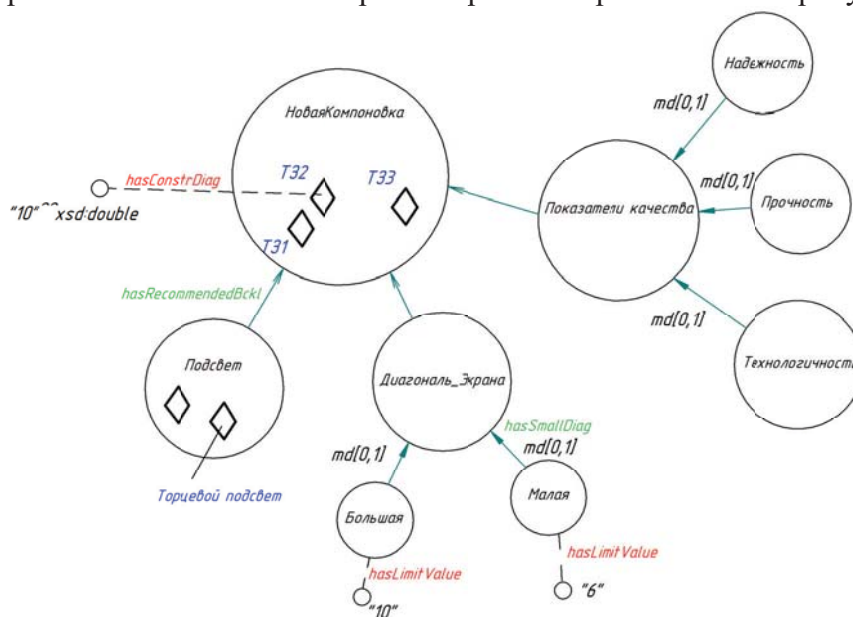


Рисунок 5 – Структура нечёткой онтологии авиационного индикатора

В качестве примера функционирования ЭС помощи при проектировании индикатора рассмотрен процесс рекомендации выбора типа подсвета при различной диагонали экрана и требуемой яркости проектируемых изделий. Размер диагонали и яркость экрана определены техническим заданием на проектирование. Для занесения этой информации в онтологию был создан класс «НоваяКомпоновка», в котором содержатся некоторые индивиды, представляющие проектируемые индикаторы. Индивидам класса «НоваяКомпоновка» присвоено свойство «hasConstrDiag» со значением диагонали экрана и свойство «hasBrightness» со значени-

ем требуемой яркости. Создан класс «Диагональ_экрана» с лимитирующими концептами «Малая» и «Большая» с соответствующими значениями 6 и 10, привязанными через свойство «hasLimitValue». Лимитирующие значения необходимы для задания функции принадлежности диагонали экрана к классу «Малая» или «Большая», также индивидам присвоено свойство «hasSmallDiag» для представления функции $f_{\text{малая_диагональ}}$, определяющей степень принадлежности значения диагонали экрана к нечёткому понятию «Малая». Графическое изображение функции принадлежности нечёткого множества $f_{\text{малая_диагональ}}$ представлено на рисунке 6. Аналогичным образом задаётся функция принадлежности значения яркости к классу «Высокая» $f_{\text{высокая}}$ с лимитирующими значениями a и b .

Занесение функций принадлежности в онтологию осуществляется посредством написания SWRL-правил [7, 8], которые выглядят следующим образом:

НоваяКомпоновка(?c)^hasLimitValue(Small,?s)^hasConstrDiag(?c,?d)^swrlb:lessThanOrEqualTo(?d,?s) ->hasSmallDiag(?c,1.0)

НоваяКомпоновка(?c)^hasLimitValue(Large,?l)^hasConstrDiag(?c,?d)^swrlb:greaterThanOrEqualTo(?d,?l) ->hasSmallDiag(?c,0.0)

НоваяКомпоновка(?c)^hasLimitValue(Small,?s)^hasLimitValue(Large,?l)^hasConstrDiag(?c,?d)^swrlb:greaterThan(?d,?s)^swrlb:lessThan(?d,?l)^swrlb:subtract(?r,?l,?d)^swrlb:subtract(?sub,?l,?s)^swrlb:divide(?div,?r,?sub) ->hasSmallDiag(?c,?div).

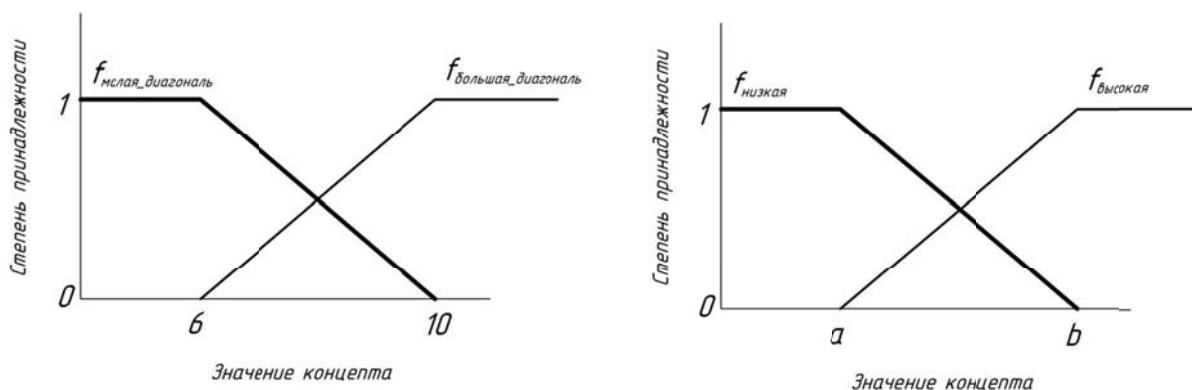


Рисунок 6 – Графическое изображение функций принадлежности нечётких множеств

Для создания правила могут использоваться встроенные в SWRL функции для вычислений: swrlb:add (сумма), swrlb:subtract (разница), swrlb:divide (отношение), swrlb:multiply (произведение).

Согласно заданному правилу, диагональ экрана меньше 6 дюймов принадлежит к понятию «Малая» со значением функции принадлежности равным 1, диагональ больше 10 дюймов имеет степень принадлежности к классу «Малая» равную 0. Диагонали экрана от 6 до 10 дюймов относятся к понятию «Малая» со значением функции принадлежности в диапазоне 0...1.

Выбор рекомендуемого типа подсвета происходит в зависимости от диагонали экрана и яркости. Условно для индикаторов с малой диагональю рекомендовано использовать задний тип подсвета, т.к. он обеспечивает лучшие светотехнические характеристики, а для больших диагоналей предпочтительнее использовать торцевой подсвет, ввиду его меньшего энергопотребления и меньших габаритов. Однако торцевой подсвет можно использовать и при малых диагоналях, если не требуется высокая яркость.

Для выбора рекомендуемого типа подсвета необходимо задание нескольких нечётких «ЕСЛИ-ТО» правил вывода ЭС.

Правило 1. Если Диагональ_Экрана Малая И Яркость Высокая, то Подсвет Задний.

Правило 2. Если Диагональ_Экрана *Малая* И Яркость *Низкая*, то Подсвет *Задний/Торцевой*.

Правило 3. Если Диагональ_Экрана *Большая* И Яркость *Низкая*, то Подсвет *Торцевой*.

Правило 4. Если Диагональ_Экрана *Большая* И Яркость *Высокая*, то Подсвет *Торцевой*.

Правила представляются в виде таблицы 1.

Таблица 1 – Правила вывода экспертной системы

Нечёткое правило, №	1	2	3	4
Диагональ_экрана	Малая	Малая	Большая	Большая
Яркость	Высокая	Низкая	Низкая	Высокая
Рекомендуемый_подсвет	Задний	Торцевой/Задний	Торцевой	Торцевой

Для комбинирования нецелочисленных значений истинности в нечёткой логике определяется эквиваленты операций И, ИЛИ, НЕТ [9,10]:

$p1 \text{ И } p2 = \min(p1, p2)$ (т.е. меньшее);

$p1 \text{ ИЛИ } p2 = \max(p1, p2)$ (т.е. большее);

НЕ $p1 = 1 - p1$ (т.е. обратное значение).

Оператор «И» в правилах 1-4 указывает на то, что коэффициент истинности правил и значение нечёткой переменной «Рекомендуемый_подсвет» рассчитывается как меньшее из значений нечётких переменных «Диагональ_экрана» и «Яркость» для каждого правила.

В дальнейшем необходимо провести процесс дефазификации для получения чёткого (численного) значения рекомендации. Используем одноэлементную функцию для подсчёта общего коэффициента рекомендации (RV) [7], показанную на рисунке 7, и произведём средневзвешенный расчёт по формуле (1). Степень рекомендации разделена на три уровня: задний (20 единиц), торцевой/задний (50 единиц) и торцевой (80 единиц).

$$(1) \quad RV = \frac{(zRV \times 20) + (tzRV \times 50) + (tRV \times 80)}{zRV + tzRV + tRV},$$

где: RV – общий коэффициент рекомендации,

zRV – коэффициент истинности для первого правила (Рекомендуемый_подсвет *Задний*),

tzRV – коэффициент истинности для второго правила (Рекомендуемый_подсвет *Торцевой/Задний*),

tRV – коэффициент истинности для третьего и четвёртого правил (Рекомендуемый_подсвет *Торцевой*).

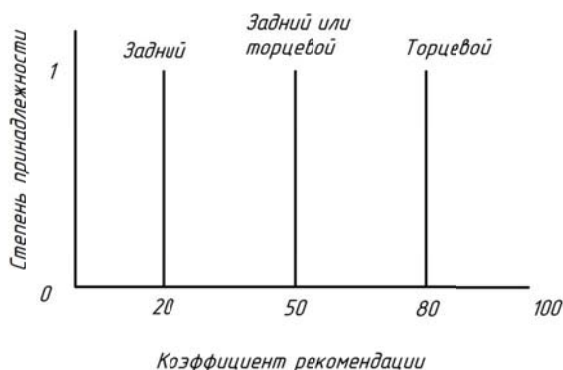


Рисунок 7– График функции для расчёта коэффициента рекомендации

Нечёткие правила 1-4 и расчёт общего коэффициента рекомендации RV заносятся в онтологию при помощи языка SWRL:

$hasSmallDiag(?c,?smd) \wedge hasHighBrightness(?c,?hbr) \wedge swrlb:lessThanOrEqual(?smd,?hbr) \rightarrow hasZRecommendationValue(?c,?smd)$

$$\begin{aligned}
& hasSmallDiag(?c,?smd) \wedge hasHighBrightness(?c,?hbr) \wedge swrlb:lessThanOrEqual(?hbr,?smd) \rightarrow \\
& hasZRecommendationValue(?c,?hbr) \\
& hasSmallDiag(?c,?smd) \wedge hasHighBrightness(?c,?hbr) \wedge swrlb:subtract(?lbr,1,?hbr) \wedge \\
& swrlb:lessThanOrEqual(?smd,?lbr) \rightarrow hasTZRecommendationValue(?c,?smd) \\
& hasSmallDiag(?c,?smd) \wedge hasHighBrightness(?c,?hbr) \wedge swrlb:subtract(?lbr,1,?hbr) \wedge \\
& swrlb:lessThanOrEqual(?lbr,?smd) \rightarrow hasTZRecommendationValue(?c,?lbr) \\
& hasSmallDiag(?c,?smd) \wedge swrlb:subtract(?larmd,1,?smd) \wedge hasHighBrightness(?c,?hbr) \wedge \\
& swrlb:subtract(?lbr,1,?hbr) \wedge sqwrl:makeBag(?bag,?lbr,?hbr) \wedge sqwrl:max(?max,?bag) \wedge \\
& swrlb:lessThanOrEqual(?larmd,?max) \rightarrow hasTRecommendationValue(?c,?larmd) \\
& hasSmallDiag(?c,?smd) \wedge swrlb:subtract(?larmd,1,?smd) \wedge hasHighBrightness(?c,?hbr) \wedge \\
& swrlb:subtract(?lbr,1,?hbr) \wedge sqwrl:makeBag(?bag,?lbr,?hbr) \wedge sqwrl:max(?max,?bag) \wedge \\
& swrlb:lessThanOrEqual(?max,?larmd) \rightarrow hasTRecommendationValue(?c,?max) \\
& hasZRecommendationValue(?c,?zRV) \wedge hasTZRecommendationValue(?c,?tzRV) \wedge hasTRecom- \\
& mendationValue(?c,?tRV) \wedge \\
& swrlb:multiply(?mul1,?zRV,20.0) \wedge swrlb:multiply(?mul2,?tzRV,50.0) \wedge \\
& swrlb:multiply(?mul3,?tRV,80.0) \wedge swrlb:(?add1,?mul1,?mul2,?mul3) \wedge \\
& swrlb:add(?add2,?zRV,?tzRV,?tRV) \wedge swrlb:divide(?RV,?add1,?add2) \rightarrow hasRecommendationVal- \\
& ue(?c,?RV).
\end{aligned}$$

На основе общего коэффициента рекомендации ЭС выбирает тип подсвета. При значении $20 \leq RV \leq 45$ рекомендуемый подсвет Задний, при $45 < RV < 55$ – рекомендуется как Задний, так и Торцевой подсвет, при $55 \leq RV \leq 80$ – рекомендуемый подсвет Торцевой.

Свойства «hasRecommendedBckl» и «hasRecommendationValue» интерпретируют рекомендуемый тип подсвета и общий коэффициент рекомендации RV. Для этого необходимо задать SWRL правило:

$$НоваяКомпоновка(?c) \wedge hasRecommendationValue(?c, ?x) \wedge swrlb:lessThanOrEqual(?x, 45) \rightarrow hasRecommendedBckl(?c, «Задний»)$$

$$НоваяКомпоновка(?c) \wedge hasRecommendationValue(?c, ?x) \wedge swrlb:lessThan(?x, 55) \wedge swrlb:greaterThan(?x, 45) \rightarrow hasRecommendedBckl(?c, «Торцевой или Задний»)$$

$$НоваяКомпоновка(?c) \wedge hasRecommendationValue(?c, ?x) \wedge swrlb:greaterThan(?x, 55) \rightarrow hasRecommendedBckl(?c, «Торцевой»).$$

Вывод результатов производился при помощи SQWRL-запросов [11], создание которых, как и правил SWRL, происходит в плагине к редактору онтологий Protégé. В таблице 2 представлен результат SQWRL-запроса, отображающий рекомендуемый тип подсвета на основе оценки значения диагонали экрана и яркости проектируемой компоновки. SQWRL-запрос:

$$\begin{aligned}
& НоваяКомпоновка(?n) \wedge hasConstrDiag(?n, ?z) \wedge hasBrightness(?n, ?x) \wedge \\
& hasRecommendedBckl(?n, ?bck) \wedge hasRecommendationValue(?n, ?v) \rightarrow sqwrl:select(?n, ?z, ?x, \\
& ?bck, ?v) \wedge sqwrl:columnNames(«ТЗ», «Диагональ», «Яркость», «Рекомендуемый тип подсвета», «Коэффициент RV»).
\end{aligned}$$

Таблица 2- Результат запроса пользователя к системе

№ п/п	Диагональ экрана, дюймы	$f_{\text{малая_диагональ}}$	$f_{\text{высокая_яркость}}$	zRV	tzRV	tRV	RV	Рекомендуемый тип подсвета
1	5	1	0,8	0,8	0,2	0	26	Задний
2	7	0,75	0,7	0,7	0,3	0,25	39,2	Задний
3	7,8	0,55	0,2	0,2	0,55	0,45	56,3	Торцевой
4	8,5	0,4	0,6	0,4	0,4	0,6	54,3	Задний/торцевой
5	10	0	0,35	0	0	0,65	80	Торцевой
6	15	0	0,9	0	0	0,9	80	Торцевой

Рассмотрим методику расчёта коэффициентов рекомендации на примере случая №3 из таблицы 2. Исходными данными в этом случае были диагональ, равная 7,8 дюйма, и условно низкая яркость подсвета.

1) Значение размера диагонали экрана равно 7,8 дюйма необходимо фазифицировать. Согласно заданной выше функции принадлежности с лимитирующими значениями 6 и 10 дюймов (рисунок 6) происходит расчет функции принадлежности нечёткого множества $f_{\text{малая_диагональ}} = \frac{(10-7,8)}{(10-6)} = 0,55$. Соответственно значение функции принадлежности $f_{\text{большая_диагональ}} = 1 - f_{\text{малая_диагональ}} = 0,45$.

2) По условию высокая яркость подсвета не требуется, поэтому зададим напрямую $f_{\text{высокая_яркость}} = 0,2$. Такой способ определения значения функции принадлежности $f_{\text{высокая_яркость}}$ использован как пример возможности задания конкретного значения степени принадлежности классов онтологии. Соответственно $f_{\text{низкая_яркость}} = 1 - f_{\text{высокая_яркость}} = 0,8$.

Для расчёта коэффициентов zRV, tzRV и tRV необходимо заполнить таблицу 1, используя полученные значения $f_{\text{малая_диагональ}}$, $f_{\text{высокая_яркость}}$ и т.д., согласно Правилам 1-4.

3) В результате имеем таблицу 3 с внесёнными значениями функций принадлежности.

Таблица 3 - Расчёт коэффициентов рекомендации по правилам вывода

Нечёткое правило, №	1	2	3	4
Диагональ экрана	0,55	0,55	0,45	0,45
Яркость	0,2	0,8	0,8	0,2
Рекомендуемый подсвет	zRV	tzRV	tRV	

В соответствии с нечёткими правилами ЭС:

правило 1: $zRV = \min(0.55, 0.2) = 0.2$;

правило 2: $tzRV = \min(0.55, 0.8) = 0.55$;

правило 3 и 4: $tRV = \max(\min(0.45, 0.8), \min(0.45, 0.2)) = 0.45$.

4) Расчёт общего коэффициента рекомендации RV ведётся по формуле (1) на основе полученных zRV, tzRV, tRV и ЭС делается вывод о рекомендуемом типе подсвета, который в дальнейшем учитывается проектантом при компоновке конструкции дисплея авиационного индикатора.

На рисунке 8 изображена конструкция авиационного дисплея, в которой плата подсвета располагается с одного из торцов ЖК-панели. Такая компоновка имеет ряд преимуществ: малые габариты, высокая энергоэффективность. Однако применение торцевого типа подсвета имеет и некоторые недостатки, связанные с оптическими характеристиками, а именно неравномерность подсвета и относительно низкую яркость. Улучшить оптические свойства дисплея с торцевым подсветом удаётся благодаря применению оптических рассеивающих и светоусиливающих полимерных плёнок, а также при помощи определённой формы светопровода (например, клиновидной). Для минимизации потерь яркости подсвета при прохождении света через границы раздела сред воздух-стекло может применяться склейка ЖК-панели с защитным стеклом. Оптическая склейка является сложной технологически, однако позволяет значительно улучшить световые характеристики дисплея, а так же увеличить ударопрочность, превращая панель в триплекс.

На рисунке 9 в разрезе показан авиационный ЖК-дисплей с задним подсветом. Такая компоновка применяется при высоких уровнях внешней засветки. Для отведения тепла плата со светодиодами установлена на радиатор, что приводит к увеличению массогабаритных характеристик индикатора. Поэтому для принятия проектного решения необходимо также учи-

тывать и ограничения по массе, что является особо важным для изделия авиационного назначения.

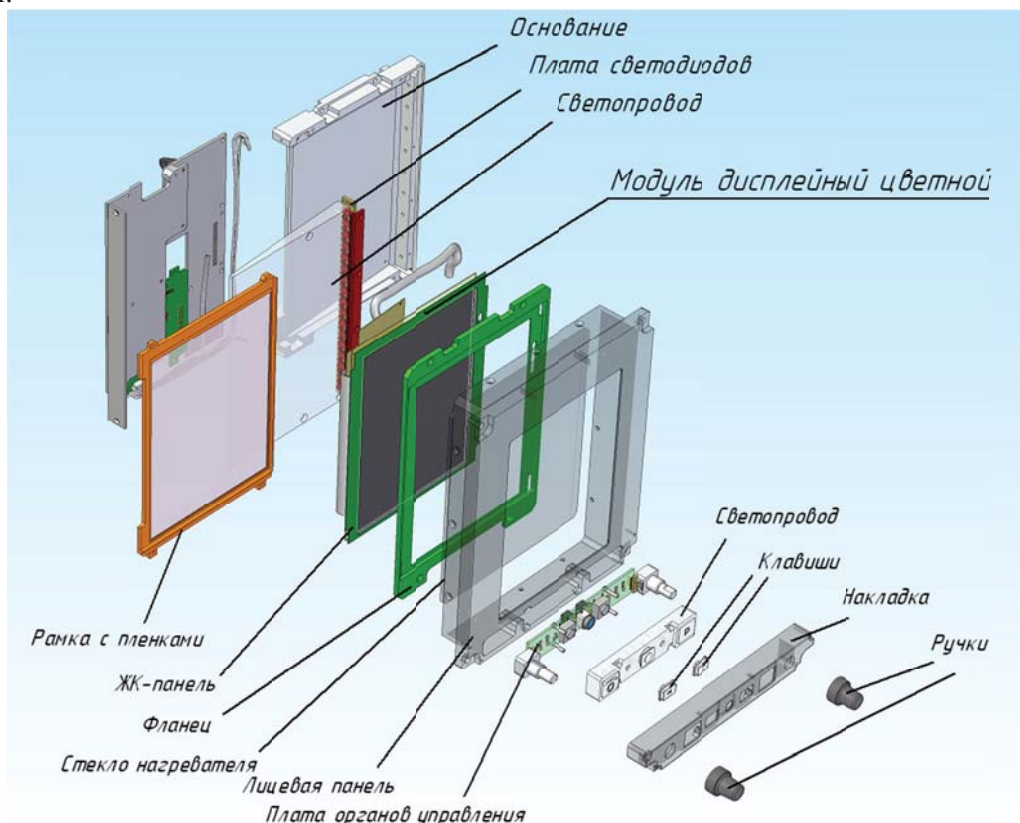


Рисунок 8 - Конструкция авиационного дисплея с торцевым подсветом

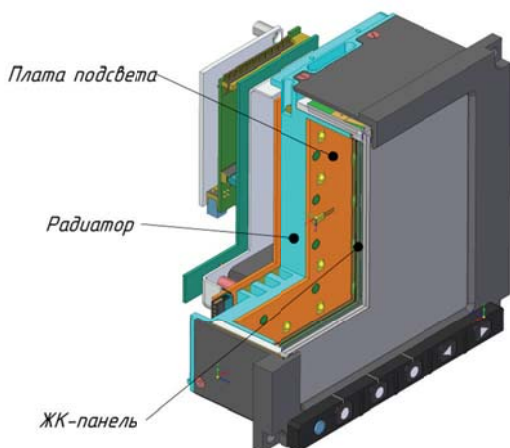


Рисунок 9 - Задний подсвет ЖК-дисплея

Для учёта различных факторов, влияющих на выдачу рекомендации, ЭС на основе онтологии дополняется наборами нечётких правил вывода, которые определяют взаимосвязь различных характеристик индикатора и применяемых в нём конструктивных решений. Так удаётся достичь оптимального соотношения по нескольким характеристикам проектируемого изделия: стоимости, технологичности, прочности, светотехнике и т.д.

Заключение

Предлагается использовать онтологический подход для решения задачи концептуального и эскизного проектирования многофункционального индикатора авиационного назначения.

Разрабатываемая на основе онтологии ЭС помощь при проектировании индикатора поможет инженеру в разработке изделия на начальных этапах. Показана возможность внедрения аппарата нечёткой логики в разработанную онтологию при помощи SWRL правил с целью формирования рассуждений, принятия решений и выдачи рекомендаций при помощи SQWRL-запросов. В качестве примера функционирования ЭС рассмотрен процесс подготовки рекомендации выбора типа подсвета в зависимости от размера диагонали и требуемой яр-

кости экрана. Дальнейшая работа направлена на дополнение онтологии и расширение применения нечёткой логики. Кроме того, необходимо создание удобного интерфейса для введения исходных данных и отображения рекомендаций и проектных решений ЭС.

Список источников

- [1] *Добров, Б.В.* Онтология и тезаурусы: модели, инструменты, приложения. / Б.В. Добров, В.В. Иванов, Н.В. Лукашевич, В.Д. Соловьев. – М.: Бинوم. Лаборатория знаний, 2009. – 173 с.
- [2] *Шустова, Д.В.* Подход к разработке семантических основ информационных систем для проектирования и производства авиационной техники. / Онтология проектирования. – 2015. – Т.5, №1(15). – С. 70-84.
- [3] *Гаврилова, Т.А.* Базы знаний интеллектуальных систем. / Т.А. Гаврилова, В.Ф. Хорошевский.– Санкт-Петербург: Питер, 2000. – 382 с.
- [4] *Корнеев, В.В.* Базы данных. Интеллектуальная обработка информации. / В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх. – Москва: Нолидж, 2000. – 352 с.
- [5] *Calegari S* Integrating Fuzzy Logic in Ontologies / S. Calegari, D. Ciucci // In Proceedings of the Eighth International Conference on Enterprise Information Systems - 2006. Volume 2: ICEIS, - P.66-73. DOI: 10.5220/0002496100660073.
- [6] *Валькман, Ю.Р.* Проектирование онтологии для правовой предметной области на основе текстового контента с использованием нечеткой логики. / Ю.Р. Валькман, Е.А. Хала // Онтология проектирования. – 2014. – №2(12). – С. 19-39
- [7] *HattaFudholi D.* Application of Protégé, SWRL and SQWRL in fuzzy ontology-based menu recommendation. / D Hatta Fudholi, N. Maneerat, R. Varakulsiripunth. - ISPACS 2009. - DOI: 10.1109/ISPACS.2009.5383759.
- [8] *Черняховская, Л.Р.* Поддержка принятия решений для оценки качества предоставления государственных услуг с применением онтологического подхода. / Л.Р. Черняховская, В.Е. Гвоздев, А.Ф. Галиуллина // Онтология проектирования. – 2016. – Т.6, №2(20). – С. 193-204.
- [9] *Змитрович, А.И.* Интеллектуальные информационные системы. / А.И. Змитрович. – Минск: Тетра Системс, 1997. – 367 с.
- [10] *Форсайд, Р.* Экспертные системы. Принципы работы и примеры / Р. Форсайд // пер. с англ. – Москва: Радио и связь, 1987. – 221 с.
- [11] *O'Connor M.* SQWRL: a query language for OWL. / M. O'Connor, A. Das // Proceedings of the 6th International Conference on OWL: Experiences and Directions (OWLED-2009). - 2009. -Vol. 529. - P. 208–215.

THE USE OF ONTOLOGY APPROACH FOR AIRCRAFT DISPLAY DESIGN

Y.K. Gorelov¹, S.K. Kiselev²

Ulyanovsk State Technical University, Ulyanovsk, Russia

¹gur-ok@yandex.ru, ²ksk@ulstu.ru

Abstract

The authors suggest an ontological approach to the creation of expert decision-making support system for aircraft displays' schematic conceptual design. Protégé, an open-source ontology design software, is used to develop display ontology. An implementation of fuzzy logic in display of ontology using SWRL (Semantic Web Rule Language) is shown. Structure of relationships between objects in fuzzy ontology is presented schematically. SQWRL (Semantic Query Web Rule Language) is used to perform a query in display design recommendation. As an example, a recommendation feature to give display backlight design recommendation by few factors, such as display diagonal and brightness level, is developed using fuzzy ontology. Membership functions are defined in Protégé editor by putting the property values along with the property. The defuzzification process is based on the set of "IF-TO" fuzzy rules that are used for the calculation of total recommendation crisp value. SWRL built-in functions are used to simulate the calculation process. An implementation of expert system recommendation is illustrated in the aircraft display design projects, in particular, the design of backlight module is given. Using an ontology with fuzzy logic in aircraft display design is new.

Keywords: ontology, knowledge, aircraft display, expert system, fuzzy logic.

Citation: Gorelov YK, Kiselev SK. The use of ontology approach for aircraft display design [In Russian]. *Ontology of designing*. 2018; 8(3): 400-411. DOI: 10.18287/2223-9537-2018-8-3-400-411.

References

- [1] **Dobrov BV, Ivanov VV, Lukashovich NV, Solovyev VD.** *Ontology and Thesaurus: Models, Tools, Applications*. [In Russian]. Moscow: Binom; 2009. – 173 p.
- [2] **Shustova DV.** Approach to developing a semantic basis of information systems for aircraft systems design and production. [In Russian]. *Ontology of designing*. 2015; 5(1): 70-84.
- [3] **Gavrilova TA, Horoshevskiy VF.** *Intellectual Systems Databases*. [In Russian]. St. Petersburg: Piter; 2001.
- [4] **Korneev VV, Gareev AF, Vasyutin SV, Reich WW.** *Databases. Intelligent processing of information*. [In Russian]. Moscow: Nolidg; 2000. – 352 p.
- [5] **Calegari S, Ciucci D.** Integrating Fuzzy Logic in Ontologies. In *Proceedings of the Eighth International Conference on Enterprise Information Systems - 2006*. Volume 2: ICEIS, - P.66-73. DOI: 10.5220/0002496100660073.
- [6] **Valkman IR, Hala CA.** Ontology design based on the text content with fuzzy logic for legal domain. [In Russian]. *Ontology of designing*. 2014; 2(12): 19-39.
- [7] **Hatta Fudholi D, Maneerat N, Varakulsiripunth R.** Application of Protégé, SWRL and SQWRL in fuzzy ontology-based menu recommendation. *ISPACS*; 2009. - DOI: 10.1109/ISPACS.2009.5383759.
- [8] **Chernyakhovskaya LR., Gvozdev VE, Galiullina AF.** Decision support for assessing of public services provision quality using ontological analysis [In Russian]. *Ontology of designing*. 2016; 6(2): 193-204. - DOI: 10.18287/2223-9537-2016-6-2-193-204.
- [9] **Zmitrovich AI.** *Intelligent information systems* [In Russian]. Minsk: Tetra Systems; 1997.
- [10] **Forsyth R.** *Expert systems. Principles and case studies*. [In Russian]. Moscow: Radio I Svyaz; 1987.
- [11] **O'Connor M., Das A.** SQWRL: a query language for OWL // *Proceedings of the 6th International Conference on OWL: Experiences and Directions (OWLED-2009)*. 2009; 529: 208–215.

Сведения об авторах



Горелов Юрий Константинович, 1988 г. рождения. Окончил факультет информационных систем и технологий Ульяновского государственного технического университета (УлГТУ). Аспирант кафедры «Измерительно-вычислительные комплексы» УлГТУ. Инженер-конструктор АО «Ульяновское конструкторское бюро приборостроения». Работает в области проектирования бортовых систем электронной индикации.

Yuriy Konstantinovich Gorelov (b. 1988) graduated from the Faculty of Information Systems and Technologies of Ulyanovsk State Technical University; Postgraduate Student of the Department of Measuring and Computing Systems of Ulyanovsk State Technical University; Design-Engineer of the Ulyanovsk Instrument Manufacturing Design Bureau; works in designing of aircraft electronic display systems.

Киселев Сергей Константинович, 1966 г. рождения. Доктор технических наук, окончил Ульяновский государственный политехнический институт по специальности «Авиаприборостроение». Профессор и заведующий кафедрой «Измерительно-вычислительные комплексы» УлГТУ. Имеет монографии, статьи, патенты в области приборостроения.

Sergei Konstantinovich Kiselev (b. 1966) Doctor of Engineering; graduated from Ulyanovsk Polytechnic Institute with a specialty in Aircraft Instruments Engineering; Professor and Head of the Department of Measuring and Computing Systems of Ulyanovsk State Technical University; an author of monographs, articles, and patents in the field of instrument engineering.



УДК 004:81'37

СЕМАНТИЧЕСКОЕ ЯДРО ЦИФРОВОЙ ПЛАТФОРМЫ

Н.В. Максимов^{1,a}, О.Л. Голицына^{1,b}, М.Г. Ганченкова^{1,c}, Д.В. Санатов^{2,3,d}, А.В. Разумов^{3,e}

¹ НИЯУ МИФИ, Высшая инженеринговая школа, Москва, Россия

² ООО «Медиа-Лаб», Москва, Россия

³ Фонд «Центр стратегических разработок «Северо-Запад», Санкт-Петербург, Россия

^anv-maks@yandex.ru, ^bolgolitsina@yandex.ru, ^cMGGanchenkova@mephi.ru,

^ddmsa@yandex.ru, ^ea.razumov@csr-nw.ru

Аннотация

Дано обобщённое описание интегрированного комплекса декларативных и процедурных средств, в совокупности обеспечивающих согласованное информационное (семантическое) представление сложных объектов на всех этапах их жизненного цикла. Знания (факты) представляются системой онтологий разного уровня, а структура - таксономиями и классификациями. Онтология определяется как система трёх взаимосвязанных систем (функциональной, понятийной и знаковой), а система таксономий представляет классы объектов и процессов, характерных для основных «координат» деятельности. Информационный поиск, как компонент семантического ядра, рассматривается как сложный самосогласованный процесс конструирования нового знания, где знание – это информация (тексты находимых документов), связываемая с контекстом задачи и представлениями пользователя. Такой контекст целенаправленно или косвенно задаётся пользователем посредством предопределённых семантических структур (таксономий, онтологий) либо посредством динамически формируемых компонентов (словников, выборок и т.д.). Это составляет существо семантического когнитивного поиска, когда система не только реализует отбор документов традиционными методами поиска, но и формирует образ информационной потребности, что, в свою очередь, позволит системе синтезировать комплексные, аспектно-ориентированные ответы. Предлагаются автоматизированные технологии поддержки лингвистического обеспечения, основанные на дистрибутивно-статистическом анализе как потоков объектного знания и неявного знания (извлекаемого системой в процессе взаимодействия), так и компонентов понятийно-терминологических систем. Представленные в статье средства апробированы в рамках разработанного программного комплекса xIRBIS-ML, предназначенного для организации семантического поиска в массивах данных сложных инженерных объектов.

Ключевые слова: семантическое ядро, цифровые платформы, семантический поиск, онтологии.

Цитирование: Максимов, Н.В. Семантическое ядро цифровой платформы / Н.В. Максимов, О.Л. Голицына, М.Г. Ганченкова, Д.В. Санатов, А.В. Разумов // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.412-426. – DOI: 10.18287/2223-9537-2018-8-3-412-426.

Введение

Согласно государственной программе «Цифровая экономика Российской Федерации» [1] эффективное развитие рынков и отраслей (сфер деятельности) в цифровой экономике возможно только при наличии развитых платформ, технологий, институциональной и инфраструктурной сред.

Для сложных технических и социальных систем, к которым относятся и указанные в упомянутой государственной программе цифровые платформы (ЦП), характерно то, что они состоят из большого числа неоднородных элементов и подсистем, которые обладают значительным разнообразием внутренних и внешних связей и могут находиться во множестве состояний, в том числе в процессах становления и эволюции. Поэтому важнейшей подзадачей задачи управления жизненным циклом (ЖЦ) сложных систем является, с одной стороны,

развитие Интернета вещей, а с другой - управление знаниями на всех этапах ЖЦ этих систем. Именно последнее обеспечит возможность саморазвития сложных систем, синтетически включающих компоненты уровня вещей и уровня знаний. То есть управление знаниями на базе общей информационной ЦП в течение всего ЖЦ – это то, что обеспечивает единый, интегрированный подход к созданию, сбору, организации, распределению знаний, а в итоге приведёт к эффективному взаимодействию и использованию информационных и технологических ресурсов.

Рост актуальности вопросов развития ЦП, трансформация рынков проектирования и эксплуатации сложных инженерных объектов в таких отраслях, как атомная промышленность (где в настоящее время идёт активный поиск новых технологий управления знаниями), авиационное судостроение, энергетика, существенно повысил актуальность тематики семантического поиска и сформировал запрос на соответствующие промышленные технологии.

Настоящая статья представляет собой обобщённое описание семантического ядра - комплекса декларативных, процедурных и информационных средств, которые в совокупности могут обеспечить согласованное семантическое представление сложных объектов, к множеству которых относятся и системы представления и управления знаниями. Разработанные¹ и использованные подходы и решения представлены кратко, в объёме, который необходим для указания места и роли компонентов.

1 Цифровая платформа и семантическое ядро

ЦП – система алгоритмизированных взаимоотношений значимого количества участников рынка, объединённых единой информационной средой, приводящая к снижению транзакционных издержек за счёт применения пакета цифровых технологий и изменения системы разделения труда [1]. Другими словами, ЦП – это площадка, поддерживающая комплекс автоматизированных процессов и согласованное использование широкого спектра цифровых продуктов (услуг) значительным количеством потребителей.

Одним из вопросов развития ЦП является взаимодействие технологических платформ между собой и взаимодействие ЦП с экосистемой. Основой такого взаимодействия являются декларативные и процедурные средства, обеспечивающие интероперабельность информационных систем (ИС), а также стандартизация информационных технологий (ИТ). При этом, согласно [2], одним из ключевых условий интероперабельности современных платформ является наличие семантического слоя организации данных.

Функциональность семантического слоя определяется характеристиками семантического ядра, которое представляет собой систему декларативных и процедурных средств, обеспечивающих согласованную автоматическую и автоматизированную (в том числе и интерактивную) идентификацию, поиск и анализ информации и знаний.

Семантический слой организации данных, по сути являющийся субплатформой, – это синтетическое соединение трёх компонентов:

- документальных информационно-поисковых систем и баз данных, обеспечивающих углублённый семантический поиск и анализ разнородной информации;
- человеко-машинных информационных интерфейсов, обеспечивающих персонализируемое представление познавательной траектории пользователя;

¹Работа основана на результатах научных исследований и практических разработок в области информационного поиска и управления знаниями, проводимых коллективом под руководством проф. Н.В. Максимова в течение более двух десятилетий. В ней также учтён опыт разработки и внедрения Информационно-аналитической системы IRBIS (©1992-2018), используемой рядом ведущих информационных центров и организаций для создания промышленных документальных баз данных.

- лингвистического обеспечения, построенного на базе гибридных методов лингвистического, лингво-статистического и структурного анализа текстов, обеспечивающих построение адекватных смысловому содержанию документов их формализованных семантических образов, над которыми определены операции, в том числе, корреспондирующие с процессами познания.

Это триединство рассматривается в контексте общего процесса синтеза, представления и поиска знаний.

Для современного состояния рынка ИТ-продуктов, ориентированных на работу с семантикой (цифровой инжиниринг, управление знаниями, семантическая обработка текстов) характерно то, что эти системы, как правило, создаются либо как закрытые проприетарные промышленные продукты, либо как узкоспециализированные решения, либо как академические проекты, что накладывает существенные ограничения на интероперабельность и, как следствие, на их широкое или совместное использование. Например, для полноценной цифровизации проектов атомной энергетики необходимо в части семантики совместить решения систем проектирования от Intergraf, Siemens, Dassault Systems, информационные системы, ориентированные на сохранение и управление знаниями, как например, Temelin knowledge management system [3] или система сохранения знаний по быстрым реакторам [4], а также функциональные технологии, ориентированные на обработку текстов и извлечение знаний, как АВВУ Compreno [5] или RCO Fact Extractor [6], технологии OSTIS [7].

Особенно острой эта проблема становится в условиях смещения границы отраслевых рынков и связанных с этим смещений границ семантического описания предметных областей (ПрО). Необходимо создание семантической инфраструктуры, способной интегрировать семантические данные из разных отраслей промышленности и областей знаний, а значит быть открытой для использования различными компаниями и способной интегрировать сторонних поставщиков семантического программного обеспечения.

Другая проблема, преодоление которой обеспечит выход ЦП на новый уровень, - высокая динамика изменения данных и связанных с ними метаданных. То есть семантическое ядро ЦП должно обладать свойствами, обеспечивающими саморазвитие. В частности, в контексте задач семантического поиска и управления знаниями система должна идентифицировать (индексировать) информацию «на лету» в зависимости от специфики задачи пользователя, отражаемой в когнитивных инфраструктурах.

2 Основные принципы разработки и функционирования ЦП

Семантическая платформа xIRBIS-ML – ориентированная на управление знаниями информационно-аналитическая система, разрабатываемой коллективом специалистов НИЯУ МИФИ, Фонда «Центр стратегических разработок «Северо-Запад» и ООО «Медиа-Лаб». Семантическое ядро ЦП xIRBIS-ML проектируется как система, обладающая свойствами функциональной полноты по отношению к среде, включая пользователей, а с точки зрения наполнения – как способная гармонично представлять знания разных отраслей и на разных уровнях общности/детализации. Основными принципами разработки и функционирования являются:

- взаимодополнительность процессов анализа/синтеза восходящих/нисходящих информационных потоков (знаний/информационных потребностей);
- интеграция функций и основных элементов информационного сопровождения этапов цикла генерации-использования знаний/данных на основе модели информационных представлений, основанной на принципах общей теории систем;

- полнота и избирательность информационного поиска, обеспечиваемые системой механизмов поиска и поиском в ассоциированных внешних информационных ресурсах (ИР);
- динамическая синхронизация взаимосвязи информационных и метаинформационных компонентов, основанная на общесистемной модели информационных представлений когнитивных процессов и управления;
- технологии оперативного построения и анализа лингвистического обеспечения (словников, рубрикаторов, тезаурусов, онтологий ПрО).

3 Интерактивный семантический поиск

Одной из основных и сложных задач семантики является смысловое отождествление содержания документов поисковым запросам. Такие подходы основываются на тезисе «смысл слова определяется его окружением» и, по существу, реализуют выявление контекста и, таким образом - определение (выбор) смысла слова.

При этом, говоря о семантическом поиске, необходимо понимать две особенности, которые важны для такого рода систем.

Первая – это то, что слово (словосочетание, выражение и даже весь текст) из найденного документа будет восприниматься в контексте *пользователя*. То есть смысл будет формироваться преимущественно на основе ПрО решаемой пользователем задачи и зависеть от полноты и специфики представления ПрО. При этом контекст ПрО, а именно общепринятые понятия и отношения между ними, может быть взят из базы знаний в общем случае интегрированной онтологии ПрО. Контекст пользовательского представления может быть сформирован динамически в процессе поиска путём построения виртуальной онтологии задачи из базовых онтологий и онтологических описаний отдельных решений [8]. Это помогает учитывать знания, которые в явном виде не присутствуют в конкретном высказывании/запросе, но существенно влияют на его смысл.

Вторая - это то, что пользователь «выстраивает» образ искомого решения, извлекая из текстов смысловые (и текстовые) фрагменты и связывая их таким образом, чтобы получаемая понятийная конструкция была непротиворечивой и обеспечивала бы решение его проблемы.

Технологической основой реализованной в xIRBIS-ML среды информационного взаимодействия «пользователь-система» является поисковый интерфейс, представляющий обобщённое рабочее пространство пользователя, ориентированное на процессы генерации знаний. Принципиально рабочее пространство включает две составляющие: собственно интерфейс формирования/развития запроса и обработки выдачи, а также структуру систематизации знаний пользователя – так называемый когнитивный рубрикатор (КР).

Поисковый интерфейс, помимо фактографического, тематического, семантического поиска с использованием чётких и нечётких механизмов отбора, вербальных или гипертекстовых технологий, позволяет пользователю осуществлять комплексный поиск (мультиобъектный, многоэтапный), например, для задач мониторинга проекта. Система, формируя поисковые образы и выдачи, готовит альтернативы, а структуры систематизации, протоколирующие поиск и идентифицирующие результаты, - задают направления «предпочтительного» развития. Эта задача решается как традиционными механизмами поиска, так и за счёт интерактивных средств визуального пространства, процедурно связывающих операционные объекты разного типа. В последнем случае поисковая траектория реализуется путём перемещения по ассоциированным визуальным элементам в находимых документах. Элементами могут быть слова или фрагменты текста, выделяемые по критерию соответствия понятиям, присутствующим в запросе или в профиле пользователя, а также динамически генерируемые гипертек-

стовые ссылки. Движущей силой, инициирующей очередные шаги поиска, является проблемная ситуация – информационная неопределённость, осознаваемая пользователем, а также дисбаланс в наполнении тематических подразделов проблемы. Автоматическое «отслеживание» таких дисбалансов обеспечивается использованием в качестве интегральной информационной структуры КР, включающего как информационные (документы, запросы, ссылки на ассоциированные ресурсы и т.п.) и метаинформационные (словари ПрО, классификации, рубрикаторы, тезаурусы, онтологии) компоненты, так и результаты аналитической обработки [9]. То есть поисковая система, помимо собственно отбора документов по заданному пользователем запросу, может способствовать выявлению предмета поиска (технологии извлечения и идентификации неявных знаний), а также обеспечить согласование на концептуальном и лингвистическом уровнях представления предмета поиска на стороне системы и пользователя.

Такая структура, соединяющая интенциональное и экстенциональное начала процесса познания отдельного субъекта и при этом представленная в распределённой сетевой среде, позволит осуществлять управляемый мониторинг как документальной, так и понятийно-терминологической составляющей. Это по сути является реализацией системного подхода и даёт возможность видеть в явной форме новые характеристические признаки, определять способы выделения подсистем и на основе свойств соответствия и симметрии обнаруживать связи (в т.ч. и противоречия) с другими системами классификации. Пользователь, осуществляющий поиск интересующей его информации, получает «многослойную» картину ПрО, что в итоге позволит ему систематизировать найденное и сделать более обоснованный выбор «траектории» поиска, учитывающий не только его представление о предмете поиска, но и сопоставительные оценки состояния и тенденций ПрО.

4 Информационные и лингвистические компоненты

Блок ЦП, практически представляющий собственно семантику в реализации xIRBIS-ML, содержит два типа логически взаимосвязанных компонентов.

Первый тип – это собственно «рабочая» информация, использование которой в сфере основной деятельности обеспечивает воспроизводство целевого продукта по его описанию (документации, схеме и т.п.). Это т.н. задокументированные знания (зафиксированные на носителе определённым способом), включая фактографические (таблицы данных, свойств и т.д.) и документальные виды информации (статьи, монографии, учебные пособия и т.д.), представляющие теории, гипотезы, эксперименты, критический опыт.

Второй тип – это информация справочного характера, обеспечивающая, в основном, представление и нахождение целевой (рабочей) информации. Сюда относятся:

- понятийно-терминологические системы (словари, тезаурусы, онтологии, глоссарии), являющиеся инструментами познания и средствами фиксации знаний на носителях;
- классификационные схемы (таксономии, рубрикаторы и т.д.), обеспечивающие единообразие «членения» ПрО, исходя из целевых, организационных или методологических представлений.

Перечисленные и другие виды и формы вторичной информации представляют объективно-исторически и технологически сложившийся ряд [10] функционально-ориентированных инструментальных форм представления существа (семантики) информационных единиц с той или иной полнотой и точностью [11]. При этом с точки зрения характера источника смысла – определений, гипотез, теорий, базовых методов, частных решений и т.п. – различаются онтологии задач, решений, онтологии физических свойств и единиц измерения и т.д.

Кроме того, существуют и развиваются онтологии терминосистем, как, например, проекты WordNet [12], PyТез [13], FrameNet [14].

Практически построение онтологий осуществляется на базе следующих трёх блоков - источников информации:

- первый блок – фундаментальные знания, которые составляют статьи, монографии, отчёты, диссертации и т.п., представляющие объекты модельного уровня;
- второй блок – базовые компетентностные знания, в том числе учебная и методическая информация, которая представлена учебниками, пособиями, методическими указаниями, хрестоматиями и т.п., содержащими описание существа (содержания) предмета. Кроме того, в этот блок входят справочно-методические материалы – учебные программы, оценочные средства, которые определяют структуру знания и технологию познания, в том числе опорные и контрольные точки;
- третий блок – знания практического уровня, которые представлены проектной, конструкторской, технологической, нормативной документацией, экономическими, экологическими, аналитическими отчётами и т.п.

При этом в практике ИС понятийная основа устойчиво фиксируется в информационно-поисковых тезаурусах – наиболее известной и технологичной упорядоченной форме понятийно-знаковых систем.

Таксономические системы через упорядоченность концептуальных (классов) и реальных (экземпляров) составляющих, а также признаков классификации (существенных свойств) представляют целостность семантического пространства. В свою очередь онтологии различных уровней дискретно представляют «фрагменты» ПрО через ситуативные связи экземпляров объектов (знаний) различного уровня.

Роли лингво-семантических компонентов определяются исходя из того, что основными пространствами жизни объекта являются:

- пространство основной деятельности субъекта, где по форме существования выделяются два подпространства – абстрактных объектов (концептуальные модели, теории, составляющие предмет деятельности) и конкретных объектов (физические объекты ЖЦ);
- пространство информационной деятельности субъекта, предопределяющее способы и формы представления объекта в виде информационных сообщений;
- время как фактор, обуславливающий изменение знания и условий его применения, и как свойство, позволяющее фиксировать знание в виде дискретных макрообъектов.

Таким образом, для структурно-систематизированного представления состоявшегося (сгенерированного, систематизированного, проверенного, задокументированного) знания используется следующая «сетка координат»:

- координата «объект», задаваемая структурной таксономией, представляющей составные части объекта (узлы, детали, технологии и т.п.) с точки зрения совокупного процесса ЖЦ;
- координата «предмет», задаваемая функциональной таксономией (как «абстрактная модель ПрО») – структурой, представляющей теоретические и иные знания, относящиеся к этапам ЖЦ изделия;
- координата этапов работ (как фактор, отражающий разделение работ, предопределяемое специализацией субъекта, в процессах ЖЦ), задаваемая таксономией стадий и этапов ЖЦ и так или иначе связанной с ней таксономией форм представления знания – типов и видов документов как специфических форм и способов описания объекта.

В качестве «координат» можно использовать и любые другие системы классификации. Например, ГРНТИ, УДК, патентные и др. классификации. Кроме того, объект может идентифицироваться также и ключевыми словами, а глубинные семантические связи могут отражаться онтологиями.

При этом, подобно соотношению практической и модельной составляющих в целенаправленной деятельности в информационной деятельности выделяются ситуационная (на вторичном уровне наиболее полно представлена онтологиями) и структурно-концептуальная (на вторичном уровне представлена таксономиями) составляющие. Первая отражает комбинаторную природу деятельности и языка, позволяющую выражать (обозначать) новые смыслы «старыми» знаками. Вторая представляет базис, который обеспечивает преемственность и «вектор» развития ПрО, а в части процессов информационных коммуникаций является основой для «узнавания» нового за счёт апелляций к общим смыслам.

С точки зрения идентификации информации и знаний важным и перспективным свойством онтологического представления семантики является то, что, в отличие от линейного поискового образа в нём представлены ситуации (факт имманентной или ситуативной взаимосвязи двух объектов).

Онтология с точки зрения общей теории систем определена в [15] как совокупность трёх взаимосвязанных систем:

$$O = \langle S_f, S_c, S_t \rangle,$$

где S_f - функциональная система (объекты и связи действительности) определяется как

$$S_f = \langle M_f, A_f, R_f, Z_f \rangle,$$

где M_f множество объектов, A_f множество характеристических свойств, R_f – множество функциональных отношений, Z_f закон композиции, т.е., правил и схем упорядочения объектов (таксономия ПрО);

S_c - понятийная система, определённая как

$$S_c = \langle M_c, A_c, R_c, Z_c \rangle,$$

где M_c – множество понятий ПрО, A_c – множество признаков систематизации понятий (мерономия), R_c – классы/подклассы парадигматических отношений, Z_c – закон композиции (схема упорядочения);

S_t - терминологическая система, определённая как

$$S_t = \langle M_t, A_t, R_t, Z_t \rangle,$$

где M_t - множество терминов, A_t – множество свойств, R_t – множество отношений эквивалентности и включения, а также лингвистических отношений, Z_t – закон композиции (грамматика);

\equiv - операция сопоставления элементов различных систем на уровне знаков, обеспечивающая их тождество в функциональной, понятийной и терминологической системах.

Автоматическое построение онтологий на основе представленных на естественном языке текстов, т.е. формирование связей между выделенными в тексте понятиями, основывается на преобразовании лингвистических отношений в функциональные. При этом набор типовых, так называемых функциональных, связей для ПрО обычно ограничен (подробно см. [16, 17]).

Автоматически построенные понятийно-терминологические графы могут быть отредактированы средствами интерактивных человеко-машинных процедур, которые в этом случае реализуют принцип дополнительности: представление знания в виде извлекаемых из текста ключевых слов и отношений система осуществляет с «устоявшейся» точки зрения (статистической значимости), а человек, внося изменения и дополнения в построенный системой образ (список терминов, граф), фиксирует отличия, характеризующие новизну и специфику по отношению к устоявшемуся и усредненному представлению знаний.

Однако при практическом построении таких объектов возникают существенные сложности. Богатство лексики естественного языка приводит к тому, что применение к построенным по текстам онтологиям теоретико-графовых операций (например, объединение или пересечение) для определения их семантической близости не даёт результата: понятия (точнее, обозначающие их знаки), выделенные в сопоставляемых текстах, не пересекаются. Определённый выход из этой ситуации заключается в приведении лексики построенных онтологий к лексике согласованной понятийно-терминологической структуры (например, тезауруса).

Представление систем, входящих в определение онтологии, в виде графов позволит использовать при формализации выполнения операций над онтологиями аксиомы теоретико-графовых операций. В качестве основных операций используются бинарные операции объединения и пересечения и унарные - построения аспектного представления и масштабирования онтологий, с помощью которых можно, в том числе, синтезировать новые онтологии (так называемые прикладные онтологии), отражающие ПрО в заданном аспекте. Отличительной особенностью реализации операций над онтологиями является возможность использования для повышения семантической связности структур понятийного и терминологического уровней [18].

На рисунке 1 представлен фрагмент результата операции объединения онтологий, построенных по текстам конструкторских документов ПрО «Атомная энергетика»: «Компоновка основного оборудования», «Прочность и сейсмостойкость», «Внутрикорпусные устройства», «Технические решения при модернизации реакторной установки», «Источники излучения». Общие узлы, входящие в нескольких онтологий функционального уровня, выделены на рисунке темно-серым фоном, а узлы, соответствующие понятийной системе – серым. Тезаурусные связи обозначены: *RT* (Related Term) – связь с ассоциативным дескриптором, *BT* (Broader Term) – связь с вышестоящим дескриптором, *NT* (Narrower Term) – связь с нижестоящим дескриптором. В результате использования в качестве общего понятийного базиса тезауруса INIS МАГАТЭ [19] удалось повысить семантическую связность объединения путём добавления тезаурусных маршрутов.

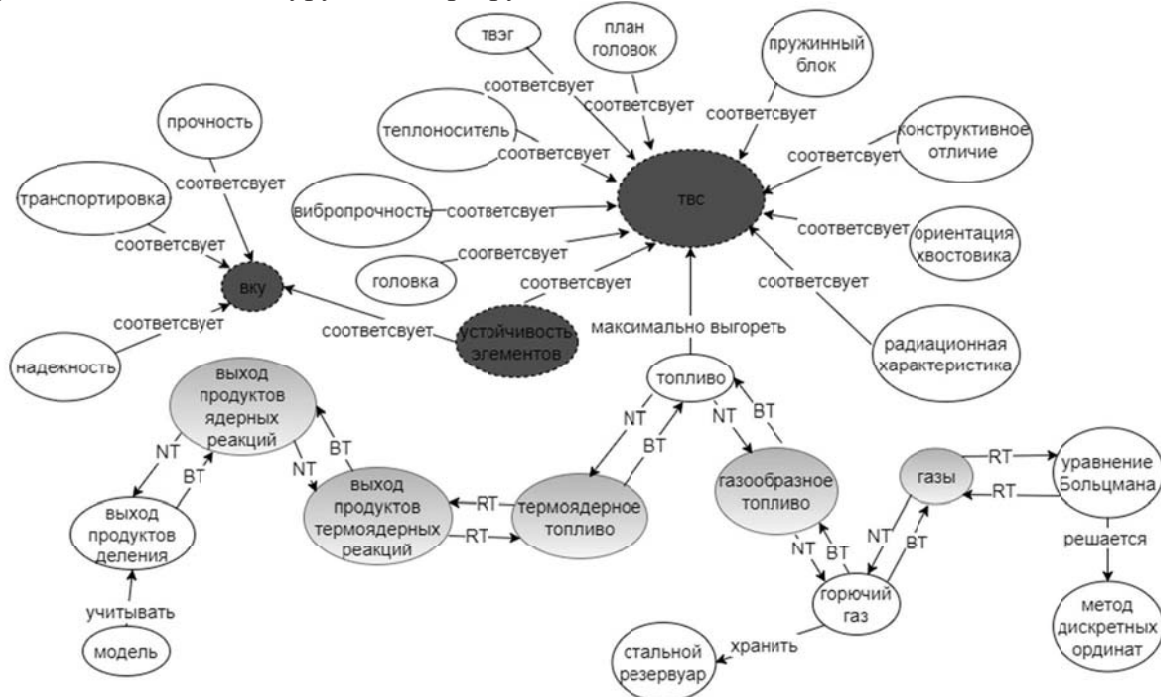


Рисунок 1 – Фрагмент графа - результата операции объединения онтологий

Следует отметить, что при построении онтологии не всегда удаётся передать контекст конкретного термина, если на структурном уровне этот термин (как отдельная вершина) не будет связан доопределяющим термином. Для преодоления подобных проблем предлагается использовать взвешенный ориентированный метаграф

$$MG_f = \langle V_f, V_f^M, X_f \rangle,$$

где V_f - множество вершин (терминов); V_f^M - множество метавершин, каждая из которых в свою очередь может быть метаграфом; X_f - множество дуг (отношений), определённых на множестве $V_f \cup V_f^M$, и $\forall x_i \in X_f: x_i = \{vb_i, ve_i, \langle tr_i, A_i \rangle\}$, где $tr_i \in TR$ (TR - множество типов отношений), A_i - множество характеристических атрибутов отношений, соответствующих дугам.

На рисунке 2 приведён пример метаграфа, построенного для следующего фрагмента текста: «Клапаны на вертикальных сосудах следует устанавливать на верхнем днище. Клапаны не допускается использовать для регулирования давления в сосуде или группе сосудов. Изготовитель обязан поставлять клапаны с паспортом и руководством по эксплуатации».

Выше отмечалось, что онтология обладает свойствами системы. Определяя системный базис - подмножества свойств и типов отношений, соответствующих аспекту, или функцию отображения на другую онтологию, можно построить подсистемы и проекции, которые сами являются системами. Это означает, что приведённое выше графовое представление онтологии обладает характерным свойством мультиграфов. Отметим также, что поскольку пара объектов ПрО может быть связана несколькими типами отношений, то граф можно классифицировать как гиперграф.

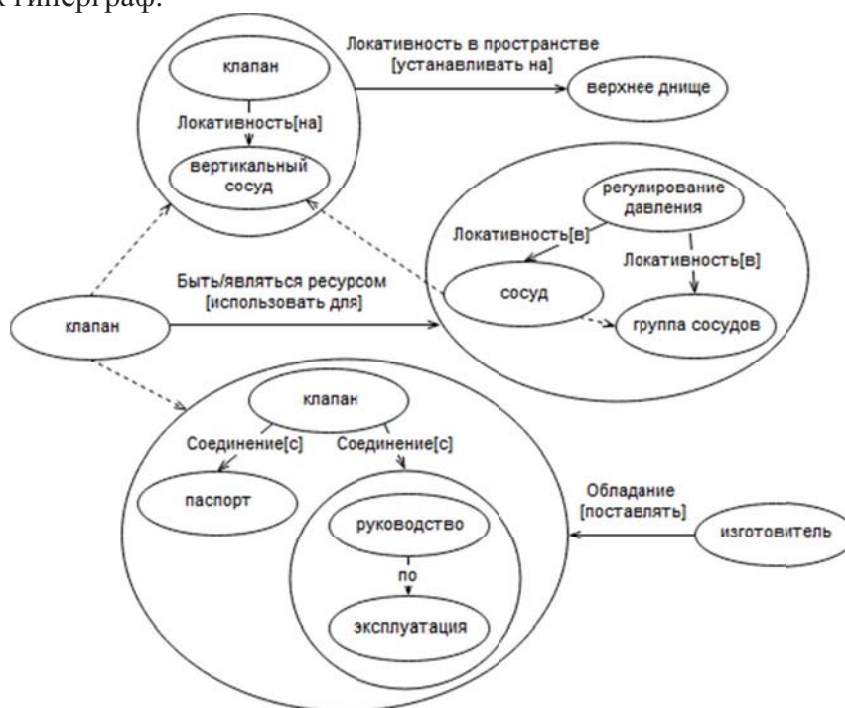


Рисунок 2 - Пример метаграфа с типизированными отношениями (в квадратных скобках приведена лингвистическая конструкция из текста, пунктирными линиями обозначены иерархические связи)

5 Архитектура и функциональные подсистемы

ИТ-среда всегда отличалась своей гетерогенностью, несмотря на многочисленные попытки её глобальной стандартизации. Современная тенденция, диктуя свои требования, за-

ставляет отвечать на подобный вызов созданием кроссплатформенных систем, способных работать с одинаковой эффективностью независимо от ИТ-инфраструктуры конечного пользователя. Одним из ответов на подобное требование является агентный принцип и наличие развитого программного интерфейса, позволяющего гибко и эффективно подстраиваться под текущие пользовательские тренды и ставшего неотъемлемой частью сложных систем. В частности, платформа xIRBIS-ML позволяет интегрировать лингвистические компоненты нескольких производителей.

Основными отличиями xIRBIS-ML являются:

- семантическая интеграция информации, относящейся к разным видам деятельности и всем этапам ЖЦ, при которой информационные объекты рассматриваются как имеющие двойное назначение: порождаются и используются в целевой деятельности пользователя и являются метаинформацией в процессах информационного обеспечения этой деятельности;
- ориентированное на задачи когнитивного поиска многоаспектное индексирование разнородной информации, в том числе с использованием взаимодополняющих технологий лингвистических процессоров и эвристических методов, обеспечивающих глубокое семантическое индексирование и последующее выделение аспектов (динамическое индексирование);
- комплексное использование разнотипных методов и моделей отбора документов;
- гибкий поисковый интерфейс, ориентированный на задачи анализа и синтеза информации;
- объединённая объектно-ориентированная модель, совмещающая как данные информационных процессов, так и метаданные, определяющие параметры обработки.

5.1 Основные подсистемы xIRBIS-ML

Подсистема сбора и обработки документов обеспечивает импорт документов с преобразованием в унифицированный хорошо структурированный формат путём выделения структурных элементов, а также многоаспектное индексирование содержания, в том числе с возможностью интерактивного редактирования автоматически приписываемых классификационных кодов, списков ключевых слов и онтологий. Построение многоаспектного понятийного образа документа производится путём представления содержания в виде гиперграфа извлекаемых из текста понятий и типизируемых отношений, что позволяет не только динамически формировать онтологию знаний и выявлять новые понятия и связи, но также перейти к графически управляемым навигационным технологиям концептуального поиска. В итоге это обеспечивает более точный поиск, а интерактивные технологии индексирования – привлечение экспертных знаний для актуализации семантических компонентов.

Подсистема информационного поиска в распределённых ресурсах, помимо классических механизмов поиска по чётким/нечётким критериям и с реформулированием запроса по обратной связи, также обеспечивает переадресацию и адаптацию запроса для проведения поиска во внешних Интернет-ресурсах с учётом их особенностей, в том числе синтаксиса поисковых языков, что обеспечит более точный индивидуализируемый отбор информации.

Подсистема статистического анализа документальных потоков и лексики обеспечивает формирование распределений различных информационных срезов, а также построение и комплексный анализ временных рядов профилированных потоков документов и лексики. Это позволяет средствами *комплексного* дистрибутивно-статистического анализа потоков документов и запросов выявлять тенденции, источники и взаимозависимости, свойственные ПрО. Для отображения и анализа используются компоненты деловой графики.

Подсистема анализа и ведения лингвистического обеспечения ориентирована на поддержку пользовательского информационного пространства и обеспечивает построение и ведение иерархических терминологических (словарных и классификационных) структур ПрО, которые могут быть использованы в качестве средств систематизации ПрО, а также для автоматической классификации документов. Динамическое обновление состава и связей терминов основано на лингвистическом и статистическом анализе текстов документов и баз данных.

Подсистема мониторинга хода выполнения и анализа содержания проектов реализует: мониторинг показателей по всем этапам ЖЦ проекта; статистический анализ и систематизацию информацию о проекте и ассоциированных ПрО в локальной БД, а также в других ИР; автоматическое формирование сводных статистических портретов основных действующих лиц и т.п.

5.2 Объектная модель xIRBIS-ML

Основу интеграции информационной среды и управления поисковой навигацией xIRBIS-ML составляет объектная модель информационной среды, включающая расширенную объектную модель документа. Структура среды определяется трёхзвенностью совокупной ИС «пользователь – автоматизированная ИС (АИС) – ИР», где АИС, как промежуточное звено, должна обеспечивать согласование сторон – пользователя и ИР – каждая из которых имеет свою специфику организации и поведения.

Объектная модель трёхзвенной информационной среды специфицирует взаимодействие рабочего пространства пользователя (объект КР) и пользовательского интерфейса системы (задаваемого матрицей взаимосвязи функций поиска и обработки с интерфейсными операционными объектами).

Информационный ресурс, как объект рабочего пространства, характеризуется тремя группами параметров, определяющими:

- собственно ресурс (идентификация, адрес, метод доступа, тип);
- содержание ИР (тематика и характер ресурса, а также метаданные об элементах данных, документах и справочных ресурсах);
- взаимодействие с ИР (поисковый вход и метод; синтаксис поискового языка; метод доступа к содержанию).

Базовым объектом является документ, характеризующийся формой представления (так называемой схемой документа), которых может быть несколько. Документ представляет собой иерархию элементов данных, для каждого из которых специфицируются: идентификатор(ы), тип, метод(ы) преобразования на входе/выходе, метод(ы) индексирования.

Заключение

Семантическое ядро, назначением которого является обеспечение семантически согласованной идентификации, поиска и анализа информации и знаний, в итоге должно обеспечивать «содержательное» отождествление текстов, сходных по смыслу, но вариантно представленных (описаний объекта в разных аспектах, разными словами, в разных формах и т.п.). В рамках сформулированных выше предложений это достигается использованием системного подхода: состоявшееся знание (факты) представлено онтологиями разного уровня, а его структура - таксономиями и классификациями. Причём онтология – это, в соответствии с семиотической моделью, система трёх взаимосвязанных систем (функциональной, понятийной и знаковой), а система таксономий представляет классы объектов и процессов, характерных для основных «координат» деятельности. Исходя из того, что новое знание всегда связано с

известным, с помощью введённых операций над онтологиями становится возможным приведение описаний (их онтологических образов) к сопоставимым формам. Процесс конструирования нового знания, где знание – это информация, связанная с контекстом, реализуется средствами информационного поиска как компонента семантического ядра.

В рамках предложенных решений контекст явно или неявно определяется пользователем с помощью либо семантических структур (таксономий, онтологий ПрО), либо динамических компонентов (словников, КР), что позволяет говорить о полноценном семантическом поиске, когда система не только реализует более или менее интеллектуальный отбор документов, но и формирует выражения самой информационной потребности (в терминах, понятных системе), что позволит синтезировать *комплексные*, аспектно-ориентированные ответы.

Изложенные положения и решения семантического ядра в той или иной степени были использованы при разработке ряда опытных образцов и промышленных систем.

Список источников

- [1] Государственная программа «Цифровая экономика Российской Федерации». Утверждена распоряжением Правительства Российской Федерации от 28 июля 2017 г. № 1632-р.
- [2] ГОСТ Р 55062-2012 Информационные технологии (ИТ). Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения. Утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 13 ноября 2012 г. № 751-ст. Дата введения 2013-09-01.
- [3] *Kostiha, F.* Knowledge management at Czech nuclear power plants / F. Kostiha // *Bezpecnost Jaderne Energie*, 2010, 18(7-8), P.203-209.
- [4] *Pryakhin, A.* Fast reactor knowledge inventory / A. Pryakhin, A. Stanculescu, Y. Yanev // *International Journal of Nuclear Knowledge Management*, 2009, 3.2, P.199-209.
- [5] Технология анализа и понимания текстов на естественном языке ABBYY Compreno. - <https://www.abbyy.com/ru-ru/infoextractor/compreno/>.
- [6] Инструментарий компьютерного анализа текстовой информации RCO Fact Extractor. - <http://www.rco.ru/>
- [7] *Голенков, В.В.* Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания / В.В. Голенков, Н.А. Гулякина // *Онтология проектирования*. – 2014. - №1(11). – С. 42-64.
- [8] *Левенчук, А.* Онтология, схема/онтология инженерного проекта и схемное/онтологическое мышление. - <https://ailev.livejournal.com/1159110.html>.
- [9] *Максимов, Н.В.* Структура и компоненты операционного визуального пространства интерактивного поиска научной информации / Н.В. Максимов, О.Л. Голицына, А.Л. Усенко // *Научная визуализация*. – 2014. – Т. 6, № 4. – С. 96-106.
- [10] *McGuinness, D.L.* Ontologies Come of Age. In In D. Fensel, J. Hendler, H. Lieberman, & W. Wahlster (Eds.), *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press, 2002. - P.171-191.
- [11] *G. Van Heijst, A. Th. Schreiber, and B. J. Wielinga.* Using Explicit Ontologies in KBS development. *International Journal of Human-Computer Studies*. 1997; Vol. 46, Iss. 2-3, P.183-292.
- [12] WordNet. A Lexical Database for English. - <https://wordnet.princeton.edu/>.
- [13] *Лукашевич, Н.В.* Проектирование лингвистических онтологий для информационных систем в широких предметных областях / Н.В. Лукашевич, Б.В. Добров // *Онтология проектирования*. – 2015. - Т.5 - №1(15). – С. 47-69.
- [14] FrameNet Project. - <https://www.framenet.icsi.berkeley.edu/fndrupal/>.
- [15] *Голицына, О.Л.* Онтологический подход к идентификации информации в задачах документального поиска. / О.Л. Голицына, Н.В. Максимов, О.В. Окропишина, В.И. Строгонов // *Научно-техническая информация*. Сер. 2. – 2012. – № 5. – С.1-9.
- [16] *Golitsina O.L., Maksimov N.V., Okropishina O.V., Okropishin A.E.* Semantic Identification of Text in the Class of Tasks of Information Retrieval // *Proceedings of ICAI'14, WORLDCOMP'14, July 21-24, 2014, Las Vegas, Nevada, USA*. CRSEA Press, USA. – 2014. – Vol. I. – P.47-52. – DOI: 10.3103/s0005105512030028.
- [17] *Голицына, О.Л.* Онтологический подход к идентификации информации в задачах документального поиска: практическое применение / О.Л. Голицына, Н.В. Максимов, О.В. Окропишина, В.И. Строгонов // *Научно-техническая информация*. Сер. 2. – 2013. – № 3. – С.1-8. – DOI: 10.3103/S0005105513020027.

- [18] **Максимов, Н.В.** Методологические основы онтологического моделирования документальной информации // *Научно-техническая информация. Серия 2: – 2018. – №3. – С.6-22. – DOI: 10.3103/S0005105518020036.*
- [19] INIS/ETDE Thesaurus. - <https://inis.iaea.org/inis/m/products-services/thesaurus/index.html>.
-

SEMANTIC CORE OF DIGITAL PLATFORM

N.V. Maksimov^{1,a}, O.L. Golitsina^{1,b}, M.G.Ganchenkova^{1,c}, D.V. Sanatov^{2,3,d}, A.V. Razumov^{3,e}

¹ *High Engineering School of National Research Nuclear University MEPhI, Moscow, Russia*

² *Media-Lab LLC, Moscow, Russia*

³ *Center for Strategic Research "North-West", St. Petersburg, Russia*

^a*nv-maks@yandex.ru*, ^b*olgolitsina@yandex.ru*, ^c*MGGanchenkova@mephi.ru*, ^d*dmsa@yandex.ru*,

^e*a.razumov@csr-nw.ru*

Abstract

This article gives a generalized description of the integrated complex of declarative and procedural means, which together provide a consistent information (semantic) representation of complex objects at all stages of their life cycle. Knowledge (facts) is represented by a system of ontologies of different levels, and the structure - by taxonomies and classifications. Ontology is defined as a system of three interrelated systems (functional, conceptual and sign), and the system of taxonomies represents classes of objects and processes' characteristic of the main "coordinates" of activity. Information retrieval, as a component of the semantic core, is considered as a complex self-consistent process of constructing new knowledge, where knowledge is information (texts of found documents) associated with the context of the task and the user's views. Such context is purposefully or indirectly determined by the user by means of predefined semantic structures (taxonomies, ontologies) or by means of dynamically formed components (dictionaries, samples, etc.). This is the essence of the semantic cognitive search, where the system not only implements the selection of documents by traditional retrieval methods, but also forms an image of information needs, which, in turn, will allow the system to synthesize complex, aspect-oriented answers. Automated technologies of linguistic support based on distributive-statistical analysis of both object knowledge and implicit knowledge flows (extracted by the system in the interaction process) and components of conceptual-terminological systems are proposed. Scientific and educational potential can be widely used to create a basic component of information and linguistic support in the early stages. The tools presented in the article are tested in the framework of the developed software xIrbis-ML, designed for semantic search of data in knowledge bases for complex engineering objects.

Key words: *semantic core, digital platforms, semantic search, ontology.*

Citation: *Maksimov NV, Golitsina OL, Ganchenkova MG, Sanatov DV, Razumov AV. Semantic core of digital platform [In Russian]. Ontology of designing. 2018; 8(3): 412-426. - DOI: 10.18287/2223-9537-2018-8-3-412-426.*

References

- [1] State Program «Digital Economy of the Russian Federation» [In Russian]. Approved by Russian Federation Government Executive Order dated July 28, 2017 No. 1632-r.
- [2] GOST R 55062-2012 Industrial automation systems and integration. Interoperability. General position. [In Russian] Approved and put into action by Order of the Federal Agency for technical regulation and metrology dated November 13, 2012. N 751-st. Date of implementation 2013-09-01.
- [3] **Kostiha, F.** Knowledge management at Czech nuclear power plants. *Bezpečnost Jaderne Energie*, 2010; 18(7-8): 203-209.
- [4] **Pryakhin A, Stanculescu A, Yanev Y.** Fast reactor knowledge inventory // *International Journal of Nuclear Knowledge Management*, 2009, 3.2: 199-209.
- [5] Technology of analysis and understanding of natural language texts ABBYY Compreno. - <https://www.abbyy.com/ru-ru/infoextractor/compreno/>.
- [6] Tools for computer analysis of text information RCO Fact Extractor. - <http://www.rco.ru/>.
- [7] **Golenkov VV, Guliakina NA.** The project is an open semantic technology component of designing intelligent systems. Part 1: Principles of creation [In Russian] *Ontology of designing. – 2014; 1(11): 42-64.*

- [8] **Levenchuk A.** Ontology, scheme / ontology of engineering project and schematic / ontological thinking [In Russian]. - <https://ailev.livejournal.com/1159110.html>.
- [9] **Maksimov NV, Golitsina OL, Usenko AL.** The structure and components of the operational visual space for scientific interactive information retrieval. *Scientific visualization*. – 2014; 6(4): 72-95.
- [10] **McGuinness D.** Ontologies Come of Age. In In D. Fensel, J. Hendler, H. Lieberman, & W. Wahlster (Eds.), *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press, 2002. - P.171-191.
- [11] **G. Van Heijst, Schreiber Ath, Wielinga BJ.** Using Explicit Ontologies in KBS development. *International Journal of Human-Computer Studies*. 1997; 46(2-3): 183-292.
- [12] WordNet. A Lexical Database for English. - <https://wordnet.princeton.edu/>.
- [13] **Loukachevitch NV, Dobrov BV.** Developing linguistic ontologies in broad domains [In Russian]. *Ontology of designing*. – 2015; 5(1): 47-69.
- [14] FrameNet Project. - <https://www.framenet.icsi.berkeley.edu/fndrupal/>.
- [15] **Golitsina OL, Maksimov NV, Okropishina OV, Strogonov VI.** The ontological approach to the identification of information in tasks of document retrieval. *Automatic Documentation and Mathematical Linguistics*. – 2012; 46(3): 125-132. – DOI: 10.3103/s0005105512030028.
- [16] **Golitsina OL, Maksimov NV, Okropishina OV, Okropishin AE.** Semantic Identification of Text in the Class of Tasks of Information Retrieval // Proceedings of ICAI'14, WORLDCOMP'14, July 21-24, 2014, Las Vegas, Nevada, USA. CRSEA Press, USA. – 2014; I: 47-52. - DOI: 10.3103/s0005105512030028.
- [17] **Golitsina OL, Maksimov NV, Okropishina OV, Strogonov VI.** An ontological approach to information identification in tasks of document retrieval: A practical application. *Automatic Documentation and Mathematical Linguistics*. – 2013; 47(2): 45-51. – DOI: 10.3103/S0005105513020027.
- [18] **Maksimov NV.** The Methodological Basis of Ontological Documentary Information Modeling. *Automatic Documentation and Mathematical Linguistics*. – 2018; 52(2): 57–72. – DOI: 10.3103/S0005105518020036.
- [19] INIS/ETDE Thesaurus. - <https://inis.iaea.org/inis/m/products-services/thesaurus/index.html>.

Сведения об авторах



Максимов Николай Вениаминович в 1975 г. окончил Московский инженерно-физический институт по специальности «Прикладная математика». Д.т.н. (2002г.). Профессор кафедры системного анализа МИФИ. Область научных интересов: моделирование и разработка документальных информационно-поисковых систем и баз данных, лингвистическое обеспечение документальных информационно-поисковых систем и систем управления знаниями; человеко-машинные информационные системы, интерфейсы на основе когнитивных и поведенческих моделей. Автор более 120 научных работ, учебников и учебных пособий.

Nikolai Veniaminovich Maksimov graduated from the Moscow Engineering Physics Institute in 1975, D. Sc. Eng. (2002). Professor of System Analysis Department of National Research Nuclear University MEPHI. Research interests: modeling and development of documentary information search systems and databases, linguistic support of documentary information search systems and knowledge management systems; human-machine information systems, interfaces based on cognitive and behavioral models. Author of more than 120 scientific papers, textbooks and tutorials.



Голицына Ольга Леонидовна в 1980 г. окончила Московский государственный университет по специальности «Прикладная математика». Кандидат технических наук (2004 г.). Доцент кафедры системного анализа МИФИ. Область научных интересов: моделирование и разработка документальных информационно-поисковых систем и баз данных, лингвистическое обеспечение документальных информационно-поисковых систем и систем управления знаниями; проектирование баз данных. Автор более 50 научных работ, учебников и учебных пособий.

Olga Leonidovna Golitsyna graduated from the Moscow State University in 1980, Cand.Sc. (2004). She is associate professor of System Analysis Department of National Research Nuclear University MEPHI. Research interests: modeling and development of documentary information search systems and databases; linguistic support of documentary information

search systems and knowledge management systems; database design.



Ганченкова Мария Герасимовна в 1997 г. окончила Московский инженерно-физический институт. К.т.н. (2002 г.). С 1995 по 2000 гг. – внедрение ИТ-систем в российском финансовом секторе. С 2000 по 2014 гг. исследовательская работа в МИФИ (Россия), Королевском институте технологий (Швеция), Университете Аальто (Финляндия). С 2017 г. – директор Высшей инженеринговой школы МИФИ. Более 70 научных работ. Область научных интересов: физика материалов, компьютерный инжиниринг, цифровые платформы.

Maria Gerasimovna Ganchenkova graduated from the Moscow State University in 1980. Can.Sc. (2002). From 1995 to 2000 – implementation of IT-systems in the Russian financial sector. From 2000 to 2014 active research work at MEPHI (Russia), Royal Institute of technology (Sweden), Aalto University (Finland). From 2017 - Director of Higher engineering schools

MIFI. More than 70 scientific works. Research interests: physics of materials, computer engineering, digital platforms.



Санатов Дмитрий Васильевич в 2006 г. окончил Санкт-Петербургский государственный университет. В 2018 г. получил дополнительное образование в МГТУ им. Баумана по направлению анализа данных. С 2006 г. по настоящее время работает в сфере стратегического консалтинга. С 2017 г. реализует проекты коммерциализации цифровых технологий в компании Медиа-Лаб, созданной как спин-офф проект ЦСР «Северо-Запад». Область научных интересов: управление изменениями в корпоративном секторе и региональном развитии, цифровизация экономики, промышленные технологические платформы.

Dmitry Vasilievich Sanatov graduated from the St.Petersburg State University in 2006. Now he is Deputy Director in the Center for strategic research “North-West” and CEO in Media-Lab, LLC. Research interests: change management in corporate sector and regional development, digitalization of economy, industrial technology platforms.



Разумов Антон Витальевич в 2002 г. окончил Московский государственный горный университет по специальности «Автоматизированные системы управления». С 2002 по 2017 гг. – развитие ИТ-инфраструктуры промышленных компаний, разработка программного обеспечения в среде Delphi, Power Builder, внедрение информационных систем уровня ERP и CRM. Область научных интересов: базы данных, системы автоматизированного проектирования.

Anton Vitalievich Razumov graduated from the Moscow State Mining University in 2002. From 2002 to 2017, he worked on the development of it infrastructure in industrial companies, software development in Delphi, Power Builder, implementation of information systems ERP and CRM. Research interests: databases, computer-aided design.

УДК 004.02

РЕАЛИЗАЦИЯ ОБОЛОЧКИ И ПОРТАЛА ЗНАНИЙ ПО ВЕРИФИКАЦИИ МАТЕМАТИЧЕСКИХ ДОКАЗАТЕЛЬСТВ НА ПЛАТФОРМЕ IASRAAS

А.С. Клещёв¹, В.А. Тимченко²*Институт автоматизации и процессов управления ДВО РАН, Владивосток, Россия*¹kleshev@iacp.dvo.ru, ²vadim@dvo.ru

Аннотация

Представлена концептуальная архитектура оболочки для интерактивных систем верификации математических доказательств и создаваемого с её помощью развиваемого тематического портала знаний. Описан процесс реализации всех программных и информационных компонентов оболочки на облачной платформе IASPaaS с использованием предоставляемых ею технологий и инструментальных средств их поддержки. Рассмотрен процесс разработки начального состояния портала знаний по верификации математических доказательств с использованием средств оболочки, способ использования портала знаний заинтересованными членами математического сообщества, а также механизмы изменения состояния портала его администратором. В состав начального состояния портала знаний входят: модель онтологии базы математических знаний, включающая спецификацию начального состояния языка представления математических знаний, редактор модели онтологии базы математических знаний, редактор базы математических знаний, редактор базы способов рассуждений, решатель задач оболочки, реализующий процесс конструирования доказательств в терминах модели онтологии доказательств. Также в состав начального состояния портала знаний входят начальное состояние базы математических знаний и начальное состояние базы способов рассуждений. Развитие портала знаний осуществляется по названным информационным компонентам. В этом процессе могут принимать участие все заинтересованные члены математического сообщества с помощью системы личных кабинетов платформы IASPaaS, в которых каждый пользователь может независимо развивать свою персональную копию текущего состояния общего портала знаний. Передача новых результатов в общий портал контролируется его администратором.

Ключевые слова: верификация интуитивных доказательств, программная оболочка, портал знаний, редактор базы знаний, облачные сервисы.

Цитирование: Клещёв, А.С. Реализация оболочки и портала знаний по верификации математических доказательств на платформе IASPaaS / А.С. Клещёв, В.А. Тимченко // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.427-448. – DOI: 10.18287/2223-9537-2018-8-3-428-448.

Введение

Одной из важнейших задач в математических исследованиях является обеспечение правильности (верификация) доказательств теорем, публикуемых в математической литературе. Одним из перспективных путей решения этой задачи является разработка интерактивных систем поддержки построения доказательств (СППД) [1–3]. Следствием потребности экспериментировать с построением (верификацией) доказательств в рамках разных формальных моделей (специализированных логик) явилась разработка программных оболочек для создания интерактивных СППД (LCF, PVS, Coq, Twelf, HOL Light, Isabelle/HOL и др.) [4–9].

В [10] обсуждалась основная на сегодняшний день причина, по которой СППД, создаваемые с использованием существующих оболочек, остаются мало востребованными в математическом сообществе. Ещё одна причина такого положения дел состоит в том, что большинство таких систем являются однопользовательскими и требуют сложной установки

на персональном компьютере. Это требует от математиков достаточной компьютерной грамотности, поэтому предпочтительнее оболочки и прикладные СППД, реализованные и предоставляемые пользователю как облачные сервисы. К настоящему моменту известны две реализации, удовлетворяющие этим требованиям: Isabelle, использующая подсистему Clide [11] и ProofPeer [12]. Но они не имеют расширяемой общей облачной базы знаний, которая могла бы совместно развиваться математическим сообществом, а не только разработчиками этих систем.

Заметим, что для Isabelle существует так называемое централизованное *хранилище формальных доказательств* (*Archive of Formal Proofs, AFP*). Пользователи могут направлять запросы на добавление формализованных ими теорий (каждая теория представляет собой набор определений, аксиом, теорем/лемм, доказательств) в это хранилище и в случае успешного прохождения процедуры рецензирования эти доказательства становятся частью AFP. Однако данное хранилище не содержит знаний о способах рассуждения, используемых при построении доказательств. Работа с ним строится на тех же принципах, что и работа с распределёнными системами управления версиями, что опять же требует от пользователей высокой компьютерной грамотности. В [12] отмечаются недостатки AFP, связанные со сложностями поиска нужных пользователю теорий, а также проблемами, возникающими при обновлении версий Isabelle. Исследователям нужна база знаний, организованная таким образом, чтобы они могли достаточно легко ориентироваться в ней при поиске необходимой информации.

В [10] представлена концепция программной оболочки для интерактивных систем верификации интуитивных математических доказательств, а также приближенная к математической практике конструирования доказательств формальная система и механизмы её расширения, которые могут быть положены в основу этой оболочки. Настоящая работа посвящена описанию разработки оболочки на облачной платформе IASaaS [13], тематического портала знаний по верификации интуитивных математических доказательств с использованием средств этой оболочки, а также принципов взаимодействия пользователей с порталом знаний и механизмов его развития.

1 Концептуальная архитектура оболочки для интерактивных систем верификации математических доказательств

Интерактивные системы верификации интуитивных математических доказательств могут быть отнесены к классу систем, основанных на знаниях (СОЗ), а компоненты описанной в работе [10] оболочки отображены на архитектурные компоненты типичных оболочек СОЗ [14]. В дальнейшем вся терминология работы [10] будет использоваться без ссылок на неё. На рисунке 1 представлена концептуальная архитектура оболочки для интерактивных систем верификации математических доказательств.

В состав оболочки входят: *модель онтологии базы математических знаний*, включающая *пустую спецификацию языка представления математических знаний*¹; *редактор модели онтологии базы математических знаний*; *модель онтологии базы способов рассуждений*, включающая спецификации *языка представления пропозициональных тавтологий* и *метаязыка*; *редактор базы математических знаний* и *редактор базы способов рассуждений*. Перечисленная совокупность *моделей онтологий* и *редакторов* представляет собой систему управления базами знаний (СУБЗ) оболочки. В состав оболочки

¹ Это означает, что в оболочке задана структура разделов математики (базы математических знаний), в которой порождающая грамматика языка представления математических знаний задаёт только общую структуру (вид *математического утверждения (предложения)*), а конкретные типы формул и термов языка ещё не определены.

также входят: модель онтологии доказательств; таблица соответствий для модели онтологии доказательств; мультиагентный интерактивный верификатор доказательств (подробнее см. раздел 3). Данная совокупность компонентов представляет собой решатель задач оболочки, имеющий пользовательский интерфейс. Наконец, оболочка включает в себя пустую базу математических знаний и пустую базу способов рассуждений.

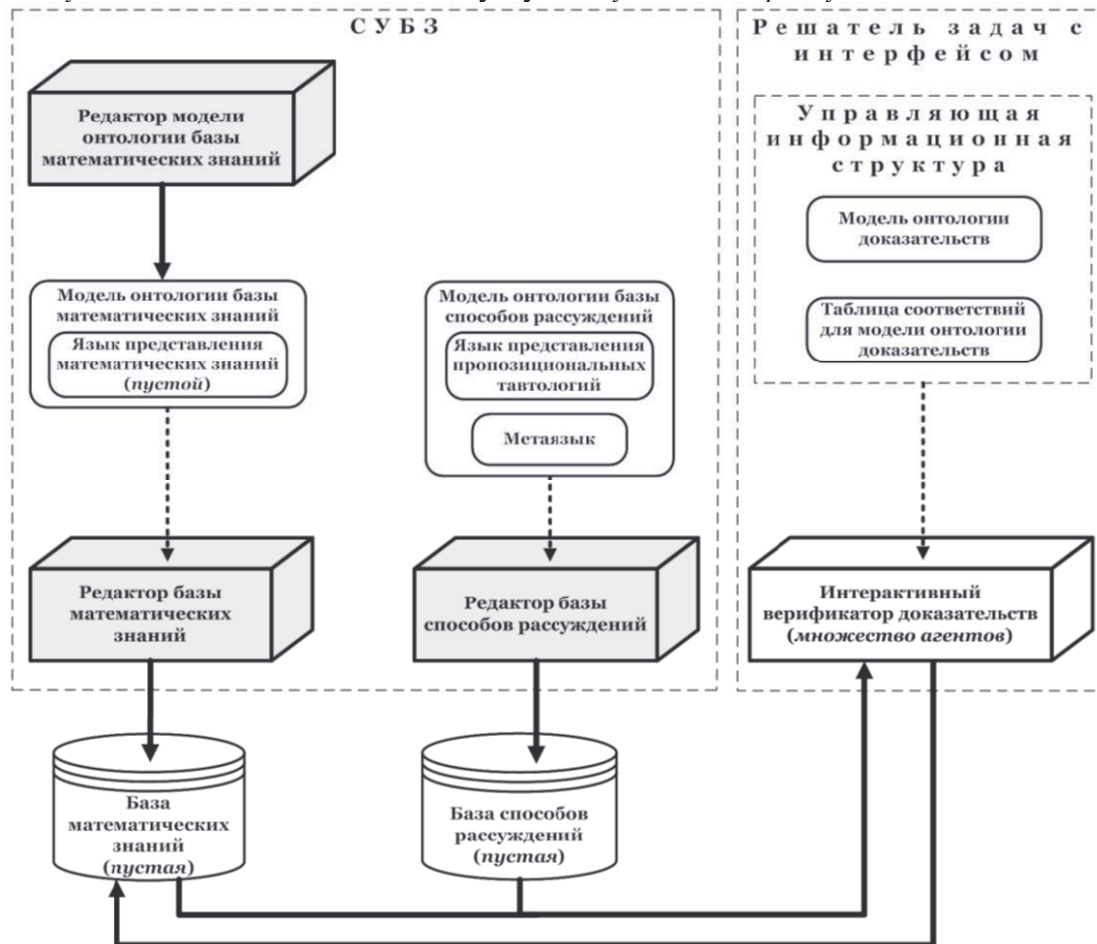


Рисунок 1 – Концептуальная архитектура оболочки для интерактивных систем верификации математических доказательств

Редактор модели онтологии базы математических знаний оболочки предназначен для формирования (спецификации) некоторого *начального состояния* языка представления математических знаний². *Редактор базы математических знаний* оболочки предназначен для формирования некоторого *начального состояния* базы математических знаний. Процесс редактирования в нём управляется моделью онтологии базы математических знаний (используемой в качестве метаинформации (метамодели) – на рисунке 1 этот факт демонстрируется пунктирной стрелкой, входящей в соответствующий программный компонент). *Редактор базы способов рассуждений* оболочки предназначен для формирования *начального состояния* базы способов рассуждений, процесс редактирования в нём управляется моделью онтологии базы способов рассуждений.

Интерактивный верификатор доказательств реализует итеративный процесс конструирования доказательств в терминах модели онтологии доказательств. Он начинается

² Это подразумевает описание в порождающей грамматике языка представления математических знаний конкретных типов формул и термов.

с задания названия доказательства и выбора (из базы математических знаний) доказываемого утверждения – теоремы, леммы, некоторого следствия. Далее на каждой итерации последовательно выполняются следующие действия.

- 1) Выбор *текущей цели* – очередного доказываемого математического утверждения (формулы). В начале доказательства текущей целью является доказываемая теорема (лемма, следствие).
- 2) Выбор метода доказательства.
- 3) Выбор способа рассуждения (если метод доказательства не основан на *правилах доказательства импликации (естественного вывода)*).
- 4) Выбор значений для посылок при использовании правила *Modus ponens* (если данное правило было выбрано для *вывода* или *декомпозиции цели*).

Данный процесс заканчивается, когда достигнуто состояние, в котором нет недоказанных целей. Результатом процесса является орграф полного доказательства. Ссылка на этот орграф добавляется в множество ссылок на доказательства утверждения (теоремы, леммы или следствия), принадлежащего базе математических знаний. При этом такая ссылка создаётся не пользователем при формировании базы математических знаний, а интерактивным верификатором доказательств в результате проверки условия завершения процесса конструирования доказательства.

2 Концептуальная архитектура начального состояния портала знаний по верификации математических доказательств

«Пустые» прикладные системы (сервисы) верификации интуитивных математических доказательств, в которых язык представления математических знаний, базу математических знаний и базу способов рассуждений необходимо формировать с нуля, не практичны. Поэтому использование оболочки для создания прикладных сервисов нецелесообразно. В связи с этим возникает потребность в концепте более высокого уровня абстракции (разрабатываемой с использованием средств оболочки), в такой как, например, тематический (специализированный) портал знаний [15] по верификации математических доказательств, который содержал бы некоторое состояние языка представления математических знаний, базы математических знаний и базы способов рассуждений, а также средства их расширения.

На рисунке 2 представлена концептуальная архитектура *начального состояния* развиваемого портала знаний по верификации математических доказательств. В этом смысле оболочка может рассматриваться как *предначальное состояние* портала знаний.

В состав начального состояния портала знаний входят: модель онтологии базы математических знаний, включающая спецификацию начального состояния (ядра) языка представления математических знаний; редактор модели онтологии базы математических знаний; редактор базы математических знаний; редактор базы способов рассуждений; решатель задач оболочки. Также в состав начального состояния портала знаний входят начальное состояние базы математических знаний и начальное состояние базы способов рассуждений.

Редактор модели онтологии базы математических знаний (уже как средство портала знаний) предназначен для расширения состояния языка представления математических знаний. Редактор базы математических знаний портала знаний предназначен для расширения состояния базы математических знаний. Процесс редактирования в нём управляется моделью онтологии базы математических знаний. Редактор базы способов рассуждений портала знаний предназначен для расширения состояния базы способов рассуждений. Процесс редактирования в нём управляется соответственно моделью онтологии базы способов рассуждений. Данные редакторы также рассматриваются уже как средства портала знаний.

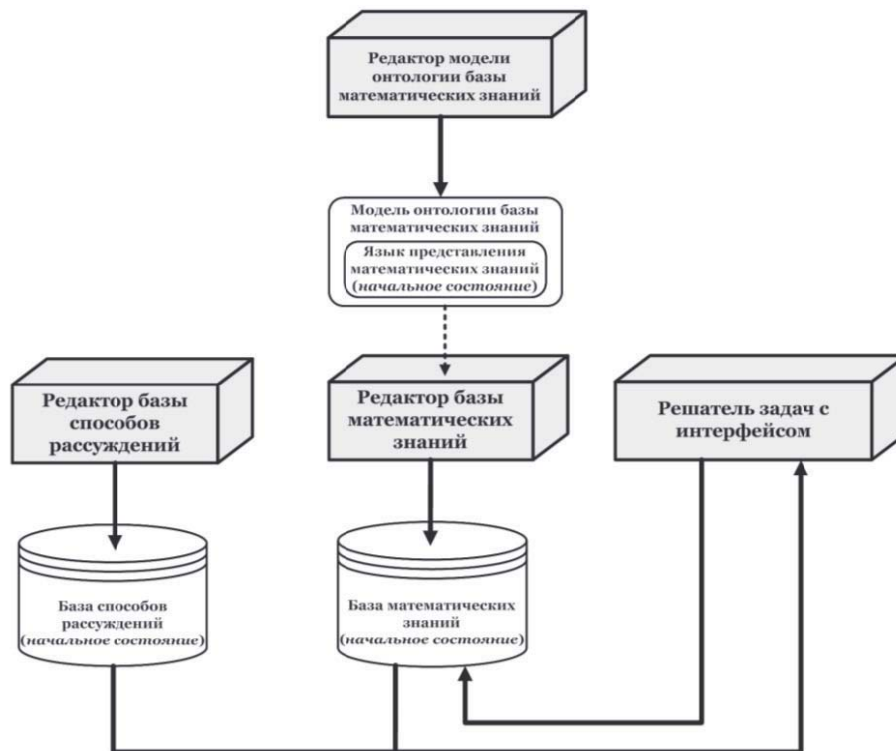


Рисунок 2 – Концептуальная архитектура начального состояния портала знаний по верификации математических доказательств

Все интегрированные в портал ресурсы структурированы в соответствии с их типом и назначением с целью организации удобного поиска и навигации по ним.

3 Разработка оболочки на облачной платформе IASaaS

Реализация оболочки для интерактивных систем верификации математических доказательств на облачной платформе IASaaS с использованием предоставляемых ею средств и технологий (инструментария) разработки направлена на обеспечение возможности предоставления оболочки и создаваемого с её использованием начального состояния портала знаний по верификации математических доказательств как облачных сервисов этой платформы (см. рисунок 3).

Разработка всех компонентов СУБЗ оболочки, представленных на рисунке 1, а также пустых базы математических знаний и базы способов рассуждений выполняется с использованием базовой (универсальной) технологии разработки оболочек CO3 на платформе IASaaS и инструментальных средств её поддержки. Разработка интерактивного верификатора доказательств и его интерфейса выполняется с использованием специализированной технологии разработки оболочек CO3 на платформе IASaaS – на основе информационных структур, управляющих процессом вычисления результата решения задачи (расширенных графовых грамматик) [16].

Использование данной технологии ориентировано на класс задач, для которых явным образом можно специфицировать *структуру* результата их решения в форме связного орграфа – орграфа грамматики (метаинформации, модели), а процесс вычисления самого результата, который также представляет собой связный орграф (орграф информации, компонента), организовать в виде его пошагового формирования «сверху-вниз».

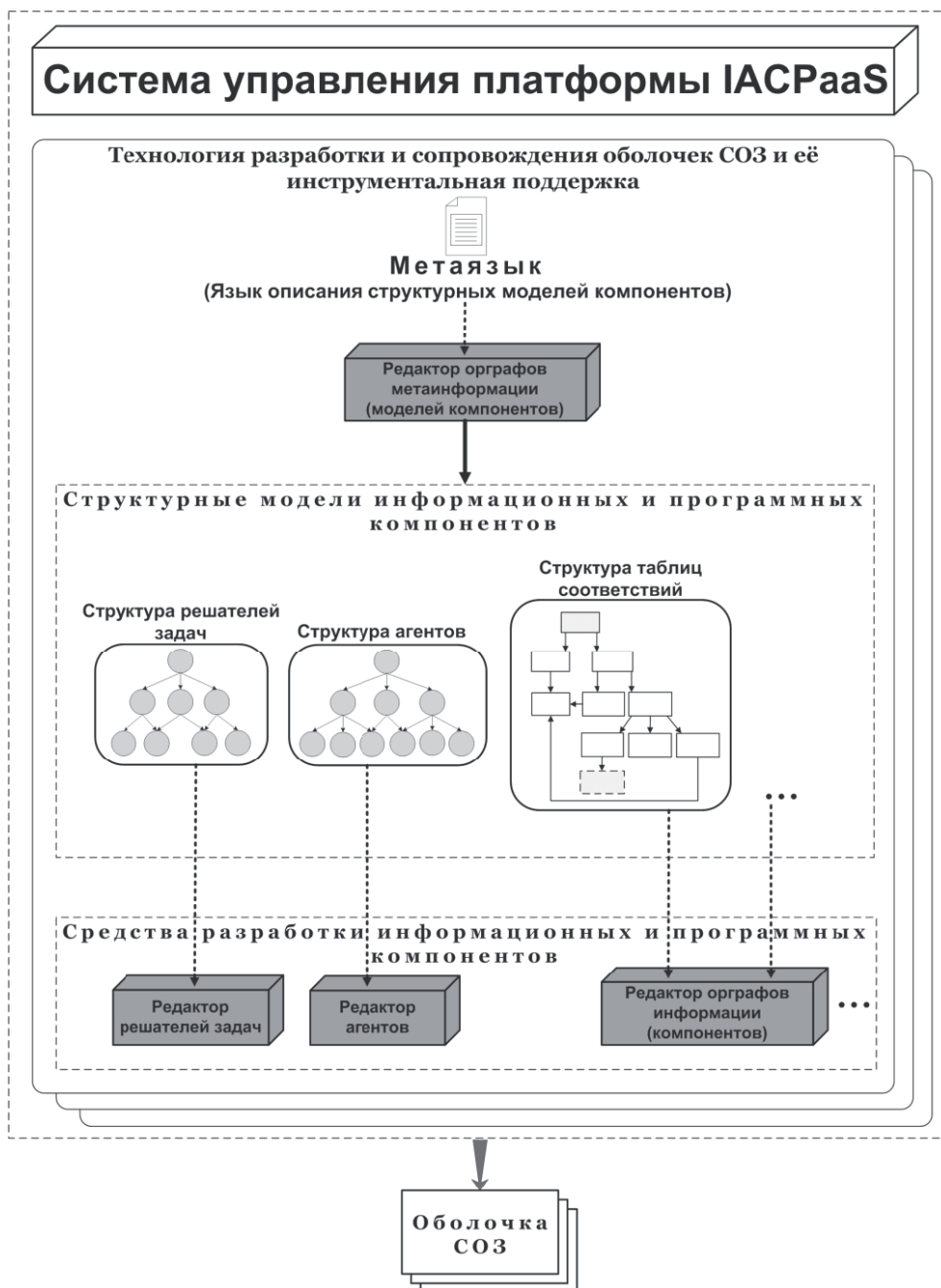


Рисунок 3 – Инструментарий для разработки оболочек СОЗ на облачной платформе IACPaas

В основу технологии положено явное описание *орграфа информационной структуры, управляющей процессом решения задачи* (либо нескольких таких взаимосвязанных орграфов). Расширение графовой грамматики представляет собой орграф специального вида, называемый *таблицей соответствий*. Она сопоставляет вершинам орграфа грамматики программные модули (агенты), которые формируют результат решения задачи, предоставляя, возможно, для этого необходимый пользовательский интерфейс. Обращения к агентам осуществляет обеспечивающий инструментальную поддержку данной технологии *интерпретатор орграфов управляющих структур*. Он обеспечивает взаимодействие пользователя (пользовательский интерфейс) с управляющей структурой и является ядром решателей задач оболочек, разрабатываемых по данной технологии. При этом

предоставляемый пользовательский интерфейс интерпретатора может расширяться интерфейсными возможностями, которые реализуют агенты, указанные в таблице соответствий.

В случае с разработкой интерактивного верификатора доказательств оргграфом управляющей структуры является оргграф грамматики, представляющий модель онтологии полных доказательств. Он связан с оргграфами, представляющими модель онтологии базы математических знаний и модель онтологии базы способов рассуждений. Посредством таблицы соответствий некоторым вершинам оргграфа модели онтологии полных доказательств сопоставляются обращения к агентам, которые реализуют семантику методов доказательств целей, а также алгоритмы однонаправленной унификации (сопоставления) и применения подстановки. В процессе работы эти агенты формируют фрагменты оргграфа доказательства в соответствии с синтаксической структурой методов доказательств (описанных в модели онтологии доказательств).

Разработка каждого информационного и программного компонента оболочки подразумевает: 1) создание с использованием *Системы управления* на web-сайте облачной платформы IASPaas в соответствующем разделе личного кабинета разработчика оболочки новой представляющей этот компонент *единицы хранения* (оргграфа грамматики или оргграфа информации) нужного типа («*метаинформация*», «*информация*», «*агент*», «*решатель*») с пустым содержимым (оргграф представлен единственной корневой вершиной)³; 2) формирование содержимого созданной единицы хранения «сверху-вниз» (начиная с корневой вершины оргграфа) с использованием соответствующего инструментального средства платформы.

3.1 Разработка компонентов системы управления базами знаний

Функциональность и интерфейс редактора «*Редактор модели онтологии базы математических знаний*» полностью реализуется инструментальным средством платформы IASPaas «*Редактор оргграфов метаинформации*». Модель процесса редактирования, положенная в основу данного редактора, построена на семантике конструкций декларативного *метаязыка*, используемых для спецификации оргграфов грамматики (метаинформации). Метаязык, используемый здесь как средство спецификации формальных систем, состоит из трёх подязыков: языка описания порождающих графовых грамматик, языка описания контекстных условий и языка описания порождающих текстовых грамматик, описанных в [10, 17].

Разработка оргграфа грамматики, представляющего *модель онтологии базы математических знаний*, включающей описание пустого языка представления математических знаний, состоит в создании в разделе «*Онтологии*» личного кабинета разработчика оболочки новой единицы хранения типа «*метаинформация*» и формировании её содержимого с использованием редактора «*Редактор оргграфов метаинформации*» как показано на рисунке 4.

Аналогично выполняется разработка оргграфа грамматики, представляющего модель онтологии базы способов рассуждений, включающей описание языка представления пропозициональных тавтологий и метаязыка (языка представления метаматематических утверждений): их абстрактного и конкретного синтаксиса, а также контекстных условий для этих языков (рисунок 5).

³ При этом задаётся её название и описание. Если тип единицы хранения есть «*информация*», то также в качестве её *метаинформации* указывается хранящийся в Фонде платформы нужный оргграф грамматики.

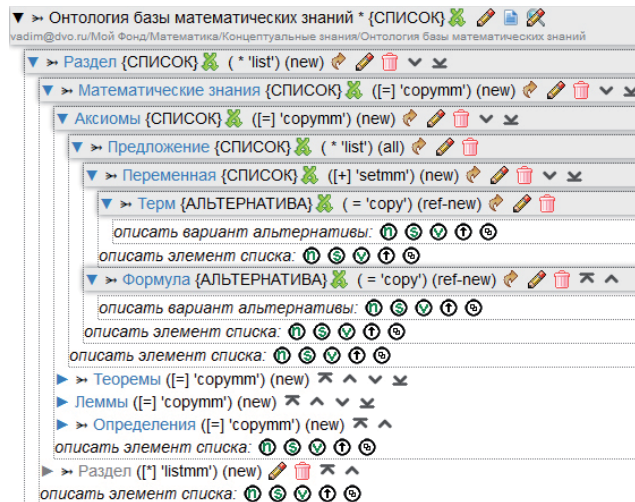


Рисунок 4 – Структура базы математических знаний и математического утверждения в интерфейсе инструментального средства платформы «Редактор орграфов метаинформации»

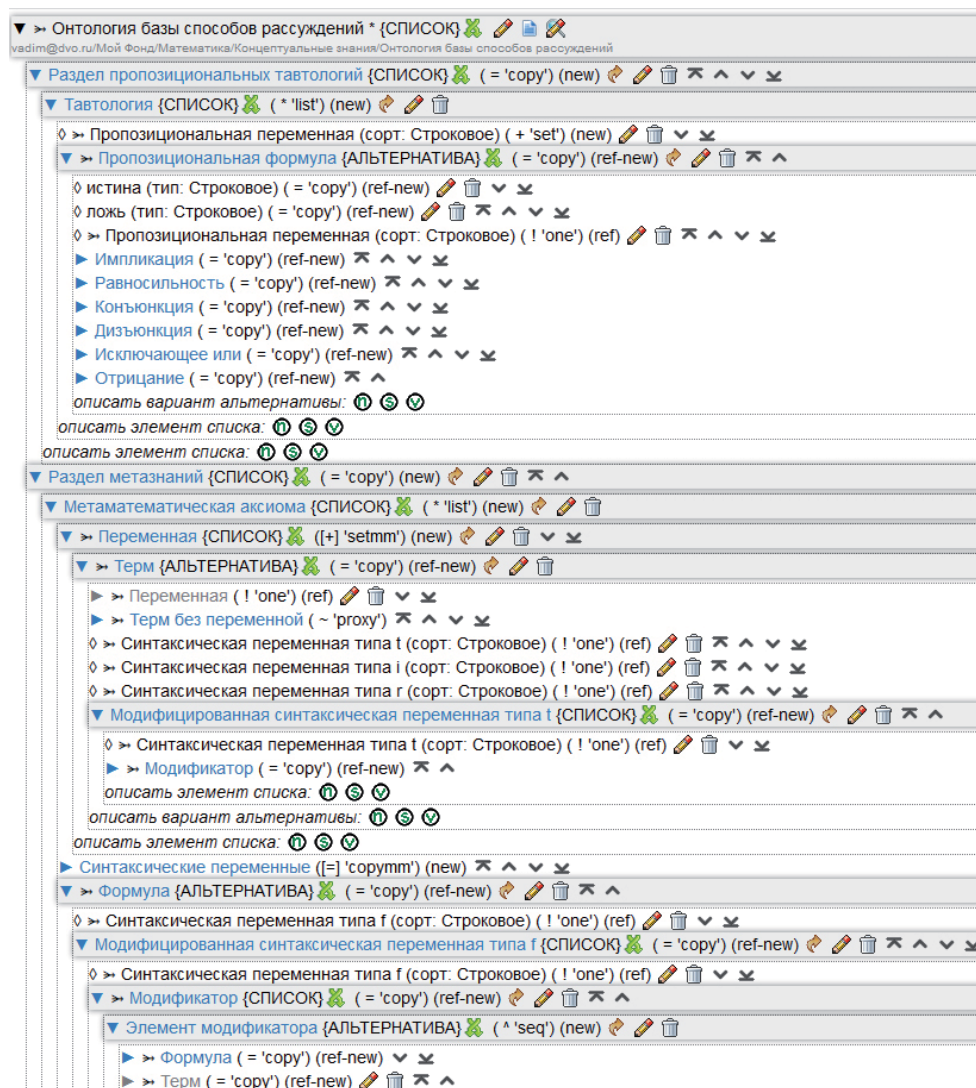


Рисунок 5 – Модель онтологии базы способов рассуждений в интерфейсе инструментального средства платформы «Редактор орграфов метаинформации»

«Редактор базы математических знаний» создаётся путём подключения к инструментальному средству платформы IASaaS «Редактор орграфов информации» модели онтологии базы математических знаний в качестве информации, управляющей генерацией пользовательского интерфейса и процессом формирования базы математических знаний с проверкой её полноты и заданных в модели онтологии контекстных условий. Аналогично, «Редактор базы способов рассуждений» создаётся путём подключения к средству «Редактор орграфов информации» модели онтологии базы способов рассуждений.

3.2 Разработка интерактивного верификатора доказательств

Разработка интерактивного верификатора доказательств с интерфейсом включает создание: орграфа грамматики, представляющего модель онтологии полных доказательств; таблицы соответствий для него; решателя задач – множества агентов, обращения к которым выполняются (непосредственно или опосредованно) интерпретатором орграфов управляющих структур через таблицу соответствий; декларативной спецификации решателя задач, описывающей его связь с формальными параметрами⁴, таблицей соответствий, базой математических знаний и базой способов рассуждений, а также пользовательским интерфейсом.

3.2.1 Разработка модели онтологии доказательств

Разработка орграфа грамматики, представляющего модель онтологии доказательств, состоит в создании в разделе «Онтологии» личного кабинета разработчика оболочки новой единицы хранения типа «метаинформация» и формировании её содержимого с использованием инструментального средства платформы «Редактор орграфов метаинформации» (рисунок 6).

3.2.2 Разработка таблицы соответствий для модели онтологии доказательств

Разработка орграфа информации, представляющего таблицу соответствий, состоит в создании в разделе «Решатели» новой единицы хранения типа «информация»⁵ и формировании её содержимого с использованием инструментального средства платформы «Редактор орграфов информации». Орграф грамматики «Структура таблиц соответствий» подключён к данному редактору в качестве информации, управляющей процессом формирования таблиц соответствий. На рисунке 7 показана таблица соответствий между понятиями модели онтологии доказательств и агентами решателя задач оболочки, представленная в форме орграфа.

Для каждого обращения к агенту вершина «интерактивное» представляет собой логический признак. Он определяет, как в процессе формирования орграфа информации (орграфа доказательства) это обращение должно выполняться: автоматически или по инициативе пользователя. Корневая вершина орграфа закрашена серым цветом. Вершины, изображенные пунктирными прямоугольниками, принадлежат орграфам, отличным от данного орграфа, т.е. представляющим другую метаинформацию или информацию. На рисунке 7 таковыми являются вершины, принадлежащие орграфу метаинформации, представляющему модель онтологии доказательств, а также корневые вершины орграфов информации, описывающих декларативные спецификации агентов решателя задач. Символ «@» в метках вершин, представленных пунктирными прямоугольниками, разделяет метку корневой вершины стороннего орграфа и метку той его вершины (не совпадающей с первой), которая является корневой

⁴ Формальный параметр представляет собой орграф грамматики (метаинформации), а фактический параметр – некоторый порожденный по ней орграф информации, обрабатываемый решателем задач.

⁵ При этом в качестве метаинформации указывается хранящийся в Фонде орграф грамматики «Структура таблиц соответствий», описывающий абстрактный синтаксис языка представления таблиц соответствий.

вершиной повторно используемого подграфа. В квадратных скобках у вершин указаны метки соответствующих им вершин из орграфа грамматики (метаинформации).

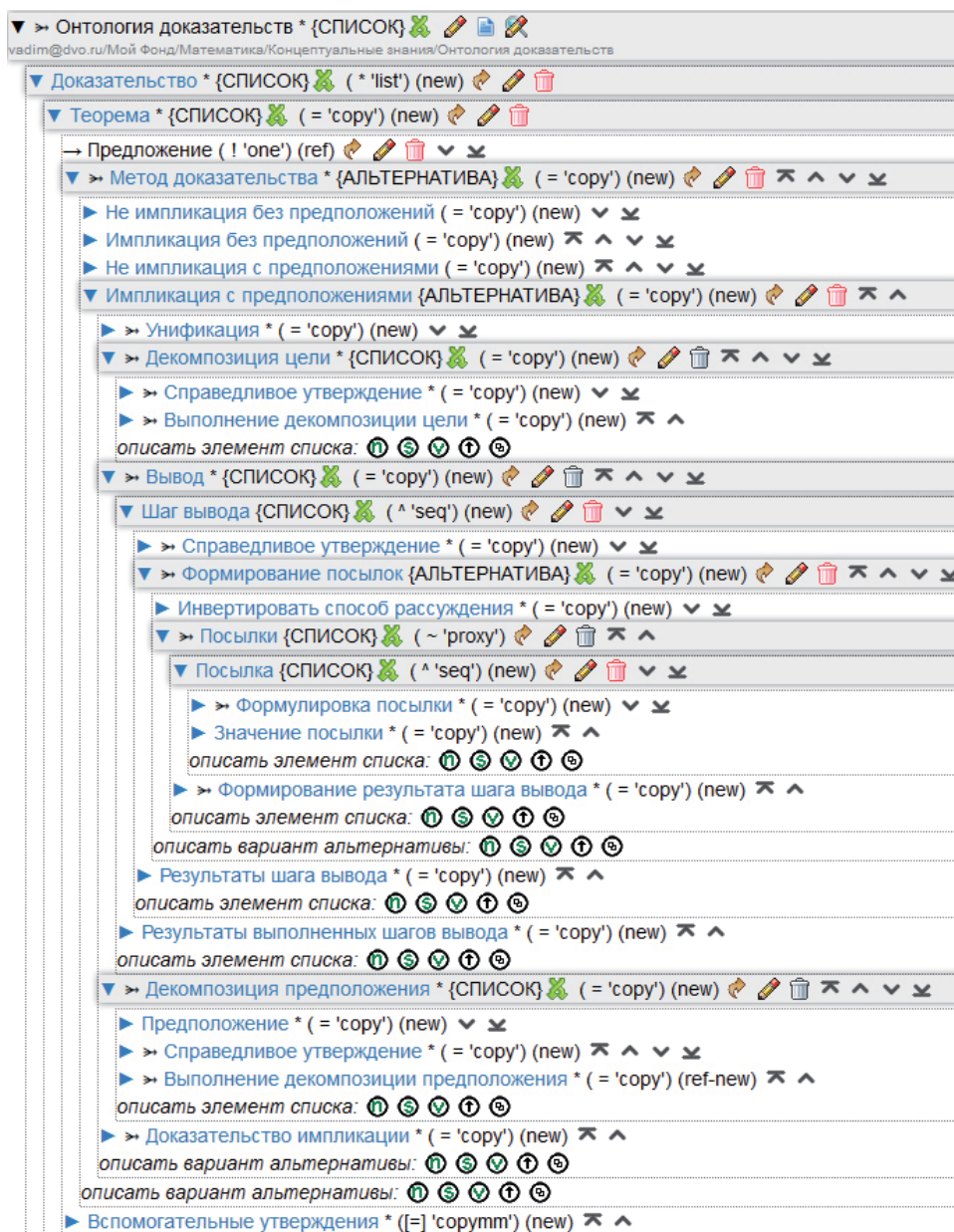


Рисунок 6 – Модель онтологии доказательств в интерфейсе инструментального средства платформы «Редактор орграфов метаинформации»

3.2.3 Разработка агентов решателя задач

В данном разделе описывается обобщённый процесс разработки агентов, указанных на рисунке 7. Обращения к этим агентам выполняются при создании вершин в орграфе доказательства, соответствующих вершинам из орграфа грамматики, представляющего модель онтологии полных доказательств, также указанных на рисунке 7. Шаблоны сообщений, посредством которых интерпретатор орграфов управляющих структур взаимодействует с описанными в орграфе агентами, доступны в Фонде платформы IASaaS для использования.

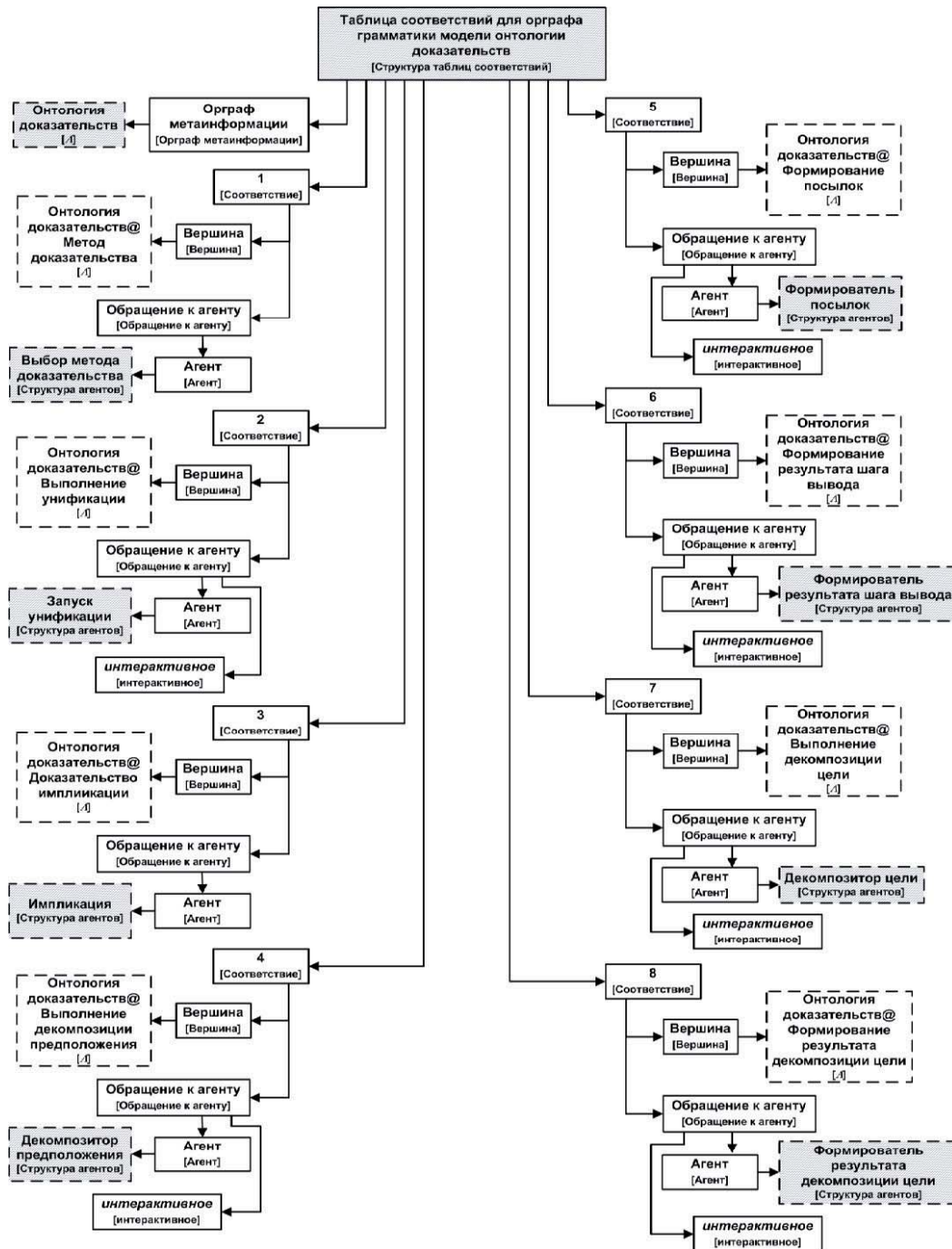


Рисунок 7 – Орграф таблицы соответствий между понятиями модели онтологии доказательств и агентами решателя задач оболочки

Разработка агента состоит в формировании в разделе «Агенты» единицы хранения - орграфа информации, представляющего декларативную спецификацию агента⁶; генерации заготовки исходного кода агента по его декларативной спецификации; написании исходного кода агента, реализующего его семантику (с использованием предоставляемого платформой API); получении байт-кода агента (в результате компиляции исходного кода) и загрузки его в Фонд – в соответствующую единицу хранения.

⁶ Создаётся новая единица хранения типа «агент» (с названием, совпадающим с названием агента) и формируется её содержимое с использованием инструментального средства платформы «Редактор агентов».

Агент «Выбор метода доказательства» предоставляет множество альтернативных вариантов для выбора метода доказательства текущей цели в зависимости от синтаксической формы её утверждения (имеет ли оно форму импликации или нет) и наличия или отсутствия у неё предположений.

Если утверждение имеет вид импликации и список предположений не пуст, то *Метод доказательства* \in {"Унификация", "Декомпозиция цели", "Вывод", "Декомпозиция предположения", "Доказательство импликации"}. Если утверждение не является импликацией и список предположений не пуст, то *Метод доказательства* \in {"Унификация", "Декомпозиция цели", "Вывод", "Декомпозиция предположения"}. Если утверждение имеет вид импликации и список предположений пуст, то *Метод доказательства* \in {"Унификация", "Декомпозиция цели", "Вывод", "Доказательство импликации"}. Если утверждение не является импликацией и список предположений пуст, то *Метод доказательства* \in {"Унификация", "Декомпозиция цели", "Вывод"}.

Агент «Запуск унификации» подготавливает входные данные для агента, реализующего алгоритм унификации (точнее, её так называемый однонаправленный вариант – сопоставление [18]) – справедливое утверждение, выбранное пользователем из базы математических знаний или базы способов рассуждений, и частное утверждение – доказываемую цель. Принимает от агента, реализующего алгоритм унификации, результат, и если унификация не выполнена, то отображает пользователю соответствующее сообщение; если унификация выполнена, то формирует множество новых целей (если в процессе унификации появились вспомогательные утверждения).

Агент «Импликация» реализует правило доказательства импликации (естественного вывода).

Агент «Декомпозитор предположения» реализует семантику применения правила *Modus ponens* (правило отсечения) для декомпозиции предварительно выбранного пользователем предположения из списка предположений доказываемой цели; если в процессе удачно выполненной унификации появились вспомогательные утверждения, то формирует множество соответствующих новых целей.

Агенты «Формирователь посылок» и «Формирователь результата шага вывода» реализуют семантику применения правила *Modus ponens* для выполнения шагов вывода. Агент «Формирователь посылок» организует выбор пользователем справедливого утверждения в форме импликации или равносильности. При этом если выбранное утверждение имеет форму равносильности, то пользователю задаётся вопрос о необходимости инвертирования данной равносильности. Если равносильность необходимо инвертировать, то правая часть равносильности считается условием, а левая – заключением. По количеству конъюнктов в условии импликации (или части равносильности, считающейся условием) агент формирует множество посылок. Формулировка каждой посылки есть формулировка соответствующего конъюнкта. Значение посылки в зависимости от наличия предположений у доказываемой цели и уже выполненных шагов вывода может быть выбрано пользователем из условий:

- если у доказываемой цели есть предположения и выполнен хотя бы один шаг вывода – базы математических знаний, списка предположений цели, результатов выполненных шагов вывода;
- если у доказываемой цели есть предположения, но еще не выполнено ни одного шага вывода – базы математических знаний, списка предположений цели;
- если у доказываемой цели отсутствуют предположения и выполнен хотя бы один шаг вывода – базы математических знаний, результатов выполненных шагов вывода;
- если у доказываемой цели отсутствуют предположения и еще не выполнено ни одного шага вывода – базы математических знаний.

Агент «*Формирователь результата шага вывода*» даёт задание на унификацию (агенту, реализующему данный алгоритм) условия справедливого утверждения с формулой, представляющей собой конъюнкцию выбранных значений посылок. Если унификация выполнена, то формирует результат шага вывода, представляющий собой результат подстановки унификатора в заключение справедливого утверждения, а также множество новых целей (если в процессе унификации появились вспомогательные утверждения). В противном случае отображает пользователю сообщение о невозможности унификации формул. После формирования результата очередного шага проверяется условие окончания процесса вывода.

Агенты «*Декомпозитор цели*» и «*Формирователь результата декомпозиции цели*» реализуют семантику применения правила *Modus ponens* для выполнения декомпозиции цели. Выполняется унификация доказываемой цели и заключения импликации (или части равносильности) выбранного пользователем справедливого утверждения из базы математических знаний или базы способов рассуждений. При этом если последнее имеет форму равносильности, в которой одна из её частей имеет форму конъюнкции, а другая не имеет, то частью равносильности, участвующей в унификации – заключением равносильности, является часть, не имеющая формы конъюнкции. Если унификация не выполнена, то, если справедливое утверждение имеет форму импликации, пользователю отображается сообщение о невозможности унификации формул, а если справедливое утверждение имеет форму равносильности, то даётся задание на унификацию доказываемой цели и другой части равносильности. Если унификация выполнена, то формируется множество новых целей (если в процессе унификации появились вспомогательные утверждения), в противном случае пользователю отображается сообщение о невозможности унификации формул.

Выполняется подстановка (агентом, реализующим алгоритм подстановки) вычисленного унификатора во все конъюнкты условия импликации (или части равносильности, не участвовавшей в последней унификации). Результат подстановки есть множество новых целей, количество которых равно количеству конъюнктов. Список предположений каждой новой цели совпадает со списком предположений доказываемой цели.

3.2.4 Разработка декларативной спецификации решателя задач интерактивного верификатора доказательств

Разработка в разделе «*Решатели*» орграфа информации, представляющего декларативную спецификацию решателя задач интерактивного верификатора доказательств, состоит в создании новой единицы хранения типа «*решатель*» и формировании её содержимого с использованием инструментального средства платформы «*Редактор решателей задач*».

Корневой агент решателя задаётся путём создания ссылки на корневую вершину орграфа информации в Фонде платформы IASaaS, представляющего декларативную спецификацию агента «*Универсальный корневой агент редакторов и просмотрщиков*». Агент *Интерфейсный контроллер* решателя задаётся путём создания ссылки на корневую вершину орграфа информации в Фонде платформы IASaaS, представляющего декларативную спецификацию агента «*Интерфейсный контроллер расширяемых редакторов и просмотрщиков единиц хранения*» (рисунок 8). Данный агент реализует функциональные возможности интерпретатора орграфов управляющих структур.

В качестве *выходного формального параметра* указывается (путём создания ссылки на его корневую вершину) орграф грамматики, представляющий модель онтологии доказательств. *Выходными фактическими параметрами*, указываемыми при создании прикладных сервисов с использованием оболочки, являются орграфы информации, представляющие формируемые по модели онтологии доказательств базы доказательств.

Среди собственных информационных ресурсов необходимо указать орграфы информации, представляющие соответственно *таблицу соответствий для орграфа грамматики модели онтологии доказательств*, *базу математических знаний* и *базу способов рассуждений*.

Связывание пользовательского интерфейса с решателем задач состоит в создании web-страницы с названием «*Интерактивный верификатор доказательств*» и содержимым, показанным на рисунке 8. Необходимые интерфейсные элементы создаются с использованием предоставляемого платформой API при написании исходного кода агентов, реализующих соответствующую семантику.

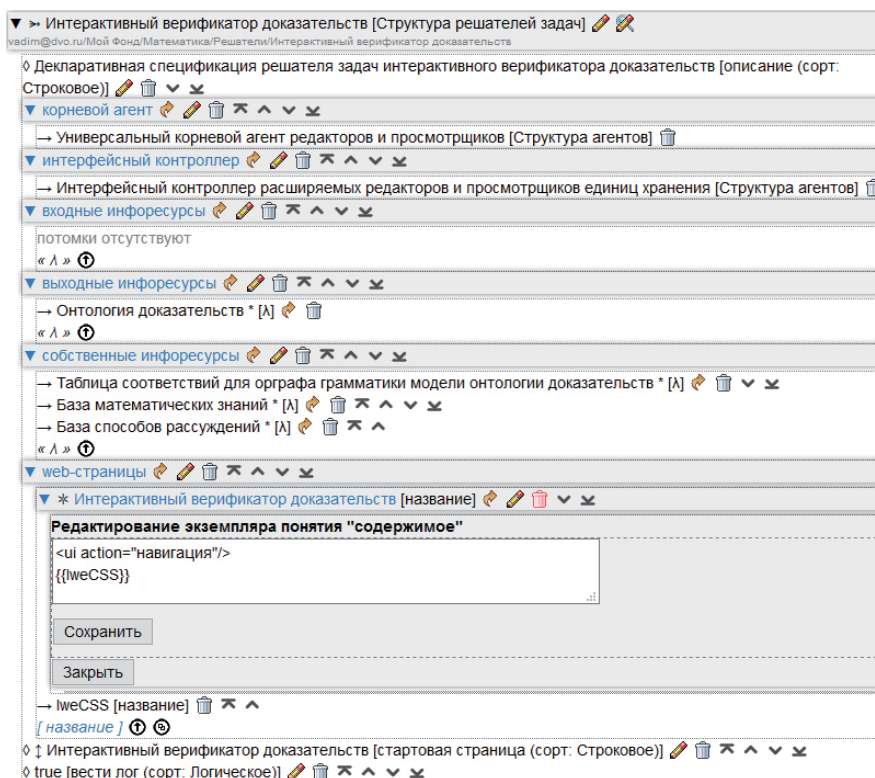


Рисунок 8 – Декларативная спецификация решателя задач интерактивного верификатора доказательств в интерфейсе инструментального средства «*Редактор решателей задач*»

3.3 Разработка пустых базы математических знаний и базы способов рассуждений

Разработка орграфа информации, представляющего базу математических знаний, состоит в создании в разделе «*Базы знаний*» новой единицы хранения типа «информация». В качестве метainформации при этом указывается хранящийся в разделе «*Онтологии*» орграф грамматики «*Онтология базы математических знаний*». Аналогично, разработка орграфа информации, представляющего базу способов рассуждений, состоит в создании в разделе «*Базы знаний*» новой единицы хранения типа «информация» с указанием в качестве метainформации орграфа грамматики «*Онтология базы способов рассуждений*».

4 Разработка начального состояния портала знаний с использованием оболочки

Разработка начального состояния портала знаний по верификации математических доказательств выполняется с использованием средств, входящих в состав оболочки, а также *Су-*

стемы управления облачной платформы IASaaS. Структура портала знаний по верификации математических доказательств формируется с помощью Системы управления на веб-сайте облачной платформы IASaaS в личном кабинете разработчика портала. Структура представляет собой дерево разделов, терминальными вершинами которого являются информационные и/или программные компоненты (ресурсы) портала, отнесённые к определённому разделу по некоторому отличительному признаку: типу, целевому назначению и т.п. Также с помощью Системы управления начальное состояние портала знаний переносится в Фонд платформы, где он становится общедоступным, и назначается ответственный за данный портал администратор, который контролирует его дальнейшее развитие.

Разработка *начального состояния (ядра)* языка представления математических знаний состоит в расширении орграфа грамматики, представляющего модель онтологии базы математических знаний, конкретными типами формул и термов с использованием редактора оболочки «Редактор модели онтологии базы математических знаний» (рисунок 9).

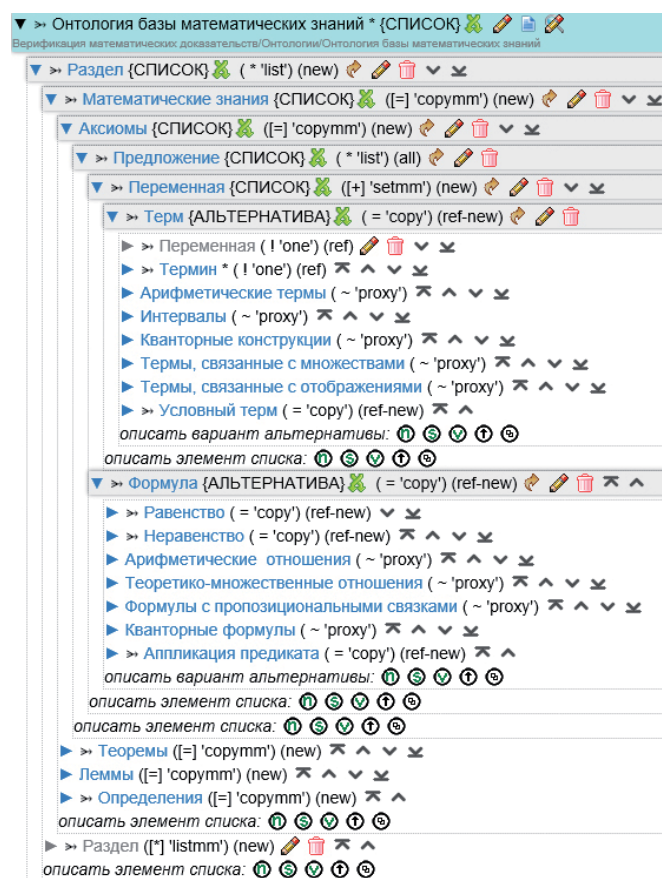


Рисунок 9 – Начальное состояние языка представления математических знаний в интерфейсе средства портала «Редактор модели онтологии базы математических знаний»

Разработка *начального состояния (ядра)* базы математических знаний состоит в формировании содержимого представляющего её орграфа информации с использованием редактора оболочки «Редактор базы математических знаний» (рисунок 10). Разработка *начального состояния (ядра)* базы способов рассуждений состоит в формировании содержимого представляющего её орграфа информации с использованием редактора оболочки «Редактор базы способов рассуждений» (рисунок 11).

Начальное состояние базы математических знаний содержит раздел «Арифметика», который содержит 29 аксиом, 17 теорем, 3 определения и 4 леммы. Начальное состояние

базы формализованных способов рассуждений содержит 29 способов рассуждений: 5 пропозициональных тавтологий и 24 метаматематических утверждения.

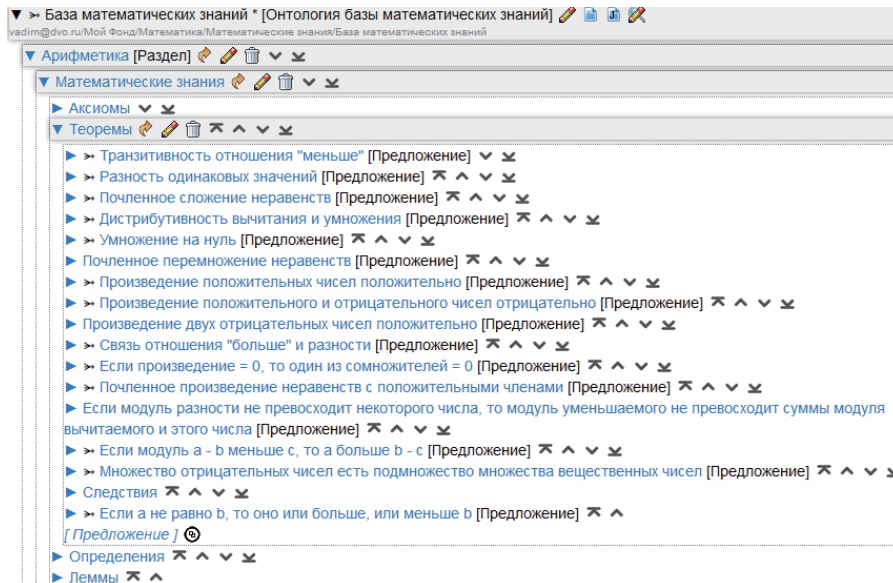


Рисунок 10 – Начальное состояние базы математических знаний в интерфейсе средства портала «Редактор базы математических знаний»

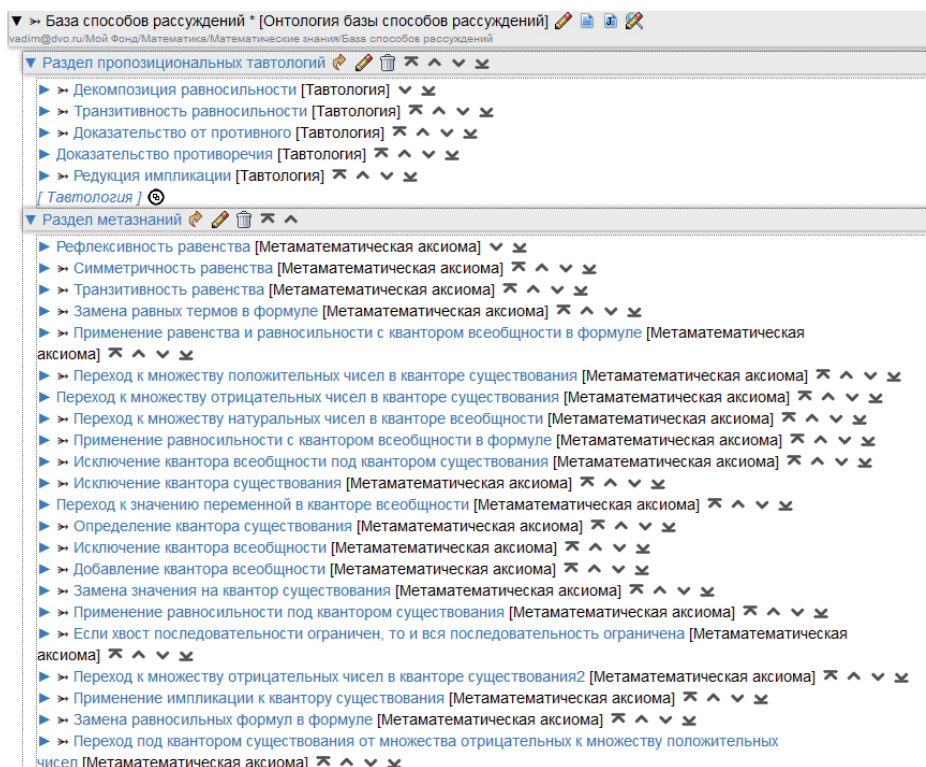


Рисунок 11 – Начальное состояние базы способов рассуждений в интерфейсе средства портала «Редактор базы способов рассуждений»

Все подграфы, корневые вершины которых соответствуют вершинам «Предложение», «Тавтология» и «Метаматематическая аксиома» в соответствующих орграфах грамматик, отображаются в текстовом представлении. На рисунке 12 показан пример для пропозициональных тавтологий.

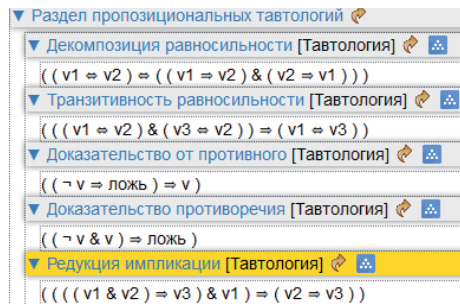


Рисунок 12 – Текстовое представление пропозициональных тавтологий в базе способов рассуждений в интерфейсе редактора «Редактор базы способов рассуждений»

Все редакторы и решатель задач оболочки включаются в состав программных компонентов портала знаний.

5 Использование текущего состоянием портала знаний исследователями

Для использования текущего состояния портала знаний по верификации математических доказательств пользователю сначала необходимо пройти процедуру регистрации на веб-сайте платформы IASaaS, после чего для него создаётся *Личный кабинет*, включающий *Персональный Фонд*. С помощью *Системы управления* пользователь может произвольным образом структурировать свой Персональный Фонд: создать древовидную структуру разделов.

Дальше у пользователя появляется возможность *скопировать текущее состояние портала знаний* из Фонда платформы в Персональный Фонд своего личного кабинета. Получение копии текущего состояния портала знаний состоит в следующем. В разделе Персонального Фонда «*Загрузки*» создаётся копия единицы хранения, представляющей решатель задач «*Интерактивный верификатор доказательств*». При создании копии решателя он связывается с копией текущего состояния базы математических знаний, для которой метаинформацией (орграфом грамматики) является уже копия модели онтологии базы математических знаний; с копией текущего состояния базы способов рассуждений; а также со всеми остальными хранимыми в портале знаний единицами хранения, указанными в декларативной спецификации решателя задач «*Интерактивный верификатор доказательств*». Полученные копии единиц хранения можно перенести из раздела «*Загрузки*», в нужные разделы Персонального Фонда. Редакторы соответствующих баз и модели онтологии базы математических знаний в силу способа своей реализации не копируются: они доступны из личных кабинетов и настроены уже на копии этих информационных компонентов.

Результатом выполнения операции *копирования текущего состояния портала знаний* является также создание в личном кабинете пользователя персонального прикладного сервиса. Создание сервиса подразумевает создание орграфа информации, представляющего персональное хранилище доказательств⁷, и его связывание с копией единицы хранения, представляющей решатель задач оболочки, с помощью которого это хранилище можно наполнять. Связывание состоит в создании специального вида единицы хранения, представляющей запускаемую и исполняемую на платформе IASaaS сущность – *сервис*. На рисунках 13, 14 продемонстрированы примеры работы такого сервиса при формировании доказательства теоремы о том, что если последовательность имеет предел, то она является ограниченной.

⁷ Его метаинформацией является орграф грамматики, представляющий модель онтологии доказательств.

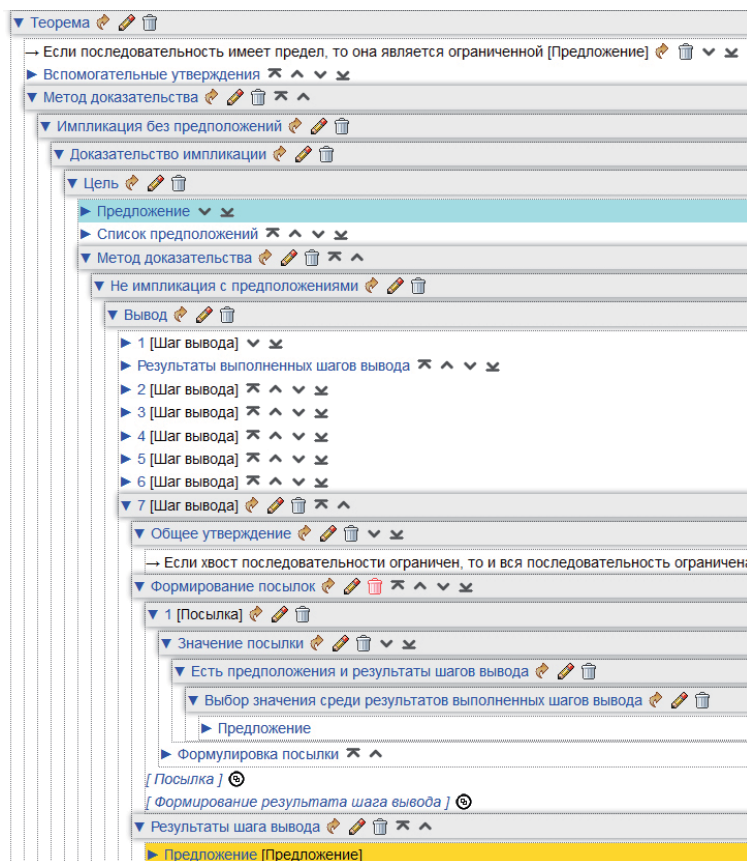


Рисунок 13 – Работа прикладного сервиса при формировании доказательства теоремы (пример)

С этого момента пользователь-исследователь может независимо развивать свою персональную базу математических знаний и базу способов рассуждений, а также язык представления математических знаний в Персональном Фонде с помощью соответствующих редакторов и доказывать математические утверждения.

6 Изменение текущего состояния портала знаний администратором

Изменение текущего состояния портала знаний может выполняться двумя способами: по инициативе пользователя либо по инициативе администратора. В обоих случаях для этого предназначен специальный сервис администратора, с помощью которого последний может получить доступ на чтение к персональным базам личных порталов пользователей. Данный сервис позволяет просматривать персональные базы математических знаний, показывая только те математические утверждения (вместе с разделами, в которых они находятся), доказательства которых являются полными. Эти доказательства должны быть рекурсивно-полными, т.е. используемые в них математические утверждения, отсутствующие в общей базе математических знаний, тоже должны иметь полные доказательства.

Если в доказательствах используются способы рассуждений, отсутствующие в общей базе способов рассуждений, то они должны быть верифицированы администратором, прежде чем эти способы попадут в соответствующие общие базы.

Если при формулировке некоторых математических утверждений используются конструкции языка представления математических знаний, отсутствующие в общей онтологии базы математических знаний, и администратор согласен с этими утверждениями, то они включаются в общую онтологию.

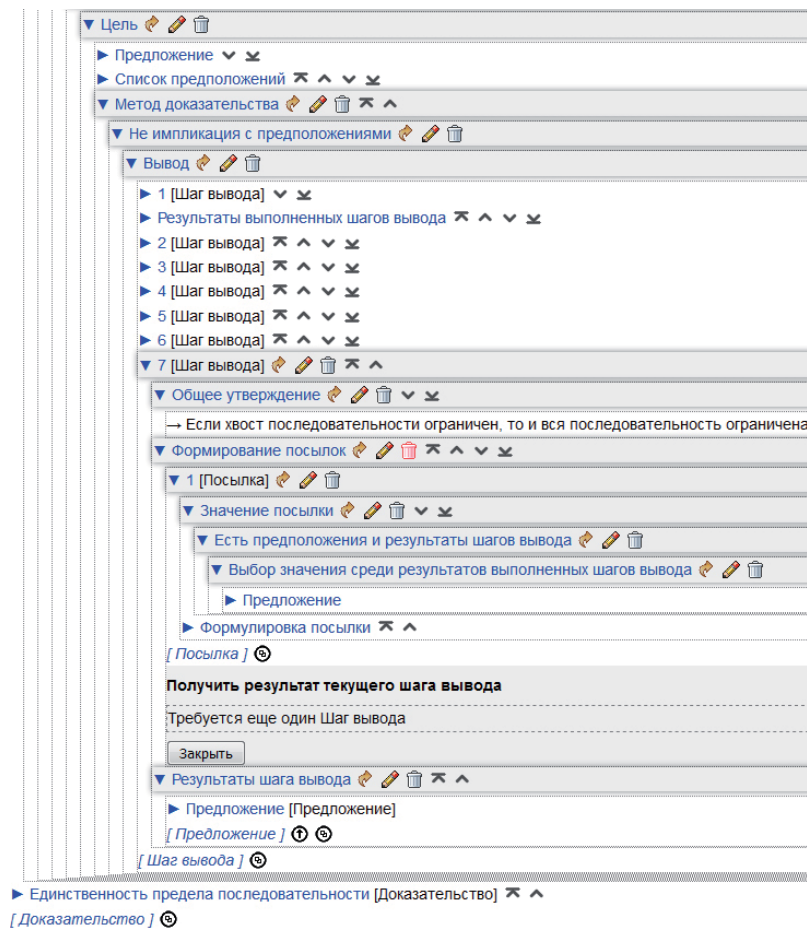


Рисунок 14 – Результат работы агента «Формирователь результата шага вывода» (пример)

Заключение

В работе представлена концептуальная архитектура оболочки для интерактивных систем верификации математических доказательств и создаваемого с её помощью развиваемого тематического портала знаний. Описан процесс реализации всех программных и информационных компонентов оболочки на облачной платформе IASaaS с использованием предоставляемых ею технологий и инструментальных средств их поддержки. Рассмотрен процесс разработки начального состояния портала знаний, способ использования текущего состояния портала знаний заинтересованными членами математического сообщества, а также механизмы изменения текущего состояния портала его администратором.

В основу оболочки и, как следствие, портала знаний положена явно представленная расширяемая формально-логическая система, приближенная к математической практике конструирования доказательств. Модель онтологии базы математических знаний, база математических знаний и база способов рассуждений расширяются с помощью соответствующих специализированных редакторов, входящих в состав портала знаний. При расширении модели онтологии новыми конструкциями языка представления знаний под эти изменения автоматически адаптируется управляемый ей соответствующий редактор, а решатель задач портала инвариантен по отношению к такому расширению.

В развитии портала знаний по верификации математических доказательств могут принимать участие все заинтересованные члены математического сообщества. Это осуществляется с помощью системы *личных кабинетов* платформы IASaaS, в которых

каждый пользователь может независимо развивать свою персональную копию текущего состояния общего портала знаний, и контролируется администратором портала знаний с помощью специального сервиса портала. Таким образом поддерживается синхронизация коллективного доступа к информационным компонентам общего портала и обеспечивается защита этих компонентов от их умышленного или непреднамеренного разрушения пользователями.

В результате развития портала может быть накоплена библиотека способов рассуждений, которыми математики пользуются в своей практике при доказательстве теорем. Эта библиотека может анализироваться специалистами по математической логике на предмет правильности содержащихся в ней способов рассуждений.

Благодарности

Работа выполнена при частичной поддержке РФФИ (проекты 17-07-00299 и 18-07-01079) и КПФИ «Дальний Восток» (проект 18-5-078).

Список источников

- [1] *Maric, F.* A Survey of Interactive Theorem Proving / F. Maric // Zbornik Radova, 2015, 18(26). Pp. 173-223.
- [2] *Asperti A.* A Survey on Interactive Theorem Proving. 2009. – <http://www.cs.unibo.it/~asperti/SLIDES/itp.pdf>.
- [3] *Harrison, J.* History of Interactive Theorem Proving / J. Harrison, J. Urban, F. Wiedijk // In Jörg Siekmann (ed.) Handbook of the History of Logic, 2014, vol. 9: Computational Logic. Elsevier. P.135-214.
- [4] *Paulson, L.* Logic and Computation: Interactive Proof with Cambridge LCF / L.C. Paulson // Cambridge University Press, New York, 1987.
- [5] *Owre, S.* PVS: A prototype verification system / S. Owre, J.M. Rushby, N. Shankar // In D. Kapur (ed.) Automated Deduction – CADE-11, LNCS 607, 1992, Springer, Berlin-Heidelberg. P.748-752.
- [6] *Bertot, Y.* Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions / Y. Bertot, P. Cast'eran // Springer, Berlin-Heidelberg, 2004.
- [7] About The Twelf Project. – <http://twelf.org/>.
- [8] *Harrison, J.* HOL Light: An Overview / J. Harrison // In S. Berghofer et al. (eds.) Theorem Proving in Higher Order Logics, LNCS 5674, 2009, Springer, Berlin-Heidelberg. - P.60-66.
- [9] *Рязанов, А.Е.* Система Буцефал: комбинирование дедуктивных процедур и описание стратегий поиска доказательств / А.Е. Рязанов. – Новосибирск, 1998. Препр./ Сиб. Отд-ние РАН. ИСИ; №50 – 42 с.
- [10] *Клещев, А.С.* Теоретические основы оболочки для интерактивных систем верификации интуитивных математических доказательств / А.С. Клещев, В.А. Тимченко // Онтология проектирования. - 2018. - Т. 8, № 2(28). - С. 219-239. - DOI: 10.18287/2223-9537-2018-8-2-219-239.
- [11] *Ring, M.* Collaborative Interactive Theorem Proving with Clide / M. Ring, C. Lüth // In Klein G., Gamboa R. (eds) Interactive Theorem Proving. ITP 2014. Lecture Notes in Computer Science, 2014, vol. 8558. Springer, Cham. P.467-482.
- [12] *Obua, S.* ProofPeer – A Cloud-based Interactive Theorem Proving System. 2012. – <https://arxiv.org/ftp/arxiv/papers/1201/1201.0540.pdf>.
- [13] *Gribova, V.* The IACPaaS cloud platform: features and perspectives / V. Gribova, A. Kleschev, Ph. Moskalenko, V. Timchenko, L. Fedorischev, E. Shalfeeva // In proc. of Second Russia and Pacific Conference on Computer Technology and Applications (25-29 Sept. 2017, Vladivostok, Russky Island, Russia). IEEE, 2017. Pp.80-84. ISBN: 978-153861206-4 - DOI: 10.1109/RPC.2017.8168073.
- [14] *Kumar, S.* Importance of Expert System Shell in Development of Expert System / S. Kumar, R. Prasad // International journal of innovative research & development, 2015, vol. 4, issue 3. P.128-133.
- [15] *Загоруйко, Ю.А.* Семантическая технология разработки интеллектуальных систем, ориентированная на экспертов предметной области / Ю.А. Загоруйко // Онтология проектирования. - 2015. - Т. 5, № 1(15). - С.30-46.
- [16] *Грибова, В.В.* Управляемая графовыми грамматиками разработка оболочек интеллектуальных сервисов на облачной платформе IACPaaS / В.В. Грибова, А.С. Клещёв, Ф.М. Москаленко, В.А. Тимченко, Л.А. Федорищев, Е.А. Шалфеева // Программная инженерия. - 2017. Т.8, №10. - С.435-447.

- [17] **Gribova, V.V.** A Two-level Model of Information Units with Complex Structure that Correspond to the Questioning Metaphor / V.V. Gribova, A.S. Kleshev, F.M. Moskalenko, V.A. Timchenko // Automatic Documentation and Mathematical Linguistics. 2015. Vol. 49. No.5. - P.172-181.
- [18] **Knight, K.** Unification: A Multidisciplinary Survey / K. Knight // ACM Computing Surveys, 1989, 21(1). - P.93-124.

IMPLEMENTATION OF THE SHELL AND KNOWLEDGE PORTAL FOR MATHEMATICAL PROOFS VERIFICATION ON THE IACPaaS PLATFORM

A.S. Kleshev¹, V.A. Timchenko²

Institute of Automation and Control Processes of the FEB RAS, Vladivostok, Russia

¹kleshev@iacp.dvo.ru, ²vadim@dvo.ru

Abstract

The paper presents a conceptual architecture of the shell for interactive systems for mathematical proofs verification and an evolutionary thematic knowledge portal created with its help. The process of implementation of all software and information components of the shell on the IACPaaS cloud platform with the use of its technologies and tools of their support is described. The process of development of the initial state of the knowledge portal for mathematical proofs verification using shell tools is considered. The way to use the knowledge portal by members of the mathematical community as well as mechanisms for changing the state of the portal by its administrator is discussed. The initial state of the knowledge portal includes: the ontology model of the mathematical knowledge base, including the specification of the initial state of the language for mathematical knowledge representation, the editor for the ontology model of the mathematical knowledge base, the mathematical knowledge base editor, the base of methods of reasoning editor, and the problem solver of the shell that implements the process of constructing proofs in terms of ontology model of proofs. The initial state of the knowledge portal also includes the initial state of the mathematical knowledge base and the initial state of the base of the methods of reasoning. The evolution of the knowledge portal consists in the development of the three listed information components. All interested members of the mathematical community can participate in this process. It is implemented with the use of personal cabinets of the IACPaaS platform, where each user can independently develop his personal copy of the current state of the common knowledge portal. The transfer of new results to the common knowledge portal is controlled by its administrator.

Key words: verification of intuitive proofs, specialized software shell, knowledge portal, knowledge base editor, cloud services.

Citation: Kleshev AS, Timchenko VA. Implementation of the shell and knowledge portal for mathematical proofs verification on the IACPaaS platform [In Russian]. *Ontology of designing*. 2018; 8(3): 427-448. - DOI: 10.18287/2223-9537-2018-8-3-427-448.

Acknowledgment

The work was partially supported by the Russian Foundation for Basic Research (projects 17-07-00299 and 18-07-01079) and the KFD "Far East" (project 18-5-078).

References

- [1] **Maric F.** A Survey of Interactive Theorem Proving. *Zbornik Radova*. 2015; 18(26): 173-223.
- [2] **Asperti A.** A Survey on Interactive Theorem Proving. 2009. - <http://www.cs.unibo.it/~asperti/SLIDES/itp.pdf>.
- [3] **Harrison J, Urban J, Wiedijk F.** History of Interactive Theorem Proving. In Jörg Siekmann (ed.) *Handbook of the History of Logic, Computational Logic*. Elsevier; 2014; 9: 135-214.
- [4] **Paulson LC.** *Logic and Computation: Interactive Proof with Cambridge LCF*. Cambridge University Press, New York, 1987.
- [5] **Owre S, Rushby JM, Shankar N.** PVS: A prototype verification system. In D. Kapur (ed.) *Automated Deduction – CADE-11, LNCS 607*, Springer, Berlin-Heidelberg; 1992; 748-752.

- [6] **Bertot Y, Cast'eran P.** Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions, Springer, Berlin-Heidelberg, 2004.
- [7] About The Twelf Project. Source: <http://twelf.org/>.
- [8] **Harrison J.** HOL Light: An Overview. In S. Berghofer et al. (eds.) Theorem Proving in Higher Order Logics, LNCS 5674, Springer, Berlin-Heidelberg; 2009; 60-66.
- [9] **Ryazanov AE.** System butsephalas: defining proof-search strategies and combining deductive procedures [In Russian]. Novosibirsk: Prepr. / Sib. Branch of the RAS. IIS; 1998; 50: 42.
- [10] **Kleshev AS, Timchenko VA.** Theoretical foundations of the shell for interactive systems of intuitive mathematical proofs verification [In Russian]. Ontology of designing. 2018; 8(2): 219-239. DOI: 10.18287/2223-9537-2018-8-2-219-239.
- [11] **Ring M, Lüth C.** Collaborative Interactive Theorem Proving with Clide. In Klein G., Gamboa R. (eds) Interactive Theorem Proving. ITP 2014. Lecture Notes in Computer Science, 2014, vol. 8558. Springer, Cham. P.467-482.
- [12] **Obua S.** ProofPeer – A Cloud-based Interactive Theorem Proving System. 2012. Source: <https://arxiv.org/ftp/arxiv/papers/1201/1201.0540.pdf>.
- [13] **Gribova V, Kleshev A, Moskalenko Ph, Timchenko V, Fedorischev L, Shalfeyeva E.** The IACPaaS cloud platform: features and perspectives. In proc. of Second Russia and Pacific Conference on Computer Technology and Applications (25-29 Sept. 2017, Vladivostok, Russky Island, Russia). IEEE; 2017: 80-84. ISBN: 978-153861206-4 DOI: 10.1109/RPC.2017.8168073.
- [14] **Kumar S, Prasad R.** Importance of Expert System Shell in Development of Expert System. International journal of innovative research & development 2015; 4(3): 128-133.
- [15] **Zagorulko YuA.** Semantic technology for development of intelligent systems oriented on experts in subject domain [In Russian]. Ontology of designing. 2015; 5(1): 30-46.
- [16] **Gribova VV, Kleshchev AS, Moskalenko FM, Timchenko VA, Fedorishchev LA, Shalfeyeva EA.** A Graph Grammar Managed Development of Intelligent Service Shells on the IACPaaS Cloud Platform [In Russian]. Software Engineering 2017; 10: 435-447.
- [17] **Gribova VV, Kleshchev AS, Moskalenko FM, Timchenko VA.** A Two-level Model of Information Units with Complex Structure that Correspond to the Questioning Metaphor. Automatic Documentation and Mathematical Linguistics. 2015; 49(5): 172-181.
- [18] **Knight K.** Unification: A Multidisciplinary Survey. ACM Computing Surveys 1989; 21(1): 93-124.

Сведения об авторах



Клещев Александр Сергеевич, 1940 г. рождения. Окончил математико-механический факультет Ленинградского государственного университета в 1964 г., д.ф.-м.н. (1990). Главный научный сотрудник лаборатории интеллектуальных систем Института автоматизации и процессов управления Дальневосточного отделения РАН, профессор, заслуженный деятель науки РФ. В списке научных трудов более 300 работ в области искусственного интеллекта, информатики, медицинской и биологической кибернетики.

Alexander Sergeevich Kleshev (b. 1940) graduated from the Leningrad State University in 1964, Professor's degree (1990). He is Chief Researcher at lab. of intelligent systems in the Institute of Automation and Control Processes of the FEB RAS, professor, Honored Scientist of Russian Federation. He is co-author of more than 300 publications in the field of biological and medical cybernetics, informatics and AI.



Тимченко Вадим Андреевич, 1983 г. рождения. Окончил Институт математики и компьютерных наук Дальневосточного государственного университета по специальности «Математическое обеспечение и администрирование информационных систем» (2005), к.т.н. (2011). Старший научный сотрудник лаборатории интеллектуальных систем Института автоматизации и процессов управления Дальневосточного отделения РАН. В списке научных трудов более 50 работ в области искусственного интеллекта, проблемно-ориентированных систем, основанных на знаниях, специализированных программных моделей и систем.

Vadim Andreevich Timchenko (b. 1983) graduated from the Far Eastern State University (Vladivostok-city) on a speciality «Mathematical support and administration of the informative systems» (2005), Ph.D. (2011). He is Senior Researcher at lab. of intelligent systems in the Institute

of Automation and Control Processes of the FEB RAS. He is co-author of more than 50 publications in the field of AI, informatics, program models, technologies and systems.

УДК 519.5

МЕТОД УНИВЕРСАЛЬНЫХ КОЭФФИЦИЕНТОВ ПРИ ПРИНЯТИИ МНОГОКРИТЕРИАЛЬНЫХ РЕШЕНИЙ

С.А. Пиявский

Поволжский государственный университет телекоммуникаций и информатики, Самара, Россия
spiyav@mail.ru

Аннотация

Проблема многокритериального выбора является ключевым элементом принятия сложных решений и уже более полувека не теряет актуальности. Предложен целый ряд подходов и методов, позволяющих предполагать, что принимаемые с их использованием решения наиболее рациональны. Представлен обзор наиболее распространённых методов и их приложений. Основным элементом большинства этих методов является линейная свёртка частных критериев, а их различие состоит в тех или иных эвристических или экспертных способах задания числовых коэффициентов важности критериев. Автором разработан подход, который позволяет использовать при формировании линейной свёртки заранее рассчитанные универсальные таблицы числовых коэффициентов важности частных критериев, что существенно уменьшает как трудоёмкость процесса подготовки принятия решения, так и неизбежный субъективизм, возникающий при эвристическом подборе или экспертном назначении её коэффициентов. Разработанный подход формирования универсальных коэффициентов важности для каждого вида свёртки критериев в различных публикациях называется как минимаксная, гарантирующая свёртка (свёртка Гермейера). Предложенный новый общий метод принятия решений и сравнения многокритериальных альтернатив основан на совместном использовании обоих видов свёрток. Его применение продемонстрировано на двух практически важных задачах – рейтинговой оценке университетов и анализе различных проектных концепций высотных беспилотных летательных аппаратов.

Ключевые слова: принятие решений, многокритериальный выбор, универсальные коэффициенты важности, минимакс.

Цитирование: Пиявский, С.А. Метод универсальных коэффициентов при принятии многокритериальных решений / С.А. Пиявский // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.449-468. – DOI: 10.18287/2223-9537-2018-8-3-449-468.

Постановка задачи

Рассмотрим классическую постановку задачи многокритериальной оптимизации. Пусть

- Y - множество допустимых альтернатив;
- $f_i(y)$, $y \in Y$, $i = 1, \dots, n$ - частные критерии оптимальности альтернатив $y \in Y$, для определённости желательным является уменьшение значений частных критериев:

$$(1) \quad f_i(y) \longrightarrow \min, \quad i = 1, \dots, n,$$

и также

$$(2) \quad 0 \leq f_i(y) \leq 1, \quad i = 1, \dots, n.$$

Требуется найти «наиболее рациональную» альтернативу $\bar{y} \in Y$, удовлетворяющую интуитивному представлению лица, принимающего решение (ЛПР), о том, что она «в наибольшей степени» отвечает условию (1).

Среди формальных методов, направленных на то, чтобы облегчить ЛПР решение этой задачи (например, [1-7]), наиболее широко распространены методы формирования на основе

системы частных критериев $f_i(y)$, $i = 1, \dots, n$ некоторого комплексного скалярного критерия (свёртки) $F(f_1(y), \dots, f_n(y)) \equiv F(y)$, отражающего условие (1) в форме, наиболее адекватной, по мнению ЛПР, его интуитивному представлению о «наиболее рациональной» альтернативе. В этом случае искомая альтернатива $\bar{y} \in Y$ определяется чисто математически из условия

$$(3) \quad F(\bar{y}) = \min_{y \in Y} F(y).$$

На практике наиболее часто используются два вида свёртки:

$$(4) \quad F = \sum_{i=1}^n \alpha_i f_i \text{ - средневзвешенная (линейная),}$$

$$(5) \quad F = \max_{i=1, \dots, n} \alpha_i f_i \text{ - гарантирующая (свёртка Гермейера).}$$

В (4), (5) α_i , $i = 1, \dots, n$ - коэффициенты важности соответствующих частных критериев, удовлетворяющие условиям

$$(6) \quad \begin{aligned} \alpha_i &\geq 0, \quad i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i &= 1 \end{aligned}$$

При таком подходе формальная незамкнутость задачи принятия решения переносится на проблему определения числовых значений коэффициентов важности частных критериев α_i , $i = 1, \dots, n$. Однако, как будет ниже показано, в этой сложной и мало достоверной операции нет необходимости, если ЛПР попросту отнесёт частные критерии к различным группам сравнительной важности (возможно, по несколько критериев в одну и ту же группу). Это позволит ему естественным путём формализовать своё интуитивное представление о сравнительной важности критериев, не затрудняясь числовой оценкой.

1 Метод вычисления коэффициентов важности критериев в гарантирующей свёртке

Обозначим:

q - номер группы важности критериев (более важной группе соответствует больший номер);

Q - количество групп важности критериев, тем самым одновременно и номер наиболее важной из групп важности;

$r(i)$ - номер группы важности, к которой отнесён критерий $f_i(y)$, $i = 1, \dots, n$, $r(i) \in \{1, \dots, Q\}$;

n_q - количество частных критериев, входящих в группу важности q , $q = 1, \dots, Q$.

$$(7) \quad n_q = \sum_{\substack{i=1 \\ r(i)=q}}^n 1$$

$$(8) \quad \sum_{q=1}^Q n_q = n.$$

Очевидно, что учёт групп важности критериев добавляет к ограничениям (6) дополнительные ограничения

$$(9) \quad \alpha_k \geq \alpha_j \quad \forall r, j: r(k) > r(j), \quad k = 1, \dots, n, \quad j = 1, \dots, n.$$

Тем самым в рассмотрение вводятся не конкретные числовые значения коэффициентов важности $\alpha_i, i=1, \dots, n$, а некоторое множество S их допустимых значений, определяемое условиями (6), (9).

Как использовать с учётом этого свёртки (4), (5)?

Случай линейной свёртки рассмотрен в [8,9]. Исходя из интегративного характера этой свёртки, основанной на осреднении взвешенных значений частных критериев, предлагается перейти к рассмотрению взвешенных значений этих критериев на всём множестве S . Это позволит получить геометрическую интерпретацию результирующих коэффициентов важности частных критериев как координат центра масс множества S в n -мерном пространстве. Соответственно становится возможным рассчитать универсальные таблицы (наподобие таблицы умножения или логарифмов) коэффициентов важности критериев в зависимости от числа частных критериев и их распределения по группам важности. Таким образом, использование линейной свёртки при принятии решений стало чрезвычайно простым делом для ЛПР: он освобождён от необходимости определения значений коэффициентов важности, ему достаточно распределить частные критерии по группам важности, а затем найти соответствующие значения коэффициентов важности в универсальной таблице.

В настоящей статье рассмотрен случай гарантирующей свёртки. Учитывая, что эта свёртка ориентирована на наихудшее (наименьшее) из взвешенных значений частных критериев, естественно ориентироваться на максимизацию этих значений на множестве S :

$$(10) \quad F = \max_{i=1, \dots, n} \max_{\alpha \in S} \alpha_i f_i \equiv \max_{i=1, \dots, n} (\max_{\alpha \in S} \alpha_i) f_i,$$

где $\alpha = (\alpha_1, \dots, \alpha_n)$. Максимальное на S значение $\max_{i=1, \dots, n} \alpha_i$ достигается при максимально возможном значении α_i из вектора α , удовлетворяющего (6) и (9).

Если представить значения $\alpha_k, k=1, \dots, n$ как результат распределения некоторого единичного ресурса (в соответствии с (6)) между частными критериями, то на i -й критерий должна быть направлена максимально возможная величина этого ресурса, допускаемая (9). Для этого на критерии, относящиеся к меньшим (по номерам) группам важности, чем группа важности критерия с номером i , следует направить нулевой ресурс, так же как и на критерии, имеющие ту же группу важности, что и i -й критерий. Что же касается критериев из групп большей важности, чем группа важности критериев с номером i , то ресурс, направляемый на каждый из них, должен быть равен ресурсу, направляемому на критерий с номером i , так как, в соответствии с (9), он не может быть меньше и не обязан быть больше.

Исходя из этого, можно записать для критериев, не относящихся к группе максимальной важности с номером Q .

$$\alpha_i + \alpha_i \sum_{q=r(i)+1}^n n_q = 1$$

Если же критерий с номером i относится к группе максимальной важности, то его коэффициент важности должен быть принят за 1, в то время как коэффициенты важности всех остальных критериев, в том числе и входящих в ту же группу максимальной важности, должны быть приняты нулевыми:

$$\alpha_i = 1 \quad \text{при } r(i) = Q.$$

Таким образом,

$$(11) \quad \alpha_i = \begin{cases} \frac{1}{1 + \sum_{q=r(i)+1}^Q n_q} & \text{при } r(i) < Q \\ 1 & \text{при } r(i) = Q \end{cases}, \quad i = 1, \dots, n.$$

Смысл комплексного критерия (10) не изменится, если его умножить на любое положительное число C . Выберем C так, чтобы сумма весовых коэффициентов при критериях в (10) равнялась единице:

$$C \left(\sum_{i=1}^n \frac{1}{1 + \sum_{\substack{q=r(i)+1 \\ r(i) < Q}}^Q n_q} + \sum_{\substack{i=1 \\ r(i)=Q}}^n 1 \right) = 1.$$

Отсюда

$$(12) \quad C = \frac{1}{\sum_{\substack{i=1 \\ r(i) < Q}}^n \frac{1}{1 + \sum_{\substack{q=r(i)+1 \\ r(i) < Q}}^Q n_q} + n_Q}.$$

Теперь гарантирующая свёртка приобретает вид

$$(13) \quad F = \max_{i=1, \dots, n} \beta_i f_i,$$

где коэффициенты важности критериев, с учётом (11) и (12), равны

$$(14) \quad \beta_i = \begin{cases} \frac{C}{1 + \sum_{q=r(i)+1}^Q n_q} & \text{при } r(i) < Q \\ C & \text{при } r(i) = Q \end{cases}, \quad i = 1, \dots, n,$$

при этом

$$(15) \quad \begin{aligned} \beta_i &\geq 0, \quad i = 1, \dots, n \\ \sum_{i=1}^n \beta_i &= 1 \end{aligned}$$

Пример расчёта коэффициентов важности критериев в гарантирующей свёртке приведён в таблице 1. Рассмотрен случай пяти частных критериев, из которых первый критерий отнесён ЛПР-ом к первой группе важности, второй и третий критерии – ко второй группе важности, остальные два критерия – соответственно к третьей и четвёртой группам важности.

Таблица 1 – Пример расчёта коэффициентов важности критериев

Номера критериев i	Номера групп важности критериев				$r(i)$	$\sum_{\substack{q=r(i)+1 \\ r(i)<Q}}^Q n_q$	$\frac{1}{1 + \sum_{\substack{q=r(i)+1 \\ r(i)<Q}}^Q n_q}$	$C = \frac{1}{\sum_{\substack{i=1 \\ r(i)<Q}}^n \frac{1}{1 + \sum_{\substack{q=r(i)+1 \\ r(i)<Q}}^Q n_q} + n_Q}$	β_i
	1	2	3	4					
1	*				1	4	$1/5=0,2$	$\frac{1}{1,367+1} = 0,4225$	0,0845
2		*			2	2	$1/3=0,33\dots$		0,1408
3			*		2	2	$1/3=0,33\dots$		0,1408
4				*	3	1	$1/2=0,5$		0,2113
5					4	-	-		0,4225
n_q	1	2	1	1		Сумма	1,367		

2 Универсальные таблицы коэффициентов важности критериев в свёртках критериев

Как видно из предыдущего пункта, алгоритм вычисления коэффициентов важности критериев в отдельной задаче принятия решения чрезвычайно прост, однако от него можно вообще отказаться, используя заранее вычисленные универсальные таблицы коэффициентов важности критериев для гарантирующей свёртки, подобные тем, которые рассчитаны в [8] для средневзвешенной свёртки. Для расчёта таких таблиц преобразуем (12), сгруппировав в знаменателе слагаемые под знаком суммы по равным значениям $r(i)$:

$$\sum_{\substack{i=1 \\ r(i)<Q}}^n \frac{1}{1 + \sum_{\substack{q=r(i)+1 \\ r(i)<Q}}^Q n_q} = \sum_{q=1}^{Q-1} \frac{n_q}{1 + \sum_{t=q+1}^{Q-1} n_t}.$$

Тогда

$$(16) \quad C = \frac{1}{\sum_{q=1}^{Q-1} \frac{n_q}{1 + \sum_{t=q+1}^{Q-1} n_t} + n_Q}$$

и вычисление коэффициентов важности β_q^* критериев, входящих в различные группы важности q , $q = 1, \dots, Q$ производится по формулам:

$$(17) \quad \beta_q^* = \begin{cases} \frac{C}{1 + \sum_{t=q+1}^Q n_t} & \text{при } q < Q \\ C & \text{при } q = Q \end{cases} \quad q = 1, \dots, Q.$$

В таблицах 2-4 приведены рассчитанные по (16), (17) универсальные таблицы весовых коэффициентов критериев в гарантирующей свёртке для числа критериев от двух до шести. Для удобства комплексного использования при решении практических задач в этих же таблицах приведены универсальные коэффициенты важности критериев и для средневзвешенной свёртки, полученные в [8].

Таблица 2 – Универсальные коэффициенты важности критериев для гарантирующей и средневзвешенной свёртки с двумя, тремя и четырьмя критериями (рассчитаны по точным формулам)

Число частных критериев в задаче принятия решений	Количество критериев в каждой группе важности				Коэффициенты важности критериев для гарантирующей свёртки				Коэффициенты важности критериев для средневзвешенной свёртки			
	Группа важности критериев				Группа важности критериев				Группа важности критериев			
	B1	B2	B3	B4	B1	B2	B3	B4	B1	B2	B3	B4
2	1	1			0,333	0,666			0,250	0,750		
	2				0,500				0,500			
3	1	1	1		0,182	0,273	0,545		0,111	0,278	0,611	
	1	2			0,143	0,429			0,111	0,444		
	2	1			0,250	0,500			0,194	0,611		
	3				0,333				0,333			
4	1	1	1	1	0,120	0,160	0,240	0,480	0,063	0,146	0,271	0,521
	1	1	2		0,097	0,129	0,387		0,063	0,146	0,396	
	1	2	1		0,111	0,222	0,444		0,063	0,208	0,521	
	2	1	1		0,154	0,231	0,462		0,104	0,271	0,521	
	1	3			0,077	0,308			0,063	0,313		
	2	2			0,125	0,375			0,104	0,396		
	3	1			0,200	0,400			0,167	0,521		
	4				0,250				0,250			

Таблица 3 – Универсальные коэффициенты важности критериев для гарантирующей и средневзвешенной свёртки с пятью критериями (рассчитаны приближённо)

Количество критериев в каждой группе важности					Коэффициенты важности критериев для гарантирующей свёртки					Коэффициенты важности критериев для средневзвешенной свёртки				
Группа важности критериев					Группа важности критериев					Группа важности критериев				
B1	B2	B3	B4	B5	B1	B2	B3	B4	B5	B1	B2	B3	B4	B5
1	1	1	1	1	0,088	0,109	0,146	0,219	0,438	0,038	0,087	0,154	0,256	0,464
1	1	1	2		0,072	0,090	0,120	0,359		0,038	0,087	0,153	0,361	
1	1	2	1		0,082	0,102	0,204	0,408		0,038	0,087	0,205	0,464	
1	1	3			0,058	0,072	0,290			0,038	0,085	0,292		
1	2	1	1		0,085	0,141	0,211	0,423		0,038	0,121	0,255	0,466	
1	2	2			0,070	0,116	0,349			0,038	0,121	0,361		
1	3	1			0,074	0,185	0,370			0,038	0,165	0,466		
1	4				0,048	0,238				0,037	0,238			
2	1	1	1		0,107	0,143	0,214	0,429		0,064	0,155	0,254	0,464	
2	1	2			0,088	0,118	0,353			0,063	0,153	0,361		
2	2	1			0,100	0,200	0,400			0,063	0,204	0,467		
2	3				0,071	0,286				0,062	0,292			
3	1	1			0,133	0,200	0,400			0,093	0,254	0,465		
3	2				0,111	0,333				0,094	0,359			
4	1				0,167	0,333				0,135	0,460			
5					0,200					0,200				

Таблица 4 – Универсальные коэффициенты важности критериев для гарантирующей и средневзвешенной свёртки с шестью критериями (рассчитаны приближённо)

Количество критериев в каждой группе важности							Коэффициенты важности критериев для гарантирующей свёртки						Коэффициенты важности критериев для средневзвешенной свёртки					
Группа важности критериев							Группа важности критериев						Группа важности критериев					
B1	B2	B3	B4	B5	B6	B1	B2	B3	B4	B5	B6	B1	B2	B3	B4	B5	B6	
1	1	1	1	1	1	0,068	0,082	0,102	0,136	0,204	0,408	0,026	0,057	0,098	0,154	0,240	0,427	
1	1	1	1	2		0,056	0,068	0,085	0,113	0,339		0,026	0,058	0,099	0,154	0,332		
1	1	1	2	1		0,064	0,076	0,096	0,191	0,382		0,026	0,058	0,099	0,198	0,422		
1	1	1	3			0,046	0,055	0,069	0,276			0,026	0,057	0,098	0,273			
1	1	2	1	1		0,066	0,079	0,132	0,197	0,395		0,026	0,058	0,128	0,241	0,420		
1	1	2	2			0,055	0,066	0,110	0,330			0,026	0,057	0,127	0,332			
1	1	3	1			0,058	0,070	0,174	0,349			0,026	0,057	0,165	0,421			
1	1	4				0,038	0,046	0,229				0,025	0,056	0,230				
1	2	1	1	1		0,067	0,100	0,133	0,200	0,400		0,026	0,079	0,157	0,24	0,418		
1	2	1	2			0,056	0,083	0,111	0,333			0,026	0,079	0,155	0,330			
1	2	2	1			0,063	0,094	0,188	0,375			0,026	0,079	0,199	0,419			
1	2	3				0,045	0,068	0,273				0,025	0,078	0,273				
1	3	1	1			0,063	0,125	0,188	0,375			0,025	0,105	0,241	0,417			
1	3	2				0,053	0,105	0,316				0,025	0,104	0,330				
1	4	1				0,053	0,158	0,316				0,025	0,139	0,418				
1	5					0,032	0,194					0,025	0,195					
2	1	1	1	1		0,081	0,101	0,134	0,201	0,403		0,043	0,101	0,156	0,239	0,417		
2	1	1	2			0,067	0,084	0,112	0,335			0,043	0,101	0,155	0,330			
2	1	2	1			0,075	0,094	0,189	0,377			0,043	0,101	0,198	0,418			
2	1	3				0,055	0,068	0,274				0,043	0,099	0,272				
2	2	1	1			0,078	0,130	0,195	0,390			0,043	0,129	0,24	0,416			
2	2	2				0,065	0,109	0,326				0,042	0,128	0,330				
2	3	1				0,069	0,172	0,345				0,042	0,166	0,417				
2	4					0,045	0,227					0,042	0,229					
3	1	1	1			0,097	0,129	0,194	0,387			0,063	0,157	0,239	0,414			
3	1	2				0,081	0,108	0,324				0,062	0,156	0,328				
3	2	1				0,091	0,182	0,364				0,063	0,199	0,415				
3	3					0,067	0,267					0,062	0,271					
4	1	1				0,118	0,176	0,353				0,087	0,241	0,413				
4	2					0,100	0,300					0,086	0,328					
5	1					0,143	0,286					0,117	0,414					
6						0,167						0,167						

3 Пример расчёта рейтинга университетов с использованием универсальных таблиц коэффициентов важности критериев

Рассмотрим использование универсальных коэффициентов важности критериев на конкретном примере. В его основу положены данные Национального рейтинга университетов России за 2018 год [10]. В демонстрационных целях ограничимся данными десяти передовых университетов из 164 университетов, представленных в [10]. Далее они обозначаются кодами от У1 до У10. Подчеркнём, что целью примера является исключительно рассмотрение возможностей предлагаемого метода универсальных коэффициентов, а не реальная «расстановка» конкретных университетов.

В [10] университеты сравниваются по шести сферам: образование; исследования; социальная среда; интернационализация; инновации и предпринимательство; бренд университета.

В каждой из них используется ряд критериев с использованием линейной комбинации с назначенными весовыми коэффициентами (т.е. средневзвешенной свёрткой критериев) и рассчитывается параметрический рейтинг университетов. Так же с использованием средневзвешенной свёртки на основе параметрических рейтингов в различных сферах рассчитывается сводный рейтинг университета. В таблице 5 в разделе «Используются в [10]» показаны применяемые в [10] весовые коэффициенты. В таблице 6 приведены баллы, рассчитанные на этой основе в [10] для десяти выбранных университетов. Используя эти данные по методике, принятой в [10], рассчитаны баллы для определения сводного рейтинга и определён сам сводный рейтинг. Эти результаты показаны в таблице 7 (второй и третий столбцы).

Таблица 5 – Сферы деятельности университетов и коэффициенты важности критериев

№	Параметрический показатель	К-во критериев	Используются в [10]								Предлагаются в настоящей статье							
			Распределе- ние критериев по группам важности				Коэффициенты важности критериев по НРУ2018 (%)				Коэффициенты важности критериев для средневзвешенной свёртки (%)				Коэффициенты важности критериев для гарантирующей свёртки (%)			
			В1	В2	В3	В4	В1	В2	В3	В4	В1	В2	В3	В4	В1	В2	В3	В4
0	Сводный рейтинг НРУ2018	6	4	2			15	20			8,6	32,8			10	30		
1	Оценка деятельности университета в сфере «Образование»	6	4	2			15	20			8,6	32,8			10	30		
2	Оценка деятельности университета в сфере «Исследования»	8	5	2	1		10	15	20		5,9	18,3	33,9		7,7	15,4	30,7	
3	Оценка деятельности университета в сфере «Социальная среда»	5					20				20				20			
4	Оценка деятельности университета в сфере «Интернационализация»	5	1	3	1		15	20	25		4	16,8	45,7		7,4	18,5	37	
5	Оценка деятельности университета в сфере «Инновации и Предпринимательство»	6	1	3	1	1	10	15	20	25	2,8	10,7	24,1	40,7	6,3	12,5	18,8	37,5
6	Оценка бренда университета	5	1	1	3		10	15	25		4	9	29		5,8	7,2	29	

Таблица 6 – Баллы и параметрический рейтинг университетов в различных сферах деятельности (по [10])

Университет	Балл университета в различных сферах деятельности:					бренд университета
	образование	исследования	социальная среда	интернационализация	инновации и предпринимательство	
У1	864	424	487	363	501	195
У2	842	364	512	465	691	147
У3	826	704	593	879	761	250
У4	937	586	822	862	584	309
У5	833	539	821	602	968	163
У6	874	450	882	835	729	494
У7	825	409	497	353	568	213
У8	885	653	785	791	713	255
У9	824	607	809	767	777	304
У10	831	450	788	565	517	330

Таблица 7 – Сводный рейтинг университетов при различных коэффициентах важности критериев

Университет	Рассчитан по методике [10] с принятым в ней соотношением коэффициентов важности критериев 2:1		Рассчитан по методике [10] с принятым в методе анализа иерархий Т.Саати соотношением коэффициентов важности критериев 3:1	
	Балл	Сводный рейтинг	Балл	Сводный рейтинг
У1	489	10	541	9
У2	513	8	543	8
У3	678	5	707	3
У4	691	2	714	2
У5	657	6	667	6
У6	705	1	691	5
У7	491	9	533	10
У8	689	3	715	1
У9	684	4	695	4
У10	586	7	604	7

Проанализируем приведённые данные.

1. Видно, что некоторые университеты (отмечены жирным шрифтом в столбце 1 таблицы 6) являются неэффективными по Парето и потому могут далее не рассматриваться, если требуемое решение сводится к выбору наиболее рациональной альтернативы. Это университеты У1 (доминируются университетами У4, У6, У8), У2 (У6, У8), У7 (У2, У4, У6, У8) и У10 (У6), однако они оставлены в рассмотрении, поскольку преимуществом использования методологии комплексной свёртки критериев является возможность линейной зависимости трудоёмкости процесса от количества альтернатив. Она позволяет использовать достаточно большое количество вариантов решения в отличие от квадратичной зависимости, характерной для методов принятия решений, основанных на попарном сравнении альтернатив.

2. Анализ коэффициентов важности различных критериев, численно заданных в [10] для расчёта комплексных критериев эффективности в рамках отдельных областей деятельности университетов и сводного рейтинга, демонстрирует, что авторы методики на самом деле использовали разбиение критериев на группы важности, а затем присвоили коэффициентам важности этих групп эмпирически выбранные значения. Этот подход реализуется в подавляющем числе практических задач. Его использование свидетельствует о том, что в реальности методика экспертного определения числовых значений коэффициентов важности критериев не имеет практической перспективы.

3. Данные, отвечающие принятому в [10] разбиению критериев на группы важности, восстановлены в таблице 5. Именно они должны считаться исходной информацией в рассматриваемой задаче принятия решения. Но в этом случае становится ясно, что результаты, полученные в [10], легко могут быть поставлены под сомнение, поскольку принятые в ней соответствующие числовые значения коэффициентов важности ничем не обоснованы. Например, при расчёте сводного рейтинга соотношение коэффициентов важности двух групп критериев: более важной и менее важной, - принято 2:1 (20% и 10%). Если ориентироваться на имеющий мировую известность и достаточно широко применяемый метод анализа иерархий Т. Саати [11-13], это соотношение должно быть 3:1 (30% и 10%)? А при таком соотношении результаты относительно первых мест становятся совершенно иными (таблица 7)! Университет У6, занимавший первое место по сводному рейтингу, оказывается лишь на пятом месте, а с третьего места на первое место перемещается университет У8, не имеющий, в частности, среди исходных баллов по отдельным областям деятельности ни одного рекордного значения.

4. В таблице 8 приведены значения комплексных критериев и отвечающие им рейтинги, рассчитанные с использованием коэффициентов важности методики [10], а также универсальных коэффициентов важности для средневзвешенной и гарантирующей свёртки. Как и следовало ожидать, полученные рейтинги разнятся. И при этом результаты, использующие универсальные коэффициенты, обладают более высокой достоверностью, поскольку исключено недеklarированное субъективное воздействие на них в процессе организации принятия решения.

Таблица 8 – Сводный рейтинг университетов при различных коэффициентах важности критериев

Университет	Рассчитан по методике [10] с принятыми в ней коэффициентами важности критериев 20% и 10%		Рассчитан для линейной свёртки с универсальными коэффициентами важности 32,8% и 8,6%		Рассчитан для гарантирующей свёртки с универсальными коэффициентами важности 30% и 10%	
	Балл	Сводный рейтинг	Свёртка	Сводный рейтинг	Свёртка	Сводный рейтинг
У1	489	10	555	8	259	4
У2	513	8	551	9	252	5
У3	678	5	715	3	247	8-10
У4	691	2	721	2	281	1
У5	657	6	669	6	249	6-7
У6	705	1	687	5	262	3
У7	491	9	545	10	247	8-10
У8	689	3	723	1	265	2
У9	684	4	697	4	247	8-10
У10	586	7	609	7	249	6-7

5. После перехода к универсальным коэффициентам важности критериев в методике принятия решения перед ЛПР остаётся лишь один вид неопределённости: какой из подходов: средневзвешенный или гарантирующий - он считает необходимым использовать при принятии решения. Но можно сразу указать, что, безусловно, его заинтересуют результаты применения обоих подходов, тем более, что использование таблиц универсальных коэффициентов делает это нетрудоёмким.

Исходными данными для дальнейшего анализа служит часть таблицы 8, выделенная жирным шрифтом. При этом целесообразно перейти от задачи на максимум к привычной для теории оптимизации задаче на минимум и к нормированным величинам, показывающим

долю относительного отклонения значения свертки $f(y)$ для варианта решения y между минимальным и максимальным значениями среди всех рассматриваемых вариантов решения Y :

$$\bar{f}(y) = 1 - \frac{f(y) - \min_{y \in Y} f(y)}{\max_{y \in Y} f(y) - \min_{y \in Y} f(y)}$$

Тогда данные, отвечающие таблице 8, будут представлены во втором и третьем столбцах таблицы 9. Учитывая, что при любом числе частных критериев таблица 9 содержит лишь два столбца данных – значения двух свёрток, удобно представить их графически (см. рисунок 1). На рисунке 1 видно, что Парето-оптимальными являются два университета: У4 и У8. После их исключения из рассмотрения Парето-оптимальными становятся У3 и У6, затем последовательно У9, У5, У1, далее У10, У2 и наконец У7. Таким образом, получен сводный рейтинг университетов, отражённый в четвёртом столбце таблицы 9.

Таблица 9 – Относительные значения средневзвешенной и гарантирующей свёрток критериев (относительно минимального значения среди всех рассматриваемых альтернатив) и сводный рейтинг университетов по Парето

Университет	Средневзвешенная свёртка	Гарантирующая свёртка	Сводный рейтинг университетов по Парето
У1	0,943	0,647	5-6
У2	0,966	0,853	7-8
У3	0,045	1	3-4
У4	0,011	0	1 -2
У5	0,303	0,941	5-6
У6	0,202	0,559	3-4
У7	1	1	9
У8	0	0,470	1 - 2
У9	0,146	1	5 - 6
У10	0,640	0,941	7-8

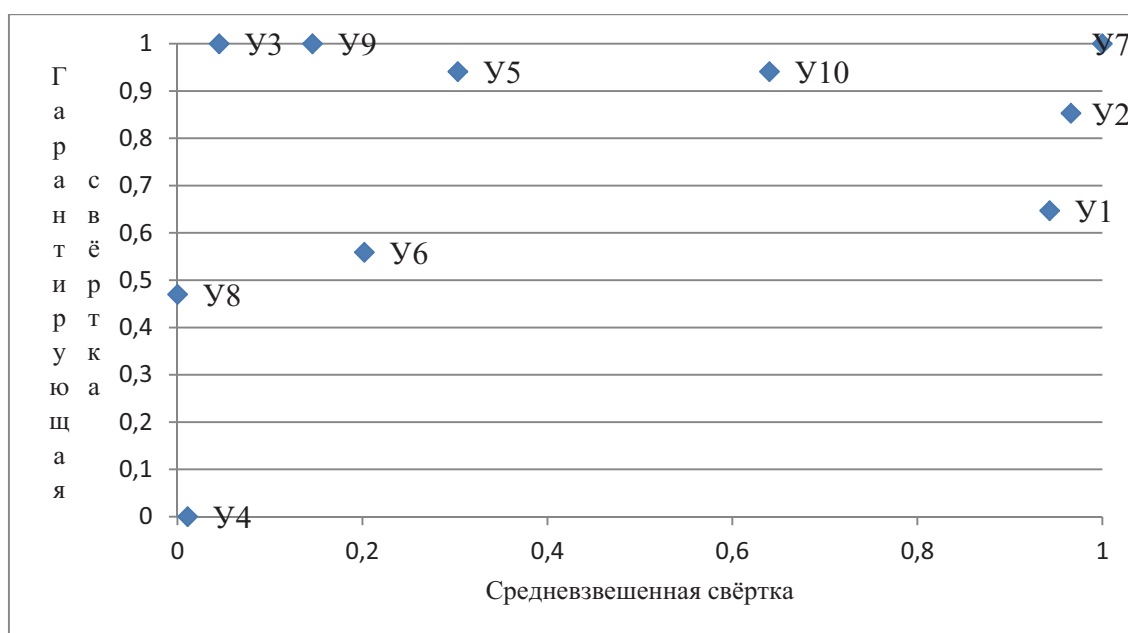


Рисунок 1 - Относительные значения средневзвешенной и гарантирующей свёрток критериев при сравнительной оценке университетов

Обратим внимание на то, что этот рейтинг обладает максимально возможной достоверностью, поскольку при его расчёте не пришлось требовать от ЛПП никакой дополнительной информации сверх исходных данных, приведённых в таблице 6, и распределения критериев по группам важности – ему даже не потребовалось определиться, какой из двух видов свёртки он считает предпочтительным! Столь подробного по детализации предпочтительности университетов рейтинга не удалось бы получить, если бы пытались изначально построить рейтинг, последовательно используя принцип оптимальности Парето. В этом случае в первый слой Парето-оптимальных альтернатив вошли бы сразу шесть университетов – все, которые не отмечены жирным шрифтом в таблице 6.

6. Можно ещё повысить детализацию рейтинга, приведённого в четвёртом столбце таблицы 9, если ЛПП сочтёт возможным выбрать одно из следующих суждений:

- использовать средневзвешенную свёртку;
- использовать гарантирующую свёртку;
- средневзвешенная свёртка предпочтительнее гарантирующей;
- гарантирующая свёртка предпочтительнее средневзвешенной;
- обе свёртки равноценны.

Тогда рейтинг определяется на основе вспомогательного критерия VK , который в первых двух случаях равен соответствующей свёртке. В третьем и четвёртом случаях естественно использовать универсальные коэффициенты важности двух критериев. Тогда в случае предпочтительности средневзвешенной свёртки

$$(18) \quad VK = \frac{3}{4}SK + \frac{1}{4}GK \quad ,$$

где SK и GK - значения соответственно средневзвешенной и гарантирующей свёрток, а в случае предпочтительности гарантирующей свёртки

$$(19) \quad VK = \max\left(\frac{1}{3}SK; \frac{2}{3}GK\right) \quad .$$

В пятом случае вспомогательный критерий следует принять равным среднему арифметическому средневзвешенной и гарантирующей свёрток. Результаты показаны в таблицах 10, 11.

Таблица 10 – Значения вспомогательного критерия при выраженном ЛПП отношении к видам свёртки критериев

Университет	Утверждение ЛПП				
	Использовать средневзвешенную свёртку	Использовать гарантирующую свёртку	Средневзвешенная свёртка предпочтительнее гарантирующей	Гарантирующая свёртка предпочтительнее средневзвешенной	Обе свёртки равноценны
У1	0,943	0,647	0,869	0,431	0,795
У2	0,966	0,853	0,938	0,568	0,9095
У3	0,045	1	0,284	0,667	0,522
У4	0,011	0	0,008	0,003	0,005
У5	0,303	0,941	0,463	0,627	0,622
У6	0,202	0,559	0,291	0,372	0,380
У7	1	1	1,000	0,667	1
У8	0	0,47	0,118	0,313	0,235
У9	0,146	1	0,360	0,667	0,573
У10	0,64	0,941	0,715	0,627	0,790

Таблица 11 – Сводный рейтинг университетов при выраженном ЛПР отношении к видам свёртки критериев

Университет	Утверждение ЛПР				
	Использовать средневзвешенную свёртку	Использовать гарантирующую свёртку	Средневзвешенная свёртка предпочтительнее гарантирующей	Гарантирующая свёртка предпочтительнее средневзвешенной	Обе свёртки равноценны
<i>У1</i>	8	4	8	4	8
<i>У2</i>	9	5	9	5	9
<i>У3</i>	3	8 - 10	3	7 - 10	4
<i>У4</i>	2	1	1	1	1
<i>У5</i>	6	6 - 7	6	6	6
<i>У6</i>	5	3	4	3	3
<i>У7</i>	10	8 - 10	10	7 - 10	10
<i>У8</i>	1	2	2	2	2
<i>У9</i>	4	8 - 10	5	7 - 10	5
<i>У10</i>	7	6 - 7	7	7 - 10	7

4 Метод универсальных коэффициентов

Изложенное выше суммируем в виде следующего метода универсальных коэффициентов при принятии многокритериальных решений.

Этап 1. ЛПР обеспечивает формирование набора альтернатив, перечня критериев их эффективности и таблицы значений критериев для различных альтернатив.

Этап 2. ЛПР относит критерии эффективности к различным группам важности.

Этап 3. С использованием таблиц универсальных коэффициентов важности критериев для средневзвешенной и гарантирующей свёрток (до шести критериев можно использовать таблицы 2 – 4 настоящей статьи) рассчитываются значения этих свёрток для всех альтернатив. При этом используются нормированные значения критериев, приведённые к требованию минимума. Если количество критериев превышает шесть, универсальные коэффициенты для средневзвешенной свёртки можно рассчитать по методике, приведённой в [8], а для гарантирующей свёртки – по формулам (14), (15) или (16), (17) настоящей статьи.

Этап 4. По рассчитанным значениям средневзвешенной и гарантирующей свёрток с использованием принципа оптимальности Парето определяются рейтинги альтернатив и тем самым альтернативы упорядочиваются по степени их рациональности. Если получившийся при этом уровень детализации удовлетворяет ЛПР, процесс завершается, либо ЛПР обращается к этапам 1, 2 и модифицирует их. Поскольку этапы 3, 4 могут быть полностью автоматизированы, подобный метод даёт в руки ЛПР гибкий инструмент всестороннего анализа проблемы, в отношении которой ищется наиболее рациональное решение.

Этап 5. Если ЛПР желает увеличить степень детализации результата, он выбирает одно из суждений о предпочтительности использования в решаемой задаче одной из свёрток. С учётом этого производится расчёт вспомогательного критерия и новый расчёт рейтинга альтернатив по правилу, описанному в п.6 предыдущего раздела.

5 Пример сравнительного многокритериального анализа концепций высотных беспилотных летательных аппаратов

Рассмотрим исследованную в [14] задачу анализа различных концепций высотных беспилотных летательных аппаратов (БПЛА). Используем для этого метод МУК.

В таблице 12 (таблица 1 из [14]) показаны 15 сравниваемых концепций БПЛА, а в таблице 13 (на основе таблиц 2 – 4 из [14]) значения частных критериев, характеризующих эффективность соответствующих БПЛА.

Таблица 12 – Кодировка концепций БПЛА (по [14])

Способ преобразования энергии	Аэродинамическая схема		
	Нормальная	Летающее крыло	Сочленённая
1.С двигателем ВС (дизель)	1 Н	1 Л	1 С
2.С двигателем ВС (жидкий водород)	2 Н	2 Л	2 С
3. С топливными элементами и электрическим двигателем	3 Н	3 Л	3 С
4. С фотоэлектрическими преобразователями солнечной радиации, электрическим двигателем, химическими накопителями	4 Н	4 Л	4 С
5. С фотоэлектрическими преобразователями солнечной радиации, электролизером, топливными элементами, электрическим двигателем	5 Н	5 Л	5 С

Таблица 13 – Частные критерии эффективности БПЛА различных концепций (по [14])

Код концепции БПЛА	Производительность км ² /день	Длительность летной операции, день	Временная эффективность летной операции	Относительная масса целевой аппаратуры	Взлетная масса, кг	Размах крыла, м	Степень покрытия пространственно-временной области заданий	Стоимость часа работы целевой аппаратуры, \$/км ²
Направление оптимизации	макс	макс	макс	макс	мин	мин	макс	мин
Группа важности	4	1	1	3	1	1	3	2
1Н	140000	2	0,5	0,1	11000	30	1	25
2Н	140000	5	0,6	0,1	12000	30	1	20
3Н	80000	9	0,8	0,15	10000	30	1	18
4Н	80000	180	0,99	0,24	1000	80	0,5	5
5Н	80000	160	0,99	0,2	1100	80	0,45	6
1Л	140000	41852	0,5	0,15	11000	30	1	30
2Л	140000	41824	0,6	0,15	12000	30	1	28
3Л	80000	41859	0,8	0,18	10000	30	1	27
4Л	80000	175	0,99	0,26	1000	80	0,48	6
5Л	80000	155	0,99	0,22	1100	80	0,42	7
1С	140000	41760	0,5	0,1	12000	25	1	34
2С	140000	4	0,6	0,1	13000	25	1	32
3С	80000	8	0,8	0,15	11000	25	1	30
4С	80000	160	0,99	0,25	1100	60	0,45	7
5С	80000	130	0,99	0,2	1200	60	0,4	8

Как следует из третьей строки таблицы 13, из восьми используемых критериев четыре относятся к первой (наименьшей) группе важности, один – ко второй группе важности, два – к третьей и один – к четвёртой группам важности. Универсальные коэффициенты важности критериев для этого случая показаны в таблице 14.

Таблица 14 – Универсальные коэффициенты важности критериев при сравнительном анализе концепций БПЛА

Количество критериев	Количество критериев в каждой группе важности				Коэффициенты важности критериев для средневзвешенной свёртки				Коэффициенты важности критериев для гарантирующей свёртки			
	B1	B2	B3	B4	B1	B2	B3	B4	B1	B2	B3	B4
8	4	1	2	1	0,0454	0,1099	0,1824	0,3378	0,0656	0,0819	0,1639	0,3279

В таблице 15 приведены нормированные значения частных критериев оптимальности концепций БПЛА, направленные на минимизацию каждого критерия.

Таблица 15 – Нормированные частные критерии эффективности БПЛА различных концепций, приведённые к минимизации

Код концепции БПЛА	Производительность км ² /день	Длительность лётной операции, день	Временная эффективность лётной операции	Относительная масса целевой аппаратуры	Взлётная масса, кг	Размах крыла, м	Степень покрытия пространственно-временной области заданий	Стоимость часа работы целевой аппаратуры, \$/км ²
Направление оптимизации	мин	мин	мин	мин	мин	мин	мин	мин
Группа важности	4	1	1	3	1	1	3	2
Коэффициент важности для средневзвешенной свёртки	0,3378	0,0454	0,0454	0,1824	0,0454	0,0454	0,1824	0,1099
Коэффициент важности для гарантирующей свёртки	0,3279	0,0656	0,0656	0,1639	0,0656	0,0656	0,1639	0,0819
1Н	0,000	1,000	1,000	1,000	0,167	0,909	0,000	0,310
2Н	0,000	1,000	0,796	1,000	0,083	0,909	0,000	0,483
3Н	1,000	1,000	0,388	0,688	0,250	0,909	0,000	0,552
4Н	1,000	0,996	0,000	0,125	1,000	0,000	0,833	1,000
5Н	1,000	0,996	0,000	0,375	0,992	0,000	0,917	0,966
1Л	0,000	0,000	1,000	0,688	0,167	0,909	0,000	0,138
2Л	0,000	0,001	0,796	0,688	0,083	0,909	0,000	0,207
3Л	1,000	0,000	0,388	0,500	0,250	0,909	0,000	0,241
4Л	1,000	0,996	0,000	0,000	1,000	0,000	0,867	0,966
5Л	1,000	0,996	0,000	0,250	0,992	0,000	0,967	0,931
1С	0,000	0,002	1,000	1,000	0,083	1,000	0,000	0,000
2С	0,000	1,000	0,796	1,000	0,000	1,000	0,000	0,069
3С	1,000	1,000	0,388	0,688	0,167	1,000	0,000	0,138
4С	1,000	0,996	0,000	0,063	0,992	0,364	0,917	0,931
5С	1,000	0,997	0,000	0,375	0,983	0,364	1,000	0,897

В таблице 16 приведены рассчитанные значения линейной и гарантирующей свёрток, их нормированные значения как относительные отклонения от минимального значения среди всех рассматриваемых альтернатив, а также сводный рейтинг концепций БПЛА по Парето, определённый на основе этих данных (рисунок 2).

Таблица 16 – Относительные значения средневзвешенной и гарантирующей свёрток критериев (относительные отклонения от минимального значения среди всех рассматриваемых альтернатив) и сводный рейтинг концепций БПЛА по Парето

Код концепции БПЛА	Средневзвешенная свёртка	Гарантирующая свёртка	Нормированная средневзвешенная свёртка	Нормированная гарантирующая свёртка	Сводный рейтинг концепций БПЛА по Парето
1Н	0,356	0,164	0,225	0,000	4 - 5
2Н	0,362	1,000	0,235	1,000	6 - 7
3Н	0,640	1,000	0,727	1,000	9
4Н	0,713	1,000	0,857	1,000	11
5Н	0,770	1,000	0,958	1,000	12
1Л	0,235	0,258	0,010	0,113	1 - 3
2Л	0,229	0,826	0,000	0,792	1 - 3
3Л	0,526	0,826	0,525	0,792	6 - 7
4Л	0,693	1,000	0,821	1,000	10
5Л	0,752	1,000	0,927	1,000	13
1С	0,277	0,250	0,084	0,103	1 - 3
2С	0,317	1,000	0,155	1,000	4 - 5
3С	0,594	1,000	0,647	1,000	8
4С	0,726	1,000	0,880	1,000	12
5С	0,794	1,000	1,000	1,000	15

Детализация полученного решения вполне удовлетворительна – лишь в двух случаях две концепции делят одно и тоже место, и в одном случае на одном и том же месте находятся три концепции: 1Л, 2Л и 1С.

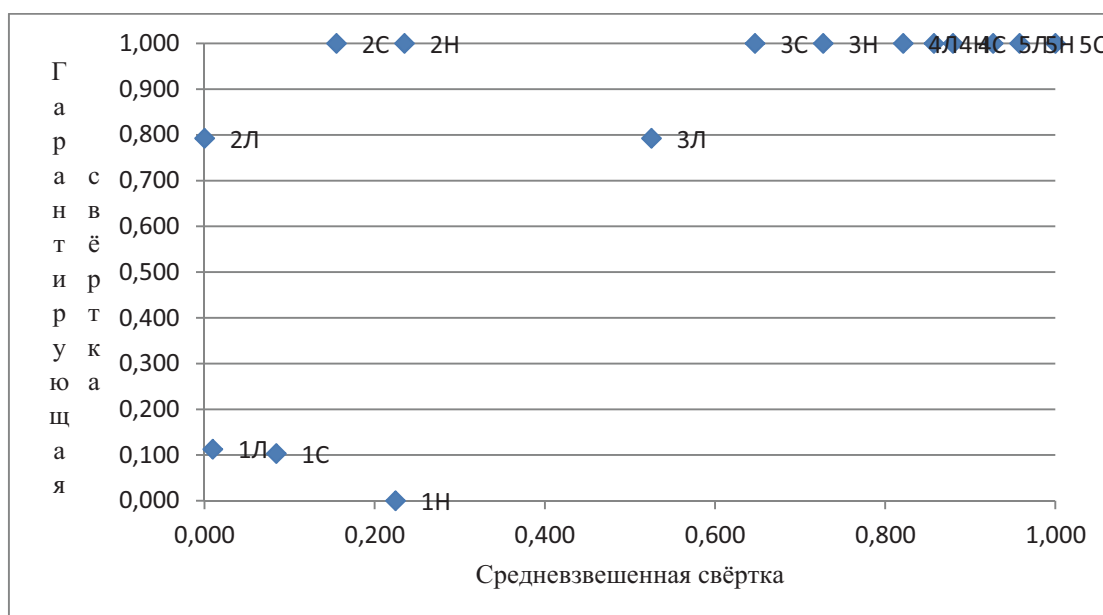


Рисунок 2 - Относительные значения средневзвешенной и гарантирующей свёрток критериев при сравнительной оценке концепций БПЛА

Это решение частично упорядочивает альтернативы по комплексной оценке, однако не даёт им количественной оценки. Для того, чтобы получить и такую оценку, необходимо привлечь дополнительно утверждение ЛПР о некоторой предпочтительности для него одного из двух видов свёртки: средневзвешенной и гарантирующей. В рассматриваемом примере положим, что средневзвешенная свёртка для ЛПР предпочтительнее. Отвечающие этому соображению итоговая комплексная оценка и рейтинг концепций БПЛА приведены в таблице 17 (второй и третий столбцы). Для сравнительного анализа там же показаны соответствующие значения для средневзвешенной и гарантирующей свёрток.

Таблица 17 – Итоговая комплексная оценка (на минимум) и рейтинг концепций БПЛА

Код концепции БПЛА	Используются обе свертки, но средневзвешенная представляется ЛПР более предпочтительной		Средневзвешенная свертка		Гарантирующая свертка	
	Итоговая комплексная оценка	Итоговый рейтинг	Нормированное значение свертки	Рейтинг	Нормированное значение свертки	Рейтинг
1Н	0,168	3	0,225	5	0,000	1
2Н	0,426	6	0,235	6	1,000	6 - 15
3Н	0,795	9	0,727	9	1,000	6 - 15
4Н	0,893	11	0,857	11	1,000	6 - 15
5Н	0,968	14	0,958	14	1,000	6 - 15
1Л	0,035	1	0,010	2	0,113	3
2Л	0,198	4	0,000	1	0,792	4 - 5
3Л	0,592	7	0,525	7	0,792	4 - 5
4Л	0,866	10	0,821	10	1,000	6 - 15
5Л	0,945	13	0,927	13	1,000	6 - 15
1С	0,089	2	0,084	3	0,103	2
2С	0,366	5	0,155	4	1,000	6 - 15
3С	0,735	8	0,647	8	1,000	6 - 15
4С	0,910	12	0,880	12	1,000	6 - 15
5С	1,000	15	1,000	15	1,000	6 - 15

Заключение

Математическая незамкнутость задачи многокритериальной сравнительной оценки альтернатив предполагает активную роль ЛПР в принятии решения. Однако следует различать ту часть этой роли, которая опирается на его неотчуждаемую суверенную волю и никому не должна делегироваться и ту, которая продиктована чисто технологическим стремлением замкнуть проблему с тем, чтобы сделать её решение полностью формализуемым.

Автор полагает [9, 15, 16], что первая часть роли ЛПР ограничивается лишь двумя типами так называемых «уверенных суждений»:

Уверенное суждение первого типа. При своей уверенности ЛПР может отнести различные частные критерии к различным группам важности. Например, «критерии 1 и 4

наиболее важны, критерии 2 и 6 просто важны, а критерий 5 имеет наименьшую важность». Подчеркнём, что не предполагается, что ЛПР даёт количественную оценку степени сравнительной важности частных критериев, речь идёт лишь об их качественном сравнении.

Уверенное суждение второго типа. При желании ЛПР может сконструировать пары Парето-несравнимых векторов частных критериев, в отношении которых он уверен, что один из векторов «лучше» другого. При этом не требуется, чтобы эти векторы обязательно отражали эффективность каких-либо реальных объектов.

Этого достаточно для того, чтобы методами «шансов» и «уверенных суждений» получить комплексные количественные оценки эффективности альтернатив.

Однако результаты, полученные в [8,17] и в настоящей статье позволяют значительно продвинуться в сторону упрощения оценки многокритериальных альтернатив. ЛПР получает возможность практически мгновенно «проигрывать» результаты корректировки своей целевой установки и тем самым не задавать её изначально, а гибко формировать совместно с содержательной неформальной оценкой последствий её применения. Это позволяет поставить перед методологами принятия решений проблему более высокого порядка – о поддержке процесса формирования ЛПР своей целевой установки при принятии решения.

Благодарности

Работа выполнена при поддержке РФФИ, научный проект № 18-08-00858 А, 09.02.2018.

Список источников

- [1] *Ларичев, О.И.* Теория и методы принятия решений / О.И. Ларичев. – М.: Логос, 2002. – 392 с.
- [2] *Ларичев, О.И.* Вербальный анализ решений / О.И. Ларичев // ИСИ РАН. – М.: Наука, 2006. – 181 с.
- [3] *Черноруцкий, И.Г.* Методы принятия решений / И.Г. Черноруцкий. – СПб.: БХВ-Петербург, 2005. – 416 с.
- [4] *Лебедев, А.А.* Курс системного анализа / А.А. Лебедев. - М.: Машиностроение/Машиностроение-Полет, 2010. – 256 с.
- [5] *Johannes, J.* Vector Optimization: Theory, Applications, and Extensions / J. Johannes. - Berlin, Heidelberg, New York: Springer-Verlag, 2010. 460 p.
- [6] *Ansari, H.Q.* Recent Developments in Vector Optimization / H.Q. Ansari, Yao. Jen-Chih. - Heidelberg, Dordrecht, London, New York: Springer-Verlag, 2010. 550 p.
- [7] *Hirota, N.* Sequential Approximate Multiobjective Optimization Using Computational Intelligence / N. Hirota, Y. Yeboon, Y. Min. - Berlin, Heidelberg: Springer-Verlag, 2009. - 197 p.
- [8] *Пиявский, С.А.* Как «нумеризовать» понятие «важнее» / С.А. Пиявский // Онтология проектирования. – 2016. – Т.6, №4(22). – С.414-435. – DOI: 10.18287/2223-9537-2016-6-4-414-435.
- [9] *Malyshev, V.V.* The confident judgment method in the selection of multiple criteria solutions / Malyshev V.V., Piyavsky S.A. // Journal of Computer and Systems Sciences International. 2015. v. 54. No 5. - P.754-764.
- [10] Методика и процедуры формирования Национального рейтинга университетов 2018 года - <http://university-rating.ru/txt.asp?rbr=30&txt=Rbr30Text4021&lng=0>.
- [11] *Saaty, T.L.* The Analytic Hierarchy Process. McGraw Hill. 1980. [Reprinted by RWS Publications, available electronically free, 2000].
- [12] *Саати, Т.* Аналитическое планирование. Организация систем / Т. Саати, К. Кернс. Пер. с англ – М.: Радио и связь, 1991. – 224 с.
- [13] *Саати, Т.* Об измерении неосязаемого. Подход к относительным измерениям на основе главного собственного вектора матрицы парных сравнений / Т. Саати // Электронный журнал Cloud of Science. 2015. Т. 2. № 1, - <http://cloudofscience.ru>.
- [14] *Brusov, V.S.* Multi-criteria analysis of high-altitude UAV concepts / V.S. Brusov, S.A. Piyavsky // Russian Aeronautics v.59, No.4, 2016. – P.447-452.

- [15] *Malyshev, V.V.* A decision making method under conditions of diversity of means of reducing uncertainty / V.V. Malyshev, B.S. Piyavsky, S.A. Piyavsky // Journal of Computer and Systems Sciences International. 2010. v.49. No 1. - P.44-58.
- [16] *Пиявский, С.А.* Два новых понятия верхнего уровня в онтологии многокритериальной оптимизации / С.А. Пиявский // Онтология проектирования. №1(7), 2013. – С.65-85.
- [17] *Пиявский, С.А.* Вычислительные аспекты формирования универсальных таблиц коэффициентов важности критериев / С.А. Пиявский // Онтология проектирования. – 2017. – Т. 7, №3(25). - С.284-295. – DOI: 10.18287/2223-9537-2017-7-3-284-295.

METHOD OF UNIVERSAL COEFFICIENTS FOR THE MULTI-CRITERIAL DECISION MAKING

S.A. Piyavsky

Povolzhsky State University of Telecommunications and Informatics, Samara, Russia
spiyav@mail.ru

Abstract

The problem of multi-criteria choice is a key element in making complex decisions and it has not lost its relevance for more than half a century. A number of approaches and methods suggest that the decisions made with their use are most rational. An overview of the most common methods and their applications is presented. The main element of most of these methods is the linear convolution of particular criteria, and their difference consists in one or another heuristic or expert way of specifying the numerical coefficients of the importance of the criteria. Author developed an approach that allows the pre-calculated universal tables of numerical coefficients of the importance of particular criteria to be used in the formation of a linear convolution, which significantly reduces both the complexity of the decision-making process and the inevitable subjectivism arising from heuristic selection or expert assignment of its coefficients. In the paper, an analogous approach is developed for an equally theoretically and practically important convolution of criteria, which in different publications is called differently: minimax, guarantee, Hermieier convolution. This allowed us to propose a new general method for making decisions and comparing multicriteria alternatives (the MUC method), based on the joint use of both types of bundles. Its application was demonstrated on two practically important tasks - the rating evaluation of universities and the analysis of various design concepts of high-altitude unmanned aero vehicles.

Key words: *decision making, multi-criteria choice, universal importance criterion, minimax.*

Citation: *Piyavsky SA.* Method of universal coefficients for the multi-criterial decision making [In Russian]. *Ontology of designing*. 2018; 8(3): 449-468. - DOI: 10.18287/2223-9537-2018-8-3-449-468.

Acknowledgment

The work was supported by the Russian Foundation for Basic Research, scientific project No. 18-08-00858 A, 09.02.2018.

References

- [1] *Larichev OI.* Theory and methods of decision making [In Russian]. – М.: Logos, 2002. – 392 p.
- [2] *Larichev OI.* Verbal Decision Analysis [In Russian]. – М.: Nauka, 2006. – 181 p.
- [3] *Chrnorutskiy IG.* Methods of decision making [In Russian]. – SPb.: BHV-Peterburg, 2005. - 416 p.
- [4] *Lebedev AA.* Course of the system analysis [In Russian]. – М.: Mashinostroyeniye / Mashinostroyeniye-Polyot, 2010. – 256 p.
- [5] *Johannes J.* Vector Optimization: Theory, Applications, and Extensions. Berlin, Heidelberg, New York: Springer-Verlag, 2010. - 460 p.

- [6] *Ansari HQ, Jen-Chih Yao*. Recent Developments in Vector Optimization. Heidelberg, Dordrecht, London, New York: Springer-Verlag, 2010. - 550 p.
 - [7] *Hirota N, Yeboon Y, Min Y*. Sequential Approximate Multiobjective Optimization Using Computational Intelligence. Berlin, Heidelberg: Springer-Verlag, 2009. - 197 p.
 - [8] *Piyavsky SA*. How do we digitize the concept of “more important” [In Russian]/ *Ontology of Designing*. 2016; 6(4): 414-435. DOI: 10.18278/2223-2016-6-4-414-435.
 - [9] *Malyshev VV, Piyavsky SA*. The confident judgment method in the selection of multiple criteria solutions - *Journal of Computer and Systems Sciences International*. 2015; 54(5): 754-764.
 - [10] National Rating of Universities - <http://univer-rating.ru/txt.asp?rbr=30&txt=Rbr30Text4021&lng=0>.
 - [11] *Saaty TL*. The Analytic Hierarchy Process. McGraw Hill. 1980. [Reprinted by RWS Publications, available electronically free, 2000].
 - [12] *Saaty TL, Kearns KP*. Analytical Planning, The Organizations of Systems. - Pergamon Press. Oxford New York Toronto Sydney Frankfurt. Elsevier Ltd/ 1985. – 208 p.
 - [13] *Saaty TL*. On the Measurement of Intangibles. A Principal Eigenvector Approach to Relative Measurement Derived from Paired Comparisons. *Notices of the American Mathematical Society*. 2013. - DOI: 10.1090/noti944.
 - [14] *Brusov VS, Piyavskii SA*. Multi-criteria analysis of high-altitude UAV concepts, *Russian Aeronautics*. 2016; 59(4): 447-452.
 - [15] *Malyshev VV, Piyavsky BS, Piyavsky SA*. A decision making method under conditions of diversity of means of reducing uncertainty - *Journal of Computer and Systems Sciences International*. 2010; 49(1): 44-58.
 - [16] *Piyavsky SA*. Two new top-level concepts in the ontology of multiobjective optimization [In Russian]. - *Ontology of designing*. 2013; 1(7): 65-85.
 - [17] *Piyavsky SA*. Computational aspects of establishing universal tables of criterion's importance [In Russian]. *Ontology of Designing*. 2017; 7(3): 284-295. DOI: 10.18287/2223-9537-2017-7-3-284-295.
-

Сведения об авторе



Пиявский Семён Авраамович. Окончил факультет летательных аппаратов Куйбышевского авиационного института в 1964 году, аспирантуру при кафедре Динамики полёта Московского авиационного института им. С. Орджоникидзе в 1967 году. Доктор технических наук, профессор Поволжского государственного университета телекоммуникаций и информатики. Почётный работник высшей школы РФ, академик Академии наук о Земле и Академии нелинейных наук. Опубликовал более 350 научных работ в области системного анализа, методов оптимизации и принятия решений, математического моделирования, образовательных систем и технологий. Основные научные результаты: онтологии образовательного процесса, методы многоэкстремальной оптимизации, принятия решений в условиях неустраняемой неопределённости, оптимизации многоцелевых систем летательных аппаратов; теория многоцелевых систем,

компьютерная технология технического творчества, теория оптимального управления развитием научных способностей молодёжи и др.

Semen Avraamovich Piyavsky. Graduated from Kuibyshev Aviation Institute in 1964 and the graduate school at the Flight Dynamics Department at the Moscow Aviation Institute named Ordzhonikidze in 1967. Doctor of Technical Sciences, Professor of the Povolzhsky State University of Telecommunications and Informatics. Honored Worker of Higher School of Russia, Academician of the Academy of Earth Sciences and Academy of Nonlinear Sciences. He has published over 350 scientific papers in field of system analysis, optimization techniques and decision-making, mathematical modeling, education systems and technologies. Basic scientific results: education ontologies, Multiple-optimization techniques, decision making under fatal uncertainty, computer technology of engineering creation, the optimal control theory of young people' academic abilities development, etc.

УДК 004.021

ПАРЕТО-ОПТИМИЗАЦИЯ ПРОИЗВОДСТВЕННОГО РАСПИСАНИЯ НА ОСНОВЕ МЕТОДА МУРАВЬИНЫХ КОЛОНИЙ

Ю.А. Скобцов¹, О.В. Ченгарь²

¹ Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия
ya_skobtsov@list.ru

² Севастопольский государственный университет, Севастополь, Россия
OVChengar@sevsu.ru

Аннотация

Для оптимизации работы автоматизированного технологического участка механообработки наряду с модифицированным муравьиным алгоритмом, разработана объектно-ориентированная модель организационно-технологического процесса загрузки оборудования, представляющая систему взаимодействующих классов её типовых компонентов. Объектная модель описывает структуру классов, составляющих систему производственного процесса, их атрибуты, операции, взаимосвязи с другими классами. Модель позволяет рассчитать значения целевой функции и оценить качество потенциальных решений. Впервые предложено использование муравьиных алгоритмов совместно с объектно-ориентированным имитационным моделированием для оптимизации производственного расписания. Каждый искусственный муравей находит потенциальное решение задачи. Концентрация искусственного феромона определяется качеством решения относительно используемых критериев оптимизации. Для искусственных муравьев предложены формулы расчёта концентрации феромона и определены правила перехода, которые управляют процессом поиска оптимального решения. Предложена многокритериальная оптимизация с адаптивными весами, где в процессе решения корректируются веса целевой функции. Рассмотрены варианты выбора критериев оптимальности: максимизация среднего коэффициента загрузки технологического оборудования, минимизация нарушения крайних сроков изготовления заказа при минимальной длительности цикла изготовления деталей, минимизация нарушения крайних сроков изготовления заказа при минимизации времени переналадок оборудования, минимизация нарушения крайних сроков изготовления заказа при минимальной длительности изготовления деталей и времени переналадок оборудования. Экспериментальные исследования выполнены для решения задачи двух и трёхкритериальной оптимизации на примере автоматизированного технологического комплекса механообработки.

Ключевые слова: производственное расписание, многокритериальная оптимизация, муравьиный алгоритм, автоматизированный машиностроительный комплекс, объектно-ориентированные модели.

Цитирование: Скобцов, Ю.А. Парето-оптимизация производственного расписания на основе метода муравьиных колоний / Ю.А. Скобцов, О.В. Ченгарь // Онтология проектирования. – 2018. – Т. 8, №3(29). – С.469-479. – DOI: 10.18287/2223-9537-2018-8-3-469-479.

Введение

В работе рассматривается задача многокритериальной оптимизации производственного расписания с помощью муравьиного алгоритма (МА) с адаптивными весами. Присущие МА свойства способствуют их эффективному применению при решении задач многокритериальной оптимизации, так как МА используют множество потенциальных решений – популяцию и глобальный поиск в нескольких направлениях [1-3]. Кроме этого, МА не предъявляют требований к виду целевых функций и ограничений. МА хорошо показали себя для решения за-

дач комбинаторной оптимизации, одной из которых является задача составления производственных расписаний технологического оборудования гибких производственных систем (ГПС) автоматизированного технологического комплекса (АТК) механообработки [4, 5]. Известны применения МА при решении задач как комбинаторной [6], так и численной [7] оптимизации.

Для оценки качества потенциальных решений в процессе поиска решений на основе МА используются объектно-ориентированные модели, которые описывают важнейшие характеристики систем и позволяют с приемлемой достоверностью проводить моделирование их функционирования. Многокритериальная оптимизация получила широкое применение при решении различных задач [8-10].

1 Производственное расписание

Задача синтеза расписания работы ГПС заключается в том, чтобы для производственного участка с заданным технологическим маршрутом составить порядок обработки деталей, учитывая ограничения реальных производственных ситуаций за данное время. Модель такого процесса удобно представить в виде ориентированного графа, построение которого эквивалентно определению чисел t_{ij} – моментов начала технологической операции O_{ij} . В статье совокупность чисел $\{t_{ij}\}$ ($i = 1, 2, \dots, n; j = 1, 2, \dots, m_i$), удовлетворяющая производственным условиям, называется расписанием работы ГПС или его графовой моделью $G(i)$, где i – тип детали, n – количество деталей, j – технологическая операция, m – количество технологических операций [1, 4, 5].

В МА каждый искусственный муравей находит своё потенциальное решение, концентрация искусственного феромона определяется качеством построенного решения в соответствии с используемыми формулами коррекции концентрации феромона. Важнейшими компонентами МА являются правила (и соответствующие формулы расчёта вероятностей) для выбора следующей вершины графа в процессе поиска решения, которые определяют последовательность рассмотрения потенциальных решений. Для адаптации МА к решению поставленной задачи необходимо выполнить следующую последовательность действий.

- 1) разработать схему представления потенциального решения.
- 2) определить правила коррекции концентрации искусственного феромона, учитывающие качество потенциального решения.
- 3) разработать глобальные эвристики для определения выбора дуги при поиске пути в графе.
- 4) определить локальную эвристику поведения муравья в виде вероятности перехода при построении решения.
- 5) определить инструмент проверки выполнимости потенциального решения с учётом ограничений задачи.
- 6) проверить адекватность модели.

2 Графоаналитическая модель

На основе разработанной графоаналитической модели (см. рисунок 1) построена объектно-ориентированная модель организационно-технологического процесса загрузки оборудования, которая представляет систему взаимодействующих классов её типовых компонентов. Эта модель описывает структуру классов, составляющих систему производственного процесса, их атрибуты, операции, взаимосвязи с другими классами.

В работе [1] рассмотрено построение графоаналитической модели $G=(V, D, P)$, где V – множество вершин, каждая из которых представляет позицию обработки деталей; D – множество дуг графа, представляющих время перехода от одной технологической операции на другую; P – матрица правил перехода, где каждой дуге $(i, j) \in D$ приписывается вес P_{ij} .

Представленная на рисунке 1 графоаналитическая модель предусматривает распределение оборудования по технологическим операциям согласно плану выпуска деталей, а также позволяет варьировать его количеством, в зависимости от исправности оборудования и плановых профилактических работ.

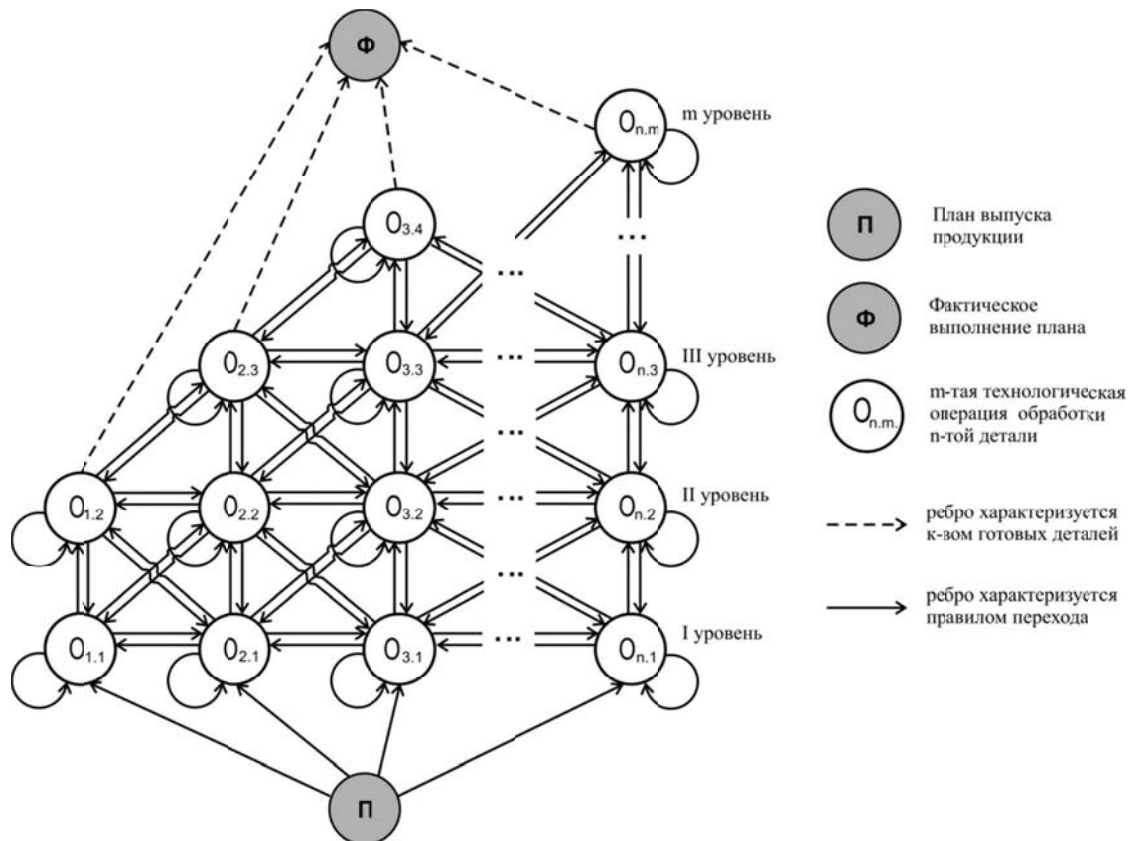


Рисунок 1 - Графоаналитическая модель организационно-технологического процесса загрузки гибких производственных модулей

Исходная вершина графа определяет начало выполнения плана производства (согласно производственной программе и технологическим картам изготовления деталей) – стартовую точку, которая характеризуется следующими параметрами:

$$(1) \quad \Pi = (Kv, D, Npl_i, Od_i, Rd_i, Ts_i),$$

где Kv – количество гибких производственных модулей (ГПМ) на производственном участке;

D – общее количество деталей d_i по производственной программе;

Npl_i – количество деталей i -того типа по производственной программе;

Od_i – количество операций O_{ij} , для изготовления детали i -того типа;

Rd_i – размер партии запуска деталей i -того типа;

Ts_i – срок изготовления партии деталей i -того типа.

Остальные вершины графа разбиты на уровни, каждый из которых соответствует отдельной технологической операции O_{ij} (согласно технологической карте). Каждая вершина O_{ij} однозначно определяется параметрами:

$$(2) \quad O_{ij} = (N_{ij}, n_{ij}, Tv_{ij}, Tn_{ij}, Tnp_{ij}, L1_{ij}, L2_{ij}, L3_{ij}),$$

где N_{ij} – номер ГПМ, на котором выполняется операция O_{ij} ;

n_{ij} – количество запущенных в обработку деталей d_i ;

Tv_{ij} – время выполнения технологической операции O_{ij} ;

Tn_{ij} – время наладки ГПМ для выполнения операции O_{ij} ;

Tnp_{ij} – время простоя ГПМ перед выполнением операции O_{ij} ;

$L1_{ij}$ – объём свободного места в лотке для заготовок;

$L2_{ij}$ – объём свободного места в лотке для готовой продукции;

$L3_{ij}$ – объём свободного места в лотке для инструментов.

Правило перехода ГПМ из стартовой точки в вершины первого уровня не является случайным событием, а рассчитывается с использованием временных параметров технологических карт изготовления деталей и с учетом сроков выполнения заказа:

$$(3) \quad P_{ij} = \frac{K_{ij}}{\sum K_{ij}}, \quad \sum P_{ij} = 1$$

где K_{ij} – коэффициент перехода:

$$(4) \quad K_{ij} = \frac{To_{ij} * kf}{Ts_i}$$

$$(5) \quad To_{ij} = \sum_{j=1}^m Tn_{ij} + Tv_{ij} * n_{ij},$$

где To_{ij} – время выполнения операции O_{ij} над партией деталей i -того типа;

kf – коэффициент возможности перехода.

$$(6) \quad Ts_i = k_i * dl,$$

где k_i – количество рабочих дней для выполнения заказа; dl – длительность рабочей смены, ч.

Время освобождения ГПМ из вершины рассчитывается с учётом величины максимальной партии запуска и временных параметров выполнения технологических операций:

$$(7) \quad Tсв_{ij} = L_{ij} * Tv_{ij} + Tn_{ij} + Tnp_{ij},$$

где L_{ij} – максимально допустимая партия запуска деталей d_i .

3 Направленный муравьиный алгоритм

Для оптимизации расписания производственного участка АТК предложен «направленный» МА, для которого: 1) разработана схема представления потенциального решения; 2) определены правила коррекции концентрации искусственного феромона, которые определяют положительную обратную связь; 3) разработана эвристика выбора дуги при поиске пути в графе; 4) определено «направленно-пропорциональное» правило перехода в следующую вершину графа; 5) выбраны «глобальные правила» для расчёта концентрации искусственного феромона, которые способствуют направленному поиску; 6) разработаны объектные модели для проверки выполнимости потенциального решения с учётом ограничений задачи; 7) произведена проверка адекватности предложенной модели.

Преимуществом разработанного алгоритма является то, что он не требует построения структурной модели самого производственного участка и допускает простые модификации, которые позволяют эффективно решать оптимизационные задачи подобного класса. Эта модификация отличается от [3] следующими особенностями:

1) Предложен метод вычисления вероятностей перехода искусственного муравья по вершинам графоаналитической модели, основанный на анализе текущей производственной ситуации; вероятность перехода k -того муравья в вершину O_{ij} определяется соотношением

$$(8) \quad \begin{cases} P_{ij,k}(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{k=1}^1 [\tau_{ij}]^\alpha [\eta_{ij}(t)]^\beta}, & O_{ij} \in N_{ij}^k \\ P_{ij,k}(t) = 0, & O_{ij} \notin N_{ij}^k \end{cases},$$

где α – коэффициент, определяющий степень влияния феромона; β – коэффициент, определяющий влияние эвристической информации; τ_{ij} – концентрация феромона на дуге графа; η_{ij} – эвристическая информация (обычно, длина дуги); N_{ij}^k – перечень вершин графа O_{ij} , доступных для k -того муравья, t – номер итерации в процессе поиска решения.

2) Предпочтительность выбора вершины графа задаётся «направленно-пропорциональным» правилом перехода между вершинами, основанным на эвристической информации. Она определяется отношением времени выполнения технологической операции To_{ij} к запланированному времени изготовления детали Ts_i , которое в свою очередь корректируется после выполнения каждой технологической операции над партией деталей

$$(9) \quad \eta_{ij} = \frac{To_{ij}}{Ts_i}.$$

3) Для расчёта концентрации искусственного феромона при переходе муравья на следующий узел графа используются «глобальные» правила, которые способствуют направленному поиску. Эти правила заставляют муравья двигаться в сторону найденных «худших» решений с целью их «улучшения». Такая стратегия стимулирует эксплуатацию пространства поиска и применяется после того, как решение построено, т.е. после прохождения всеми муравьями своего пути. В этом случае концентрацию феромона разрешается корректировать только «худшим» муравьям, построившим неоптимальный путь. Тогда для каждой дуги графа концентрация феромона корректируется в соответствии с правилом

$$(10) \quad \tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}(t), \Delta\tau_{ij}(t) = \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t),$$

где $\Delta\tau$ – количество искусственного феромона, откладываемое муравьями, определяется в зависимости от заданного критерия оптимальности и вычисляется по одному из следующих правил:

- для задачи максимизации среднего коэффициента загрузки оборудования

$$(11) \quad \Delta\tau_{ij}^k(t) = \frac{1}{\sum_{i,j} Tnp_{ij}^k},$$

где Tnp_{ij}^k – время простоя k -го оборудования перед выполнением операции O_{ij} ;

- для задачи минимизации времени переналадок оборудования

$$(12) \quad \Delta\tau_{ij}^k(t) = \frac{1}{\sum_{i,j} Tn_{ij}^k},$$

где Tn_{ij}^k – время наладки ГПМ для выполнения операции O_{ij} на k -ом оборудовании;

- для задачи минимизации длительности цикла изготовления деталей

$$(13) \quad \Delta\tau_{ij}^k(t) = \frac{1}{\sum_{i,j} \text{Тпр}_{ij}^k + \text{То}_{ij}^k},$$

где То_{ij} – времени выполнения технологической операции O_{ij} на k -ом оборудовании.

4 Парето-оптимизация производственного расписания на основе «направленного» муравьиного алгоритма

При использовании метода взвешенной функции для решения задач многокритериальной оптимизации «общая» целевая функция $F(x)$ строится из целевых функций $f_i(x)$ в виде их взвешенной суммы

$$(14) \quad F(x) = \sum_{i=1}^q w_i f_i(x).$$

Каждой целевой функции $f_i(x)$ присваивается свой вес w_i , и задача сводится к скалярному случаю. При этом различные веса w_i дают разные решения в смысле Парето. Скалярное значение общей целевой функции вычисляется путём суммирования взвешенных значений q критериев оптимальности. Для параллельного поиска кратных решений веса не фиксируются, что даёт возможность «направленному» МА расширять фронт по всем направлениям.

В предложенном алгоритме на каждой итерации по определённому критерию оптимизации формируется множество решений на основе «направленного» МА. Далее для исследуемых решений определяются максимальная и минимальная экстремальные точки в пространстве заданных критериев (Z). В итоге получаем гиперплоскость, которая определяется двумя экстремальными точками, и содержит все текущие решения, показанные на рисунке 2. Указанные две экстремальные точки обновляются на каждой итерации. При этом адаптивный вес k -ой цели определяется соотношением

$$(15) \quad w_k = \frac{1}{z_k^{\max} - z_k^{\min}}, \quad k = 1, 2, \dots, q.$$

Для каждой цели на основе выбранных критериев эффективности устанавливаются средневзвешенные весовые коэффициенты значимости, которые нормируются внутри группы.

$$(16) \quad w_k^{\text{норм.}} = \frac{w_k}{\sum_{i=1}^q w_i}, \quad k = 1, 2, \dots, q.$$

Тогда на каждой итерации взвешенная целевая функция определяется согласно следующему выражению

$$(17) \quad z(x) = \sum_{k=1}^q w_k^{\text{норм.}} (f_k(x) - z_k^{\min}).$$

Вначале алгоритма предполагается инициализация - обнуление всех значений Z_{\min} и Z_{\max} . Поскольку экстремальные точки обновляются на каждой итерации, то соответственно обновляются и веса.

В течение процесса поиска решения в пространстве z каждой популяцией муравьёв адаптивная гиперплоскость последовательно приближается к положительной (или отрицательной) идеальной точке (см. рисунок 2).

Поскольку для каждого из критериев оптимальности может быть задана своя точность решения, которая достигается при различных количествах итераций n_i , для их определения проведены экспериментальные исследования всех однокритериальных задач и определено количество итераций, при котором будет достигнута точность, соответствующая технологическим условиям. В качестве критерия останова при решении многокритериальной задачи выбирается максимальное значение количества итераций, полученной из решения однокритериальных задач.

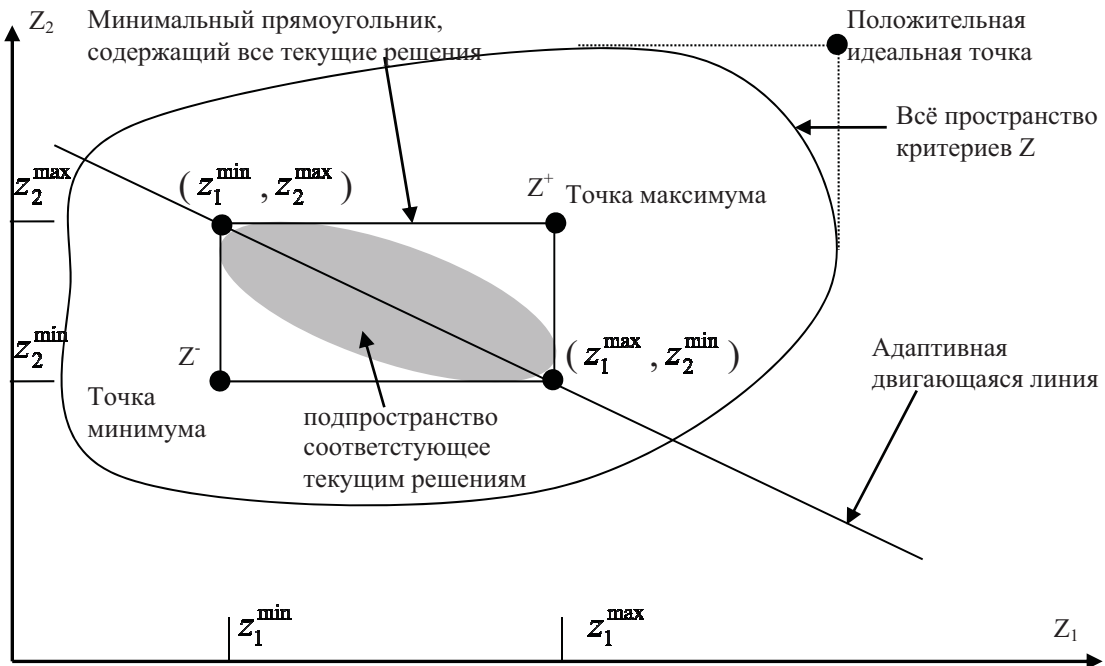


Рисунок 2 - Адаптивные веса и адаптивная гиперплоскость

Таким образом, описанный выше метод многокритериальной оптимизации позволяет перебирать всё пространство решений для каждой популяции муравьёв и выбирать из них пути, оптимальные в смысле Парето.

5 Экспериментальные исследования

В ходе эксперимента на объектной модели АТК решены многокритериальные задачи с применением «направленного» МА с адаптивными весами. Анализ результатов проводился по сравнению с решениями, полученными при решении однокритериальной задачи. Рассмотрим следующие варианты выбора критериев оптимальности.

I. Максимизация среднего коэффициента загрузки технологического оборудования.

II. Минимизация нарушения крайних сроков изготовления заказа при минимальной длительности цикла изготовления деталей.

III. Минимизация нарушения крайних сроков изготовления заказа при минимизации времени переналадок оборудования.

IV. Минимизация нарушения крайних сроков изготовления заказа при минимальной длительности изготовления деталей и времени переналадок оборудования.

Для проведения анализа полученных решений были рассмотрены зависимость среднего коэффициента загрузки технологического оборудования ($K_{зср}$) и времени нарушения крайних сроков заказа ($T_{ср}$) в зависимости от количества популяций муравьёв n_i для каждого из вари-

антов выбора критериев оптимальности. Отметим, что максимизация $K_{з.ср}$ включает в себя и минимизацию производственного цикла и настроек оборудования одновременно. Полученные результаты представлены на рисунках 3 и 4 соответственно.

Анализ полученных результатов показывает, что предложенный «направленный» МА с адаптивными весами для многокритериальной задачи требует большего количества итераций, чем подобный алгоритм для однокритериальной задачи. Причём с увеличением числа критериев увеличивается количество популяций муравьёв для нахождения Парето-оптимального решения. Так, для однокритериальной задачи (I вариант) количество итераций, необходимое для достижения субоптимального решения, $n_t = 140$, для двухкритериальных задач (II и III варианты) $n_t = 190$, а для трёхкритериальной задачи (IV вариант) $n_t = 240$. Однако если предоставить «направленному» МА данное количество ресурса n_t , он неизменно показывает свою эффективность и позволяет получить «хорошие» решения.

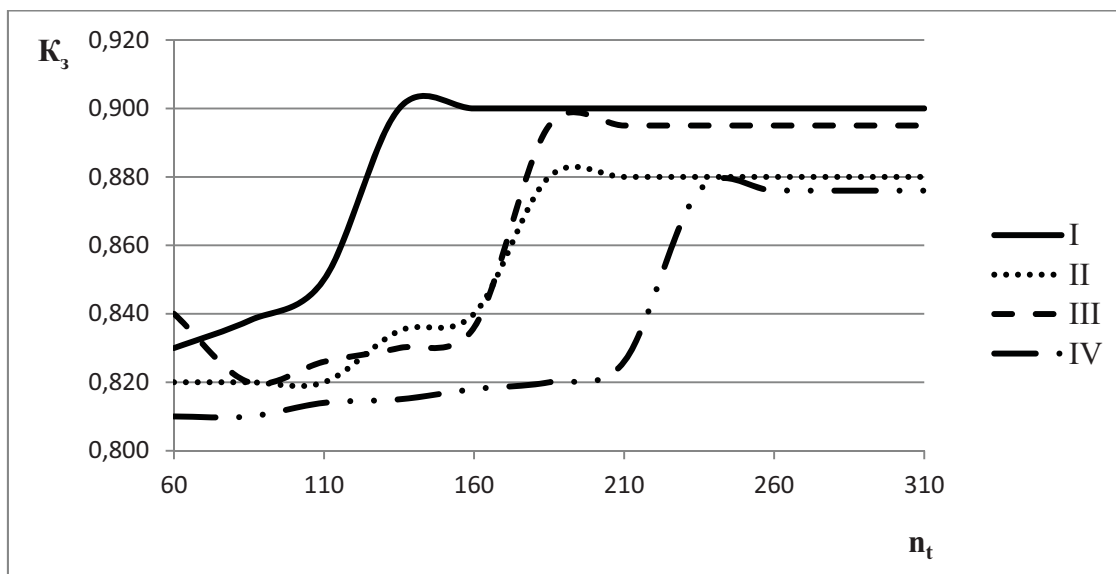


Рисунок 3 – Зависимость среднего коэффициента загрузки оборудования от количества итераций для различных вариантов выбора критериев оптимальности

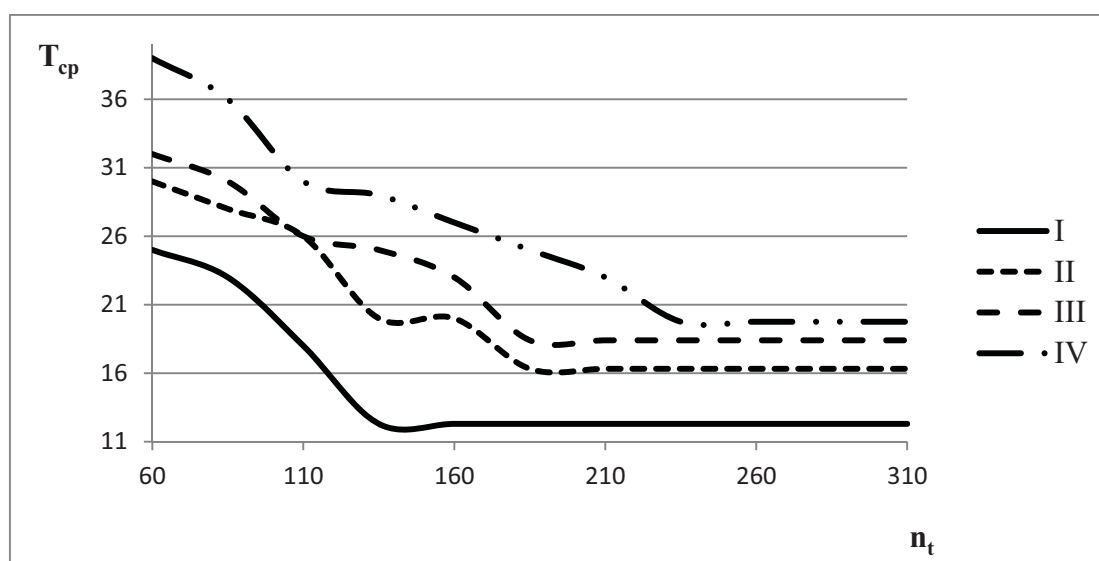


Рисунок 4 – Зависимость времени нарушения крайних сроков изготовления заказа от количества итераций для различных вариантов выбора критериев оптимальности

Заключение

Проведённые исследования основных временных параметров, полученных в результате решения задачи оптимизации производственного расписания, показали, что при введении дополнительных критериев эффективности задача выбора оптимальных решений значительно усложняется, однако эффективность методов существенно не снижается. Так, на рисунке 5 представлены диаграммы, отражающие разницу между значениями основных параметров, полученных путём однокритериальной и многокритериальной оптимизаций.

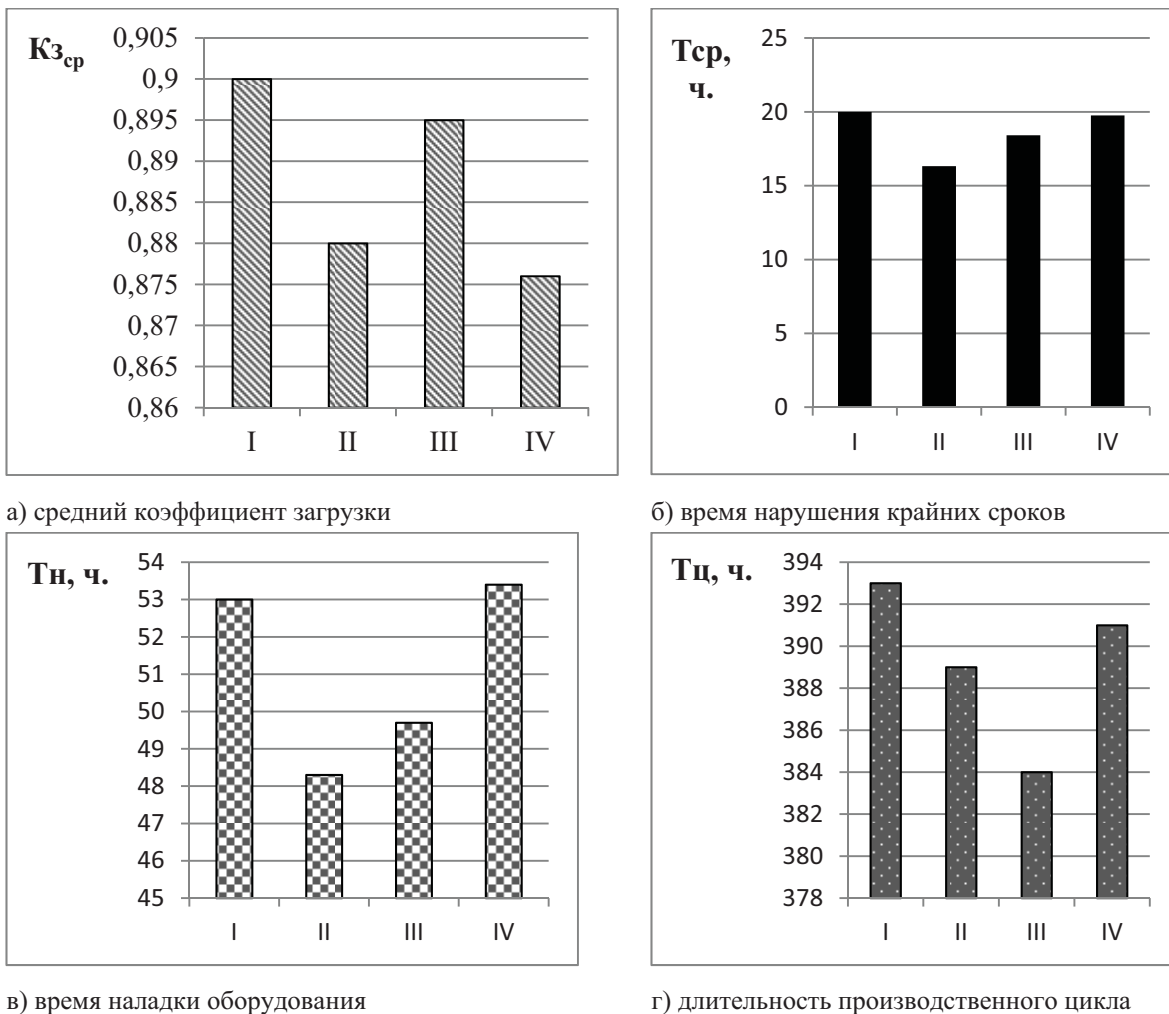


Рисунок 5 – Значения среднего коэффициента загрузки оборудования и временных параметров, полученных в результате оптимизации производственного расписания, в зависимости от варианта выбора критериев эффективности

Таким образом, можно сделать вывод о том, что применение метода Парето-оптимизации производственного расписания АТК на основе «направленного» МА с адаптивными весами для многокритериальной задачи позволяет получать результаты, в наименьшей степени отличающиеся от решения однокритериальной задачи. Т.е. при оптимизации одновременно по нескольким критериям, качество полученных решений по каждому критерию в отдельности значительно не снижается, что говорит о выборе верной тактики решения подобных задач.

СПИСОК ИСТОЧНИКОВ

- [1] **Скобцов Ю.А.** Объектно-ориентированное моделирование и эволюционные алгоритмы / Ю.А. Скобцов А.И. Секирин С.Ю. Землянская О.В. Ченгарь В.Ю. Скобцов // Труды 7-й Всероссийской научно-практической конференции «Имитационное моделирование» (ИММОД-2015) // ISBN 978-5-91450-173-7 // Т.2.-М.: ИПУ РАН, 2015. - С.338-343.
- [2] **Скобцов Ю.А.** Эволюционные вычисления: учебное пособие/ Ю.А Скобцов., Д.В. Сперанский. - М.: Национальный Открытый Университет «ИНТУИТ», 2015. - 331 с.
- [3] **Dorigo M.** Swarm Intelligence, Ant Algorithms and Ant Colony Optimization. Reader for CEU Summer University Course “Complex System”/ М. Dorigo. - Budapest, Central European University, 2001. - P.1–3.
- [4] **Скобцов Ю.А.** Многокритериальный муравьиный алгоритм оптимизации производственного расписания / Ю.А. Скобцов, О.В. Ченгарь, А.Н. Скаковская // Труды XXIX Международной научной конференции «Математические методы в технике и технологиях - ММТТ-29». - СПб.: СПбГТИ(ТУ). Том.9. - С.245-253.
- [5] **Скобцов Ю.А.** Многокритериальные муравьиные алгоритмы и объектно-ориентированные модели / Ю.А. Скобцов, О.В. Ченгарь // Труды 8-й Всероссийской научно-практической конференции «Имитационное моделирование» (ИММОД-2017). - СПб.: СПИИРАН, 2017. - С.162-166.
- [6] **Курейчик В.М.** О некоторых модификациях муравьиного алгоритма / В.М. Курейчик, А.А. Кажаров // Известия ЮФУ. Технические науки». Тематич. выпуск «Интеллектуальные САПР». 2008, № 4(81) – 268 с.
- [7] **Карпенко А.П.** Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой / А.П. Карпенко. - Москва: Изд-во МГТУ им. Н.Э. Баумана, 2014.-446с.
- [8] **Пиявский С.А.** Прогрессивность многокритериальных альтернатив / С.А. Пиявский // Онтология проектирования. - 2013. - №4(10). - С.53-59.
- [9] **Пиявский С.А.** Два новых понятия верхнего уровня в онтологии многокритериальной оптимизации / С.А. Пиявский // Онтология проектирования. – 2013. - №1(7). – С.65-85.
- [10] **Ларионов И.П.** Парето-оптимизация в области принятия решений при проектировании комплексной системы защиты предприятия/ И.П. Ларионов, П.Б. Хорев // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №2 (2016). - <http://naukovedenie.ru/PDF/118TVN216.pdf>. - DOI: 10.15862/118TVN216.

PARETO-OPTIMIZATION OF THE PRODUCTION SCHEDULE BASED ON THE METHOD OF ANTS OF COLONIES

Y.A. Skobtsov¹, O.V. Chengar²

¹ *St. Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia, ya_skobtsov@list.ru*

² *Sevastopol State University, Sevastopol, Russia, OVChengar@sevsu.ru*

Abstract

To optimize the functioning of the automated machining technological complex together with a modified ant algorithm an object model of the organizational and technological process of loading equipment was developed, representing a system of interacting classes of its typical components. The object model describes the structure of the classes that make up the production process system, their attributes, operations, relationships with other classes. The model allows to calculate the values of the objective function and evaluate quality of potential solutions. For the first time, the use of ant algorithms in conjunction with object-oriented simulation modeling is proposed to optimize the production schedule. Every artificial ant finds a potential solution to the problem. The concentration of artificial pheromone is determined by the quality of the solution with respect to the optimization criteria used. For artificial ants formulas for calculating the concentration of pheromone are proposed and transition rules that control the process of searching for the optimal solution are defined. Multi criteria optimization with adaptive weights is proposed, where in the search process the weights of the objective function are corrected. The following variants of choosing the optimality criteria are considered: maximization of the average load factor of the process equipment, minimization of the violation of the deadlines for manufacturing the order with the minimum duration of the manufacturing cycle of the parts, minimization of the violation of the deadlines for the manufacture of the order, minimization of the time limits for manufacturing the order, and the time of adjustment of equipment. Experimental studies were performed to solve the problem of two and three-criteria optimization using the example of an automated technological complex for machining.

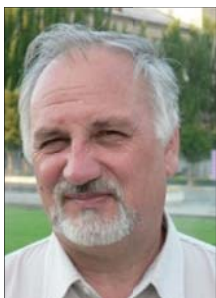
Key words: production schedule, multi-criteria optimization, ant algorithm, automated machine building complex, object-oriented models.

Citation: Skobtsov YA, Chengar OV. Pareto-Optimization of the production schedule based on the method of ants of colonies [In Russian]. *Ontology of designing*. 2018; 8(3): 469-479. DOI: 10.18287/2223-9537-2018-8-3-469-479.

References

- [1] *Skobtsov YA, Sekirin AI, Zemlyanskaya SY, Chengar OV, Skobtsov VY*. Object-oriented modeling and evolutionary algorithms [In Russian]; Proc. of the 7th Scientific and Practical Conference «Imitation Modeling» (IMMOD-2015) // ISBN 978-5-91450-173-7 // V.2. - Moscow: IPP RAS, 2015. - P.338-343.
- [2] *Skobtsov YA, Speransky DV*. Evolutionary computing: a tutorial [In Russian]. - Moscow: National Open University "INTUIT", 2015. – 331 p.
- [3] *Dorigo M*. Ant Algorithms Solve Difficult Optimization Problems. In: Kelemen J., Sosík P. (eds) Advances in Artificial Life. ECAL 2001. Lecture Notes in Computer Science, vol. 2159. Springer, Berlin, Heidelberg. – DOI: 10.1007/3-540-44811-X_2.
- [4] *Skobtsov YA, Chengar OV, Skakovskaya AN*. Multicriteria ant algorithm for optimization of production schedules [In Russian]; Proc. of the XXIX - International scientific conference "Mathematic methods in engineering and technology - MMET-29". - St. Petersburg: SPbGTI (TU). V.9. - P.245-253.
- [5] *Skobtsov YA, Chengar OV*. Multicriteria ant algorithms and object-oriented models [In Russian]; Proc. of the 8th Scientific and Practical Conference "Imitation Modeling" (IMMOD -2017). - St. Petersburg: SPIIRAN, 2017. - P.162-166.
- [6] *Kureichik VM, Kajarov AA*. On some modifications of the ant algorithm [In Russian]; News of SFedU. Technical science". Thematic issue "Intellectual CAD". - Taganrog, 2008; 4(81) – 268 p.
- [7] *Karpenko AP*. Modern algorithms of search optimization. Algorithms inspired by nature [In Russian]. - Moscow: Pub.MSTU named after Bauman, 2014. – 446 p.
- [8] *Piyavsky SA*. Progressivity of multicriteria alternatives [In Russian]. *Ontology of designing*, 2013; 4(10): 53-59.
- [9] *Piyavsky SA*. Two new upper level concepts for the multi-criteria optimization ontology [In Russian]. *Ontology of designing*, 2013; 1(7): 65-85.
- [10] *Larionov IP, Khorev PB*. Pareto-optimization in the field of decision-making when designing an integrated system of enterprise protection [In Russian] / Internet-journal "Naukovedenie". 2016; 8(2). - <http://naukovedenie.ru/PDF/118TVN216.pdf>. - DOI: 10.15862/118TVN216.

Сведения об авторах



Скобцов Юрий Александрович, 1946 г. рождения. Окончил Московский физико-технический институт в 1971 г., д.т.н. (1994), профессор (2000). Профессор кафедры «Компьютерные технологии и программная инженерия» Санкт-Петербургского государственного университета аэрокосмического приборостроения. В списке научных трудов более 300 работ (включая 10 монографий и учебных пособий) в области искусственного интеллекта, объектно-ориентированного моделирования, эволюционных алгоритмов.

Yuri Alexandrovich Skobtsov (b. 1946). He graduated from the Moscow Institute of Physics and Technology in 1971, Doctor of Technical Sciences (1994), Professor (2000). Professor of the department "Computer Technologies and Software Engineering" of St. Petersburg State University of Aerospace Instrumentation. In the list of scientific works more than 300 scientific papers (including 10 monographs and textbooks) in the field of artificial intelligence, object-oriented

modeling, evolutionary algorithms.



Ченгарь Ольга Васильевна, 1977 г. рождения. Окончила Донецкий государственный технический университет в 1999 г., к.т.н. (2013). Доцент кафедры информационных технологий и компьютерных систем ФГАОУ ВО «Севастопольский государственный университет». В списке научных трудов более 30 работ в области имитационного моделирования, искусственного интеллекта, биоинспирированных вычислений.

Olga Vasilievna Chengar (b. 1977). Graduated from Donetsk State Technical University in 1999, Ph.D. (2013). Associate Professor of the Department of Information Technologies and Computer Systems of the Federal State Unitary Enterprise "Sevastopol State University". In the list of scientific works more than 30 works in the field of imitation modeling, artificial intelligence, natural computing.

Конференция ИИ в бизнес – AI CONFERENCE

22 ноября 2018 г., Москва, Россия

AI Conference — международная выставка-конференция по применению искусственного интеллекта в бизнесе. Организатор мероприятия – **Smile-Expo**.



Темы конференции:

- Как изменяются бизнес-процессы в связи с внедрением искусственного интеллекта.
- Когда чат-бот становится бизнес-ассистентом.
- С помощью каких инструментов внедрить искусственный интеллект в свои проекты.
- Какие возможности открывает применение технологий распознавания лиц и эмоций в ретейле.
- Как технология глубокого обучения помогает повысить эффективность онлайн-маркетинга.
- AI-алгоритмы для диагностики заболеваний.
- Применение AI в автомобильной индустрии и сервисной робототехнике.
- Применение AI-технологий в обработке естественной речи в голосовом поиске и переводе.
- Как технологии машинного обучения влияют на процесс обслуживания пользователей.
- Как AI-решения для распознавания лиц повышают качество обслуживания клиентов.

<https://aiconference.ru/ru>

2nd International Conference on Intelligent Human Systems Integration: Integrating People and Intelligent Systems (IHSI 2019)

February 7-10, 2019 ~ Hilton San Diego Bayfront, California USA

IHSI 2019 Conference aims to provide a global forum for presenting and discussing novel approaches, design tools, methodologies, techniques, and solutions for integrating people, intelligent technologies and automation, and artificial cognitive systems in all areas of human endeavor in industry, economy, government, and education, including but not limited to energy, transportation, urbanization and infrastructure development, digital manufacturing, social development, human health, sustainability, new generation of service systems, as well as safety, risk assurance, autonomy, and cybersecurity in both civilian and military contexts.

<http://www.ihsint.org/>

The 10th International Multi-Conference on Complexity, Informatics and Cybernetics: IMCIC 2019

March 12 - 15, 2019 ~ Orlando, Florida, USA

Motives and Purpose. Conceptual and instrumental relationships among Complexity, Informatics and Cybernetics are continuously growing, cross-fertilizing each other and generating important theoretical and technological synergies. Similar concepts and theories support both Complexity Science and Cybernetics. Informatics and cybernetics have been increasingly related via computing, communications and control technologies; and even some authors conceive them as the same but with different names, emphasis or orientation. Complexity and Informatics have been supporting each other in theory and practice.

<http://www.2019iisconf.org/imcic>

The 23rd World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2019[©]

July 6 - 9, 2019 ~ Orlando, Florida, USA

The **purpose** of WMSCI 2019 is to promote discussions and interactions between researchers and practitioners focused on disciplinary, interdisciplinary and transdisciplinary issues, ideas, concepts, theories, methodologies and applications. We are particularly interested in fostering the exchange of concepts, prototypes, research ideas, and other results which could contribute to the academic arena and also benefit business, and the industrial community.

<http://www.iis2019.org/wmsci/website/about.asp?vc=1>

Индекс 29151

СОДЕРЖАНИЕ

От редакции

20 ЛЕТ «ПУМСС»: КРАТКИЕ ИТОГИ

С. Фогель, С. Рудольф (*Штутгарт, Германия*)

ПРОЕКТИРОВАНИЕ СЛОЖНЫХ СИСТЕМ ПРИ ПОМОЩИ ЯЗЫКОВ
ПРОЕКТИРОВАНИЯ: МЕТОДЫ, ПРИМЕНЕНИЕ
И ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ

Д.А. Губанов, Д.А. Новиков (*Москва, Россия*)

МЕТОДЫ ИЗВЛЕЧЕНИЯ И АНАЛИЗА ТЕРМИНОЛОГИЧЕСКИХ СТРУКТУР
СМЕЖНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ (НА ПРИМЕРЕ МЕТОДОЛОГИИ)

В.А. Головкин, В.В. Голенков, В.П. Ивашенко, В.В. Таберко, Д.С. Иванюк,

А.А. Крощенко, М.В. Ковалёв (*Минск, Брест, Республика Беларусь*)
ИНТЕГРАЦИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С БАЗАМИ ЗНАНИЙ

О.Ю. Марьясин (*Ярославль, Россия*)

ПРОЕКТИРОВАНИЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЗДАНИЕМ
С ИСПОЛЬЗОВАНИЕМ ОНТОЛОГИЙ

Ю.К. Горелов, С.К. Киселев (*Ульяновск, Россия*)

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЧЕСКОГО ПОДХОДА ПРИ ПРОЕКТИРОВАНИИ
МНОГОФУНКЦИОНАЛЬНОГО АВИАЦИОННОГО ИНДИКАТОРА

Н.В. Максимов, О.Л. Голицына, М.Г. Ганченкова, Д.В. Санатов, А.В. Разумов

(*Москва, Санкт-Петербург, Россия*)
СЕМАНТИЧЕСКОЕ ЯДРО ЦИФРОВОЙ ПЛАТФОРМЫ

А.С. Клещев, В.А. Тимченко (*Владивосток, Россия*)

РЕАЛИЗАЦИЯ ОБОЛОЧКИ И ПОРТАЛА ЗНАНИЙ ПО ВЕРИФИКАЦИИ
МАТЕМАТИЧЕСКИХ ДОКАЗАТЕЛЬСТВ НА ПЛАТФОРМЕ IASRAAS

С.А. Пиявский (*Самара, Россия*)

МЕТОД УНИВЕРСАЛЬНЫХ КОЭФФИЦИЕНТОВ ПРИ ПРИНЯТИИ
МНОГОКРИТЕРИАЛЬНЫХ РЕШЕНИЙ

Ю.А. Скобцов, О.В. Ченгарь (*Санкт-Петербург, Севастополь, Россия*)

ПАРЕТО-ОПТИМИЗАЦИЯ ПРОИЗВОДСТВЕННОГО РАСПИСАНИЯ НА ОСНОВЕ
МЕТОДА МУРАВЬИНЫХ КОЛОНИЙ

НАУЧНЫЕ КОНФЕРЕНЦИИ



Ontologists and designers of all countries and subject areas, join us!



Издательство "Новая техника" - Publisher «New Engineering» Ltd

Россия, 443010, Самара, ул.Фрунзе 145 - 145, Frunze Str., Samara, 443010, Russia