

# БИОНИКА ИНТЕЛЛЕКТА

ИНФОРМАЦИЯ, ЯЗЫК, ИНТЕЛЛЕКТ

---

---

№ 1 (80)

2013

---

---

НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

**Основан в октябре 1967 г.**

Учредитель и издатель  
*Харьковский национальный университет радиоэлектроники*

Периодичность издания – *3 раза в год*

## СОДЕРЖАНИЕ

### **СТРУКТУРНАЯ, ПРИКЛАДНАЯ И МАТЕМАТИЧЕСКАЯ ЛИНГВИСТИКА**

<i>Широков В.А.</i>	
Грамматика как феноменологическая проблема.....	3
<i>Вороной С.М., Егошина А.А.</i>	
Предварительная кластеризация текстовых документов для повышения качества автоматического построения онтологий.....	15
<i>Лазаренко О.В., Панченко Д.И.</i>	
Семантико-контекстна модель реферування .....	19
<i>Бабкова Н.В., Канищева О.В., Кочуева З.А.</i>	
Моделирование семантических отношений в онтологии .....	25
<i>Бегунов А.Б., Заболотня Т.М.</i>	
Агентно-орієнтований метод автоматизованого вилучення фактів з природномовних текстових даних.....	29

### **ТЕОРЕТИЧЕСКИЕ ОСНОВАНИЯ ИНФОРМАТИКИ И КИБЕРНЕТИКИ. ТЕОРИЯ ИНТЕЛЛЕКТА**

<i>Скопа О.О., Вавілов Є.В.</i>	
Интеллектуальні автономні системи: концептуальні положення створення та функціонування .....	35

### **МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ. СИСТЕМНЫЙ АНАЛИЗ. ПРИНЯТИЕ РЕШЕНИЙ**

<i>Ерохин А.Л.</i>	
Моделирование взаимодействий внешних факторов с системой когнитивных функций человека-оператора.....	41
<i>Дрюк А.Д., Кучеренко Е.И.</i>	
Стохастические модели и методы субоптимальной маршрутизации сложных объектов.....	45
<i>Керносов М.А.</i>	
Математическая модель контента информационно-аналитической системы .....	54
<i>Литвин О.О.</i>	
Інтегральне представлення залишку наближення диференційовних функцій інтерполяційними кубічними поліномами Зламала-Женішека на трикутнику .....	62
<i>Mashtalir S.V., Mikhnova O.D.</i>	
Stabilization of key frame descriptions with higher order voronoi diagram .....	68
<i>Писклакова О.А.</i>	
Модели распределения ресурсов в сильно централизованных системах .....	73
<i>Чайников С.И., Солодовников А.С.</i>	
Формализованное описание граф-модели предметной области .....	77

### **ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА ИНФОРМАЦИИ. РАСПОЗНАВАНИЕ ОБРАЗОВ**

<i>Танянский С.С.</i>	
Формализация средств доступа к базам данных произвольной структуры.....	82
<i>Чала Л.Е., Попаденко П.Ю.</i>	
Метод виявлення нечітких дублікатів електронних текстових документів.....	88
<i>Власенко Н.В.</i>	
Построение информативных компактных описаний и классификация объектов путем представления в ортогональном базисе.....	93
<i>Аксак Н.Г., Коргут С.А., Стрельцова Н.Г.</i>	
Расширенный browsebank на основе учета поведенческих факторов пользователей.....	99
<i>Дударь З.В., Егоров С.В.</i>	
Семантическое аннотирование в информационно-поисковых системах.....	104
<i>Кораблев Н.М., Иващенко Г.С.</i>	
Применение модели клонального отбора, использующей вывод по прецедентам, для прогнозирования временных рядов.....	108
<i>Мантула Е.В.</i>	
Прогнозирующая нейронная сеть с переменной структурой для контроля показателей загрязнения окружающей среды .....	112
<i>Ручкін К.А.</i>	
Аналіз та проектування багатоагентної системи прогнозування поведінки нелінійної динамічної системи у реальному часі .....	117

### **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММНО-ТЕХНИЧЕСКИЕ КОМПЛЕКСЫ**

<i>Боргест Н.М., Коровин М.Д.</i>	
К вопросу базовой онтологии машиностроительного предприятия .....	128
Об авторах .....	133
Правила оформлення рукописів для авторів науково-технічного журналу «Біоніка інтелекту».....	135

В.А. Широков

Украинский языково-информационный фонд НАНУ, г. Киев, Украина

## ГРАММАТИКА КАК ФЕНОМЕНОЛОГИЧЕСКАЯ ПРОБЛЕМА

### Логико-феноменологические основания грамматического описания языка

Грамматическая проблематика в последнее время интересует не только теоретиков языкознания, которые обязаны ею заниматься, так сказать, по долгу службы, но всё больше — прикладников, создающих системы компьютерной обработки языка. В понятие такой обработки входит широкий круг задач, связанных с информационным поиском, переводом с одного языка на другой, инженерией знаний, распознаванием и синтезом речевых и текстовых образов, интерпретацией содержания текстов, проведением различных лингвистических экспертиз, обучением языку и т.д. Грамматическое описание для обозначенного круга задач представляет, по преимуществу, интерес в том смысле, что оно обеспечивает интерпретацию языковых объектов со стороны формы, что принципиально важно при формализации данного описания, поскольку формализация (и как можно более глубокая) является необходимым условием создания эффективных, наделенных интеллектуальными свойствами артефактов, производящих языковую обработку.

Опыт построения любых теорий свидетельствует, что успех формализации описания той или иной предметной области непосредственно связан с выбором адекватных объектов концептуального представления ее феноменов. Сказанное целиком относится и к грамматике. Следует, однако, признать, что при создании грамматических схем их авторы зачастую склонны сужать область языковой феноменологии, так что эти построения носят вполне камерный характер. Разумеется, такое ограничение значительно упрощает задачу исследователей, за такое упрощение приходится расплачиваться адекватностью самого описания, его неполным, недостаточным для приложений соответствием феноменологии языка и, в конце концов, эффективностью конечных продуктов — лингвистических технологий.

Очевидно, что границы области языковой феноменологии не являются постоянными и подлежат уточнению в связи с теми задачами, которые предстоит решать в процессе исследования. Более того, возможны и весьма различающиеся между собой подходы к определению самих принципов, на основании которых предполагается осуществить упомянутое уточнение. В этой связи хотим

привести слова Исаака Ньютона из его первой, написанной в восемнадцатилетнем возрасте научной работы, которая называлась «Об универсальном языке»<sup>1</sup> и была посвящена фундаментальным грамматическим проблемам: *«Диалекты отдельных языков так сильно различаются, что всеобщий Язык не может быть выведен из них столь верно, как из природы самих вещей, которая едина для всех народов и на основе которой весь Язык был создан вначале. Система бытия построена из отдельных субстанций (духов и тел), каждая из которых находится в том или ином положении, состоянии или кондиции. Использование языка в том и состоит, что один человек может обозначить для другого, в каком состоянии та или иная субстанция есть, была, будет, может быть, должна быть, желаемая быть, побуждается быть и т. д. Этого достаточно в качестве вступления к тому, что последует».*

Из приведенного краткого фрагмента со всей однозначностью следует, что Исаак Ньютон, формулируя начальные принципы своей теоретико-лингвистической концепции, предполагал недостаточность того, что в наше время называют «языковым материалом», и расширил границы феноменологии языка вплоть до «природы самих вещей, которая едина для всех народов и на основе которой весь Язык был создан вначале». В XXI веке понятна вся колоссальная сложность попытки И. Ньютона вывести систему языка из «природы самих вещей», но и сегодня мы не можем не восхититься интеллектуальным атлетизмом юного гения, предпринявшего столь смелую попытку более 350 лет тому назад.

С другой стороны, данную попытку можно охарактеризовать и как апофеоз феноменологического подхода к исследованию языка. Ведь природа вещей «является» субъекту, то есть «дается» ему через освоение совокупности своих явлений, которые, в конце концов, становятся фактами языка. Методологическая установка на выведение системы языка из природы вещей представляется тем более ценной, что она вводит язык в общий круг

<sup>1</sup> «Об универсальном языке»: (Newton I. Of an Universal Language // Modern Language Review). Данная работа была опубликована лишь в 1957 году. Ее перевод на русский язык, выполненный Л.В.Кнориной (с её же примечанием и послесловием, а также предисловием В.А.Успенского), был впервые опубликован в сб. «Семиотика и информатика», вып. 28, М., 1986.

явлений и сущностей как одно из проявлений общей природы вещей.

Разумеется, при определении границ языковой феноменологии нужно соблюдать и определенную умеренность, дабы теоретические построения по возможности заканчивались не только формулировкой общих принципов, но и полезными для практических приложений результатами. И несмотря на то, что самолюбию многих лингвистов должно льстить изречение Апостола Иоанна «В начале было Слово, и Слово было у Бога, и Слово было Бог», вводящее языковые средства в первоначальный арсенал Творца, целесообразные начала природы вещей применительно к изучению языка следует, возможно, отнести к более поздним этапам эволюции мира.

Однако, говоря об эволюции языка, нельзя забывать и о креационистских взглядах на его природу. Вот как, несколько иронически, описывает акт творения языка Умберто Эко устами одного из главных героев романа «Имя розы» distinguished Вильгельма Баскервилльского: «*Бог предупредил, чтобы Адам не вкушал от древа познания добра и зла, и это был божий закон. Но в то же время он сам позволил Адаму, более того, уполномочил его раздавать имена вещам в гражданском мире. И в этом отношении – в мирском отношении – предоставил полную свободу своему подданному. Да, именно так, несмотря на то, что некоторые наши современники утверждают, будто *nominas unt consequentia rerum*<sup>2</sup>. Однако книга Бытия на сей счет гласит достаточно ясно: Господь подвел к человеку всех тварей, чтобы узнать, какое имя тот им даст, и как наименовал человек какую живущую тварь, таково ей и именоваться вековечно. И хотя не подлежит никакому сомнению, что первый из человеков подошел к делу сугубо ответственно и, называя на своем эдемском языке всякую вещь и всякое животное, руководствовался природой называемого, все-таки ничем не отменяется то обстоятельство, что, занимаясь этим, он облакался некоторой верховной властью: решать, которое из многих имен, по его усмотрению, лучше всего соответствует природе называемого предмета. Ибо в самом деле установлено, что имена, которыми разные люди пользуются для описания одних и тех же понятий, различны, а неизменны и едины для всех только понятия, то есть знаки вещей. И слово *потеп* (имя) бесспорно происходит от *potos*, то есть по-гречески «закон», как раз потому, что *potina* создаются группами людей *adplacitum*, то есть по свободному совместному решению»<sup>3</sup>.*

Таким образом, можно полагать, что Исаак Ньютон предпринял попытку, уже, по сути, осуществленную Адамом. Ведь, как утверждает

Вильгельм Баскервилльский, Адам «называя на своем эдемском языке всякую вещь и всякое животное, руководствовался природой называемого». И хотя не вполне ясно, откуда первочеловеку была известна эта природа (возможно, его консультировал сам присутствовавший при этом Творец?), методологическая установка на руководство природой вещей при их назывании становится совершенно понятной. Точно так же, как становится понятным, почему Ньютон, не обладая в семнадцатилетнем возрасте достаточным знанием природы вещей, но желая воплотить такое знание в своей теории языка, большую часть своей научной жизни посвятил именно познанию природы вещей.

Говоря об эволюционизме и креационизме в отношении к языку, нельзя не вспомнить и суждение на данную тему такого авторитета как Вильгельм фон Гумбольдт, который отмечал: «Каким естественным не казалось бы нам предположение о постепенном образовании языка, он мог возникнуть только сразу. <...>Для того, чтобы человек мог осознать хотя бы одно-единственное слово, весь язык целиком и во всех своих взаимосвязях уже должен быть заложен в нем»<sup>4</sup>. Наши собственные наблюдения над системными отношениями языка, проявляющимися в больших толковых словарях<sup>5</sup>, подтверждают изложенную мысль Гумбольдта, хотя мы сегодня готовы отдать предпочтение в данном предмете ни эволюционизму, ни креационизму.

В связи с вопросом о феноменологии языка обратимся к взглядам наиболее яркого представителя феноменологического направления в философии XX века Эдмунда Гуссерля. В третьем томе своих Логических исследований<sup>6</sup> он пишет: «Современная грамматика полагает, что она должна быть основана исключительно на психологии или иных эмпирических науках. В противоположность этому мы усматриваем здесь, что старая идея *общей* и в особенности *априорной грамматики* обретает (благодаря тому, что мы обнаруживаем законы, определяющие возможные формы значений) не подлежащий сомнению фундамент и во всяком случае некоторую определенным образом выделенную сферу своей значимости (*Gültigkeit*). В какой степени могут быть обнаружены еще и другие сферы принадлежащего грамматике *Apriori*, выходит за рамки наших интересов. Внутри чистой логики существует сфера законов, отвлекающихся от любой предметности. Эти законы, в отличие от

<sup>4</sup> В. Гумбольдт. Об отличиях в строении человеческих языков и их влиянии на духовное развитие человечества. — СПб., 1859. — 366 с.

<sup>5</sup> V. Shyrovokov. System relations in explanatory dictionaries. Proceedings of the VI International Scientific and Technical Conference. CSIT 2011. Lviv, 2011, pp.260-264.

<sup>6</sup> Э. Гуссерль. Собрание сочинений. Том III (1). Логические исследования. Исследования по феноменологии и теории познания. Пер. с немецкого В.И. Молчанова. — М.: Дом интеллектуальной книги, 2001.

<sup>2</sup> Имена суть производные от вещей (*лат.*)

<sup>3</sup> Умберто Эко. Имя розы. Перевод с итальянского Елены Костюкович. Санкт-Петербург. Symposium. 2005. С.440.

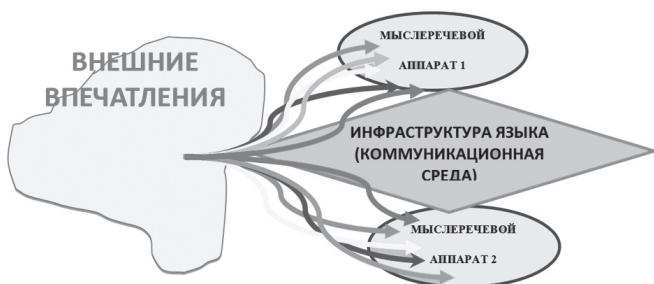
логических законов в обычном и точном смысле, можно было бы оправданно назвать {чистыми логико-грамматическими (reinlogisch grammatische)}<sup>238</sup>. {И еще лучше было бы противопоставить чистому учению о формах значений предполагающее его чистое учение о значимости (reine Geltungslehre).}»

На первый взгляд кажется, что в данном случае феноменологист Э. Гуссерль делает выбор в пользу априоризма. Однако прошлое столетие с его великими открытиями в области логики (связанными с именами Г. Фреге, Г. Кантора, Б. Рассела, А. Уайтхеда, Д. Гильберта, К. Гёделя, Л. Витгенштейна, А. Тарского, Л. Брауэра, Л. Заде и других), а также квантово-релятивистской физики (А. Эйнштейн, А. Пуанкаре, М. Планк, Н. Бор, Л. де Бройль, В. Гейзенберг, Э. Шрёдингер, П.А.М. Дирак, М. Борн, Дж. фон Нейман, Р.Фейнман и другие) настолько расширили понятие логики, одновременно приблизив ее к субстанции и субъекту, что сочетание логики и грамматики по Э. Гуссерлю уже выглядит вовсе не таким, каким оно виделось, например, рационалисту Р. Декарту в XVII веке. Так что замечание Гуссерля, как нам представляется, вполне интерпретируемо в феноменологическом духе, и «чистые логико-грамматические (reinlogisch grammatische) законы» Гуссерля по своей концептуальной эпистемологии вполне сопоставимы с «природой вещей» Ньютона.

Какими же являются общие черты последней, которые находят свое отражение в языковых формах? Для ответа обратимся к тому, что мы называем картиной языкового мира. Как афористически высказался Л. Витгенштейн: «Границы моего языка определяют границы моего мира», а ранее В. фон Гумбольдт в письме к Ф. Вольфу в 1804 году писал: «Мне удалось открыть – и сознанием этого я проникаюсь всё больше, что через посредство языка можно окинуть взглядом наиболее высокие и глубокие сферы и всё многообразие мира».

### Картина языкового мира и ее концептуальное представление

Картину языкового мира наглядно можно представить следующим образом:



На данной схеме внешние впечатления мира через перцептивно-сенсорный аппарат человека (субъекта мыслеречевого процесса), преобразуясь, попадают в

его мыслеречевой аппарат, где подвергаются обработке, обретая языковую форму. Через коммуникационную систему – инфраструктуру языка – осуществляется обмен вербально выраженной информацией с другим субъектом мыслеречевого процесса.

Даже из такой лапидарной схемы следуют достаточно определенные выводы относительно строения системы языка. Во-первых, это вывод, касающийся относительной стабильности форм, приобретаемой языковыми структурами в процессе эволюции. Ведь коммуниканты (участники коммуникационного процесса) обязаны осуществлять обмен информацией в согласованных «форматах» и «протоколах» данных, поскольку без этого коммуникация вообще невозможна. Очевидно, что упомянутые средства «стандартизации» обмена данными по определению не должны меняться (или меняться слабо) в процессе коммуникации, что обеспечивает требуемую стабильность форм языковых структур. Второе касается специфики, своеобразия и разнообразия форм, в которых реализуются языковые данные в контурах и элементах трактов мыслеречевого и коммуникационного процессов. Как отмечали Питтс и Мак-Наллок<sup>7</sup>: «Язык, с помощью которого передается информация (в мозге) ... не соответствует и не должен соответствовать тому языку, которым люди пользуются в общении друг с другом». Это понятно, поскольку в мыслеречевом и коммуникационном процессах решаются, по сути, разные задачи, требующие различных средств для своего разрешения. Если функцией коммуникационной системы есть всего лишь передача языковых данных без или с минимальными искажениями, то в мыслеречевом аппарате решаются гораздо более разнообразные задачи: кодирование-декодирование, анализ, синтез, преобразование, интерпретация, концептуализация, сравнение, запоминание, коротко- и долгосрочное хранение и т. д. Происходят отмеченные процессы в разных средах и осуществляются в разных, так сказать, аппаратно-программных реализациях. Мы полагаем, что свойства информации и особенности протекания информационных процессов, вообще говоря, зависят от физического субстрата, в котором они реализуются. Последнее утверждение находит свое обоснование в квантовой информации, исследования и разработки которой особо активизировались в последнее время<sup>8</sup>.

<sup>7</sup> Цитируется по книге: К. Прибрам. Языки мозга. Экспериментальные парадоксы и принципы нейробиологии. Перевод с английского Я. Н. Даниловой и Е. Д. Хомской. Под редакцией и с предисловием действительного члена АПН СССР Р. Лурия. «Прогресс». Редакция литературы по философии. – М.: 1975.

<sup>8</sup> См., например, книгу: Хренников А.Ю., Введение в квантовую теорию информации. М.: ФИЗМАТЛИТ, 2008. – 284 с., а также многочисленные ссылки в ней и статьи в журналах: Europhysics Letters, Journal of Applied Physics, Journal of the Optical Society of America, Nature Photonics, Nature Physics, New Journal of Physics и др.

Первым вопросом, возникающим при формулировке принципов моделирования языковой субстанции, является вопрос об объектах моделирования, а именно: что представляют собой объекты языка и что, собственно, мы собираемся моделировать. За исходное в данной работе мы берем положение о том, что собственными объектами языка есть определенные психофизические состояния и процессы, происходящие в мыслеречевом аппарате человека, а устная и письменная ее формы служат элементами инфраструктуры речевого процесса. Это утверждение несколько отличается от привычного для традиционной лингвистики взгляда, согласно которому субстанцией языка является звуковая субстанция, которая и формирует языковой субстрат, подлежащей изучению и моделированию. Мы исходим из того очевидного факта, что речь не возникает в голосовых связках — ее происхождение значительно глубже, а звуковая форма уже принадлежит к инфраструктуре языкового, точнее мыслеречевого процесса.

Очевидно, что мыслеречевой процесс сам по себе является интегрированным, то есть содержит как языковую, так и ментальную компоненты. В мыслеречевом аппарате он находит выражение в виде динамической системы взаимосвязанных рефлексов, содержание и характер которых исследуется, например, в книге В.М. Бехтерева<sup>9</sup>, изданной еще в 1909 году, но до сих пор не утратившей своей актуальности, а также в уже упоминавшейся монографии К. Прибрама и целом ряде других работ. Согласно взглядам В.М. Бехтерева естественный язык является одним из так называемых соединительных рефлексов, происходящих в мозге человека. Следовательно, отделение языковых процессов от ментальных, принятое многими лингвистами, а также попытки изучения языка «самого по себе» нам представляется неоправданным и методологически некорректным упрощением. Языковую систему следует рассматривать как открытую, что предполагает как существенное расширение феноменологической базы языка, так и модификацию соответствующего концептуального инструментария.

Устная и письменная формы речи в этом смысле играют роль моделей мыслеречевых процессов и одновременно — коммуникативной для них среды (инфраструктуры). При такой факторизации можно утверждать, что они представляют языковую периферию.

Сделаем, однако, оговорку о возможной недооценке инфраструктурных составляющих языка, которая может возникнуть из-за того, что «основными», первичными языковыми объектами нами признаются психофизические состояния и процессы мыслеречевого аппарата. Дело в том, что

современные данные свидетельствуют о том, что знание языка и владение им не являются врожденными свойствами человека. Врожденной является всего лишь способность к языку, для чего в мозгу человека от рождения имеются определенные участки, ответственные за речь. А процесс «инсталляции» языка у человека с необходимостью требует наличия таких инфраструктурных элементов как так называемые «внешняя» и «эгоцентрическая» речь, которые функционируют уже на ранних стадиях филогенетического развития языковой системы у ребенка и заканчиваются формированием у него «внутренней» речи, что венчает процесс создания полноценного речевого аппарата<sup>10</sup>. Таким образом, языковая периферия является неотъемлемым элементом языковой системы. Кроме того, она является одним из элементов, обеспечивающих ее информационную открытость.

Следует отметить, что психофизические состояния и процессы (а среди них и мыслеречевой), как правило, не полностью доступны для непосредственного наблюдения, а тем более — объективной фиксации. Устная и письменная формы речи, фактически, служат репрезентантами наблюдаемых компонентов состояний языковых объектов и процессов, происходящих в мыслеречевом аппарате. Как таковые, они и используются в качестве основных объектов при концептуальном моделировании языка. Отмеченное обстоятельство еще более подчеркивает значение языковой периферии в изучении системы языка. Таким образом, в картине языкового мира можно выделить наблюдаемые и непосредственно ненаблюдаемые компоненты мыслеречевого процесса:



Мы сознаем условность такой факторизации, ведь, некоторые элементы мыслеречевых процессов в наше время уже поддаются наблюдению (но всего лишь некоторые). Однако, в каком виде, форме, форматах хранятся языковые данные, каковы модели, структуры, типы этих данных — всё это пока наблюдать непосредственно нельзя, и непонятно, насколько это возможно в принципе.

В этой связи обратимся к анализу самой идеи или принципа наблюдаемости. Данный принцип,

<sup>9</sup> Бехтерев В. М. Объективная психология. — М., 1991. — 480 с.

<sup>10</sup> Л. С. Выготский. Мышление и речь. Изд. 5, испр. — Издательство «Лабиринт», М., 1999. — 352 с.

положенный А. Эйнштейном в основу специальной теории относительности и достаточно подробно разработанный при обосновании квантовой механики, оказался весьма продуктивным в методологическом отношении применительно к описанию сложных процессов. Как отмечал А. Эйнштейн в своей дискуссии с В. Гейзенбергом в 1926 г. при обсуждении основ квантовой механики<sup>11</sup>, построить теорию только из одних наблюдаемых величин нельзя (хотя без них никакая научная теория вообще немыслима). Только сама теория должна определять, какие из ее величин являются непосредственно наблюдаемыми, а какие — нет. В частности, в квантовой механике состояния, вообще говоря, не являются непосредственно наблюдаемыми величинами. Последним соответствуют определенного типа операторы, действующие в пространстве состояний; их собственные значения и представляют значения наблюдаемых.

Таким образом, в мире, а значит и в познании существует «невидимое», то есть то, что не может быть наблюдаемо непосредственно. Это дает нам основание для признания роли веры в когнитивно-интеллектуальных процессах, которая по выражению Апостола Павла является «уверенностью в невидимом»<sup>12</sup>. Следует, однако, признать, что более глубокие эпистемологические причины наблюдаемости-ненаблюдаемости не раскрываются и в квантовой теории. По нашему мнению, эти причины кроются в феноменологии сложности проявленного бытия. Любая реальность, являясь объектом наблюдения (другими словами, вступая в отношения «субъект — объект»), проявляет свою сущность в каких-то ограниченных формах, достижимых для восприятия субъектом<sup>13</sup>. Проявленная сложность отмеченных форм (потенциально она бесконечна — ведь материя не имеет границ делимости, так сказать, «вглубь» и не допускает контроля всех ее взаимосвязей и опосредствований) оказывается компенсированной, согласованной с «перцептивно-сенсорным» аппаратом воспринимающего субъекта или прибора (что, в принципе, одно и то же). Механизм самокомпенсации сложности представляется нам настолько универсальной «универсалией» отношения «субъект — объект», что его, возможно, следует отнести к основным принципам системного анализа. Отметим, что сложность здесь мы понимаем как положительную научную категорию, для которой могут быть сформулированы точные и даже

формальные определения типа имеющих место в теории А. Н. Колмогорова<sup>14</sup> и в многочисленных приложениях и обобщениях этой теории, где понятие сложности оказывается непосредственно связанным с понятием информации.

### Понятие состояния языковой единицы

Каким же образом приведенные рассуждения могут быть применены для формулировки принципов моделирования языковой субстанции? Напомним, что мы исходим из положения о том, что собственными («элементарными») объектами концептуального представления в лингвистике должны выступать не непосредственно единицы языка (единицы фонетического, морфологического, лексического и других уровней), и не грамматические или семантические категории и их значения, а некоторые «промежуточные» относительно языка объекты, феноменологическими коррелятами которых являются психофизические состояния и процессы, имеющие место в мыслеречевом аппарате человека, так что устная и письменная ее формы служат элементами инфраструктуры речевого процесса. Описанию и моделированию подлежат как, собственно, упомянутые психофизические состояния и процессы, так и их инфраструктура.

Исходя из данного положения, выясним их роль в моделировании языка и вспомним в этой связи инициативу великого русского математика А.Н. Колмогорова, который, насколько нам известно, впервые применил понятие состояния языковой единицы при попытке строгого определения понятия падежа в русском языке. Сам А.Н. Колмогоров своих трудов по лингвистике не публиковал, поэтому мы следуем в своем изложении его идей работе В.А. Успенского<sup>15</sup>, в которой они были изложены первоначально.

А. Н. Колмогоров предложил такой подход к определению падежа.

Предметы могут находиться в разных состояниях. Так, предмет, который в русском языке называется «молоко», может находиться в следующих состояниях: он может кипеть, его может не быть, его может пить кошка, его может пить собака и т. д. Состояния предмета выражаются в языке в предложениях, где участвует существительное, которое является названием этого предмета. Вышеперечисленные состояния «молока» выражаются в русском языке предложениями: «молоко кипит», «молока нет», «кошка пьет молоко», «собака пьет молоко». При выражении путем предложений состояний, в которых находится данный предмет,

<sup>11</sup> Эта дискуссия подробно описана В. Гейзенбергом в его книге «Часть и целое»: Гейзенберг В. Физика и философия // Часть и целое / Пер. с нем. — М., 1989. — С. 191–196.

<sup>12</sup> «Вера же есть осуществление упования и уверенность в невидимом». Св. Ап. Павел, К Евреям, 11, 1.

<sup>13</sup> В этой связи вспомним известное изречение А. Эйнштейна: «Бог изобретателен, но не зол», в котором подчеркивается, что на каждом этапе познания мир проявляет себя в конечных формах сложности, делающих это познание возможным.

<sup>14</sup> Колмогоров А.Н. Три подхода к определению понятия «количество информации». // «Теория информации и теория алгоритмов». — М.: Наука, 1987. — С. 213–223.

<sup>15</sup> В. А. Успенский. К определению падежа по А.Н. Колмогорову. В сборнике: Бюллетень Объединения по проблемам машинного перевода. — № 5. — М.: [И МГПИИЯ], 1957. — С. 11–18.)

его название употребляется в той или иной форме (в приведенных примерах - «молоко», «молока»).

Два состояния называются эквивалентными относительно данного предмета, если при выражении указанных состояний этого предмета в языке название этого предмета в обоих случаях употребляется в одной и той же форме. Например, два состояния, первый из которых состоит в том, что данный предмет кипит, а второй в том, что кошка пьет данный предмет, эквивалентны относительно предмета «молоко». Эти же состояния не эквивалентны относительно предмета «вода»: «вода кипит», но «кошка пьет воду». Назовем два состояния абсолютно эквивалентными, если они эквивалентны относительно любого предмета, который может находиться в этих состояниях. Так, например, два состояния, первый из которых состоит в том, что кошка любит данный предмет, а второй в том, что собака пьет данный предмет, являются эквивалентными относительно любого предмета, который может находиться в этих состояниях, и, следовательно, абсолютно эквивалентны. Совокупность всех состояний разбивается на непересекающиеся классы таким образом, что любые два состояния из одного и того же класса абсолютно эквивалентны, а любые два состояния из разных классов не являются абсолютно эквивалентными. Эти классы А. Н. Колмогоров и предложил называть падежами.

Приведенное определение, отмечает В. А. Успенский, не является вполне корректным, и далее А. Н. Колмогоровым были предложены определенные его уточнения. Интересно, что на основании этого анализа был сделан вывод, что в русском языке, кроме обычных шести падежей, должны быть еще и такие:

Местный падеж: «в лесу», «в году» и т. д.;

Количественно-отделительный падеж: «выпить чаю», «прибавить ходу», «дать воды» и т. д.

Если оба предложения «не читал газету» и «не читал газеты» есть правильные и выражают одно и то же состояние предмета «газета», то это указывает на то, что существует особый падеж («лишительный»), употребляемый после глаголов, отрицаются, и имеющий две формы (одна из них совпадает с формой «винительного», а вторая - с формой «родительного»).

Возможно, отмечает, следуя А.Н.Колмогорову, В.А.Успенский, что существуют и какие-то иные падежи<sup>16</sup>.

<sup>16</sup> В частности, в книге [Н.В.Перцов. Инварианты в русском словоизменении. — М.: Языки русской культуры, 2001, 280 с.], в главе III «Очерк системы русского словоизменения: План содержания» приводится такой состав падежной системы существительного русского языка: «Существительное. [...] Падежи: именительный (им), родительный (род), дательный (дат), винительный (вин), творительный (тв), предложный (пр), партитивный (парт) [2-й родительный; кусок сахара], локативный (лок) [2-й предложный или местный; в саду], малый нумеративный (мал-нум) [два / три / четыре часа / шага / следа / шара / ряда], большой нумеративный (больш-нум) [пять грамм / ампер / вольт], звательный (н) [вокатив: Наташ, Петь, Вась, мам, пап]».

Как видим, в подходе А.Н.Колмогорова ключевым и принципиальным является понятие *состояния* существительного. Введение этого понятия позволило на дедуктивном уровне построить «из первых принципов» содержательную грамматическую теорию.

Таким образом, в теории Колмогорова состояния занимают место, как это обозначено на нижеприведенной схеме:



Применение понятия состояния языковой единицы получило дальнейшее развитие в наших работах и работах наших сотрудников<sup>17</sup>, соглас-

Существует немало и других работ на эту тему. Следует, однако, заметить, что здесь, так же как и в других работах, как правило, не излагается концептуальный аппарат, применение которого привело к построению описания падежной — в данном случае — системы. Складывается впечатление (это касается и других грамматических трудов), что набор грамматических категорий и их значений является следствием неких полностью эмпирических процедур, в основе которых лежит только лингвистическая наблюдательность (и, разумеется, традиция). Более того, в известных грамматических трудах (за исключением, как нам представляется, замечательной книги А.А.Зализняка «Русское именное словоизменение») вообще не ставится вопрос о том, что должно быть объектом концептуального моделирования в грамматической теории. В этом смысле можно сказать, что гениальную идею А. Н. Колмогорова, изложенную в упомянутой работе В. А. Успенского, лингвистическое сообщество оставило практически незамеченной. Также, по нашему мнению, неглубоко были поняты и пионерские идеи А.А.Зализняка, базирующиеся на логике Колмогорова-Успенского и изложенные в книге «Русское именное словоизменение». Об этом, в частности, свидетельствует замечание В.А.Плунгяна в его учебнике (очень хорошо) [В.А.Плунгян. Введение в грамматическую семантику: грамматические значения и грамматические системы языков мира. — М.:, 2011, С. 183, сноска 69] касательно процедуры Колмогорова-Успенского, где, как пишет автор «вместо понятия семантической роли использовалось более импрессионистическое понятие «состояние объекта». Из дальнейшего изложения станет очевидным, что понятие состояния имеет совершенно иной феноменологический и эпистемологический статус, нежели понятие семантической роли.

<sup>17</sup> Широков В.А. Лексикографічне представлення семантичних станів. Математичні машини і системи. — 1999. — №3. — С.21-32.; Широков В.А. Феноменологія лексикографічних систем. — К.: Наукова думка, 2004. 328 с.; Широков В.А. Семантичні стани мовних одиниць та їх застосування в когнітивній лексикографії — Мовознавство/2005.; Широков К.В. Іменна словозміна у сучасній турецькій мові. —К.: Довіра, 2009, 318 с.; Потапова Е.В. Модель лингвистической онтологии предметной области с нечеткими семантическими состояниями терминов. // Научно-технический журнал «Бионика интеллекта», Харьков, ХНУРЭ. №. 2(79), 2012 г., С.95-102; В.А.Широков. Системна семантика тлумачних словників. У зб., присвяченому 75-літтю В.Г.Скляренка. —К.: Наукова думка, 2012.

но которым любое слово (вообще говоря - любая единица языка) в контексте или в речевом потоке находится в определенном семантическом состоянии. Для единиц лексического уровня это состояние представляет собой некоторым образом организованную сумму признаков грамматической и лексической семантики и предоставляет путь для обобщения понятий грамматического и лексического значения. Собственно, сам процесс понимания речи с этой позиции выглядит как редукция априорного распределения лексем по суммарным признакам грамматической и лексической семантики, присущим субъективному лексикону реципиента, к определенному грамматическому и лексическому значению, характерному именно для того контекста, который находится в поле внимания реципиента и подлечит в данный момент процессу его индивидуальной языковой обработки. Следовательно, вышеприведенная схема может быть обобщена следующим образом:



Несмотря на то, что до сих пор не создано формального аппарата для определения системы всех возможных состояний любой лексемы (более общо – любой единицы языка), естественно считать, что достаточно адекватной моделью данной системы может служить комплекс грамматических и лексических значений - так, как он представлен, скажем, в большом толковом словаре того или иного языка. В целом же понятие состояния системы (а каждую единицу языка мы рассматриваем и как систему, и как элемент системы высшего уровня) мы предполагаем использовать в рамках парадигмы современного естествознания и техники<sup>18</sup>. Таким образом, при рассмотрении формальных аспектов семантики мы будем исходить из существования соответствия между языковой единицей и ее состоянием:

<sup>18</sup> Боум А. Квантовая механика: основы и приложения. М.: Мир, 1990. — 720 с. Глава IV.

Дирак П. Принципы квантовой механики. 2-е изд. М.: Наука, 1979. — 480 с.

Ландау, Л. Д., Лифшиц, Е. М. Квантовая механика (нерелятивистская теория). — Издание 4-е. — М.: Наука, 1989. — 768 с. — («Теоретическая физика», том III).

$$s: X \rightarrow s(X), \quad (1)$$

где  $X$  – некоторая единица языка;  $s$  – соответствие между  $X$  и  $s(X)$  – формальным объектом, представляющим состояние единицы  $X$ , детерминантами которого являются элементы средств материального выражения семантики, как это обозначено на схеме. Для любой единицы  $X$  состояния образуют некоторое множество (для простоты полагаем его конечным, но неограниченным, хотя это и не принципиально) – обозначим его как обычно символом  $\{s(X)\}$ . Класс единиц определенного типа в языке  $L$  обозначим символом  $W(L)$  или просто  $W$ , если речь идет только об одном конкретном языке; принадлежность  $X$  классу  $W$  обозначим:  $X \in W$ ; множество всех состояний для всех  $X \in W$  обозначим через  $S \equiv \{s(X); X \in W\}$ .

Предположим, что существует оператор  $F$ , действие которого определено на множестве семантических состояний  $S$  и который будем интерпретировать как оператор значения некоторой семантической категории (ими могут быть и категории грамматической семантики – если, например,  $F$  является оператором части речи, то его значениями являются:  $f_1$  – «существительное»,  $f_2$  – «глагол»,  $f_3$  – «прилагательное» и т.д.). Это означает, что  $F$  представляет собой некий интеллектуальный механизм, который, анализируя состояние  $s(X)$ , в котором находится единица  $X$ , идентифицирует значение конкретной семантической категории, соответствующей именно данному состоянию. Понятно, что определение системы операторов типа  $F$  должно основываться на некой семантической теории, формализованной должным образом.

Математическим языком действие оператора  $F$  можно выразить следующим образом:

$$Fs_i(X) = f_i s_i(X), \quad (2)$$

где  $f_i$  – конкретное значение семантической категории; функции  $s_i(X)$ ,  $i = 1, 2, \dots$ , представляют состояния единицы  $X$ , маркирующие ее принадлежность значению  $f_i$  категории  $F$ . Значения  $f_i$ ,  $i = 1, 2, \dots$ , будем называть собственными значениями оператора  $F$ , соответствующими состояниям  $s_i(X)$ ,  $i = 1, 2, \dots$ . Множество всех состояний, отвечающих собственному значению  $f_i$ , будем называть множеством частных семантических состояний и обозначать символом  $S(f_i)$ :

$$S(f_i) := \{s: Fs = f_i s\}. \quad (3)$$

По определению множество  $S(f_i)$  состоит только из тех семантических состояний, которые характеризуются определенным значением семантической категории  $F$ , а именно – значением  $f_i$ .

Уравнение (2) в тех случаях, когда множество, которое пробегает индекс  $i$ , состоит более чем из одного элемента, представляет формальное

выражение явления семантической многозначности. Для единиц лексического уровня это полисемия и омонимия (как лексическая, так и грамматическая). В частности, в случае омонимии это отражает ситуацию, когда словоформа  $X$  в одном контексте может находиться в состоянии, скажем, существительного, а в другом – в состоянии, например, глагола. Такой вид омонимии обычно называется межчастеречной омонимией.

Возможны случаи, когда в уравнении (2) определенному значению  $f_i$  отвечает не одно семантическое состояние  $s_i(X)$ , а несколько:  $s_i^j(X)$ ,  $j = 1, 2, \dots$ . Такие состояния будем называть вырожденными; о вырожденности сигнализирует наличие верхнего индекса у символа семантического состояния  $s_i^j(X)$ . Например, в украинском языке словоформа «МАТИ» со значением категории «часть речи» «іменник» («имя существительное») имеет два грамматических состояния  $g$  (МАТИ):

$$g_{\text{іменник}}^{\text{жінін. рід. одна, наз. відм.}},$$

где слово „МАТИ” имеет лексическое значение „Женщина по отношению к ребенку, которого она родила” и:

$$g_{\text{іменник}}^{\text{чол. рід. множина, наз. відм.}},$$

где слово „МАТИ” имеет лексическое значение „Спортивные матрасы”. Этот пример иллюстрирует явление *внутричастеречной омонимии*.

Количество собственных семантических состояний единицы  $X$ , соответствующих определенному значению категории  $F$ , будем называть кратностью вырождения этого семантического состояния. Так, в рассматриваемом примере состояние  $g$  (МАТИ) имеет кратность вырождения 2.

Семантические состояния, в которых оператор  $F$  имеет одно определенное значение, будем называть *чистыми*. Однако, а priori нельзя запрещать существования семантических состояний, для которых оператор  $F$  не приобретает одного определенного значения, но может характеризоваться, например, двумя. Формально такую ситуацию можно изобразить с помощью соотношения:

$$F(X) = f_1 \alpha_1(s_1) s_1(X) + f_2 \alpha_2(s_2) s_2(X), \quad (4)$$

где семантическое состояние  $s$  языковой единицы  $X$  при воздействии на него оператора  $F$  расщепляется на два, а именно:  $s_1(X)$  и  $s_2(X)$ , где  $s_1(X)$  соответствует значению семантической категории  $f_1$ , а  $s_2(X)$  – значению семантической категории  $f_2$ ; лингвистический смысл функций  $\alpha_1(s_1)$  и  $\alpha_2(s_2)$  будет выяснен ниже.

Состояния, для которых оператор  $F$ , действуя на функцию семантического состояния, представляется комбинацией определенного числа чистых семантических состояний, соответствующих различным собственным значениям этого оператора, будем называть *смешанными семантическими состояниями*.

Таким образом, уравнение (4) определяет семантическое состояние, в котором находится единица  $X$  и которое представляет собой своеобразное явление *суперпозиции* («смеси») чистых семантических состояний  $s_1(X)$  и  $s_2(X)$ , отвечающих собственным значениям  $f_1$  и  $f_2$ , соответственно. Лингвистическая интерпретация уравнения (4) состоит в том, что языковая единица  $X$  имеет семантические признаки одновременно и  $f_1$  и  $f_2$ . Соответствующие показатели – идентификаторы отмеченной принадлежности находятся в выражениях для семантических состояний  $s_1(X)$  и  $s_2(X)$ .

Такая ситуация достаточно типична для языка. Например, в украинском и русском причастиях соединены свойства глагола и прилагательного. Рассмотрим, например, российские причастные лексемы «*ведущий*» и «*ведомый*». Они склоняются по словоизменительной парадигме прилагательного (шесть падежей в мужском, женском и среднем родах и во множественном числе), имея одновременно в своей структуре глагольный морфологический признак активности / пассивности – он здесь материально выражен посредством суффиксов *-ущ* и *-ом*, соответственно. Отмеченный морфологический признак не выглядит сильным с точки зрения принадлежности к глаголу, где он не получил статус словоизменительного; кроме того он характерен для полной словоизменительной парадигмы, а не только для отдельных ее членов – это дает основание для выделения класса слов с такими свойствами в самостоятельную часть речи – «причастие». В украинском языке «причастие» называется «дієприкметник», и в этом названии нашли свое отражение оба признака – глагола и прилагательного<sup>19</sup>.

#### Семантические состояния в формализме нечетких множеств

Подчеркнем принципиальное различие между описанной ситуацией и обычной омонимией. Явление омонимии можно описать уравнением типа (4):

$$Fs(X) = f_1 \alpha_1(s_1) s_1(X) + f_2 \alpha_2(s_2) s_2(X) + \dots,$$

где разные члены правой части соответствуют различным омонимичным состояниям слова  $X$  – если ему присущ тот или иной вид омонимии. Но в процессе речевой обработки, когда происходит «снятие» омонимии, правая часть этого уравнения редуцируется к одному члену, который, собственно, и представляет чистое грамматическое состояние рассматриваемого слова  $X$  в конкретном контексте.

<sup>19</sup> Пример суперпозиции грамматических состояний в турецком языке приведен в книге К.В. Широкова «Іменна словозміна у сучасній турецькій мові», –К.: Довіра, 2009, 318 с., где продемонстрировано что часть словоизменительной парадигмы турецкого имени существительного обладает определенными категориальными и материальными признаками глагола.

Совсем иная ситуация в случае, когда  $X$  находится в смешанном состоянии - в этом случае никакая языковая обработка не способна уменьшить число членов уравнения (4), которое и представляет ее конечный результат. Такое положение сигнализирует о существовании единиц, которые в контексте функционируют в смешанных семантических состояниях.

Вывод о существовании единиц, для которых язык допускает контексты с функционированием только в смешанных семантических состояниях, выглядит достаточно интересным свидетельством невозможности полной формализации языковой системы, точнее, свидетельством границ этой формализации. Это, в частности, означает, что даже наиболее формализуемая и формализованная область теории языка, каковой представляется грамматика, имеет черты нечеткости, что пребывает в некоей дисгармонии с обыкновением грамматического детерминизма и стимулирует разработку специального языка описания неоднозначных грамматических ситуаций. Концептуальную базу такого языка, по нашему мнению, дает теория нечетких множеств Л. Заде. Продемонстрируем, каким образом это достигается.

Обратимся к определению множества семантических состояний  $S$  и множеств  $S(f_i) = \{s: Fs = f_i s_i\}$  частных семантических состояний. Если бы в процессе речевой обработки существовала принципиальная возможность редукции любого семантического состояния к чистому, то множество  $S$  можно было бы представить в виде объединения непересекающихся подмножеств  $S(f_i)$ , то есть справедливой была бы формула:

$$S = \cup_f S(f_i); S(f_i) \cap S(f_j) = \emptyset \text{ при } i \neq j. \quad (5)$$

Ситуация совершенно иная, если мы учтем описанную возможность существования единиц, одновременно характеризующихся, например, двумя (или более) значениями определенной семантической категории. Семантическое состояние  $s(X)$  такой единицы  $X$  уже не будет принадлежать только одному из подмножеств  $S(f_i)$ , но одновременно двум, а возможно и больше.

Формальный механизм описания подобных явлений осуществляется следующим образом. Определим на множестве  $S = \cup_f S(f_i)$  структуру нечеткого в смысле Заде множества. Для этого на каждом из подмножеств  $S(f_i)$  определим функцию принадлежности  $\alpha_i(s)$ , которая для каждого  $s \in S(f_i)$  приобретает определенное числовое значение из интервала  $[0, 1]$ :

$$\alpha_i(s) \in [0, 1]. \quad (6)$$

При этом будем считать, что если  $\alpha_i(s) = 1$ , то состояние  $s$  является чистым. Если же  $\alpha_i(s) < 1$ , то оно отвечает состоянию  $s$ , которое является компонентой смешанного состояния; при этом вторая его

компонента  $s'$  (для двухкомпонентных состояний) принадлежит некоторому подмножеству  $S(f_j)$ ,  $i \neq j$ , со значением функции принадлежности  $\alpha_j(s')$ , также меньшим 1, но таким, что выполняется условие:

$$\alpha_i(s) + \alpha_j(s') = 1 \quad (7)$$

Тогда каждое из подмножеств  $S(f_i)$  превращается в нечеткое множество с функцией принадлежности  $\alpha_i(s)$ :

$$S(f_i) \rightarrow \{S(f_i), \alpha_i(s), s \in S(f_i)\}. \quad (8)$$

Элементами нечеткого множества служат пары  $(s(f_i), \alpha_i(s))$ . Одновременно структура нечеткого множества индуцируется и на всем множестве  $S$  как объединении нечетких множеств (8) при выполнении условия (7). Функция принадлежности  $\alpha_i(s)$  при этом интерпретируется как мера обретения семантическим состоянием  $s$  свойств значения  $f_i$  категории  $F$ . Максимальное значение, равное 1, функция принадлежности приобретает на чистых состояниях, характеризующих слова с однозначно определенным показателем принадлежности соответствующей семантической категории. Для слов со смешанными семантическими состояниями вида:

$$s(X) = \alpha_1(s_1) s_1(X) + \alpha_2(s_2) s_2(X) \quad (9)$$

величина  $\alpha_1(s_1)$  демонстрирует степень, с которой языковой объект  $X$  проявляет свойства значения  $f_1$ , а  $\alpha_2(s_2)$ , соответственно, степень, с которой  $X$  проявляет свойства значения  $f_2$ . Условие (7), а именно:  $\alpha_1(s_1) + \alpha_2(s_2) = 1$ , обеспечивает полноту семантических свойств рассматриваемого объекта и замкнутость семантического описания.

#### Замечания касательно концепции состояний и феноменологической сложности устройства мира

В данной статье мы не предполагаем рассматривать много примеров, иллюстрирующих возможности формализма состояний языковых единиц. Ряд приложений данного формализма был продемонстрирован в других работах, которые цитировались выше. Цель настоящей статьи — показать феномено-логичность грамматики естественного языка и, значит, близость науки о языке к другим дисциплинам естественнонаучного, а значит и технологического цикла. Актуальность этой задачи мы ощущаем в связи с той лингвистической нагрузкой, которая ложится на инженерию знаний, всё более явственно приобретающую черты глобальной индустриальной отрасли и требующей для своего развития всё более изощренных интеллектуальных языковых инструментов.

В связи с изложенным считаем необходимым привести некоторые общенаучные соображения о концепции состояния. Это понятие, которое используется во многих естественных, социогуманитарных и технических дисциплинах, по нашему

мнению, наиболее глубоко теоретически и практически разработано в квантовой механике, где оно является основополагающим.

Согласно канонической доктрине квантовой механики, любая система в определенный момент времени находится в определенном состоянии. Состояние системы формализуется в виде решения уравнения Шредингера для данной системы. Поскольку уравнение Шредингера является дифференциальным уравнением определенного типа в частных производных, множество его решений, которые отождествляются с состояниями рассматриваемой системы, образует бесконечномерное гильбертово пространство. Таким образом, число состояний квантовомеханической системы теоретически бесконечно. Состояние системы представляет максимально полное ее описание в теории и определяет вероятностную интерпретацию поведения системы, однако само по себе оно, вообще говоря, не является непосредственно наблюдаемой величиной. Наблюдаемые величины изображаются в квантовой механике эрмитовыми операторами, которые действуют в гильбертовом пространстве состояний, а возможные значения наблюдаемых величин выражаются через матричные элементы этих операторов в пространстве состояний. Однако в других теориях состояния системы могут быть наблюдаемыми величинами. Скажем, в классической механике состояние материальной точки задается парой координата-импульс в данный момент времени ( $x(t)$ ,  $p(t)$ ), которые являются наблюдаемыми – как по отдельности, так и вместе. В квантовой же механике существует фундаментальное ограничение на одновременное измерение координат и импульсов, определяемое соотношением неопределенности Гейзенберга.

Следовательно, понятие и статус наблюдаемой величины неинвариантны и по-разному определяются в различных естественнонаучных (и других) теориях. Это придает определенную пикантность использованию понятия состояния, например, в теории исчислений, которая в ее нынешнем виде явления наблюдаемости-ненаблюдаемости, по нашему мнению, игнорирует<sup>20</sup>.

Можно было бы потребовать для теории оперировать только наблюдаемыми величинами, но данный вопрос не является простым. Он широко дискутировался во времена становления квантовой теории и не утратил актуальности и в наше время.

<sup>20</sup> Правда, в связи с квантовыми обобщениями таких формальных конструкций как, например, сети Петри (S.Abramsky. Petri Nets, Discrete Physics, and Distributed Quantum Computation. In: Concurrency, Graphs and Models. Springer-Verlag Berlin, Heidelberg. 2008 P. 527-543) или нейронные сети (Lagaris, I. E., A. Likas, D. I. Fotiadis "Artificial Neural Network Methods in Quantum Mechanics", Comput.Phys.Commun.104 (1997) 1-14.) понятия наблюдаемости-ненаблюдаемости, по-видимому, будут введены в теорию объектов, применяемых для описания информационно-технологических процессов и систем.

Достижения этой отрасли теоретического знания действительности содержат столь общие методологические уроки и установки, что они с пользой могут и должны быть усвоены любой наукой, которая имеет амбиции касательно теоретического осознания природы изучаемых ею вещей.

Первым и самым важным из них является, пожалуй, то, что для характеристики состояний объектов действительно используются как наблюдаемые, так и непосредственно не наблюдаемые величины. Причем, по убеждению большинства ученых, строить теорию только из одних наблюдаемых величин невозможно (вспомним высказывание А.Эйнштейна по этому поводу). Но одновременно понятно, что без наблюдаемых никакая научная теория и вообще наука в принципе немыслимы. Наблюдаемые и непосредственно не наблюдаемые величины должны иметь различный логический и онтологический статус, однако, насколько нам известно, общая теория этого вопроса до сих пор в деталях не разработана.

В свете изложенного напрашивается такая интерпретация взаимоотношений между наблюдаемыми и непосредственно не наблюдаемыми величинами теории: они представляют, соответственно, «формальную» и «содержательную» стороны исследуемого объекта (в терминологии теории лексикографических систем (Л-систем)<sup>21</sup> – реестровую и интерпретационную части некоторой гипотетической Л-системы, развившейся вследствие проявления некоторого лексикографического эффекта). В применении к объектам языка такая интерпретация может быть детализирована в том смысле, что любая языковая единица имеет «двустороннюю» природу и ее состояние допускает разложение на формальную составляющую (достижимую для непосредственного восприятия субъектом – будь-то звук или графическое изображение), а содержательная представляется совокупностью «всех контекстов», в которых может функционировать данная языковая единица в данном состоянии – это обстоятельство, собственно, и делает ненаблюдаемой указанную часть данного состояния.

В научной дискуссии о логических и психологических основах феномена наблюдаемости следует упомянуть о такой философской установке как принцип Маха, согласно которому чувственные впечатления упорядочиваются в мышлении человека способом, предусматривающим максимально экономную компоновку этих впечатлений в устойчивые комплексы. Характерно, что А. Эйнштейн, считая этот принцип слишком банальным для того, чтобы он был состоятельным выполнять роль универсального гносеологического закона,

<sup>21</sup> Наиболее полное на данный момент изложение теории лексикографических систем и ее приложений представлено в нашей книге «Компьютерная лексикография» (К.: Наукова думка, 2011, 352 с.)

отмечал особую роль языка в онтологологико-психологическом развертывании процесса познания<sup>22</sup>. Языковые конструкции он полагал не только способом фиксации чувственных комплексов, но и отражением того, что существует (или даже только может существовать) вне этих комплексов и без связи с ними – вспомним в этой связи ньютоновскую «природу вещей». По нашему мнению, замечания Эйнштейна о роли языка (а он был чрезвычайно чувствительным к вопросам философии познания) не случайны – они подчеркивают тезис об универсальности информационно-культурных процессов на всех уровнях познания реальности.

Следующие замечания касаются обсуждения критерия простоты научной теории – его вовсе не обязательно связывать с принципом Маха. Простота научной теории приобретает для большинства исследователей эстетическую окраску – простота и красота математической схемы, подсказанной природой, имеют для них большую убеждающую силу. Заметим, что в эпоху формулировки квантовой теории понятие простоты (и антонимическое, и следовательно – родственное с ним понятие сложности) были общезыковыми, тогда еще не была сформулирована теория сложности – она, как известно, появилась благодаря работам А.Н. Колмогорова только в 50-годы прошлого века (см. Колмогоров А.Н. Три подхода к определению понятия «количество информации» // «Теория информации и теория алгоритмов». – М.: Наука, 1987. – С. 213-223. и ссылки в этой книге). Не было также прояснена и связь такой характеристики как сложность объектов и их описаний (а следовательно – и простота!) с понятием информации, и не были известны количественные меры для оценки этих величин и связи между ними. То, что говорилось о разработанном А.Н. Колмогоровым и другими учеными понятии сложности, его связи с информационными аспектами описания реальности и собственно с понятием информации и ее количественными характеристиками, имеет глубокую связь с критерием простоты и красоты научной теории. Минимальность описания изучаемого объекта, которая по А.Н. Колмогорову является объективной мерой количества информации об этом объекте, побуждает ученых (хотя бы на уровне подсознательной установки) к нахождению описания именно такого типа, хотя и не указывает путей и не дает рецептов, поскольку, вообще говоря, относится к алгоритмически неразрешимым проблемам. Однако отсутствие путей и рецептов не отрицает объективности существования минимального описания – оно лишь свидетельствует о несуществовании формулы или алгоритма для получения новых научных истин. И когда такое

описание найдено, оно, очевидно, обязано выглядеть как наиболее простое – в сущности, оно таким и является. Итак, критерий простоты (или красоты) научной теории, по нашему убеждению, является не столько следствием принципа экономии мышления (который Эйнштейн квалифицирует как «подозрительно коммерческий» и который, собственно, имеет лишь очень опосредованное отношение к делу, поскольку здесь, скорее, речь идет о фундаментальном информационном свойстве объективно существующих вещей, чем о свойстве мышления как субъективного процесса), сколько вытекает из общей природы информации и корреспондируется с формальным определением меры ее количества по А.Н. Колмогорову.

Действительно, когда получено описание изучаемого объекта (процесса, системы и т.п.), которое наиболее адекватно соответствует его сущности, то это описание должно быть минимальным, поскольку оно представляет только существенную информацию об исследуемом объекте и не содержит описания случайных, несущественных деталей, которые «загромождают» существенное «лишними» элементами. Ученый, так сказать, инстинктивно стремится получить именно такое описание исследуемого объекта, которое согласуется с определением информационной меры А.Н. Колмогорова, базирующейся на минимальности описания – этим, по нашему мнению, объясняется и та психологическая уверенность, которую исследователь чувствует, когда ему удается получить простую (красивую!) формулу, уравнение, вывод и т.д. Формализм теории сложности является одновременно и прозрачным, и глубоким, его следует воспринимать онтологически, как объективное свойство вещей. Одним из нетривиальных проявлений отмеченной черты является то, что сложность составного образования, вообще говоря, не равна сумме сложностей формирующих его сущностей. Выражаясь точнее, сложность не является аддитивной функцией системы. Иными словами, если имеется некая система, состоящая из других, «меньших» подсистем, являющихся ее конституентами, т.е. если:

$$D = \bigcup_i D_i$$

где символом  $D$  обозначена рассматриваемая система, а  $D_i$  – ее составляющие, то, вообще говоря:

$$K(D) \neq \sum_i K(D_i) \quad (10)$$

где  $K(D)$  – количественная мера сложности системы  $D$ , а  $K(D_i)$ , соответственно, – количественные меры сложности ее конституентов  $D_i$  (обычно  $K(D_i) < \sum K(D_i)$ ). Данные представления, разумеется, распространяются и на отдельные  $K(D_i)$ , а также на их составляющие.

<sup>22</sup> Гейзенберг В. Физика и философия. Часть и целое. Пер. с нем. – М.: Наука, 1989. – С. 191–196.

В процессе образования, функционирования и взаимодействия составных систем происходит такое явление, которое мы квалифицируем как «самокомпенсацию сложности». Содержание этого феномена сводится к следующему. Характер взаимодействия конститuentов, образующих некое единство (целостность), идентифицируемое как составной объект, является таким, что они проявляют в «связанном» состоянии лишь определенную часть их полной, «имманентной» сложности. Необходимость такого поведения можно трактовать как обеспечивающую принципиальную возможность познания «проявленного» бытия, а может даже и его существования. В противном случае сложность любого объекта была бы актуально бесконечной (потенциально она такой и есть), а так – сложности отдельных компонентов как бы «самокомпенсируются» в процессе формирования целого. Так что можно утверждать, что потенциально сложность любой вещи бесконечна, поскольку сегодня мы не видим границ делимости материи и каждый нижестоящий структурный уровень имеет свою ненулевую сложность. Но «одномоментно» все разновидности сложности компонентов не «проявляются» в целом, они выявляются лишь «поуровнево». Поэтому, сложность в каждом случае подлежит «перенормировке», если идти от аналогии с квантовой электродинамикой, где для устранения расходимостей также приходится применять процедуру «вычитания бесконечностей». Наглядный пример самокомпенсации сложности нам предоставляет язык. Так, например, мерой сложности конкретного слова можно считать длину соответствующей словарной статьи толкового словаря, где учтены эффекты грамматической и лексической семантики, в том числе, множественность грамматических значений, лексическая полисемия, фразеологическая структура лексемы т.д. Между тем, слово в предложении (конкретном контексте) функционирует только в определенном значении – одном или своеобразной «смеси» из нескольких возможных значений для полисемических лексем и, следовательно, мера его сложности в конкретном контексте определяется лишь частью словарной статьи, причем в отдельных случаях она

может составлять лишь десятые, а то и сотые доли от полной сложности лексемы. Таким образом, сложность целого предложения может оказаться меньшей, чем полная сложность отдельного слова, которое является его составной частью.

Конструкция бытия оказывается парадоксальной! Оказывается, что сложные вещи в действительности состоят из еще более сложных. В этом смысле «большее» является меньшим, чем «меньшее». Нетривиальным, на наш взгляд, подтверждением этого тезиса является известный эффект, который имеет и онтологическое, и психологическое измерение – он касается сложности научных теорий: теория атомов, например, не выглядит проще, чем теория молекул, теория ядер не кажется проще теории атомов, теория элементарных частиц не проще теории ядра и так далее. В лингвистике, например, теория слова («лексикология») также не проще теории предложения («синтаксис»). В свете сказанного принцип редукционизма, по которому сложные вещи должны состоять из более простых, выглядит не только не очевидным, но даже сомнительным, что побуждает к некоторой ревизии основ стандартного системного анализа с учетом эффектов, описываемых теорией сложности. На этом уровне последняя начинает приобретать черты и статус естественно- и общенаучной, а не только чисто математической доктрины.

В заключение добавим, что наши наблюдения над феноменологическими аспектами грамматики убеждают нас в том, что язык и наука о языке имеют гораздо больше общего с естественными явлениями и науками, чем это принято полагать в обычной лингвистике. Хочется выразить убеждение, что в дальнейшем эта общность приобретет гораздо более определенные черты.

В нынешнем, не бедном на различные юбилеи году мы, среди ряда других, отмечаем два выдающихся события: 370-летие со дня рождения Исаака Ньютона (4 января 1643 г. – 31 марта 1727 г.) и 110-летие со дня рождения Андрея Николаевича Колмогорова (25 апреля 1903 г. – 20 октября 1987 г.). Этим двум великим мыслителям автор и посвящает данную статью.

УДК 004.89

С.М. Вороной<sup>1</sup>, А.А. Егошина<sup>2</sup><sup>1</sup>ДонНТУ, г. Донецк, Украина, kafedra.sii@gmail.com;<sup>2</sup>ДонНТУ, г. Донецк, Украина, ann\_e@inbox.ru

## ПРЕДВАРИТЕЛЬНАЯ КЛАСТЕРИЗАЦИЯ ТЕКСТОВЫХ ДОКУМЕНТОВ ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА АВТОМАТИЧЕСКОГО ПОСТРОЕНИЯ ОНТОЛОГИЙ

Рассматривается онтологический инжиниринг и методы автоматического построения онтологий. Предлагается применять кластеризацию документов по общей тематике для улучшения качества онтологии с помощью алгоритма LSA/LSI. В качестве понятий, по которым будет происходить составление онтологии, предлагается использовать существенные, встречающиеся в текстах, а в качестве отношения между ними – степень их семантической связи, оцениваемой на основе закона Д. Зипфа.

ОНТОЛОГИЧЕСКИЙ ИНЖИНИРИНГ, КЛАСТЕРИЗАЦИЯ ДОКУМЕНТОВ, АЛГОРИТМ LSA/LSI, СЕМАНТИЧЕСКАЯ СВЯЗЬ

### Введение

Сегодня основная часть информации, доступной во всемирной паутине, не содержит семантики и поэтому ее поиск, релевантный запросам пользователя, а также интеграция в рамках конкретной предметной области затруднены. Для обеспечения эффективного поиска, веб-приложение должно четко понимать семантику документов, представленных в сети. Данный фактор обуславливает бурный рост и развитие технологий Semantic Web. Одним из перспективных направлений в данной области является использование онтологий, которые, являясь новым средством представления и обработки знаний, позволяют создавать интеллектуальные средства для поиска ресурсов в сети Интернет. Они способны точно и эффективно описывать семантику данных для некоторой предметной области и решать проблему несовместимости и противоречивости понятий.

### 1. Онтологический инжиниринг и существующие методы автоматического построения онтологий

Онтологический инжиниринг – одно из популярных направлений компьютерных наук, в рамках которого разрабатываются и проектируются компьютерные онтологии, соединившие в себе различные области знания: искусственный интеллект, логику и философию.

Для представления онтологий используется весь арсенал формальных моделей и языков, разработанных в области искусственного интеллекта – исчисление предикатов, системы продукций, семантические сети, фреймы и т.п.

Онтологии получили широкое распространение в решении проблем представления знаний и инженерии знаний, семантической интеграции информационных ресурсов, информационного поиска и т.д. Интеллектуальные системы на основе онтологий показали на практике свою эффективность, однако построение онтологий требует экспертных знаний в исследуемой предметной области и

занимает существенный объем времени, поэтому актуальной задачей является автоматизация процесса построения онтологий.

На данный момент существует не менее десятка зарубежных систем, относимых к классу инструментов онтологического инжиниринга, которые поддерживают различные формализмы для описания знаний и используют различные машины вывода из этих знаний. Среди уже разработанных онтологий наиболее известными и объемными являются CYC (<http://www.cyc.com>) и SUMO (<http://www.ontologyportal.org/>).

На рынке программных средств достаточно активно продвигается более 50 редакторов онтологий. Одной из наиболее популярных систем работы с онтологиями является система Protege, созданная в Стэнфордском университете (США). По версии разработчиков системы все понятия предметной области делятся на классы, подклассы, экземпляры. Экземпляры могут быть как у класса, так и подкласса и описываются они фреймом.

Сегодня существует множество подходов к автоматизации процесса построения онтологий [1 – 4]. Рассмотрим основные из них.

1. Представление онтологий в виде конечного автомата.

В работе [1] онтологии предлагается представлять в виде орграфа  $G$ , где множество вершин  $V$  представляет множество предметных областей, а множество ребер  $E$  – бинарное отношение между этими предметными областями.

Представление онтологий в виде конечного автомата без выходов позволяет ввести операции на онтологиях. Операции на автоматах означают операции на регулярных языках, которые акцептируются этими автоматами. Основными такими операциями являются следующие: объединение, пересечение, конкатенация или умножение двух автоматов, итерация, обращение.

Алгебраические свойства введенных операций на онтологиях вытекают из соответствующих

свойств операций алгебры регулярных языков. Это значит, что данные операции удовлетворяют следующим законам: коммутативность и ассоциативность операций объединения и пересечения, ассоциативность умножения, дистрибутивность операции умножения относительно операций объединения и пересечения.

Данное множество операций (в случае надобности) можно расширять, по крайней мере, в двух направлениях. Одним из таких направлений является расширение операциями на графах (введение и удаление вершины и ребра, соединение графов, изоморфного соединения декартового произведения и т. д.). Другим направлением является алгебра отношений. Поскольку каждая онтология является представлением некоторой совокупности отношений (в частности: одного), то можно вводить операции реляционной алгебры.

## 2. Построение семантической карты ресурса.

В данном методе для автоматизации процесса построения онтологии предлагается использовать текстовое содержание массива Веб-ресурсов описательного характера определенной тематики [2].

Базовой является задача разработки алгоритма автоматического построения семантической карты веб-ресурса с помощью анализа его текста. Семантическая карта ресурса – это отображение контента веб-ресурса в концептуализацию его содержания, представленное в виде OWL онтологии. Семантическая карта ресурса строится на основе особенностей языка, которые позволяют вытягивать семантические конструкции из текста. Исследования проводились следующим образом:

- формировался набор пар «текст – конструкция языка OWL»;

- по набору выявленных пар «текст – OWL конструкция» выявлялись правила, позволяющие автоматизировать процесс отображения текста в соответствующую OWL конструкцию.

Семантическая карта строится в два этапа: на первом строится формальная семантическая OWL конструкция, на втором происходит привязка полученной конструкции к конкретной предметной области. Формулируются правила, использующие синтаксис языка. Правила синтаксического уровня выявляют семантику на основе принципов построения словосочетаний и предложений. Отдельно выделяются правила, которые сами не строят семантическую конструкцию, но определяют, каким образом (к каким словам) применять правила, непосредственно выявляющие семантические конструкции.

Для того чтобы привязать полученную семантическую модель к интересующей предметной области, используется словарь соответствующей тематики. В итоговой онтологии фиксируются только те семантические конструкции, в которых участвуют термины из словаря предметной области.

Словарь может создаваться экспертом или автоматически на основе статистических методов классификации.

## 3. Подход на основе лексико-синтаксических шаблонов

Данный подход был предложен в [3] и относится к группе методов автоматического построения онтологий, использующих лингвистические средства.

Сторонники подхода утверждают, что для построения онтологий следует активно использовать все уровни анализа естественного языка: морфологию, синтаксис и семантику. Таким образом, для автоматического построения онтологии автором используется один из методов семантического анализа текстов на естественном языке – лексико-синтаксические шаблоны. На основе лексико-синтаксических шаблонов выделяются онтологические конструкции. В целом отмечается, что лексико-синтаксические шаблоны как метод семантического анализа текстов на естественном языке (в случае большого объема коллекции шаблонов) являются эффективным средством для автоматического построения онтологий.

## 4. Автоматическое построение онтологии по коллекции текстовых документов

В работе [4] предлагается подход к решению проблемы автоматического построения онтологий, преимущественно основанный на статистических методах анализа текстов на естественном языке.

Построение онтологий разделено на три этапа: предварительная подготовка коллекции; определение классов онтологии; определение отношений «is-a» и «synonym-of», построение иерархии классов.

На качество построения онтологии влияет предварительная подготовка текста, в частности, особенности коллекции документов. Кластеризация документов по общей тематике может сократить время, затрачиваемое на создание онтологии. Для улучшения получаемой в результате работы системы онтологии предлагается провести предварительную кластеризацию документов коллекции таким образом, чтобы в один кластер попадали тематически близкие документы, а дальнейшую работу проводить отдельно с каждым полученным кластером.

На первом этапе построения онтологии требуется выделить входящие в ее состав классы. Следует отметить, что понятия лингвистической онтологии строго связаны с терминами. Таким образом, данная задача сводится к определению терминов рассматриваемой предметной области.

Алгоритмы извлечения терминов из текстов на естественном языке можно разделить на две группы: статистические и лингвистические. Однако первые обладают определенным преимуществом, поскольку их использование не зависит от лингвистических особенностей конкретного языка. Подход к извлечению терминов в рассматриваемом методе является преимущественно статистическим.

Предполагается, что существующие статистические методы могут показать лучшие результаты, если дополнить их определенными эвристиками.

Предварительно в качестве базовых эвристик предлагается использовать следующие:

- имя класса содержит хотя бы одно существительное;
- общеупотребительные слова обладают большей частотой встречаемости, и приблизительно равной в документах из различных кластеров;
- количество информации термина из нескольких слов больше, чем количество информации отдельных слов.

Этап выделения отношений между классами создаст наибольшие трудности, в связи с чем первоначально имеет смысл говорить об автоматическом тезаурусе (таксономии с терминами). В качестве базовых отношений, действующих между терминами, определим отношения «is-a» и «synonym-of». Для выделения отношения «is-a» можно воспользоваться количественным подходом к информации. Для этого было использовано предположение, что количество информации термина из нескольких слов больше, чем количество информации отдельных слов, входящих в его состав.

Предложенный подход позволяет выделить только базовые отношения, необходимые для построения таксономии. Однако предполагается, что возможно его расширение для выделения других отношений.

## 2. Использование алгоритма кластеризации LSA/LSI для решения задачи автоматического построения онтологий

Предварительный этап в построении онтологии — это подготовка коллекции документов. Особенностью работы с текстами на естественном языке является необходимость предварительной обработки данных. Процесс обработки обычно состоит из нескольких этапов: приведение документов к единому формату; токенизация; стемминг (лемматизация); исключение стоп-слов.

Для улучшения качества онтологии применяется кластеризация документов по общей тематике. Кластеризация существенно сократит время, затрачиваемое на создание онтологии. В качестве алгоритма кластеризации предлагается алгоритм LSA/LSI. Данный метод кластеризации позволяет успешно преодолевать проблемы синонимии и омонимии, присущие текстовому корпусу, основываясь только на статистической информации о множестве документов/терминов.

Латентно-семантический анализ (LSA) [4] — это метод обработки информации на естественном языке, анализирующий взаимосвязь между коллекцией документов и терминами в них встречающимися, сопоставляющий некоторые факторы (тематике) всем документам и терминам.

В основе метода латентно-семантического анализа лежат принципы факторного анализа, в

частности, выявление латентных связей изучаемых явлений или объектов. При классификации/кластеризации документов этот метод используется для извлечения контекстно-зависимых значений лексических единиц при помощи статистической обработки больших корпусов текстов

Существуют два основных отличия метода LSA от прочих статистических методов обработки текстов:

- в качестве исходных данных LSA использует частоту использования слов в отрывках текста, а не частоту совместного использования слов;
- метод собирает данные не о попарной совместной используемости слов, а об используемости множества слов в большом массиве отрывков.

После кластеризации коллекции документов строим онтологию по обработанным текстам. В качестве понятий, по которым будет происходить составление онтологии, будут использоваться существительные, встречающиеся в текстах.

Отношение между двумя понятиями будем представлять степенью их семантической связи, оцениваемой на основе закона Джорджа Зипфа [5] по формуле:

$$\frac{P * R}{N} = C, \quad (1)$$

где  $P$  — частота вхождения слова в текст;  $R$  — ранг этой частоты;  $N$  — общее количество слов в тексте;  $C$  — встречаемость слова в языке. Ранг частоты по Зипфу определяется по частоте вхождения слова в текст. Наиболее часто встречающиеся слова имеют ранг, равный единице, реже встречающиеся слова — ранг, равный двум, ранг  $M$  — наименее часто встречающиеся слова, так что  $M$  — общее число рангов конкретного текста.

Джордж Зипф статистически определил, что встречаемость слова приблизительно одинакова для всех без исключения текстов в пределах одной языковой группы и подчиняется приведенному выше закону. Из закона Зипфа для одного слова следует то, что встречаемость пары слов также будет приблизительно постоянна для любых текстов. Если рассчитать величину встречаемости для слов  $A$  и  $B$  в некотором тексте по формулам (2) и (3):

$$\frac{P_A * R_A}{N} = C_A, \quad (2)$$

$$\frac{P_B * R_B}{N} = C_B, \quad (3)$$

то степень их семантической связи получим по формуле:

$$C_{AB} = \frac{C_A + C_B}{2} * \rho, \quad (4)$$

где  $C_{AB}$  — степень семантической связи между словами  $A$  и  $B$ ;  $\rho$  — количество слов в кортеже ( $A, \dots, B$ ).

Степень семантической связи учитывает влияние всех слов между исследуемой парой. Кроме

того, стоящие между двумя существительными слова имеют назначение связать их синтаксически и отразить семантическую связь. Это расстояние учитывается для того чтобы однозначно определить, встречаются ли исследуемые пары на приблизительно одинаковых расстояниях друг от друга во всех текстах предметных областей.

На рис. 1 показано, что пары с похожей степенью семантической связи хранятся группами. Такой способ хранения вместе с исследованиями статистики расстояний между словами позволит выяснить возможность автоматической классификации, исходя из степени семантической связи слов. Например, пара «животное — медведь» теоретически должна иметь степень семантической связи, близкую к паре «животное — слон» или «насекомое — муравей».

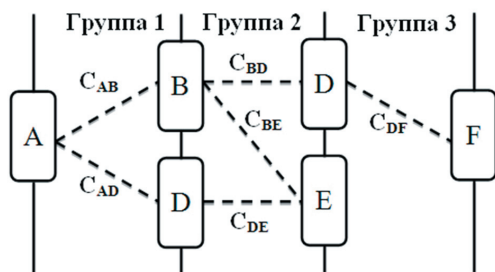


Рис. 1. Хранение отношений между понятиями

Хранение отношений между понятиями будет организовано следующим образом. Когда для пары слов  $A$  и  $B$  рассчитывается степень семантической связи  $C_{AB}$  по формуле (4), для их хранения создается группа, если только это значение не близко к одному из уже существующих. Впоследствии, если другая пара, например,  $A$  и  $D$ , после расчета получит значение, близкое к  $C_{AD}$ , то она попадет в эту же группу.

Для того чтобы построить онтологию нужно выделить отношения, которые в рамках одной группы связывают как минимум два разных понятия. На основе выделенных понятий строится онтология. Затем выделяются те понятия, которые связаны как минимум с двумя уже присутствующими понятиями, после чего эти понятия добавляются к основе, построенной на первом шаге.

### Выводы

В данной работе на основе анализа существующих методов автоматизации процесса построения онтологий установлено, что наиболее популярными являются подходы к созданию онтологий, основанные на статистическом анализе естественно-языкового текста. На качество построения онтологий влияет предварительная подготовка текста, в частности, особенности формирования коллекции текстовых документов.

Кластеризация документов по общей тематике позволит сократить время, затрачиваемое на создание онтологий. В качестве алгоритма кластеризации

предлагается использовать алгоритм LSA/LSI, который является реализацией основных принципов факторного анализа применительно к множеству документов. А отношения между понятиями предметной области предложено устанавливать по степени их семантической связи, оцениваемой на основе закона Зипфа. Предложенная технология позволяет строить онтологии, нуждающиеся лишь в незначительной корректировке эксперта.

**Список литературы:** 1. Крывый С.Л. Автоматное представление онтологий и операции на онтологиях [Электронный ресурс] / С.Л. Крывый, А.Н. Ходзинский. — Режим доступа: <http://shcherbak.net/avtomatnoe-predstavlenie-ontologij-i-operacii-na-ontologiyah>. 2. Рабчевский Е.А. Автоматическое построение онтологий [Электронный ресурс] / Е.А. Рабчевский. — Режим доступа: <http://shcherbak.net/avtomaticheskoe-postroenie-ontologij>. 3. Рабчевский Е. А. Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска / Е.А. Рабчевский // Труды 11-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL 2009. — Петрозаводск, 2009. — С. 69–77. 4. Мозжерина Е. С. Автоматическое построение онтологий по коллекции текстовых документов / Е.С. Мозжерина // Электронные библиотеки: Перспективные методы и технологии, Электронные коллекции — RCDL 2011. — Воронеж, 2011. — С. 293 – 298. 5. Закон Зипфа. Материал из Википедии — свободной энциклопедии [Электронный ресурс]. — Режим доступа: [http://ru.wikipedia.org/wiki/Закон\\_Ципфа](http://ru.wikipedia.org/wiki/Закон_Ципфа).

Поступила до редколлегии 15.10.2012

УДК 004.89

**Попередня кластеризація текстових документів для підвищення якості автоматичної побудови онтології / С.М. Вороной, А.А. Егошина // Біоніка інтелекту: наук.-техн. журнал. — 2013. — № 1 (80). — С. 15–18.**

Проведено аналіз наявних підходів до автоматизації процесу побудови онтологій. З метою зменшення часових витрат на побудову онтології пропонується провести попередню кластеризацію колекції текстових документів таким чином, щоб в один кластер потрапляли тематично близькі документи, а подальшу роботу проводити окремо з кожним отриманим кластером. У якості базового алгоритму кластеризації використовується алгоритм LSA / LSI, а в якості відношень між двома поняттями онтології — ступінь їх семантичної зв'язку, який оцінюється на основі закону Д. Зіпфа.

Л. 1. Бібліогр.: 5 найм.

UDK 004.89

**Preliminary clustering of text documents to increase quality of automatic creation of ontology / S.M. Voronoy, A.A. Yegoshina // Bionics of Intelligence: Sci. Mag. — 2013. — № 1 (80). — P. 15–18.**

In this article analysis of existing approaches to automating the process of building ontologies is made. In order to reduce the time spent on the construction of the ontology it is proposed to conduct a preliminary clustering collections of text documents so thematically similar documents would fall in the same cluster and further work is carried out individually with each cluster obtained. As a basic clustering algorithm LSA / LSI is used, and as the relationship between the two concepts of the ontology - the degree of their semantic link, which is assessed on the basis of Zipf's law.

Fig. 1. Ref.: 5 items.

УДК 81'322.2'33

О.В. Лазаренко<sup>1</sup>, Д.І. Панченко<sup>2</sup><sup>1</sup> ХГУ «НУА», м. Харків, Україна, lazolvlad@gmail.com;<sup>2</sup> ХГУ «НУА», м. Харків, Україна, panchenko\_di@yahoo.fr

## СЕМАНТИКО-КОНТЕКСТНА МОДЕЛЬ РЕФЕРУВАННЯ

Запропоновано семантико-контекстну модель реферування текстів природною мовою, що забезпечує розуміння тексту і його трансформацію за рахунок використання моделі реферату, моделі заголовка, текстової бази та онтологій.

АВТОМАТИЧНЕ РЕФЕРУВАННЯ, СЕМАНТИКО-КОНТЕКСТНА МОДЕЛЬ, ТЕКСТОВА БАЗА, ОНТОЛОГІЯ

### Вступ

Роботи у сфері створення систем автоматичного реферування (АР) мають багате минуле. Понад півстоліття проводились активні дослідження, присвячені пошукам ефективних методів АР.

Усі спроби автоматизації реферування одержали однозначний висновок: залучення виключно статистичних і позиційних методів не дозволяє створити повноцінну систему реферування, яка б замість вилучення речень із тексту стисло викладала зміст. Тому протягом останніх років увага дослідників зосереджувалася переважно на аналізі змісту текстової інформації. У сфері інформаційних технологій намітився позитивний перехід до розробок, зосереджених на знаннях як якісних складових таких процесів.

Окреме місце в цьому ряді займають системи, орієнтовані на автоматичне реферування текстів – побудову стислих, змістовних, виразних фрагментів, які в основному відображають зміст реферованих текстів.

Такі системи передбачають змістове опрацювання тексту в автоматичному режимі у декілька етапів: аналіз, компресія та синтез реферату.

Змістовий аналіз тексту є першим етапом процесу реферування, після якого йдуть не менш складні завдання стиснення змісту й синтез реферату. Отже, пошук шляхів і методів автоматичної компресії тексту є, на наш погляд, дуже важливою складовою досліджень при розробці систем автоматичного реферування. Під *компресією* розуміється сукупність операцій аналітико-синтетичного опрацювання інформації, які мають на меті створення вторинних документів чи демонстрацію змісту вихідного тексту в більш економній формі при максимальному збереженні його інформативності в похідному тексті. Реалізація механізмів компресії у системі АР дозволяє істотно прискорити й полегшити процес укладання рефератів і сприяє інтелектуалізації процесу реферування.

При розробці системи автоматичного реферування нами було сформульовано положення про те, що моделювання процесу реферування як сукупності найскладніших процесів розуміння й

компресії змісту слід починати з вивчення не самих процесів, а з їх результату – реферату. Причому не розгорнутого, інформативного, а стиснутого, індикативного, перш за все тому, що розглядаємо його як відправну точку у дослідженні цього питання, як об'єкт найбільш простий за формою, але такий, що відбиває всі особливості реферативного тексту. Відштовхуючись від цього, автором були проведені дослідження значенневої і синтаксичної структури реферату, які дозволили з'ясувати природу компресії в реферуванні та її наслідки щодо структури реферативних речень. На підставі виявлених особливостей семантико-синтаксичної структури цих речень було побудовано модель індикативного реферату, на базі якої створено першу версію системи реферування «АвтоРеферат» [1].

Аналіз результатів роботи програми «АвтоРеферат» продемонстрував правильність породження реферативних речень у відповідності до розробленої моделі реферування, але разом з тим вказав на змістову неповноту цих речень та необхідність більш глибокого змістового аналізу первинного тексту.

Для вирішення цього завдання необхідно було вивчити семантичні відношення вихідного тексту між рефератом та побудувати модель подання знань у системі автоматичного реферування, що і визначило об'єкт дослідження на наступному етапі.

Основною метою дослідження цього етапу стала побудова когнітивної або семантико-контекстної моделі, яка б дозволила забезпечити глибоке проникнення до змісту тексту та його трансформацію зі збереженням змісту. Для досягнення цієї мети було використано заголовок, текстову базу, онтологію та вивчено вплив кожного з них на аналіз змісту тексту.

### 1. Дослідження семантичних зв'язків тексту, заголовку та реферату

Для формального опису змісту тексту й формального опису смислових перетворень у процесі реферування використовувались:

1) заголовок, який презентує зміст вихідного тексту в концентрованому вигляді;

2) текстова база, яка містить речення, що є «інформаційним ядром» тексту, котре утримує інформацію, залежну від ситуації (тематики тексту);

3) онтології, що містять незалежну від тематики тексту інформацію: онтології верхнього рівня (набір смислових категорій, які описують смислові аспекти, котрі входять до реферату), онтології загальнонаукової лексики й онтології предметних галузей.

На нашу думку, аналіз змісту тексту повинен містити аналіз заголовка, тому що заголовки науково-технічних статей дають уявлення про основний напрямок змісту статті. Важлива риса заголовка – кількість значущих слів у ньому, тобто слів, які несуть основне змістове навантаження і є потенційно ключовими. При цьому заголовки, звичайно, пишуться максимально стисло, надто лаконічно, в них опущені всі семантично другорядні елементи. Тобто ми розглядаємо заголовки як концептуальні інваріанти тексту (термін введений В.А. Широковим), як камертон, що допомагає знайти необхідну для реферату інформацію із первинного тексту. Тому ми розглядаємо заголовок як реферат мінімального обсягу або як текст із максимальним рівнем згортання змісту.

У зв'язку з цим був проведений порівняльний аналіз компресії у заголовку й рефераті (рис. 1).

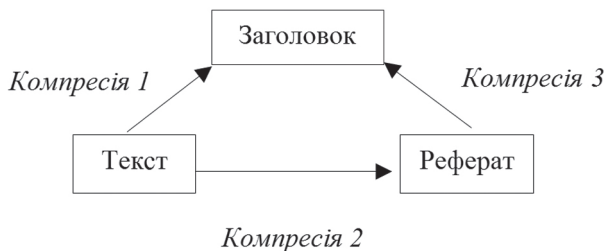


Рис. 1. Згортання інформації

У результаті дослідження змістової і синтаксичної структури заголовка виявилася його схожість зі структурою реферату. Як і в індикативному рефераті змістова структура заголовка складається з двох метазначень – «об'єкт» і «результат». Утім на відміну від реферату вони є елементами змістової структури одного речення і вживаються у зворотній, порівняно зі змістовою структурою реферату (*об'єкт – результат*), послідовності: *результат – об'єкт*. Така подібність змістових структур реферату і заголовка, що містять однакові змістові аспекти, стала підставою для вивчення взаємозв'язку текстів і заголовків, аби за допомогою інформації, що міститься в заголовку, виявити в тексті ті лексичні одиниці, які необхідні для семантичного наповнення моделі реферату даного тексту.

Для цього було побудовано класифікацію лексем синтаксичних конструкцій заголовка (СКЗ) [2]. Усі дієслова-предикати розбито на три класи зі значенням:  $m_1$  – форми подання інформації в тексті,  $m_2$  – етап роботи,  $m_3$  – порівняльна оцінка.

Лексичні значення іменників, що входять до складу іменникових груп, поділено на шість класів зі значенням:  $m_4$  – об'єкт,  $m_5$  – результат (або процес, що прагне досягнення результату),  $m_6$  – ознака,  $m_7$  – мета,  $m_8$  – засіб і  $m_9$  – місце.

Слід зауважити, що при описі заголовка використовується спільна з описом реферативних конструкцій (РК) система позначень, оскільки обидві є частинами розробки спільної моделі процесу реферування. Тому класи віддієслівних іменників визначаються при описі заголовків як клас  $V(m_5)$ , що відповідає класу дієслів, які виступають у ролі предикатів реферативних конструкцій. Усі інші визначаються як класи  $A(m_i)$  з відповідними метазначеннями  $m_i$ .

РК:  $Sr - V(m_1 - m_3) - A(m_4) - A(m_5) - A(m_6) - A(m_7) - A(m_8) - A(m_9)$ ;

СКЗ:  $O(б) / K - Sr - V(m_5) - A(m_4) - A(m_7) - A(m_9) - A(m_8)$ .

Слова, які належать до класів зі значеннями  $m_4$  і  $m_5$ , є основними компонентами структури заголовка. При цьому, на відміну від реферату в заголовковій конструкції, на першому місці, як правило, стоїть актант «результат» (процес, що прагне досягнення результату).

СКЗ:  $V(m_5) - A(m_4)$

Наприклад: *О розвитку соціума*.

Іноді, дуже нечасто, актант «результат» може бути відсутній узагалі.

СКЗ:  $A(m_4)$

Наприклад: *Ринки, правительства и глобальное руководство*.

Відмінна риса заголовків полягає в тому, що практично в усіх заголовках відсутній предикат у чистому вигляді. Він, зазвичай, трансформується у віддієслівний іменник чи іменник зі значенням дії. Це відповідає відомому у стилістиці факту: заміна віддієслівних іменників дієсловами сприяє конкретизації тексту. Звідси випливає: трансформація дієслова підвищує абстрактність, а отже, й узагальненість тексту.

Якщо розглядати класи зі значеннями  $m_1$ ,  $m_2$  і  $m_3$  як віддієслівні іменники, то можна сказати, що  $m_1$  як форма подання інформації в тексті (*изложение, освещение, описание, введение, рассмотрение*) зазвичай не наявна в заголовку,  $m_3$  демонструє суб'єктивну оцінку автора (*дополнение, изменение, исправление, улучшение*), що також зустрічається вкрай рідко, а  $m_2$  – трансформується в  $m_5$ , яка виступає у ролі результату і є одним із двох основних компонентів структури заголовка. Наприклад, у рефераті: *Анализируется инвестиционная привлекательность различных регионов Украины*. А в заголовку: *Анализ инвестиционной привлекательности различных регионов Украины*.

При аналізі заголовків з'ясувалося, що деякі лексеми, утворені від предикатів, збігаються з лексемами, що відносяться до класу  $A(m_2)$ :

$V(m_5) = \{анализ, использование, исследование, обобщение, оценка, реализация, решение, установление\}$ . Сюди також можна додати такі лексеми, які не зустрічалися в дослідженні:  $\{взаимодействие, восстановление, идентификация, моделирование, обращение, определение, организация, планирование, построение, проектирование, развитие, реструктуризация, синтез, формализация\}$ .

Інші компоненти структури заголовка можуть бути присутні, а можуть і не зустрічатися в певних заголовках. Все залежить від того, наскільки важливим є цей компонент у тексті. За результатами порівняльного аналізу з текстом реферату встановлено, що всі компоненти структури заголовку відповідають компонентам структури реферату.

При вивченні граматичних характеристик актантів заголовкових структур було проведено порівняння з рефератом: переважно всі поняття з метазначенням «результат» у заголовку стоять у називному відмінку. Однак є ряд заголовків, коли перед актантом «результат» стоїть прийменник (*о/об, про, к*) чи указник на результат з прийменником або без нього (*к вопросу о, гипотеза о, учителя о, еще раз о, к заданию, одним из вопросов*). Залежно від цього наступні за ними лексеми, що напевне виражають об'єкт чи результат, можуть виступати в трьох відмінках: родовому (*К проблеме управления (руководства) вузовским обществом, давальному (К взаимодействию религиозного и советского образования)*) і місцевому (*О развитии социума*).

Поняття з метазначенням «об'єкт» найчастіше виступає у формі іменника в родовому або називному відмінках за умови, що перед ним не стоїть актант «результат».

Поняття з метазначенням «мета» вживається тільки у формі родового відмінка з прийменником (*Процедура дедуктивного вывода для планирования работы системы управления в динамической среде*), поняття з метазначенням «засіб» — у формі орудного (*Об алгоритме исправления орфографических ошибок методом корректирующего словаря*), а поняття з метазначенням «місце» — місцевого відмінка (*Обзор исследований по автоматической обработке естественного языка в США*), що цілком збігається зі структурою реферату.

Проведені дослідження дозволили відстежити чітке продовження процедури згортання (компресії) в заголовку перед процедурою компресії в рефераті як на семантичному, так і на синтаксичному рівнях. При збереженні тих самих змістових складових у рефераті й у заголовку вони мають також синтаксичні й граматичні особливості їх вираження, які підвищують рівень компресії тексту заголовка.

Наявність одних і тих самих семантичних компонентів у заголовку, рефераті й тексті, що є різними формами вираження одного й того ж поняття,

дозволяє описати смислові структури словосполучень на різних рівнях згортання інформації, для чого необхідна наявність онтологій предметних галузей та загальнонаукової лексики. І лише тоді можна побудувати низку переходів по цих структурах від заголовка до тексту й далі до реферату при його змістовому конструюванні. Для здійснення цього переходу необхідно побудувати текстову базу, до якої входять речення, що містять слова із заголовка або ж їх смислові еквіваленти з тексту.

Таким чином, ще одним необхідним елементом у роботі системи АР є текстова база. Вона складається з фактів і тверджень, пов'язаних із певною ситуацією (конкретним текстом). І на противагу онтології, яка містить не залежну від ситуації і стану інформацію, є «інформаційним ядром», що містить залежну від ситуації і стану інформацію.

## 2. Побудова текстової бази в системі АР

Для побудови текстової бази знань ми відштовхувалися від понять, які містяться в заголовку документа. Заключовими словами, знайденими у заголовку відшукуються відповідні їм іменникові групи в тексті (будується текстова база знань), після чого формуються ланцюжки іменникових груп для реферативних конструкцій відповідно до наявної моделі реферату.

Текстова база має містити інформацію, виражену власне текстом, тобто референціальне значення тексту, оскільки, як відомо, на відміну від художньої літератури для наукової й технічної літератури характерне превалювання референціальних значень.

Згідно з концепцією, запропонованою в роботі [3], для опису глобального змісту тексту необхідна побудова схеми, що забезпечувала б «швидкий аналіз поверхових структур і побудову простої та жорсткої семантичної конфігурації». Відповідно до даної концепції, розуміння тексту припускає часткове планування (або очікування) структур і значень речень і цілого тексту. Такі структури тексту (називаються макроструктурами) являють собою узагальнений опис основного змісту дискурсу, який читач буде у процесі розуміння, і вважаються фактично рефератом або резюме.

Побудова макроструктури здійснюється в процесі застосування макроправил. До ряду таких правил належать правила скорочення (несуттєвої інформації), узагальнення (двох чи більше однотипних пропозицій) і побудова (тобто комбінації кількох пропозицій в одну), що, власне, є еквівалентом побудованим нами моделям компресії тексту й моделі реферату.

Передбачається, що послідовно застосовуючи макроправила, можна побудувати формальний перехід від вихідного тексту до реферату, який складається з кількох речень. Побудова макроструктур

читачами – це один із різновидів так званих стратегій розуміння дискурсу.

Поряд із цим схема дозволяє «забезпечувати певні загальні обмеження на можливі локальні й глобальні значення текстової бази», що відповідає обмеженню на змістові аспекти, які ми виділяємо в тексті для змістового наповнення моделі реферату.

Оскільки «процес розуміння передбачає часткове планування або очікування (у нашому випадку очікування) «структур і значень речень та цілих текстів» [3], то саме побудова текстової бази (ТБ), що являє собою основу макроструктури тексту, дозволить наблизитися до формального опису деяких аспектів процесу розуміння тексту людиною.

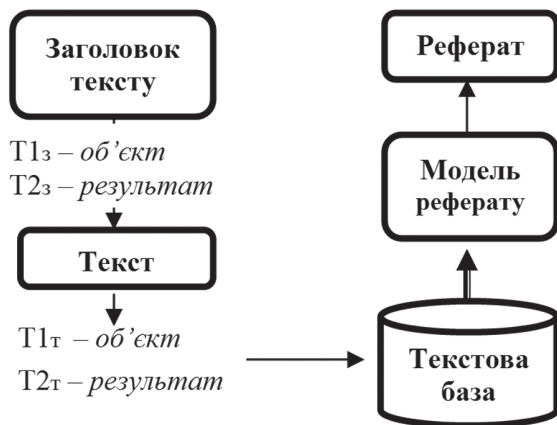


Рис. 2. Схема реферування на основі текстової бази

Відштовхуючись від цих міркувань і розроблених нами моделей різноманітних етапів реферування, ми змогли підійти до розв'язання завдання побудови семантико-контекстної моделі реферування, яка може сприяти глибинному проникненню у зміст текстів і його трансформації зі збереженням сенсу.

Із цією метою ми використали виявлену й описану схожість змістових структур реферату й заголовка для вивчення взаємозв'язку текстів та їх заголовків, для того щоб за допомогою інформації, яка міститься у заголовку, виявити в тексті ті лексичні одиниці, що є необхідними для семантичного наповнення моделі реферату екстенсійною семантикою даного тексту.

У процесі дослідження наукових текстів ми проаналізували змістову й структурну зв'язність наукового тексту. Перші результати наших досліджень показали, що змістова зв'язність заголовка, який виступає в ролі своєрідного ключа до розуміння тексту й безпосередньо тексту, здійснюється переважно за допомогою засобів лексемрекурентності, імпліцитної рекурентності й референційної рекурентності. Використання того чи іншого типу зв'язності визначається структурно-композиційною характеристикою тексту.

На сьогодні ведеться робота над побудовою схеми, яка забезпечує швидкий аналіз поверхневих

структур тексту за рахунок використання слів-вказівників на змістові аспекти в тексті, необхідних для побудови реферату (об'єкт, результат, мета, засіб).

Однак не завжди виділення речень за допомогою слів-вказівників дозволяє здійснити оптимальний вибір речень із тексту для текстової бази. Для того щоб бути впевненим у правильності вибраних речень, необхідна наявність у системі автоматичного реферування онтологій, у яких достеменно зафіксовані всі концепти відповідної предметної галузі.

### 3. Подання знань в системі АР

Для вирішення задачі подання знань у системі автоматичного реферування виникла необхідність створення декількох онтологій, які сукупно дають можливість змістового конструювання реферату (через більш точне виділення в текстовій базі необхідних для цього іменникових груп).

*Онтологія верхнього рівня* перебуває над онтологіями предметних галузей (ПГ) і є самостійною, не залежить від ПГ і конкретної задачі, оскільки описує загальні поняття (простір, час, матерія, об'єкт, подія, дія, результат тощо; в нашому випадку – об'єкт, результат, мета й засіб).

У запропонованій нами моделі такий словник використовується для опису категорій реферативних конструкцій. Він є виродженою онтологією і являє собою кінцеву множину понять верхнього рівня, що відображають змістову структуру рефератів.

*Проміжна онтологія загальнонаукової лексики*, що містить загальні поняття й відношення для різних ПГ. Певною мірою вона запроваджується як посередня ланка між онтологіями ПГ.

*Онтології предметних галузей* містять поняття певної сфери знань або сфер, які входять до неї, і складаються з об'єктів та зв'язків між ними, описаних у термінології конкретної ПГ.

Онтології-словники не вводять експліцитного значення термінів, але в тих випадках, коли терміни належать до досить вузького словника ПГ (технічного, лінгвістичного, медичного, економічного тощо), тобто їхні значення чітко визначені, такі онтології можуть використовуватися на практиці в ролі опису ПГ.

Зв'язок категорій з онтології верхнього рівня з об'єктами з онтології ПГ описується формалізмом, які задають принципи віднесення до цих категорій об'єктів світу.

Етап концептуалізації включає витяг термінів з текстів і розподіл їх за категоріями, на базі чого і будується концептуальна модель ПГ. При цьому ми розглядаємо не всі категорії, що входять до онтології верхнього рівня, а лише ті, які зустрічаються в рефератах. У результаті в процесі побудови моделі реферату ми одержали вироджену онтологію у

вигляді словника змістових категорій, притаманних рефератам – *об’єкт, результат, мета, засіб*.

При побудові онтології ПГ ми відштовхувалися від того, що в реальному світі є об’єкти, яким притаманні властивості і значення [4, ст. 20-27]. А побудова онтології ПГ зводиться до концептуалізації світу в поняттях словника для об’єктів, їхніх якісних характеристик, відмінних рис тощо. Поняття, визначені в такий спосіб у словнику, становлять прийняту в ПГ термінологію.

Як правило, онтологія містить таксономію понять, що успадковують властивості одного (проста таксономія) або ряду більш загальних понять (складна таксономія). Поняття пов’язані відношеннями з основними в системі поняттями або з невід’ємними типами. Відношення зазвичай мають кілька персональних властивостей, таких як ім’я відношення і його опис, закріплені документально. Для системи відношень установлюються семантичні обмеження.

Іншими словами, онтологічне (або концептуальне) моделювання має справу з питанням: як декларативним чином, який припускає повторне використання, описати ПГ і як обмежити

використання цих даних для вирішення конкретного завдання аналізу.

Для розв’язання цього завдання модель ПГ у нашому дослідженні базується на поняттях верхнього рівня (об’єкт, результат, мета й засіб), які потім співвідносяться з поняттями з конкретної ПГ. Онтологія такого типу є простою таксономією.

У кожній прикладній онтології вихідна множина понять формується в результаті відбору характерних предметів і явищ, після чого будуються дерева класифікації концептів. При цьому під терміном розуміють слово (або сполучення слів), що є точним позначенням якого-небудь поняття з галузі науки й техніки, а поняття інтерпретується як логічно оформлена думка про клас предметів і явищ. Тобто в понятті відбивається сукупність істотних ознак, кожна з яких, узята окремо, необхідна, а всі разом – достатні, аби з їхньою допомогою можна було виділити даний предмет серед всіх інших й узагальнити однорідні предмети в клас. Предмети і явища, об’єднані в один клас, становлять обсяг відповідного їм поняття.

При класифікації предмети і явища, які входять до (обсягу) поняття верхнього рівня, діляться на

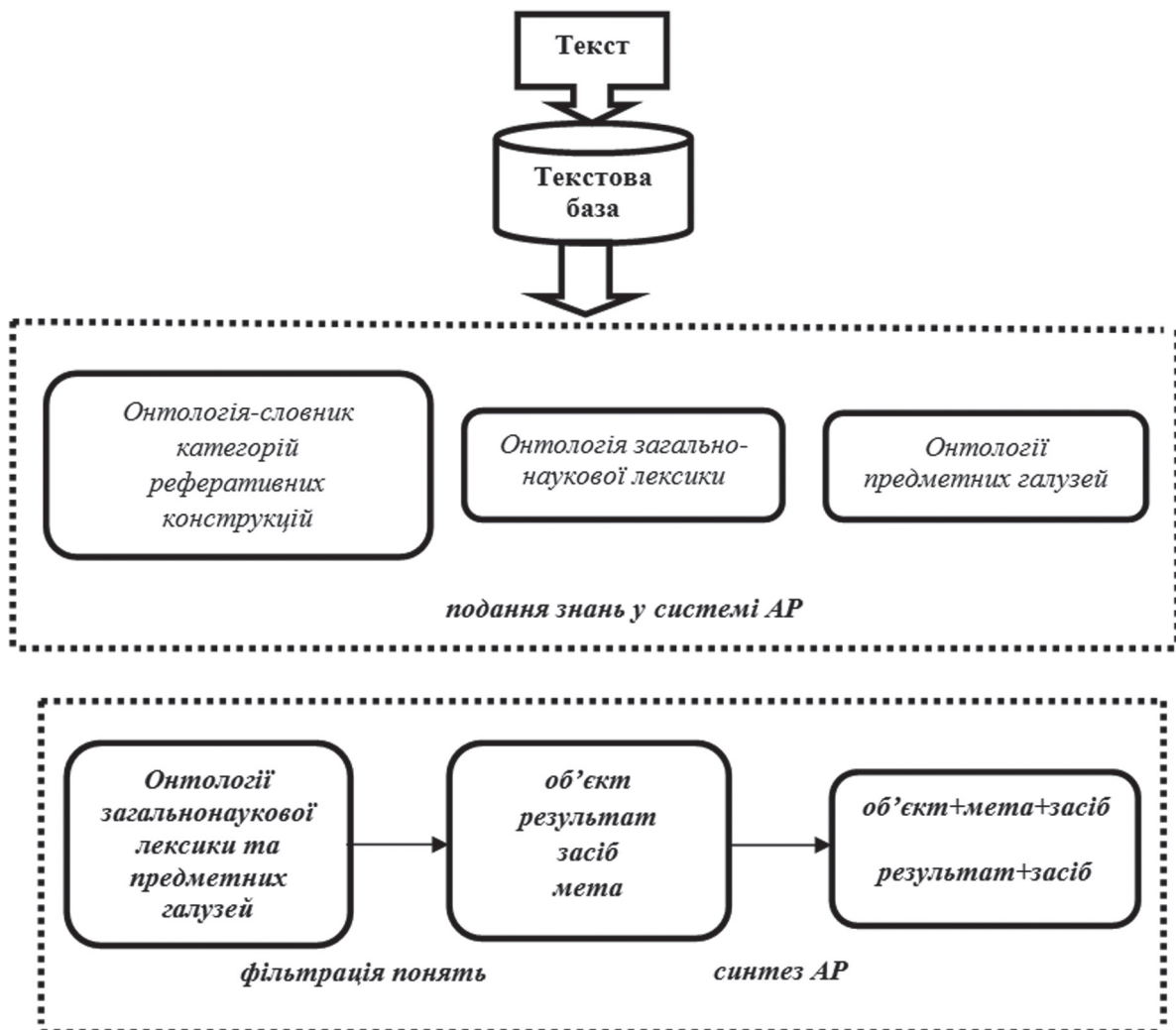


Рис. 3. Подання знань у системі автоматичного реферування

підкласи, що утворюють нові поняття, котрі є частиною поняття верхнього рівня, і так далі до рівня конкретних термінів ПГ.

У цьому аспекті таксономія розглядається як синонім систематики – надійної і зручної системи реєстрації розмаїтості предметів та явищ шляхом розподілу за класами на підставі їхньої подібності і спорідненості в найширшому сенсі [5].

В рамках даного підходу запропоновано процедуру фільтрації термінів з текстової бази знань на етапі змістового конструювання реферату. В основу процедури покладені попередньо описані структури знань, які вказують системі реферування, які поняття є характерними для рефератів і які концептуальні властивості має те чи інше поняття.

Ці структури є виродженою онтологією у вигляді словника категорій реферативних конструкцій, онтології загальнонаукової лексики та онтології ПГ у вигляді таксономії понять конкретної галузі знань.

Побудовані онтології є лише першим кроком до створення відповідної системи подання знань у системі автоматичного реферування. На нашу думку, кращі результати можливі лише за наявності повних онтологій термінологічної і загальнонаукової лексики, а також за наявності в системі АР текстової бази у вигляді «відносно простої і незмінної семантичної конфігурації» [3], що дозволяє досить швидко знаходити змістові аспекти для заповнення моделі реферату преференційними значеннями мовних одиниць тексту, які характерні для наукової літератури. Створення такої текстової бази передбачає дослідження процесів розуміння тексту, заснованих на лінгвістичному аналізі структури зв'язного тексту.

### Висновки

В статті запропоновано підхід до опису семантичних зв'язків «Текст-Реферат» [6] у вигляді семантико-контекстної моделі реферування, що забезпечує розуміння тексту і його трансформацію зі збереженням смислу за рахунок використання заголовка, який виражає зміст вихідного тексту в концентрованому вигляді; текстової бази, яка містить речення, що являють собою «інформаційне ядро» тексту й відображають його тематику, та онтологій, що містять незалежну від тематики тексту інформацію: онтологій предметних галузей,

онтології загальнонаукової лексики й онтології верхнього рівня.

Запропонована семантико-контекстна модель реферування з опертям на знання дозволяє суттєво поліпшити роботу системи «АвтоРеферат».

**Список літератури:** 1. Лазаренко О. В. Моделювання процесу узагальнення в системі автоматичного реферування / О. В. Лазаренко, А. А. Яковенко. – Х.: Изд-во НУА, 2007. – 136 с. 2. Лазаренко О. В. Аналіз смислової структури заголовка як тексту з максимальним рівнем узагальнення / О. В. Лазаренко, Т. В. Попова // Проблеми семантики слова, речення та тексту: Збірник наукових праць. – К.: КНЛУ, 2004. – Вип. 12. – С. 143–149. 3. Дейк ван Т. А. Стратегии понимания связного текста / Т. А. ван Дейк, В. Кинч // Новое в зарубежной лингвистике. – Вып. 23: Когнитивные аспекты языка. – М., 1988. – С. 153–211. 4. Клещев А. С. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия «онтология» / А. С. Клещев, И. Л. Артемьева // НТИ. – 2001. – Сер. 2. – № 2. – С. 20–27. 5. Лазаренко О. В. Классификация понятий в системе автоматического реферирования / О. В. Лазаренко // Wiek XXI. – THE 21<sup>st</sup> CENTURE. – Варшава, PWSBiA, 2002. – № 4 (6) – С.189–196. 6. Панченко Д.І. Моделювання семантичних зв'язків «Текст-Реферат» у системах автоматичного реферування: автореф. дис. ... канд. філол. наук: 10.02.21 – структурна, прикладна та математична лінгвістика. – Харків: Вид-во НУА, 2012.

Надійшла до редколегії 23.10.2012

УДК 81'322.2'33

**Семантико-контекстная модель реферирования** / О.В. Лазаренко, Д.И. Панченко // Бионика интеллекта: науч.-техн. журнал. – 2013. – № 1 (80). – С. 19-24.

В статье рассматриваются вопросы построения семантико-контекстной модели реферирования и модели представления знаний в системе АР с использованием онтологий. Предложена модель реферирования, позволяющая приблизиться к формальному описанию некоторых аспектов процесса понимания текста человеком.

Ил. 3. Библиогр.: 6 назв.

UDK 81'322.2'33

**Semantic-context model of summarization** / O.V. Lazarenko, D. I. Panchenko // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 19-24.

In article the approach to creation of creation knowledge-based automatic summarization systems is considered. The necessity of using context-semantic model has demonstrated.

Fig. 3. Ref.: 6 items.

УДК 004.934

Н.В. Бабкова<sup>1</sup>, О.В. Канишева<sup>2</sup>, З.А. Кочуева<sup>3</sup><sup>1</sup>НТУ «ХПИ», м. Харьков, Украина, nadjenna@gmail.com;<sup>2</sup>НТУ «ХПИ», м. Харьков, Украина, olya-kanisheva@rambler.ru;<sup>3</sup>НТУ «ХПИ», м. Харьков, Украина, kochueva@kochuev.com

## МОДЕЛИРОВАНИЕ СЕМАНТИЧЕСКИХ ОТНОШЕНИЙ В ОНТОЛОГИИ

В статье рассмотрен подход к моделированию семантических отношений в онтологии, проанализированы теоретические аспекты использования онтологий в семантической паутине. В работе показаны типы связей, присущие онтологиям, дано описание онтологических отношений средствами алгебры конечных предикатов. Приведен пример родо-видового дерева для лексико-семантического класса существительных.

ЕСТЕСТВЕННЫЙ ЯЗЫК, ОНТОЛОГИИ, АВТОМАТИЗИРОВАННЫЙ СЕМАНТИЧЕСКИЙ АНАЛИЗ, СЕМАНТИЧЕСКИЕ ОТНОШЕНИЯ, АЛГЕБРА КОНЕЧНЫХ ПРЕДИКАТОВ

### Введение

Развитие наукоемких областей человеческой деятельности в современном обществе сопровождается возрастанием роли компьютерных технологий. Резко возросший поток информации обусловил необходимость поиска новых способов ее хранения, представления, формализации, систематизации и автоматической обработки. Возрос интерес к всеобъемлющим базам знаний, которые можно использовать для различных практических целей. Большой интерес вызывают системы, способные без участия человека извлечь сведения из текста. Все это обусловило появление Semantic Web (SW) как расширения World Wide Web. В Semantic Web гипертекстовые страницы снабжены дополнительной разметкой, в которую включены сведения о семантике элементов страницы. Неотъемлемым компонентом Semantic Web является понятие онтологии, описывающее смысл семантической разметки.

Под онтологией (в наиболее общих чертах) понимают систему понятий некоторой предметной области, которая представляется как набор сущностей, соединенных различными отношениями [1]. Онтологии используются для формальной спецификации понятий и отношений, которые характеризуют определенную область знаний. Основным преимуществом онтологий как способа представления знаний является их формальная структура, которая упрощает их компьютерную обработку.

Онтологии используются как источники данных для многих компьютерных приложений (для информационного поиска, анализа текстов, извлечения знаний и т. д.), позволяя более эффективно обрабатывать сложную и разнообразную информацию. Этот способ представления знаний позволяет приложениям распознавать те семантические отличия, которые являются само собой разумеющимися для людей, но не известны компьютеру [2].

Онтологии получили широкое распространение в решении проблем представления знаний и инженерии знаний, семантической интеграции информационных ресурсов, информационного поиска и т.д. Интеллектуальные системы на основе

онтологий показали на практике свою эффективность, однако построение онтологии требует экспертных знаний в исследуемой предметной области и занимает существенный объем времени (возникает так называемая проблема «бутылочного горлышка»), поэтому актуальной задачей является автоматизация процесса построения онтологии.

В связи с вышеизложенным целью работы является: рассмотреть существующие подходы к моделированию семантических отношений; показать использование алгебры конечных предикатов для моделирования семантических отношений в онтологии, что позволит улучшить качество автоматического создания онтологий или автоматического пополнения онтологий и, соответственно, повысить эффективность процессов обработки данных и знаний в компьютерных системах и сетях.

### 1. Онтологии как основа создания семантической сети

Большинство электронных документов находится в сети Интернет. Современный взгляд на Интернет предполагает создание семантической паутины – это надстройка над существующей Всемирной паутиной, которая призвана сделать размещенную в ней информацию более понятной для компьютеров. Машинная обработка возможна в семантической паутине благодаря двум её важнейшим характеристикам:

1. Повсеместное использование унифицированных идентификаторов ресурсов (URI). Традиционно в Интернете эти идентификаторы используются для установки ссылок на адресуемый объект (например, веб-страницу, файл или ящик электронной почты). Поскольку URI глобально уникальны, они позволяют называть одни и те же предметы в разных местах в семантической паутине. При этом URI протокола HTTP можно одновременно использовать как адреса документов, содержащих машиночитаемые описания этих предметов [3].

2. Использование семантических сетей и онтологий. Современные методы автоматической обработки данных, доступных в Интернете, как правило,

основаны на частотном и лексическом анализе текстового содержимого, которое, прежде всего, предназначено для восприятия человеком. В семантической паутине вместо этого используется стандарт RDF, описывающий семантические сети (графы), в которых узлы и дуги имеют URI. Утверждения, кодируемые с помощью RDF, в дальнейшем можно интерпретировать с помощью онтологий, созданных по стандартам RDF Schema и OWL, чтобы получать из них логические заключения. В основе онтологий лежат математические формализмы, называемые дескрипционными логиками, которые представляют собой семейство языков, позволяющих формально и однозначно описывать понятия. Структура семантической паутины представлена на рис. 1.

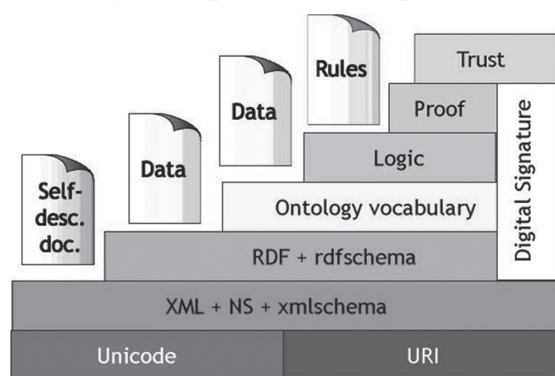


Рис. 1. Составляющие семантической паутины

Онтологии широко используются во всех областях, занимающихся обработкой данных на естественном языке. В связи с использованием онтологий в различных приложениях возникла необходимость создания стандартизированных способов их представления. Началось развитие разнообразных языков, которые могли бы применяться повсеместно во всех системах (самыми известными являются RDF и OWL). Возникло также большое количество редакторов для создания, пополнения и изменения онтологий. Каждое из этих средств обычно направлено на работу с определенным форматом данных и обладает своими особенностями.

Однако, несмотря на это, на сегодняшний день для решения практически любой задачи автоматической обработки естественного языка требуется онтология.

## 2. Семантическая сочетаемость слов

Основной лингвистической единицей, вокруг которой разрабатывается тот или иной прием определения смысла или содержания, является слово. Слова в предложении связаны друг с другом грамматически и по смыслу. Грамматические связи обеспечивают грамматическую правильность речи. Примеры грамматических связей: согласование прилагательного-определения с определяемым существительным в роде, числе и падеже, глагольное управление существительными. Задача грамматической сочетаемости, благодаря морфологическому

анализу и специфическим маркерам (например, изменение окончания), является практически идентифицируемой, особенно для часто используемых конструкций словосочетаний.

Смысловые связи обеспечивают правильность высказывания по смыслу, и смысловые отношения слов редко выражаются при их написании. Тем не менее, употребляя слово, мы должны согласовывать его по смыслу с другими словами. Это смысловое согласование выражается в двух типах словесной сочетаемости — семантической и лексической.

Семантическая сочетаемость слова — это его способность вступать в сочетания с целыми классами слов, объединяемых общностью смысла. Например, глаголы думать, полагать, радоваться, смеяться, грустить и другие описывают различные состояния человека; значит, и сочетаться они могут лишь с такими словами, которые обозначают человека (это и есть один из семантических классов): мальчик, старик, прохожий, врач, учительница и т.п. Только в сказке или в фантастической повести возможны выражения вроде коза задумалась или шкаф засмеялся.

Лексическая сочетаемость слова — это его способность вступать в сочетания не с любым словом из какого-либо семантического класса, а только с некоторыми. Например, существует класс слов, объединяемых общим смыслом «множество, совокупность»: стадо, табун, стая, рой, косяк и т.п. При необходимости обозначить множество каких-нибудь животных мы не можем сочетать название любого животного с любым из этих слов. Говорят: стадо коров, табун лошадей, стая птиц, рой пчел, косяк рыбы (но не «стая рыбы или рыб», «стадо пчел», «рой лошадей» и т.п.).

## 3. Типы связей в онтологии

Считается, что связи между понятиями в реальном мире и его моделях бывают различного типа, например, таксономические связи, выражающие отношения «является видом» или отношение «общее/специфичное», композиционные связи, выражающие отношение «является частью», и «топологические» связи (Topological relationships) или проблемно-специфичные (Domain specific relations), показывающие «пути» физических взаимодействий (interactions) между понятиями или обеспечивающие информацией о пространственном их расположении [4-6]. Каждая из таких типов связей представлена в конкретных онтологиях множеством вариантов.

**Таксономические связи.** К наиболее типичным таксономическим отношениям относится отношение, ставящее две сущности в зависимость «частное — общее» (является потомком) [3-5], исчерпывающее разбиение на подклассы (Exhaustive-Subclass-Partition). К таксономическим может быть отнесено и отношение «является представителем» (Instance-of) [5, 6].

**Композиционные связи.** Согласно [4, 6] в «искусственном интеллекте» и концептуальном

моделировании различают несколько типов отношений часть-целое (part-whole), среди которых наиболее популярны:

- *компонент-объект* (component-object) – физическая или концептуальная часть [3, 5];
- *ингредиент, вещество – для объекта* (stuff-object) (например, мука в хлебе, пластик – материал для монитора) [4, 5];
- *член из набора* (member-collection) – сущность является членом множества (набора, коллекции) однородных сущностей [3, 4]).

«Топологические» связи. Согласно анализу реальных онтологий [4, 5], «топологическими» (или проблемно-специфичными) являются, например, такие связи: сущность имеет атрибутом что-то (отношение объект – его свойство);

Следующая «группа» связей – **связи сущностей с процессами и связи сущностей с сущностями посредством процессов**: сущность поддерживает процесс, сущность является инструментом процесса, актор/сущность участвует в процессе, актор/сущность выполняет процесс, актор руководит/управляет процессом, и т.п.

Другая группа связей – **причинно-следственные связи**: процесс влияет на прохождение другого процесса (например, препятствует, оберегает от), процесс влияет на сущность, процесс вызывает другой процесс, абстракция является предпосылкой для абстракции и др.

**Временные связи** (или связи сущностей, протекающих во времени): процесс связан по времени с другим процессом, временной интервал является частью другого, временной интервал связан с другим интервалом, процесс является частью другого процесса.

**Пространственные связи**: сущность частично располагается в другой сущности, объект пространственно связан с чем-то (находится рядом, пересекается...), организм обитает в сущности, сущность наполняет другую сущность.

Все рассматриваемые отношения являются бинарными отношениями. Для анализа структуры этого достаточно, поскольку к таковым можно свести любые n-арные отношения.

#### 4. Описание онтологических отношений средствами алгебры конечных предикатов

Между словами в словосочетаниях существуют различные типы отношений (рис. 2). Отношения иерархии как разновидность сильных парадигматических отношений соответствуют отношениям подчинения или отношениям типа «выше-ниже». В пределах иерархических отношений различают отношения типа «род-вид» и «целое-часть» [7].

Отношение «род-вид» является одним из важнейших видов связей между понятиями. При этом родовым (подчиняющим) называется понятие, выражающее существенные признаки класса предметов, являющихся видами этого рода.

Соответственно видовым (подчиненным) называется понятие, которое отображает существенные признаки класса предметов, являющегося видом какого-либо рода.

В качестве математического аппарата для моделирования семантики свободных словосочетаний с родо-видовыми отношениями предлагается использовать алгебру конечных предикатов (АКП). Это универсальный математический аппарат для описания детерминированных, дискретных и конечных информационных объектов. Этот аппарат позволяет описывать различные языковые закономерности в рамках единой математической модели.

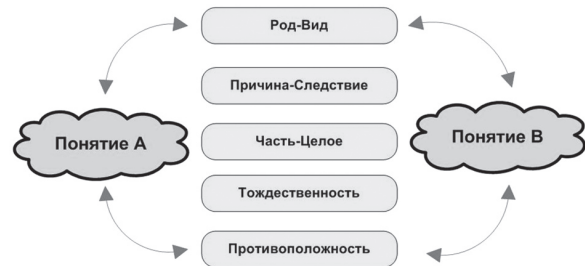


Рис. 2. Виды логических связей между понятиями

Аппарат алгебры конечных предикатов (АКП) является обобщением аппарата булевых функций и аппарата многозначной логики. АКП дает возможность перехода от алгоритмического описания лингвистических закономерностей к описанию их в виде уравнений.

Словоформы принимают свои сочетания в контексте. Контекстом в нашем случае служит словосочетание в определенной предметной области. Поэтому значение словоформы определяется лишь теми признаками, которые характеризуют ее связь со значениями слов в контексте, а также с объектами предметной области.

Семантическая сочетаемость словоформ зависит от знаний предметной области и от родо-видовой структуры [8].

Родо-видовое отношение в русском языке – это бинарное отношение  $R$ , заданное на множестве словоформ  $D$ , такое что для любого  $d \in D$ , всегда существует  $d^1 R d$  и, кроме того,  $R$  не рефлексивно ни для одного  $d \in D$ , антисимметрично и транзитивно. Отношение  $R$  называют еще отношением древовидного порядка. Для иерархического порядка, если  $d R d^1$ , то говорят, что  $d^1$  следует за  $d$ . Если, кроме того, нет такого  $d^{11}$ , что  $d R d^{11}$  и  $d^{11} R d^1$ , то говорят, что  $d^1$  непосредственно следует за  $d$ .

Использование аппарата АКП требует перехода от двухместного родо-видового отношения к  $n$ -местному. Каждому  $n$ -местному отношению  $R$ , заданному на  $n$ -той декартовой степени  $D^n$  множества словоформ  $D$ , можно поставить в соответствие каноническое уравнение, т.е. уравнение вида:

$$f_R(x_1, x_2, \dots, x_n) = 1, \quad (1)$$

которое связывает переменные  $x_1, x_2, \dots, x_n$  точно так же, как и отношение  $R$ . Предикат  $f_R$  выбирается следующим образом:

$$f_R(x_1, x_2, \dots, x_n) = \begin{cases} 1, & \text{если } (x_1, x_2, \dots, x_n) \in R \\ 0, & \text{если } (x_1, x_2, \dots, x_n) \notin R, \end{cases} \quad (2)$$

где  $(1, 2, \dots, n)$  – количество уровней иерархии. Записав предикат  $f_R$  в виде формулы АКП, получаем аналитическую запись отношения  $R$  в форме уравнения (1).

Каждый уровень иерархии можно представить следующим образом:

$$x_1^1 \sim (x_2^{a_1} \vee x_2^{a_2} \vee \dots \vee x_2^{a_i}) = 1, \text{ где } (a_1, a_2, \dots, a_i) \in A,$$

$$x_2^1 \sim (x_3^{b_1} \vee x_3^{b_2} \vee \dots \vee x_3^{b_m}) = 1, \text{ где } (b_1, b_2, \dots, b_m) \in B, m \geq k,$$

.....

$x_{n-1}^1 \sim (x_n^{z_1} \vee x_n^{z_2} \vee \dots \vee x_n^{z_l}) = 1, \text{ где } (z_1, z_2, \dots, z_l) \in Z, l \geq j,$   
 где  $A, B, C, T$  – соответственно подмножество словоформ словаря  $D$ , представленных на каждом уровне иерархии. Согласно теореме о разложении [8] уравнение (1) можно записать в следующем виде:

$$(x_1^1 \sim \vee x_2^{a_i})(x_2^1 \sim \vee x_3^{b_m}) \dots (x_{n-1}^1 \sim \vee x_n^{z_l}) = 1.$$

$a=1 \qquad b=1 \qquad l=1$

Слова, связанные родо-видовыми отношениями, представляют собой лексико-семантические классы с иерархической структурой, описываемой в виде родо-видовых деревьев [7]. Родо-видовое дерево представляет собой иерархию элементов, называемых узлами. В роли узлов выступают словоформы. Пример родо-видового дерева представлен на рис. 3.

В зависимости от контекста родо-видовое дерево с этими же словоформами будет иметь другой вид. Для приведенного дерева (рис. 3) введем переменные и области их допустимых значений. Пусть  $x_1 \in \{1\}, x_2 \in \{2, 3, 4\}, x_3 \in \{5, 6\}, x_4 \in \{7, 8, 9, 10\}$ .



Рис. 3. Родо-видовое дерево лексико-семантического класса существительных

Уравнение, описывающее дерево, имеет следующий вид:

$$(x_1^1 \sim x_2^2 \vee x_2^3 \vee x_2^4)(x_2^2 \sim x_3^5 \vee x_3^6)$$

$$(x_3^5 \sim x_4^7 \vee x_4^8)(x_3^6 \sim x_4^9 \vee x_4^{10}) = 1.$$

Решение данного уравнения соответствует узлам родо-видового дерева.

### Выводы

В работе показано использование алгебры конечных предикатов для моделирования семантических отношений в онтологии, что позволит улучшить качество автоматического создания онтологий или автоматического пополнения онтологий и, соответственно, повысить эффективность процессов обработки данных и знаний в компьютерных системах и сетях. В работе представлено только один вид связи “род-вид”. В дальнейшем планируется описать и другие типы связей. Моделирование семантических отношений может использоваться в задаче автоматического создания онтологии и других задачах автоматической обработки языка.

**Список литературы:** 1. *Simperl E., Mochol M.* Achieving Maturity: the State of Practice in Ontology Engineering in 2009. / In International Journal of Computer Science and Applications, Technomathematics Research Foundation Vol. 7 No. 1, pp. 45-65, 2010. 2. *Faatz A. and Steinmetz R.* Ontology enrichment with texts from the WWW. Semantic Web Mining 2nd Workshop at ECML/PKDD-2002, 20th August 2002, Helsinki, Finland 3. *Towards the Semantic Web: Ontology-driven Knowledge Management.* – John Wiley & Sons, 2003. – ISBN 0470848677. 4. *Gruber T.* What is an Ontology? <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>. 5. *Rogers J.E.* GALEN's Model of Parts and Wholes: Experience and Comparisons / Rogers J.E., Rector A.L. // Journal of the American Medical Informatics Association, Fall Symposium Special Issue, pp. 819-823, November, 2000. 6. *Нариньяни А. С.* ТЕОН-2: От тезауруса к онтологии и обратно // [http://www.dialog-21.ru/archive\\_article.asp?param=7360&y=2002&vol=6077](http://www.dialog-21.ru/archive_article.asp?param=7360&y=2002&vol=6077). 7. *Замаруева И.В.* Математические модели семантики свободных словосочетаний с родо-видовыми компонентами и их применение в АИС. – Дис... кан. техн. наук. – Харьков : ХТУРЭ, 1990. – 170 с. 8. *Шаронова Н.В.* Математические модели суффиксального словообразования и их использование в системах автоматической обработки текстов на русском языке. – Дис... кан. техн. наук. – Харьков : ХТУРЭ, 1984. – 217 с.

Поступила до редколлегии 05.11.2012

УДК 004.934

**Моделивання семантичних відношень в онтології /** Н.В. Бабкова, О.В. Канищева, З.А. Кочуева // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 25-28.

У статті розглянуто підхід до моделювання семантичних відношень в онтології, проаналізовано теоретичні аспекти використання онтологій у семантичній павутині. У роботі показано типи зв'язків, які властиві онтологіям, дано опис онтологічних відношень засобами алгебри кінцевих предикатів, що дозволить поліпшити якість автоматичного створення онтологій або автоматичного поповнення онтологій. Наведено приклад родо-видового дерева для лексико-семантичного класу іменників.

Іл. 3. Бібліогр.: 8 найм.

UDK 004.934

**Modeling semantic ontology relationships /** N. V. Babkova, O.V. Kanishcheva, Z.A. Kochueva // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 25-28.

The paper discussed an approach to modeling semantic ontology relationships, analyzed the theoretical aspects of the use of ontologies in the semantic web. The paper shows the types of relationships that are inherent in the ontology, a description of the means of ontological predicates of finite algebra, which will allow to improve the quality of automatic creation of ontologies and automatic replenishment of ontologies. The example of genus-species tree for the lexical-semantic class of nouns.

Fig. 3. Ref.: 8 items.

УДК 004.912

А. Б. Бегунов<sup>1</sup>, Т. М. Заболотня<sup>2</sup><sup>1</sup>НТУУ «КПІ», м. Київ, Україна, arxton@mail.ru<sup>2</sup>НТУУ «КПІ», м. Київ, Україна, tatiana104@yandex.ru

## АГЕНТНО-ОРІЄНТОВАНИЙ МЕТОД АВТОМАТИЗОВАНОГО ВИЛУЧЕННЯ ФАКТІВ З ПРИРОДНОМОВНИХ ТЕКСТОВИХ ДАНИХ

У статті наведено результати аналізу існуючих методів вилучення фактів, що засновані на цих методах. Визначено недоліки існуючих методів. Для їх усунення запропоновано агентно-орієнтований метод автоматизованого фактографічного аналізу природномовних текстових даних. Метод полягає у вилученні кожного типу фактів різними типами агентів.

МОВА ПРИРОДНА, АНАЛІЗ ФАКТОГРАФІЧНИЙ, ФАКТ, АГЕНТ, АГЕНТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

### Вступ

Важливим напрямком досліджень у галузі автоматизованого аналізу природномовних текстових даних є вилучення фактів із заданого тексту (так званий фактографічний аналіз). Програмні засоби фактографічного аналізу дозволяють приводити невпорядковану інформацію до вигляду, придатного для подальшого перетворення або використання стандартними методами обробки текстових даних.

На даний момент для отримання фактів використовуються такі базові методи їх автоматизованого вилучення: методи вилучення на основі ознак; методи вилучення, які використовують функції-ядра; методи, засновані на співставленні зразків; методи, засновані на фразових зразках [1].

Слід зазначити, що переважна більшість цих методів характеризується жорсткою прив'язкою контексту вилучення до певної предметної галузі, що призводить до необхідності реалізації окремих систем вилучення фактів для різних предметних галузей. Звичайно, така розробка вимагає значних витрат часу, а також матеріальних ресурсів.

Аналіз же існуючих систем, в яких передбачено можливість налаштування на різні предметні галузі, показав, що їх істотним недоліком є складність налаштування за умови потреби у додаванні нових структур даних, що описуватимуть нові предметні галузі. Це, в свою чергу, призводить до іншої проблеми — неможливості динамічного навчання системи на наборах даних із вже заданих предметних галузей з метою підвищення точності та повноти вилучення фактів [1].

Усунення визначених вище недоліків потребує модифікації існуючих методів фактографічного аналізу. Таким чином, пошук шляхів щодо забезпечення абстрагування механізму вилучення фактів від предметної галузі оброблюваного тексту, а також щодо підтримки можливості динамічного навчання систем фактографічного аналізу видається актуальним питанням.

### 1. Постановка задачі

Перш, ніж сформулювати задачі дослідження, дамо визначення основних понять, якими будемо оперувати у статті.

*Природномовними текстовими даними (текстом)* будемо називати сукупність речень, складених будь-якою природною мовою.

Під *фактом* будемо розуміти атомарну одиницю даних в тексті, що несе інформацію про певну сутність або об'єкт [1]. Факти розділяють на види за їх змістом (події, опис, твердження, дії тощо).

*Фрейм* — структура, що складається з поіменованих елементів (слотів). Фрейми і слоти наділяються певною семантикою в залежності від предметної області, в якій вони використовуються. Фрейми слугують для структурованого подання фактів.

Під *фактографічним аналізом* будемо розуміти процес обробки природномовних текстових даних із застосуванням певних способів та методів вилучення фактів.

Під *текстовим доменом* будемо розуміти частину текстової інформації, що містить в собі інформацію про деякий факт.

*Агент* — сутність, що функціонує в певному середовищі, сприймає своє середовище за допомогою сенсорних механізмів і впливає на це середовище за допомогою виконавчих механізмів для досягнення певної мети.

**Метою** даного дослідження є покращення характеристик масштабованості систем вилучення фактів, їх здатності до динамічного навчання під час функціонування, а також універсалізація таких систем шляхом абстрагування від контексту певної предметної галузі.

З огляду на описану вище проблематику задачею дослідження стали:

1. Вивчення методів вилучення фактів з природномовних текстових даних на предмет виявлення можливостей щодо підвищення ефективності цього процесу за критеріями повноти та точності отримуваних результатів.

2. Розробка методу фактографічного аналізу текстових даних, реалізація якого програмним забезпеченням дозволить задовольнити такі вимоги як:

– абстрагування методу вилучення фактів від конкретної предметної галузі для підтримки налаштування вже існуючої системи на вилучення фактів будь-якого змісту;

– масштабованість системи, що забезпечить підтримку додавання нових даних для вилучення фактів відносно нових предметних галузей;

– можливість динамічного навчання системи на основі результатів, отриманих на попередніх циклах її роботи.

## 2. Розв'язок задачі

Нижче наведено ряд кроків, які пропонуються авторами до виконання для досягнення мети дослідження. Для кожного кроку визначимо необхідні структури даних та дії, спрямовані на вирішення поставлених задач.

### 2.1. Вибір способу подання фактів

Перш за все, необхідним кроком є визначення структури, що буде використовуватися для опису фактів.

На сьогоднішній день найбільш поширеною і зручною для передачі на подальшу обробку формою подання фактів в задачах вилучення знань є фрейм — структура з поименованими елементами — слотами. Фрейми і слоти наділяються семантикою, що залежить від предметної галузі, в якій вони використовуються.

Декларативну частину знань про предметну галузь поділяють на дві групи: аксіоматичну і фактичну. У нашому випадку аксіоматика задає схему фреймового подання даних, тобто визначає склад кожного фрейма і область значень слотів. Вважатимемо, що вона задається людиною-експертом. Факти з точки зору фреймової моделі — це конкретні екземпляри фреймів із зазначеними слотами. Екземпляри фреймового подання можна подати у вигляді моделі [2]:

$F$  — множина екземплярів фреймів;

$S$  — множина слотів екземплярів фреймів;

$T$  — множина значень слотів;

$R_{FST} \in F \times S \times T$  — відношення, що зв'язує з екземпляром фрейма певний слот і його значення.

Таким чином, завдання вилучення фактів зводиться до виявлення в аналізованому тексті деякого екземпляра фрейма  $f_i$  і заповнення значень його слотів  $\{s_j\}$  фрагментами з тексту. Результатом вилучення є трійки  $(f_i, s_j, t_k) \in R_{FST}$ , де  $t_k$  — значення слота виявленого фрейма [3]. Тоді для певного фіксованого значення  $f_i$  фрейм можна подати як множину пар  $\{(s_j, t_k)_1, \dots, (s_j, t_k)_m\}$ .

Крім концептуального рішення щодо використання фреймів для зберігання даних про вилучені факти, необхідно також визначити, які саме типи фреймів потрібні для подання довільного факту, а також коли і якими даними будуть заповнюватися їхні слоти.

Для цього спочатку проаналізуємо характер інформації, що міститься у будь-якому тексті.

В загальному випадку текст описує певні об'єкти, кожен з яких має набір характеристик. Текст також описує відношення між об'єктами та властивості цих відношень. На основі цього введемо два типи фреймів:

– фрейми першого типу — описують об'єкт; мають групу слотів для іменування заданого об'єкту та групу слотів, що містять опис об'єкта, тобто набір його властивостей. Об'єкт може не мати властивостей, тобто слоти, відповідальні за його опис, можуть бути пустими;

– фрейми другого типу — описують зв'язок одного об'єкта з іншим. Вважатимемо, що в такій парі зв'язок є однонаправленим і він завжди направлений від першого об'єкта до другого. Такі фрейми мають три слоти. Перший та другий слоти описують, відповідно, зв'язані між собою об'єкти. Третій слот описує відношення, що пов'язує ці об'єкти. Ці три слоти є фреймами першого типу. Отже, відношення також виступає в ролі об'єкта, який має назву і може мати властивості.

Проаналізуємо, якими даними можна заповнювати слоти фреймів визначених вище типів в залежності від виду виконуваної процедури вилучення інформації з природномовних текстових даних. Зазначимо, що існує 5 видів таких процедур, які відрізняються глибиною аналізу тексту і типом інформації (видами фактів), що вилучаються [4].

1. Виділення іменованих сутностей — пошук та класифікація імен об'єктів в тексті. При цьому деякі об'єкти можуть бути подані в тексті групами слів. Ця процедура передбачає заповнення слотів, що відповідають за іменування об'єкта у фреймах першого типу.

2. Побудова елементів шаблону — зводиться до додавання описової інформації (значень деяких атрибутів) до об'єкта, знайденого на попередньому етапі аналізу. Ця процедура передбачає заповнення слотів, що відповідають за опис об'єкта у фреймах першого типу.

3. Розпізнавання зв'язків кореференції — визначення того, які імена об'єктів, знайдені в тексті, вказують на один і той самий об'єкт. Сюди входить як вилучення власне кореференції, тобто позначення одного і того ж об'єкта двома значущими лексичними структурами, так і аналіз анафоричних зв'язків (наприклад, коли при частому повторенні значуще слово замінюється займенником). Отримана інформація додається до об'єктів, знайдених на двох попередніх етапах аналізу. Ця процедура передбачає об'єднання фреймів, сформованих на попередніх етапах, що описують один і той самий об'єкт. Внаслідок цього утворюються фрейми першого типу, що більш повно описують об'єкт (наскільки це можливо в контексті певного тексту, аналіз якого проводиться).

4. Побудова зв'язків — виявлення зв'язків між

окремими об'єктами. Ця процедура передбачає встановлення зв'язків між об'єктами, для яких було сформовано фрейми на попередніх етапах. На основі цієї інформації будуються фрейми першого типу, що описують відношення.

5. Побудова сценарію – побудова повного опису події (факту) шляхом об'єднання результатів попередніх етапів. Ця процедура передбачає використання інформації, отриманої з попередніх етапів для побудови фреймів другого типу, що будуть виражати зв'язані між собою поняття з тексту.

Розглянемо, на яких етапах автоматизованого аналізу текстових даних можна отримати інформацію, необхідну для вилучення кожного з вищезгаданих видів фактів [5], а також оберемо структуру для подання природномовних текстових даних у вигляді, зручному для проведення фактографічного аналізу.

*Графематичний аналіз.* На цьому етапі з тексту виділяються структурні або синтаксичні одиниці. Характер і тип таких елементів залежить від поставленої задачі. Частіше за все таким елементом виступає слово. Цей етап слугує лише для підготовки даних для наступних етапів.

*Морфологічний аналіз.* Визначає морфологічну інформацію про словоформи, що були виділені на попередньому етапі. Дані, отримані на цьому етапі, використовуються для виділення іменованих сутностей.

*Попередній синтаксичний аналіз.* Відповідає за об'єднання окремих лексичних одиниць в одну синтаксичну. Так, наприклад, в єдину синтаксичну одиницю можуть об'єднуватися змінювані нерозривні словосполучення (наприклад, «бити байдики»). Ще одним завданням етапу попереднього синтаксического аналізу є проведення синтаксичної сегментації. Її метою є розмітка лінійного тексту на фрагменти, згідно з морфологічними правилами. На цьому етапі відбувається лише підготовка до наступного етапу і жодні процедури вилучення фактів не проводяться.

*Синтаксичний аналіз.* Результатом проведення даного етапу є набір дерев, вершинами яких є словоформи, а дугами – синтаксичні відношення. Для кожного речення будується своє, окреме дерево синтаксичного розбору. Виконання завдання ускладнюється величезною кількістю альтернативних варіантів, що виникають в ході розбору, пов'язаних як з багатозначністю вхідних даних (одна й та ж словоформа може бути отримана від різних нормальних форм), так і неоднозначністю самих правил розбору. Інформація, отримана на цьому етапі, використовується для побудови елементів шаблону. Також на цьому етапі процедури вилучення розпізнавання зв'язків кореференції та побудови сценарію можуть бути виконані частково, оскільки зв'язки між лексемами встановлюються в рамках кожного окремого речення.

*Семантичний аналіз.* На цьому етапі проводиться

аналіз змісту тексту. З одного боку, семантичний аналіз відтворює семантичні зв'язки між лексемами, за допомогою яких відображається значення слів. З іншого боку, семантичний аналіз дозволяє відфільтрувати деякі значення слів або навіть цілі варіанти розбору як «семантично незв'язні». Семантична інформація застосовується для розпізнавання зв'язків кореференції та побудови сценарію, оскільки для цього потрібні дані не тільки про синтаксичні зв'язки між лексемами, але і їх семантичний зв'язок.

В залежності від обраного методу вилучення фактів в системі можуть бути реалізовані лише деякі етапи обробки тексту. Те, які етапи потрібно реалізувати, залежить від особливостей застосування конкретного методу, необхідної точності аналізу, використовуваної структури фреймів та зв'язків між ними.

Вибір структури для подання природномовних текстових даних у вигляді, зручному для проведення фактографічного аналізу, зроблений авторами на основі міркувань, наведених нижче.

Природномовний текст містить багато посилань з одного речення на інше. Такі зв'язки носять синтактико-семантичний характер. З одного боку, між реченнями можливий лише семантичний зв'язок. З іншого боку, такий зв'язок може бути встановлений через аналіз морфологічних та синтаксичних характеристик словоформ, що входять до складу речень. Встановивши такі синтактико-семантичні зв'язки між лексемами з різних речень, отримуємо структуру, що складається з синтаксичних дерев, які пов'язані між собою синтактико-семантичними зв'язками. Ця структура є орієнтованим графом, вершинами якого є лексеми або групи лексем, а дуги відображають синтактико-семантичні зв'язки у тексті. Таке подання текстових даних є зручним для застосування різних алгоритмічних підходів до вилучення фактів. В разі використання статистичних алгоритмів кожна вершина та дуга можуть мати певну вагу, що буде враховуватись при заповненні фреймів. В разі застосування методів, заснованих на правилах (шаблонах), заповнення фреймів зводиться до пошуку підграфів певної топологічної та семантичної структури.

Для отримання коректної інформації при заповненні фреймів типи слотів фреймів ставляться у відповідність синтактико-семантичним ролям підграфів (в окремому випадку – вершин-словоформ). Задача заповнення фреймів у такому випадку зводиться до пошуку певних задалегідь заданих закономірностей на графі.

Вершини отриманого графа (лексичні вузли) можуть об'єднувати в собі декілька лексем. У випадку, коли певна група лексем виражає одне поняття, всі лексеми з цієї групи входять в один лексичний вузол. Такі лексичні вузли утворюються, наприклад, для власних назв, що описуються декількома словами. Певний лексичний вузол має наступну структуру:

лексема\_1; ознака\_11; ...; ознака\_1n;  
лексема\_2; ознака\_21; ...; ознака\_2n;  
.....;  
лексема\_m; ознака\_m1; ...; ознака\_mn.

## 2.2. Агентно-орієнтований метод фактографічного аналізу тексту

Для визначення основи нового методу фактографічного аналізу, здатного задовольнити всім вимогам, сформульованим у задачі дослідження, розглянемо існуючі у цій галузі методи та проаналізуємо їх переваги та недоліки.

### 1. Методи вилучення на основі ознак.

Методи вилучення на основі ознак мають в основі наявність фіксованого набору ознак (словник ознак) і ваг використання цих ознак в межах вилучених елементів тексту. Для кожного вилученого елемента визначається його вектор характеристик (ознак). Найбільш поширеними в даному класі є імовірнісні класифікатори Байеса і приховані Марківські моделі [6]. Вилучення фактів у межах цих методів зводиться до розпізнавання деякого сегмента тексту на основі ймовірнісного аналізу ознак, виявлених в околі цього сегмента. Недоліками такого підходу є використання обмеженого розміру околу (як правило, не більше 2-3 слів), необхідне для забезпечення заданої точності вилучення. Використання більшого розміру контексту призводить до зниження повноти розпізнавання і до збільшення розміру необхідної репрезентативної вибірки, на якій збирається статистика для розрахунку оцінок ймовірностей.

### 2. Методи вилучення, які використовують ядра.

Дані методи не мають частини недоліків попереднього класу методів за рахунок алгоритмічного визначення міри подібності між поданнями текстових сегментів, що зіставляються. Суть методів - замінити скалярний добуток векторів, що відображають ознаки розпізнаваних елементів, деякою функцією - ядром. Така функція задається алгоритмічно і враховує більш складне подання розпізнаваних елементів та їх контекстів, як правило деревоподібне, таке що описує синтаксичну структуру текстового сегмента. Для деревоподібних подань розрахунок ядра найчастіше зводиться до співставлення всіх вкладених дерев у вихідні дерева. Недоліком такого підходу є висока обчислювальна складність розрахунку ядер і визначення синтаксичної структури текстового сегмента [6].

### 3. Методи, засновані на співставленні зразків.

Дані методи оперують поняттям «зразок» та правилами співставлення зразків з фрагментами текстів. Зразки є ланцюжками обмежень (символи, слова, частини мови та семантичні класи) і свого роду шаблонами фраз. В цьому відношенні дані методи є аналогічними методам з використанням функцій-ядер за умови, що текстові сегменти мають «пласке» подання, а не деревоподібне. Особливість методів полягає у використуваному

способі навчання, в основі якого лежать прийоми індуктивного логічного програмування.

### 4. Методи, засновані на фразових зразках.

Методи, засновані на фразових зразках, є свого роду компромісом між методами вилучення, які використовують ядра, і методами, заснованими на співставленні зразків. Текстові сегменти так само, як і в підході з функціями-ядрами, подаються деревами синтаксичних зв'язків, але замість складного розрахунку ядер виконується більш проста з точки зору обчислювальної складності процедура співставлення з синтаксичним шаблоном, притаманна методам, заснованим на співставленні зразків, але доповнена морфологічними ознаками, що використовуються при співставленні синтаксичних зв'язків. Співставлення відбувається за певними, заздалегідь визначеними правилами для кожного зразка. Такі правила описують типи відношень між вузлами дерева, а також морфологічні та синтаксичні ознаки відношень та самих вузлів.

Виходячи з поставлених задач дослідження, за основу нового створюваного методу фактографічного аналізу доцільно обрати методи, засновані на фразових зразках. Такі методи використовуються для обробки текстових даних, поданих у вигляді графа синтактико-семантичних зв'язків, що відповідає обраній нами структурі внутрішнього подання даних.

У новому методі пропонується розширити поняття зразка (шаблону), з яким зазвичай працюють методи даного класу, шляхом введення поняття синтактико-семантичного шаблону. При цьому спосіб співставлення з шаблоном залишається незмінним, але до синтаксичних зв'язків та морфологічних властивостей додаються семантичні, що описують зв'язки між синтаксичними деревами і також можуть існувати в рамках окремого дерева синтаксичних зв'язків. Додавання нових фразових зразків для забезпечення вилучення нових типів фактів зводиться до додавання нових наборів правил для кожного зразка. Це дозволяє покращити масштабованість системи, побудованої на такому методі. В нашому випадку, за умови подання тексту у вигляді орієнтованого графа синтактико-семантичних відношень, фразовий зразок виглядає як шаблон підграфа, властивості якого відповідають певним вимогам. Кожна вершина цього підграфа описує ті ознаки, які може або повинен мати певний лексичний вузол. Кожна дуга цього підграфа описує можливі або необхідні відношення між відповідними лексичними вузлами.

Оскільки метод, заснований на фразових зразках, спирається на певні морфологічні, синтаксичні та семантичні властивості, він не пов'язаний з контекстом певної предметної галузі. Таким чином, застосування цього методу дозволяє побудувати гнучку систему, яка може бути налаштована на вилучення будь-яких видів фактів без модифікації структурної чи алгоритмічної складової самої системи.

На жаль, обрання за основу методу, засновано на фразових зразках, є недостатнім кроком для забезпечення ефективного вирішення поставлених задач. Масштабованість системи по відношенню до великих обсягів тексту все одно залишається обмеженою. За допомогою розробленого методу фактографічний аналіз проводиться лінійно по структурам даних, що подають текст, і, як наслідок, потребує значного часу для пошуку відповідних фразових зразків. Проте, будь-який великий за обсягом природномовний текст може бути розділений на частини, що непов'язані або слабко пов'язані між собою синтаксично або семантично. Наявність слабких зв'язків або відсутність зв'язків взагалі дає змогу аналізувати такі частини тексту паралельно. Оскільки у вилученні різних видів фактів приймають участь різні типи зв'язків, слабко зв'язані або незв'язані частини тексту слід виділяти в межах кожного виду фактів окремо.

Відповідно до задач даного дослідження необхідно забезпечити можливість динамічного навчання системи для покращення показників ефективності її функціонування. Отже, потрібен певний механізм, за допомогою якого можна інкрементно навчати систему, використовуючи попередні результати вилучення фактів та оцінку їх якості. Такий механізм закладений в агентно-орієнтованому підході до програмування [7]. Цей підхід дозволяє, по-перше, організувати паралельну обробку агентами окремих слабозв'язних або незв'язних частин тексту і, по-друге, навчати систему вилучення фактів при її функціонуванні через взаємодію з середовищем та користувачем. Таким чином, для розв'язання задачі дослідження застосування агентно-орієнтованого підходу є доцільним.

### 2.3. Опис агентів

За середовище функціонування агента пропонується вважати текст, поданий у вигляді графа синтактико-семантичних зв'язків. Тоді підграфи цього графа, що слабо зв'язані або не зв'язані між собою, відповідають окремим частинам тексту, які можуть бути оброблені паралельно окремими агентами.

Зазначимо, що для вилучення кожного виду фактів необхідне виконання різних дій з боку агентів. Зазвичай агенти здійснюють свої дії за допомогою ефекторів. У нашому випадку ефекторами агента будуть виступати процедури обходу підграфа, в межах якого функціонує агент. При цьому для вилучення кожного виду фактів пропонується використовувати окремий тип агентів.

У межах даного дослідження метою агентів є заповнення фреймів для фактів відповідного виду. Кожен такий спеціалізований агент містить когнітивні структури даних, що відповідають за його поведінку і водночас слугують тими даними, на які спирається агент для виконання дій, спрямованих на заповнення фреймів. Такі структури даних є переконаннями агента, вони можуть змінюватись в

процесі його діяльності. Це дає змогу проводити навчання агента під час його функціонування.

Для кожного виду фактів визначимо когнітивні структури агента, що буде проводити вилучення фактів цього виду, його мету і дії щодо вилучення фактів.

#### 1. Виділення іменованих сутностей.

Когнітивними структурами даних для агентів, що виконують виділення іменованих сутностей, є морфологічні характеристики лексем, які задаються у фразовому зразку. Агент шукає всі збіги морфологічних характеристик лексичних вузлів із заданими в когнітивних структурах даних морфологічними характеристиками і заносить у відповідні слоти фреймів інформацію про лексичні вузли, що описують об'єкти. Процедура обходу підграфа у даному випадку полягає у лінійному перегляді всіх лексичних вузлів і їх співставленні з морфологічними ознаками, згідно з якими ведеться пошук іменованих сутностей. Агент певним чином позначає знайдені вузли як такі, що відповідають іменованим сутностям, для подальшого використання цієї інформації іншими агентами на наступних етапах методу.

#### 2. Побудова елементів шаблону.

При вилученні фактів даного виду когнітивними структурами даних є синтаксичні та морфологічні ознаки лексем, які можуть бути пов'язані з іменованими сутностями. Процедура вилучення фактів, що є елементами шаблону, зводиться до проходження агентом вузлів, що прямо чи опосередковано зв'язані із знайденими на попередньому етапі вузлами і відміченими як іменовані сутності. Кількість вузлів, зв'язаних з іменованими сутностями, які необхідно співставити із зразком, заданим у когнітивних структурах даних, залежить від налаштувань агента і повинна задаватися користувачем як ступінь деталізації пошуку.

Метою агента у даному випадку є заповнення слотів фреймів першого типу, що відповідають за опис об'єкта. Агент помічає вже знайдені вузли для подальшого використання цієї інформації на наступних етапах вилучення фактів.

#### 3. Розпізнавання зв'язків кореференції.

Когнітивні структури даних агента на цьому етапі вилучення фактів мають містити інформацію про необхідні синонімічні відношення та відношення кореференції між іменованими сутностями. Вилучення фактів зводиться до проходження агентом всіх вузлів, що представляють іменовані сутності, і співставлення їх один з одним для визначення того, чи є вони кореферентно пов'язаними. В результаті всі пов'язані таким чином вузли помічаються як одна й та сама іменована сутність. Відповідно, ознаки таких сутностей також об'єднуються.

Метою агента є утворення нових фреймів першого типу шляхом об'єднання фреймів, утворених на попередніх етапах.

#### 4. Побудова зв'язків та побудова сценарію.

Когнітивні структури даних агента на цьому етапі повинні містити інформацію про фразові зразки. Ці зразки є певного виду синтактико-семантичними зв'язками між лексичними вузлами, а також визначеною морфологічною та синтаксичною інформацією про ці лексичні вузли. За даними зразками повинно здійснюватися вилучення фактів. На цьому етапі воно зводиться до проходження агентом всіх груп вузлів, визначених на попередньому етапі як іменовані сутності, з метою пошуку збігів між такими групами та іншими вузлами із фразовими зразками. При вдалому співставленні із фразовими зразками агент позначає певними чином всі лексичні вузли, що приймають участь у формуванні відношення між іменованими сутностями. Після цього агент намагається визначити опис вузлів, що формують відношення, переглядаючи вузли, зв'язані з ними прямо чи опосередковано. Кількість таких вузлів залежить від налаштувань агента і повинна задаватися користувачем як ступінь деталізації пошуку.

Метою агента на даному етапі є формування фреймів першого типу, що описують відношення між іменованими сутностями, та фреймів другого типу, що зв'язують між собою іменовані сутності.

Для забезпечення інкрементного навчання кожен тип агентів на кожному етапі взаємодіє з користувачем з метою отримання оцінки якості виконаного етапу фактографічного аналізу. Агент надає користувачу вилучені фрейми та відповідні фразові зразки, за допомогою яких було здійснено вилучення. Таким чином, користувач корегує правильність виконання вилучення фактів.

#### Висновки

В результаті проведеного дослідження для опису фактів у системі фактографічного аналізу природномовних текстових даних обрано фреймову модель подання інформації. Виділено два типи фреймів та визначено, якими даними можна заповнювати слоти цих фреймів в залежності від глибини аналізу текстових даних, здійснюваного у межах процедури вилучення фактів. Обґрунтовано вибір структури даних для подання природномовного тексту у вигляді, зручному для проведення фактографічного аналізу. Авторами доведена доцільність використання методів, заснованих на фразових зразках, як основи для нового методу фактографічного аналізу, що пропонується у статті.

Новий метод розроблено на основі агентно-орієнтованого підходу до програмування. Він дозволяє абстрагуватися від конкретної предметної галузі та налаштовувати існуючу систему вилучення фактів на роботу з будь-якими видами даних, а також надає можливість динамічного навчання системи в процесі роботи, спираючись на результати, отримані на попередніх циклах її роботи.

Подальшого дослідження, на думку авторів, потребують можливі модифікації базового способу

функціонування агентів. Також перспективним напрямком продовження роботи над тематикою статті є підвищення швидкодії запропонованого методу.

**Список літератури:** 1. Ландэ, Д.В. Интернетика. Навигация в сложных сетях: модели и алгоритмы [Текст] / Д.В. Ландэ, А.А. Снарский, И.В. Безсуднов. — М.: Либроком, 2009. — 264 с. 2. Bocharov V. Ontological parsing of encyclopedia information [Text] / Bocharov V., Pivovarova L., Rubashkin V., Chuprin B. Lecture Notes in Computer Science. Т. 6008 LNCS., 2010. — 650 p. 3. Пивоварова, Л. М. Фактографический анализ текста в системе поддержки принятия решений [Текст] / Л. М. Пивоварова. Вестник Санкт-Петербургского университета. Сер. 9, Филология, востоковедение, журналистика. — 2010. — Вып. 4, декабрь. — С. 190-197. — Библиогр.: с. 196-197 (21 назв.). 4. Гаспаров, Б. М. Язык, память, образ. Лингвистика языкового существования [Текст] / Б. М. Гаспаров. — М.: Новое Литературное Обозрение, 1996. — 352 с. 5. Леонтьева, Н. Н. Автоматическое понимание текста: системы, модели, ресурсы [Текст] / Н. Н. Леонтьева. — М.: Издательский центр «Академия», 2006. — 304 с. 6. Симаков К.В. Модель извлечения знаний из естественно-языковых текстов [Текст] / А.М. Андреев, Д.В. Березкин, К.В. Симаков // Информационные технологии., 2007. — С. 57—63. 7. Tomas Salamon. Design of Agent-Based Models. [Text] / Tomas Salamon. Czech Republic, Bruckner Publishing, 2011. — 220 p.

Поступила до редколегії 15.11.2012

УДК 004.912

**Агентно-ориентированный метод автоматизированного извлечения фактов из естественно-языковых текстовых данных** / А.Б. Бегунов, Т.Н. Заболотня // Бионика интеллекта: науч.-техн. журнал. — 2013. — № 1 (80). — С. 29-34.

В данной работе предложен метод извлечения фактографической информации из естественно-языковых текстовых данных, основанный на сопоставлении фразовых образцов, а также агентно-ориентированном подходе к созданию программного обеспечения. Разработанный метод дает возможность абстрагироваться от предметной области текстов, для которых проводится извлечение фактов, и реализовать на его основе программную систему, обладающую возможностью добавления новых видов фактов для извлечения, а так же способностью к динамическому обучению. Описаны этапы работы метода. Представлены особенности строения и функционирования агентов, при помощи которых предлагается производить извлечение фактов.

Библиогр.: 7 items.

UDK 004.912

**The agent-oriented method of automatic fact extraction from natural language text** / A. Begunov, T. Zabolotnia // Bionics of Intelligense: Sci. Mag. — 2013. — № 1 (80). — P. 29-34.

In this work the method of automated fact extraction from natural language texts based on phrase pattern matching and agent-oriented approach to software development is proposed. The developed method allows to abstract from the subject area of the processed texts and enables to implement the software system based on it supporting the ability to add a new types of facts to extract as well as the ability of dynamically learning. The steps of the method are described. The features of the structure and functioning of the proposed agents for performing the extraction of facts are presented.

Ref.: 7 items.

О.О. Скопа<sup>1</sup>, Є.В. Вавілов<sup>2</sup><sup>1</sup>ОНЕУ, м. Одеса, Україна, skora2003@ukr.net<sup>2</sup>ОНУ ім. І. Мечнікова, Одеса, Україна, intelevgen@gmail.com

## ІНТЕЛЕКТУАЛЬНІ АВТОНОМНІ СИСТЕМИ: КОНЦЕПТУАЛЬНІ ПОЛОЖЕННЯ СТВОРЕННЯ ТА ФУНКЦІОНУВАННЯ

На основі аналізу перспектив наукових досліджень та розробок в області штучного інтелекту робиться висновок про необхідність формулювання проблемного завдання, яке могло б послужити базою для коректного обговорення та зіставлення результатів як концептуального, так і прикладного змісту для штучних інтелектуальних систем. Пропонується використовувати в якості такого завдання проблему вивчення та створення інтелектуальних автономних систем. Розглядаються основні особливості таких систем та наводяться характерні для них приклади.

ІНТЕЛЕКТУАЛЬНА АВТОНОМНА СИСТЕМА, ШТУЧНА ІНТЕЛЕКТУАЛЬНА СИСТЕМА, ЦІЛЕОРІЄНТОВНА ЗАДАЧА

### Вступ

Вже більше, ніж півстоліття, якщо вважати з моменту виходу роботи [1], не припиняються спроби створення штучних інтелектуальних систем (ШІС).

Результати цих спроб виглядають поки достатньо скромними, якщо порівнювати їх з основною декларуючою метою – *створити ШІС, яка за рівнем інтелектуальних можливостей не буде поступатися людині*. При цьому дотепер немає ясності навіть у такому кардинальному питанні, як принципова реалізація ШІС подібного рівня. Саме це положення відображає **постановку проблеми у загальному вигляді та її зв'язок із науковими та практичними завданнями**.

Аналіз останніх досліджень та публікацій показав, що діапазон точок зору з цього питання варіюється від категоричного «ні» [2...4] до рішучого «так» [5]. Як правило, аргументування як «за», так і «проти» виглядають не дуже переконливими. Основна причина – ці аргументування найчастіше уможливлені: вони зазвичай засновані на свого роду «уявному експерименті» або є не цілком коректною екстраполяцією якихось часткових результатів на більш широку область, де, взагалі кажучи, з цими результатами треба було б поводитися з великою обережністю. Один з найбільш знаменитих прикладів такої екстраполяції – книга М. Мінського і С. Пейперта «Перцептрони» [6], яка на півтора десятиліття істотно ускладнила життя представникам нейромережного напряму досліджень в області ШІС. Таким чином, **невиділеною раніше проблемою** в області ШІС є розробка та удосконалення окремих концептуальних положень їх створення та функціонування, причому у якості базових систем можуть бути вибрані інтелектуальні автономні системи. Зважаючи на сказане, **метою та завданням** дослідження є формулювання проблемної області, яка могло б послужити базою для коректного обговорення та зіставлення результатів як концептуального, так і прикладного змісту щодо створення ШІС.

### 1. Цілеорієнтовна задача: передумови та вимоги

За минулі десятиліття запропоновано та розроблено значну кількість всіляких підходів, методів, алгоритмів, які відносяться до різних аспектів проблеми створення та функціонування ШІС. Проте аналізувати та зіставляти між собою ці підходи надзвичайно важко. Так же важко отримати оцінку внеску того або іншого результату в реалізацію «основної мети» ШІС. Це пояснюється тим, що коректне зіставлення результатів припускає наявність загальної понятійної бази, тобто основи для такого порівняння. Аналіз літературних джерел показав, що такого роду основи поки що немає.

Отже, можна стверджувати, що в даний час у області ШІС відсутня база для ефективного та коректного обговорення та зіставлення результатів, принципово важливих для ШІС як питань концептуального характеру, відповіді на які визначають зовнішність створюваних систем, так і питань конкретного характеру, що відносяться до різних аспектів проблеми створення ШІС та їх компонентів.

Обговорення питань як концептуального, так і конкретного плану повинне бути конструктивним.

Стосовно питань концептуального характеру це означає, що треба не просто сказати «так, це можливо», але й визначити спосіб побудови обговорюваного об'єкта; чи ж не просто «немає, це зробити не можна», але й показати конкретно, з яких саме причин.

Аналогічно, обговорення тих або інших запропонованих підходів, методів, алгоритмів повинне проводитися так, щоб можна було наочно показати, яку ж частину загальної проблеми створення ШІС та якими засобами дозволяє вирішити пропонуваній метод.

Базою, що полегшує пошук спільної мови представникам різних напрямів, зайнятих вивченням та створенням ШІС, могла б послужити деяка цілеорієнтовна задача, яка в англійській літературі позначається словом challenge.

У світлі вищесказаного цілком природними виглядають наступні вимоги до цілеорієнтовної задачі: з одного боку, вона повинна бути настільки багатобічною та всеосяжною, щоб в її рамках можна було б обговорювати будь-які аспекти ШІС, щоб саме існування такої задачі виправдовувало б пошуки та створення будь-яких нових та ефективних методів та засобів.

Основний принцип задачі — скільки б не зробили, багато не буде ніколи. З іншого боку, ця задача повинна легко масштабуватися та структуруватися з тим, щоб в її рамках можна було б обговорювати не тільки глобальні концепції ШІС, але й будь-які конкретні та часткові питання, пов'язані, наприклад, з алгоритмами ШІС.

Представляється, що на роль такого роду цілеорієнтовної задачі може претендувати проблема створення інтелектуальних автономних систем (ІАС), які є різновидом ШІС. Покажемо окремі (на нашу думку важливі) елементи та проблеми концепції у тому руслі, як це пропонує достатня кількість вчених, які ведуть дослідження у зазначеній області науки та практики.

## 2. Елементи та проблеми концепції створення ІАС

1 Система та середовище її існування. Загальноприйнято вважати, що світ влаштований так: є якась система — предмет нашого вивчення, і є все інше, що не входить в цю систему — зовнішнє середовище.

Система та зовнішнє середовище існують у взаємодії. Ця взаємодія може бути як статичною, коли дія середовища на систему та реакція системи на цю дію постійні, так і динамічною, коли дії та реакції — деякі функції часу.

Класи систем, що відповідають зазначеним взаємодіям, прийнято іменувати відповідно *статичними* і *динамічними* системами.

Практично переважна більшість класів систем, які для нас представляють зацікавленість, є динамічними. Саме вони і є предметом вивчення.

Як відмічалось вище, динамічна система взаємодіє з середовищем, тобто сприймає дії навколишнього середовища та відповідним чином реагує на них. Реакція системи на дії середовища може бути як пасивною, так і активною.

Пасивна взаємодія — це, наприклад, рух каменя, політ артилерійського снаряда або некерованої ракети під впливом гравітаційних та аеродинамічних сил. Інтуїтивно ясно, що динамічні системи, які пасивно взаємодіють з середовищем (тобто є некерованими системами), не представляють зацікавленості з погляду вибору цілеорієнтовної задачі.

При активній взаємодії система, отримавши будь-який вплив від середовища, за тими або іншими правилами формує та реалізує відповідь-реакцію системи на дану дію, наприклад, відхиляє кермо, компенсуючи обурення. Це означає, що динамічна система здатна активно взаємодіяти з середовищем. Вона є системою керованою.

Характер активної взаємодії з середовищем, реалізованого системою, визначається властивостями даної системи, а також ресурсами, які вона має в своєму розпорядженні. В процесі формування реакції на ту або іншу дію середовища система може як спиратися на свої ресурси (наприклад, закони управління або карту місцевості, що реалізовані в ній), так і використовувати якісь інформаційні джерела та управляючі дії зовнішнього характеру (як, наприклад, в завданні навігації з використанням супутникової системи GPS або в завданні телекерування відповідно).

Чим менше система залежить від зовнішніх інформаційних джерел та управляючих команд, тим вище ступінь її автономності.

Необхідна цілеорієнтовна задача, щоб бути нетривіальною, повинна бути пов'язана з керованими динамічними системами, які активно взаємодіють з зовнішнім середовищем, причому ця взаємодія повинна бути направлена на досягнення деякої мети.

Найбільшу практичну зацікавленість представляють ті системи, які володіють достатньо високим рівнем автономності, тобто вирішують ту або іншу прикладну задачу, в основному «спираючись на власні сили». Такий клас систем відомий як системи високої автономності (СВА).

Чи представлятиме зацікавленість той або інший клас СВА з погляду цілеорієнтовної задачі, залежить від поведінки, що реалізується системами цього класу [7...11].

2 *Поведінка системи.* Поведінкою СВА називають послідовність реакцій системи в процесі її взаємодії з зовнішнім середовищем.

Поняття поведінки як взаємодії системи з зовнішнім середовищем було сформульовано і давно досліджується стосовно людини в психології та нейрофізіології [13, 14, 16], а стосовно тварин — в етології [12]. У ширшому сенсі прийнято також говорити про поведінку й інших класів систем, наприклад, фізичних систем.

При виявленні видів поведінки систем класу СВА, що цікавлять нас, в якості відправної точки будемо використовувати той факт, що ці види систем виділяються для живих систем.

Стосовно живих систем, для яких поведінка трактується як форма адаптації, виділимо *стереотипні* та *придбані* види поведінки [12, 15], що включають такі їх різновиди:

- Стереотипні види поведінки:
  - тропізми та таксиси;
  - рефлекси;
  - інстинкти (частково);
- Придбані види поведінки, що модифікуються:
  - інстинкти (частково);
  - навчання;
  - розсудлива діяльність;
  - інсайт.

Як показано в науковій літературі, тропізми та таксиси є простими формами адаптації

«автоматичного» характеру. Тропізми характерні для рослин. Прикладом тут може служити геліотропізм – поворот суцвіття до сонця у соняшника. Таксиси властиві одноклітинним істотам, які стоять на самій нижній сходинці «сходів» тваринного світу. Приклад таксису – рух інфузорії-туфельки до їжі або від джерела роздратування, зокрема, наприклад, дуже яскравого світла.

Рефлекси – це вже форми поведінки, пов'язані з наявністю нервової системи. Вони є ланцюгом подій, коли сигнали від якого-небудь органу чуття передаються за допомогою нервової системи та викликають відповідну автоматичну реакцію. Приклади рефлексу – відсмикування людиною руки при дії на неї, що заподіює біль; розширення зіниць в темноті.

Таксиси та рефлекси відносяться до класу простих стереотипних реакцій. Разом з ними можуть бути й форми складної стереотипної поведінки – інстинкти, властиві даному виду тварин. Характер та цілі такої поведінки, а також умови його запуску детерміновані генетично. Прикладами інстинктивної поведінки є споруда кубла у птахів, міграція у лососевих риб та перелітних птахів.

Перераховані вище форми поведінки (адаптації) є природженими. У міру підйому по еволюційних сходах, з вдосконаленням нервової системи, природжена поведінка все частіше замінюється придбаними навичками. Виникає, наприклад, часткова модифікація інстинктивної поведінки. Зокрема, ссавці у ряді випадків (материнський інстинкт, наприклад) мають у своєму розпорядженні природжене знання, що доповнюється за рахунок придбаного знання.

Вищий рівень поведінки пов'язаний з навчанням, коли нервова система тварин дістає можливість накопичувати інформацію, здатну змінювати їх поведінку.

Навчання реалізується або через наслідування, або шляхом спроб та помилок. Вищі ссавці, головним чином мавпи та людина, мають у своєму розпорядженні також можливості організації поведінки шляхом розсудливої діяльності, не вдаючись до попередніх пробних дій, як це має місце при простому навчанні.

Вище наголошувалося та відмічено у науковій літературі, що предметом розгляду може бути клас систем високої автономності – штучних динамічних керованих систем, які існують в деякому середовищі. Стан цього середовища (і, отже, характер його дії на систему) і самої системи, а також ще ряд чинників, визначають поточну ситуацію. Дана система є керованою. Процес управління припускає, як відомо, наявність цілей управління: цілеорієнтація може бути як зовнішньою, коли цілі задаються системою якоюсь зовнішньою інстанцією, так і внутрішньою, коли цілі формулюються самою системою, виходячи з деякої системи цінностей, мотивацій і т.п.

Залежно від поточної ситуації, а також від поставленої мети, формується та реалізується та або інша реакція системи. Ця реакція може як вибиратися деяким способом з репертуару наперед заготовлених реакцій, так і породжуватися динамічно.

Залежно від того, які механізми залучені в те, щоб поставити у відповідність парі «ситуація-цілі» реакцію, що відповідає їй, виділимо декілька рівнів поведінки СВА:

- «механічна» поведінка;
- стереотипна поведінка на основі засобів, конструктивно вбудованих в систему;
- стереотипна поведінка, придбана шляхом навчання;
- поведінка, яка направляється знаннями та міркуваннями;
- поведінка, яка направляється досвідом;
- поведінка, яка направляється інтуїцією.

«Механічна» поведінка СВА – це реакції системи на дії зовнішнього середовища, реалізовані без залучення управління, виключно за рахунок «фізичних» чинників. Прикладом тут може служити поведінка аеродинамічно стійкого літального апарату при дії на нього обурення, наприклад, вітрового пориву. Аналогом цього виду поведінки СВА в світі живої природи можна вважати тропізми та таксиси.

Стереотипна поведінка СВА на основі засобів, конструктивно вбудованих в систему – це поведінка систем під дією традиційних «жорстких» регуляторів. Закон управління в таких регуляторах заданий при проектуванні системи, фіксований при її виготовленні та ніяким змінам надалі не підлягає. При цьому відповідні програми реагування (ланцюжки елементарних дій) можуть бути як простими, так і надзвичайно складними. Принциповим тут є лише те, що ці програми підготовлені наперед і надалі залишаються незмінними. У живій природі цьому виду поведінки СВА відповідає природжена поведінка на рівні безумовних рефлексів, а також інстинктивна поведінка в тій частині, яка визначається природженими знаннями.

Стереотипна поведінка СВА, придбана шляхом навчання, за характером реагування схожа з попереднім рівнем поведінки. Є деякий репертуар програм-реакцій, де поточному значенню пари «ситуація-цілі» ставиться у відповідність одна з реакцій даного репертуару.

Принципова відмінність даного рівня поведінки від попереднього рівня полягає в тому, що відображення множини значень пар «ситуація-цілі» в множини програм-реакцій (і самі ці програми-реакції) формуються в процесі роботи СВА шляхом навчання, на відміну від попереднього випадку, де це відображення було «природженим» і незмінним. Проте тут істотним є те, що спочатку систему треба «навчити» новим «умінням» (програмам реагування) для якихось вказаних їй класів пар «ситуація-цілі», а лише потім вимагати від неї застосування цих «умінь».

У живій природі даному виду поведінки СВА відповідає інстинктивна поведінка в тій частині, яка пов'язана з модифікацією природженої поведінки, а також поведінка на основі навчання.

Поведінка СВА, що направляється знаннями та міркуваннями — це рівень осмислених дій «за правилами», «по інструкції», коли є деякі знання та формалізовані процедури, яким треба слідувати. В даному випадку при визначенні реакції СВА на поточну пару «ситуація-цілі» використовується вже не просто відображення, згадуване раніше, а складніша конструкція, де замість одиничного відображення використовується деякий ланцюжок міркувань, тобто послідовність відображень, яка формується динамічно — приблизно так, як це робиться в машинах ведення баз знань.

Якщо на попередніх двох рівнях процеси реагування були «однокроковими» (поточній парі «ситуація-цілі» відразу ставилася у відповідність і запускала програма-реакція), то в даному випадку має місце вже багатокроковий процес пошуку адекватної реакції. Проте доступний репертуар реакцій, як і в попередніх двох випадках, повинен бути підготовлений наперед.

У живих системах даному рівню поведінки СВА відповідає розсудлива діяльність в тій її частині, яка пов'язана з діями по наперед відомим розпорядженням (правилах).

Поведінка СВА, що направляється досвідом, певною мірою нагадує попередній рівень (поведінка, що направляється знаннями та міркуваннями). Тут також важливу роль грає навчання.

Принципові відмінності між цими двома рівнями поведінки полягають в наступному.

Замість ланцюжка міркувань (у душі машинного висновку системи, яка заснована на знаннях), методом проб та помилок будується істотно менш формалізований ланцюжок асоціативних контекстно-залежних переходів (приблизно у душі гетероасоціативних нейронних мереж).

Ще істотніше те, що як програми-реакції, так і міркування для них формуються динамічно, в процесі роботи СВА. Це означає, що знімається обмеження, яке було на двох попередніх рівнях поведінки, що вимагало попередньої підготовки як реакцій, так і схем їх пошуку (відображень або дерев висновку).

Таким чином, на даному рівні поведінки відповідність між парою «ситуація-цілі» та реакцією є динамічний об'єкт, сформований в процесі роботи СВА: працюючі, система «накопичує досвід» поведінки стосовно різних ситуацій та цілей.

У живій природі цьому рівню частково відповідає розсудлива діяльність (коли суб'єкт, який може «думати», оперує поняттями, пов'язаними з невизначеністю); частково — рівень інсайту (через неусвідомлюваний характер ряду виконуваних дій).

Поведінка СВА, що направляється інсайтом, пов'язана з реалізацією дій аналогічно, шляхом внутрішнього скріплення різнорідних елементів.

Тут не використовуються такі механізми зіставлення парі «ситуація-цілі» адекватної реакції, як відображення (рівні 2 та 3), логічні міркування (рівень 4) або метод проб та помилок (рівень 5).

Цей вид поведінки треба, мабуть, пов'язувати з інтуїцією, інсайтом в живих системах, коли виникає рішення, яке «незрозуміло звідки взялося» — «осаяння». Ключ до цього виду (рівню) поведінки міститься, можливо, в наступному.

Всі придбані форми поведінки, розглянуті вище (це рівні 3, 4 та 5), мають справу з завданнями, наперед визначеними в тому сенсі, що класи вирішуваних задач та відповідні способи реагування на них задаються тим або іншим способом наперед. Деякий рівень невизначеності в динамічний механізм реагування на зміни значень пар «ситуація-цілі» вноситься тільки на рівні 5 — поведінка, яка направляється досвідом. Але радикально це суть справи не міняє — тут невизначеність грає деяку роль, в основному, в ланці, пов'язаній з «тонкістю застосування» реакцій-програм з метою їх пристосування до наявних умов — «підгонка по місцю». По загальному стилю «досвід», що накопичується і вживається на даному рівні поведінки, дуже нагадує методи евристики.

На відміну від цього, на рівні поведінки, яка направляється інтуїцією, ставляться інші цілі. Традиційна архітектура інтелектуальних систем володіє однією «природженою» особливістю, що істотно знижує їх ефективність — система, навчена (або підготовлена якимнебудь іншим способом) рішенням деякої часткової задачі, абсолютно безпорадна перед іншими частковими завданнями. В зв'язку з цим напрошується необхідність пошуку механізмів, що наділяють систему якимись метавластивостями, тобто адаптивністю до виду завдань, які пред'являються системі для вирішення — це і є основна мета для рівня поведінки, що направляється інтуїцією.

Системи високої автономності, що реалізують хоча б один з рівнів поведінки 4, 5 або 6, в літературі названо інтелектуальними автономними системами (ІАС).

Очевидно, що ІАС призначена для вирішення достатньо нетривіальних завдань. Вона реалізуватиме не один якийсь рівень поведінки, а деяку їх комбінацію — «суміш». Компоненти цієї «суміші» можуть з'являтися або поодиноці у деякій черговості, або одночасно — це буде випадок «основного завдання» для штучних інтелектуальних систем, коли потрібно створити систему, зіставну по інтелектуальним можливостям з людиною.

Рівні поведінки від третього і нижче також можуть брати участь в згаданій «суміші». У загальному випадку вони реалізуватимуться на тлі інших рівнів поведінки в поєднанні з ними на одному й тому ж відрізьку часу.

У ряді випадків СВА задіюватиме тільки «неінтелектуальні» рівні поведінки. Це може мати місце у двох випадках:

1) якщо дана система в принципі не в змозі за своїми потенційними можливостями реалізувати завдання вищих рівнів поведінки;

2) коли поточна пара «ситуація-цілі» така, що для відповідної реакції цілком достатньо нижчих рівнів поведінки (свого роду «принцип найменшої дії»).

Ці випадки СВА можна трактувати як вироджені варіанти ІАС і розглядати їх в тій мірі, що потрібно для повноти загальної картини.

3 *Властивості інтелектуальних автономних систем.* Проведений аналіз показує, що значна частина перспективних технічних систем створюватиметься або в класі ІАС і співтовариств таких систем, або міститиме в собі ІАС у якості підсистем.

ІАС – це системи, що реалізують деяку комбінацію з рівнів поведінки, перерахованих вище, які володіють високим рівнем самостійності та, зокрема, уміють:

- досягати поставлених цілей у високодинамічному середовищі зі значним числом різнорідних невизначеностей в ньому;

- коректувати поставлені цілі, а також формувати нові цілі й комплекси цілей, виходячи із закладених в ІАС ціннісних та нормативних установок (мотивації);

- здобувати нові знання, накопичувати досвід рішення різноманітних задач, навчатися на цьому досвіді, модифікувати свою поведінку (реакції на зміну ситуації) на основі отриманих знань та накопиченого досвіду;

- адаптуватися до виду завдань, в рішенні яких виникає необхідність, зокрема навчатися рішенням задач, не передбачених первинним проектом системи;

- утворювати «колективи» з ІАС (співтовариства ІАС), націлені на взаємодію їх членів при рішенні деякої загальної задачі. Ці колективи повинні мати в своєму розпорядженні можливості самоструктуризації (різнорідність елементів колективів ІАС, різнорідність та динамічність зв'язків між ІАС), виходячи з поточної і/або прогнозованої ситуації;

- здійснювати самовідтворення з залученням місцевих сировинних та енергетичних ресурсів, можливо, зі змінами в «геномі» системи (для підтримки процесів еволюції в співтовариствах ІАС).

В першу чергу ІАС – це самодостатні системи, на які може покладатися рішення певного комплексу прикладних завдань в повному обсязі. Є, проте, ще одна важлива ніша для ІАС – це інтелектуальні автономні підсистеми в складних системах, що включають людину-оператора: пілотовані літаки, вертольоти, космічні апарати, надводні, підводні або наземні апарати і т.п. Такі підсистеми націлюються на те, щоб у максимальному ступені самостійно вирішувати поставлені перед ними задачі. Їх використання дає можливість істотно підвищити якість реалізації критично важливих функцій, зменшити робоче навантаження людини-оператора та підвищити за рахунок цього безпеку та ефективність

експлуатації відповідних складних систем згідно з їх цільовим призначенням.

Структурно «діяльність» ІАС у деякому середовищі розділяються на три сфери:

а) Сприйняття поточної ситуації (ситуація = зовнішня-ситуація + внутрішня-ситуація) – сенсорні функції.

б) Формування реакції-відповіді на поточну або прогнозовану ситуацію (види можливих реакцій: зміна стану ІАС у її фазовому просторі, реконфігурація, реструктуризація, адаптація цілей, самонавчання, самоорганізація і т.п.) – управляючі функції.

в) Реалізація (формування) реакції на поточну або прогнозовану ситуацію – ефекторні функції.

Задане поєднання необхідних «умінь» ІАС та необхідних конкретних форм її «діяльності» визначає склад функцій, які повинна реалізувати ІАС того або іншого виду (приклади таких функцій: зір в різних діапазонах електромагнітного випромінювання, поєднання сенсорних даних з різних джерел в єдину інформаційну картину, виявлення відмов в системах ІАС, компенсація виявлених відмов шляхом реконфігурації/реструктуризації і т.п.).

4 *Застосування інтелектуальних автономних систем.* На сьогоднішній час широко вживається пара термінів: «автоматична система» і «автоматизована система». Перший з цих термінів відноситься до систем, які діють без якого-небудь втручання людини в процесі їх функціонування. Другий – визначає такі системи, де як активний елемент присутня людина.

По аналогії з цим можна розрізняти системи-роботи (або просто роботи) і роботизовані системи. Обидва класи систем є найбільш очевидною сферою застосування ІАС.

До систем-роботів, які доцільно було б реалізувати на рівні ІАС, можна віднести літаки-роботи, вертольоти-роботи, космічні апарати-роботи, надводні та підводні апарати-роботи, безлюдні (автоматичні) системи, призначені для вирішення складних комплексів завдань без втручання людини і т.п.

Різниця між апаратами-роботами та традиційними автоматичними апаратами полягає в істотно вищому рівні самостійності поведінки роботів у порівнянні з традиційними автоматами, у здатності роботів навчатися, накопичувати та використовувати досвід в ході рішення поставлених задач.

Прикладами роботизованих систем можуть служити пілотовані літальні апарати, до складу систем та бортового устаткування яких входять (на правах підсистем) ІАС. Тут обов'язковим та активним елементом системи, в цілому, є людина. Зовсім необов'язково при цьому, щоб людина знаходилася безпосередньо на борту роботизованого апарата. Вона може здійснювати свої функції (контроль, управління, цілеорієнтування і т.п.), знаходячись поза апаратом, у тому числі і на значному віддаленні від нього. Прикладом такого роду апаратів можуть служити роботизовані дистанційно пілотовані

літальні апарати, а також інші дистанційно керовані апарати, коли оператор та керований ним апарат рознесені в просторі.

Як приклади систем, роботизація яких дозволила б різко підвищити їх ефективність, можна назвати:

- системи організації повітряного руху;
- засоби управління енергетичними системами;
- засоби підтримки процесів контролю та управління виробництвами традиційних видів (машинобудівними, хімічними і нафтохімічними, добувними і т.п.);

- засоби управління в надзвичайних обставинах: ліквідація наслідків стихійних та техногенних катастроф та ін.;

- інші сфери, де ІАС можна було б використовувати як засоби підтримки процесів формування та ухвалення рішень («інтелектуальні помічники») в людино-машинних системах, які дозволяють працювати в середовищах зі значним числом різномірних невизначеностей, в умовах великих потоків даних та жорстких часових обмежень.

Ще одна очевидна сфера застосування ІАС – роботизована побутова техніка різного призначення, а також співтовариства роботизованих побутових пристроїв («роботизоване житло»).

### Висновки

Як слідує з приведеного вище матеріалу, завдання вивчення та створення інтелектуальних автономних систем цілком може претендувати на статус цілеорієнтовної задачі, яка може бути викликом сучасним та перспективним інформаційним технологіям. Завдання задовольняє тим вимогам, які були сформульовані у введенні до проблеми. Воно допускає широкий діапазон постановок, які можуть змінюватися як по рівню спільності (від концептуальних до конкретно-алгоритмічних), так і по складності вирішуваних задач. Т.ч. завдання забезпечує необхідну масштабованість підходу. У свою чергу, таке варіювання приводить до необхідності залучення в процес вивчення та створення ІАС всіх без виключення дисциплін, пов'язаних з ШІС.

**Список літератури:** 1. Мак-Каллок, У. С. Логическое исчисление идей, относящихся к нервной активности [Текст] / У. С. Мак-Каллок, В. Питтс // *Нейрокомпьютер*. – 1992. – №3. – С. 39-53. 2. Скопа, О. О. Глобальні властивості нейронних мереж [Текст] / О. О. Скопа, Н. Ф. Казакова // *Наукові записки УНДІЗ*. – К. : УНДІЗ. – 2008. – №3(5). – С.13-19. 3. Дрейфус, У. Чего не могут вычислительные машины: Критика искусственного разума : монографія / Переклад з англ. Н. Родмана ; під ред. Б. В. Бірюкова. – М. : Прогресс, 1978. – 334 с. 4. Скопа, А. А. Нейронна мережа як об'єкт техносфери: властивості [Текст] / А. А. Скопа, Н. Ф. Казакова // *Наукові записки Міжнародного гуманітарного університету*. – Одеса : МГУ. – 2008. – №11 : Управління проектами та програмами ; відп. за випуск А. І. Рибак. – С.44-48. 5. *McCarthy J. What is Artificial Intelligence?* [Електронний ресурс] // Портал : без назви. – Режим доступу \www/ URL : <http://www-formal.stanford.edu/jmc/whatisai.html>. – Заголовок з екрану, доступ вільний, 13.01.2010. 6. *Минский, М.* Перцеп-

троны : монографія / М. Минский, С. Пейперт : Переклад з англ. Г. Л. Гімельфарба та В. М. Шарипанова ; під ред. В. А. Ковалевського. – М. : Мир, 1977. – 262 с. 7. Скопа, О. О. Концептуальні положення створення нейронних мереж в Україні [Текст] / О. О. Скопа // *Наукові записки УНДІЗ*. – К. : УНДІЗ. – 2008. – №1(3). – С.3-19. 8. Скопа, О. О. Наукова концепція інтелектуальної мережі [Текст] / О. О. Скопа // *Наукові записки Міжнародного гуманітарного університету*. – Одеса : МГУ. – 2008. – №13 : Економіка та управління проектною діяльністю ; відп. за випуск А. І. Рибак. – С. 44-49. 9. *Вилли, К.* Биология: Биологические процессы и законы : монографія / К. Вилли, В. Детье. – М. : Мир, 1975. – 822 с. 10. Скопа, О. О. Нейронна мережа як об'єкт техносфери: концептуальні положення створення [Текст] / О. О. Скопа, Н. Ф. Казакова // *Наукові записки Міжнародного гуманітарного університету*. – Одеса : МГУ. – №10 : Управління проектами та програмами ; відп. за випуск А. І. Рибак. – С. 44-60. 11. *Баховський, П. Ф.* Концептуальні положення створення та розвитку інтелектуальних мереж: Основоположні зауваження [Текст] / П. Ф. Баховський, О. О. Скопа // *Моделювання та інформаційні технології*. – К. : ІПМЕ НАН України. – 2008. – №49. – С. 154-159. 12. *Кемп, П.* Введение в биологию : монографія / П. Кемп, К. Арме. – М. : Мир, 1988. – 671 с. 13. *Блум, Ф.* Мозг, разум и поведение : монографія / Ф. Блум, А. Лейзерсон, Л. Хофстедтер. – М. : Мир, 1988. – 248 с. 14. *Веккер, Л. М.* Психика и реальность: Единая теория психических процессов : монографія / Л. М. Веккер. – М. : Смысл; PerSe, 2000. – 685 с. 15. *Годфруа, Ж.* Что такое психология / Ж. Годфруа. – М. : Мир, 1992. – Т. 1. – 496 с. – ISBN відсутній. 16. *Выготский, Л. С.* Психология : підручник / Л. С. Выготский. – М. : ЭКСМО-Пресс, 2000. – 1008 с.

*Поступила до редколлегии 15.10.2012*

УДК 681.5.015; 519.7

**Интеллектуальные автономные системы: концептуальные положения создания и функционирования** / А. А. Скопа, Е. В. Вавилов // *Бионика интеллекта: науч.-техн. журнал*. – 2013. – № 1 (80). – С. 35-40.

На основе анализа перспектив научных исследований и разработок в области искусственного интеллекта делается вывод о необходимости формулирования проблемной задачи, которая могла бы послужить базой для корректного обсуждения и сопоставления результатов как концептуального, так и прикладного содержания для искусственных интеллектуальных систем. Предлагается использовать в качестве такой задачи проблему изучения и создания интеллектуальных автономных систем. Рассматриваются основные особенности таких систем и приводятся характерные для них примеры.

Библиогр.: 16 назв.

UDK 519.7

**Intelligent autonomous systems: conceptual principles of creation and functioning** / О. О. Скопа, Е. В. Вавилов // *Bionics of Intelligence: Sci. Mag.* – 2013. – № 1 (80). – P. 35-40.

On the basis of analysis of prospects of scientific research-and-developments in area of artificial intelligence conclusion is drawn about the necessity of problem which would serve by a base for the correct discussion and comparison of results of both conceptual and concrete plan for the artificial intellectual systems definition. It is suggested to use as such task problem of study and creation of the intellectual off-line systems. The basic features of such systems are examined and characteristic examples are made for them.

Ref.: 16 items.



А.Л. Ерохин

Харьковский национальный университет радиоэлектроники,  
г. Харьков, Украина, ayerokhin@ukr.net

## МОДЕЛИРОВАНИЕ ВЗАИМОДЕЙСТВИЙ ВНЕШНИХ ФАКТОРОВ С СИСТЕМОЙ КОГНИТИВНЫХ ФУНКЦИЙ ЧЕЛОВЕКА-ОПЕРАТОРА

В статье на основе теоретико-множественного подхода разработана оригинальная модель взаимодействия системы когнитивной деятельности человека с внешними воздействиями. Рассмотрены модели генерации дискретно-хаотических процессов. Исследованы взаимодействия дискретно-хаотических процессов с системой когнитивных функций человека-оператора.

МОДЕЛЬ КОГНИТИВНОЙ ДЕЯТЕЛЬНОСТИ, ВНЕШНИЕ ВОЗДЕЙСТВИЯ, ДИСКРЕТНО-ХАОТИЧЕСКИЕ ПРОЦЕССЫ

### Введение

При разработке систем и средств искусственного интеллекта одним из наиболее важных остается вопрос моделирования интеллектуальной деятельности человека. Важной составляющей такого класса задач является моделирование влияния различных внешних факторов на процессы принятия решений при идентификации когнитивных функций человека [1]. Актуальность теоретических и прикладных исследований взаимодействий стохастических процессов с системой когнитивных функций человека определяется усложнением задач принятия решений в условиях интенсивных внешних стохастических воздействий на комплексы психофизиологических состояний человека-оператора.

Разработка и исследование моделей таких взаимодействий дает возможность повысить эффективность функционирования систем искусственного интеллекта систем в поле воздействий стохастических факторов, получить практические средства снижения вероятности принятия ошибочных решений.

Статья посвящена разработке моделирования влияний стохастических воздействий на комплекс психофизиологических состояний человека с целью разработки интеллектуальных средств стабилизации когнитивных функций мозга человека при выполнении им функций принятия решений.

### 1. Анализ литературы и постановка задачи

Процессы принятия решений человеком являются сложноорганизованными процессами, образующими иерархическую открытую систему, которая динамически изменяет свои состояния под влиянием как экзогенных, так и эндогенных факторов [2].

Изменение этих факторов носит случайный характер, поэтому смена психических состояний человека может быть представлена последовательностью переходных состояний, у которых каждое последующее состояние зависит от предыдущего. Это свойство дает возможность характеризовать этот процесс как марковский. Большой интерес

представляют так называемые пограничные состояния, возникающие у человека, например, при длительном и постоянном воздействии стрессоров или при хронических заболеваниях.

В частности, стресс монотонии является неспецифической реакцией системы психофизиологического статуса лица, принимающего решение, на длительное воздействие повторяющихся (монотонных) стимулов. Рассматриваемый стресс можно представить как защитную реакцию центральной нервной системы, которая ставит фильтр, перекрывающий доступ указанным малоинформационным воздействиям в высшие отделы сознания [3]. Для человека-оператора основной задачей является реализация именно операторских функций

Рассмотрим некоторую формальную модель взаимодействия интеллектуальной (человеко-машинной) системы со стохастическими возмущающими влияниями. Рассматриваемая система, являясь эргатической, наряду с подсистемами управления содержит систему поддержки принятия решений и модель интеллектуальной деятельности человека в виде комплекса когнитивных функций человека-оператора.

Представим рассматриваемую систему в виде иерархии подсистем, связанных между собой отношениями координации и субординации. Представим структуру системы когнитивных функций человека-оператора в виде локализованного процесса, а процессы, происходящие в ее подсистемах – в виде топологий. В такой декомпозированной системе отношения координации и субординации между подсистемами предполагают возможность преобразований выходных параметров с нижележащей  $S$ -й подсистемы на вход  $S+1$ -й подсистемы.

Естественно предположить, что некоторое подмножество внутренних  $i$ -параметров на выходе обладает упорядоченностью. При передаче  $i$ -параметров на вход  $S+1$ -й подсистемы возможны варианты, когда упорядоченность структур-процессов выхода-входа остается без изменений или претерпевает нарушения.

Необходимо разработать модель генерации дискретно-хаотических процессов и исследовать возможные модели взаимодействий с системой когнитивных функций человека.

**2. Генерация дискретно-хаотических процессов в системе когнитивных функций человека-оператора**

Топология структуры внутренних параметров в системе связей основывается на изоморфизме отображения множества  $i$ -параметров выхода на множество входа с такой же мощностью.

Рассмотрим случай, когда структуры параметров и их флуктуаций сохраняют упорядоченность. Представим  $A_{S_j}$  множество  $a_i$ -параметров выхода  $S_j$ -подсистемы на  $k$ -уровне в виде объединения дискретных параметров :

$$\{A\}_{S_j}^k = \bigcup_{i \in M^2} (a_i)_{S_j}^k, \quad i \in M^2. \quad (1)$$

Отобразим множество  $\{A\}_{S_j}^k$  на множество  $\{B\}_{S_{j+1}}^{k+1}$  входов  $S_{j+1}$  подсистемы. Вследствие изоморфности отображения параметры  $S_j$ -подсистемы на выходе и параметры входа  $S_{j+1}$  подсистемы тождественны друг другу. Выражение (1) представим в виде

$$\{A\}_{S_j}^k = \bigcup_{i \in M^2} (a_i)_{S_j}^k \equiv \bigcup_{i \in M^2} (b_i)_{S_{j+1}}^{k+1} = \{B\}_{S_{j+1}}^{k+1} \quad (2)$$

при наличии условия взаимной сопряженности  $(a_i)_{S_j} \leftrightarrow (b_i)_{S_{j+1}}$ . Изоморфизм отображений предполагает тождественность локальных систем координат, в которых задана упорядоченность параметров. Зададим  $i$ -параметрам выхода флуктуации  $\Delta a_i$ .

Посредством подсистемы автоматического управления указанные флуктуации компенсируются на входе, и параметры выхода-входа по истечении времени  $\Delta t$  становятся равноэквивалентными с точностью до ошибки установления параметра в оптимальном диапазоне изменений

$$(a_i \pm \Delta a_i)_{S_j} \equiv (b_i)_{S_{j+1}}. \quad (3)$$

Если значение флуктуаций  $\Delta a_i$  превышают возможности авторегуляции системы, то посредством ручного управления за время, превышающее  $\Delta t$ , может быть установлен режим, близкий к оптимальному, тогда (3) можно представить в виде

$$\begin{aligned} (a_i \pm \Delta a_i)_{S_j} &\neq (b_i \pm \Delta b_i)_{S_{j+1}} \rightarrow \\ &\rightarrow (a_i)_{S_j} \approx (b_i)_{S_{j+1}} \quad \forall t > \Delta t. \end{aligned} \quad (4)$$

Из (3) и (4) видно, что превышение уровней флуктуаций не приводит к возникновению неустойчивости системы когнитивных функций человека-оператора на некотором временном промежутке. Именно в этом состоянии являются важными воздействия стохастических факторов. Однако сами флуктуации, как внутренние параметры воздействия на систему, детерминированы множеством сценариев нестандартных ситуаций.

Гораздо больший интерес может представить возникновение хаотического процесса изменения внутренних параметров, не предусмотренных указанными сценариями.

Рассмотрим вариант, когда при отображении параметров изменяется исходная упорядоченность. Нарушение упорядоченности параметров  $(a_i)_{S_j}$  выхода означает появление на множестве  $(b_i)_{S_{j+1}}$  входа новой локальной системы координат, при которой отображенному параметру выхода соответствует параметр  $b_k$  входа, локализованный в другом месте среды.

Однако нарушение упорядоченности не означает нарушение отношения сопряженности

$$(a_i \pm \Delta a_i) \leftrightarrow (b_k \pm \Delta b_k) \quad \forall i \neq k. \quad (5)$$

Если общее число временно неупорядоченных  $k$ -параметров образует подмножество  $k_0 < k \leq i$ , то при  $T \gg \Delta t$  в динамике отображения параметров приобретают хаотичный характер.

Можно рассмотреть условные системные каналы, состоящие из множества элементарных информационно независимых каналов, у которых отображение параметров  $a_i \leftrightarrow b_k \quad \forall i \neq k$  осуществляется посредством комбинаторно-топологических преобразований.

Как было показано в [4], процессы, генерируемые в результате таких комбинаторно-топологических преобразований, являются детерминированными хаотическими процессами, относительно устойчивые внутри своих аттракторов. Для комбинаторно-топологических преобразований аттракторами являются компактные объединения гомеоморфизмов топологических пространств с дискретными носителями топологии.

**3. Исследование взаимодействия дискретно-хаотических процессов с системой когнитивных функций человека**

Для исследования моделируемых взаимодействий дискретно-хаотических процессов и когнитивных функций человека применим дискретно-хаотический генератор (ДХП-генератор), который размещается между  $S_j$  и  $S_{j+1}$  подсистемами так, что выходы  $(a_i)_{S_j}$  поступают на его вход, преобразуются в нем случайным образом и поступают на вход  $S_{j+1}$  подсистемы.

Таким образом моделируется сложноорганизованная динамическая система связей, флуктуации структур-процессов параметров которой детерминированы аттракторами двумерного, фазового, дискретного пространства (метрического или логического).

Когда в рассматриваемой модели взаимодействия отключены ДХП-генераторы, то устойчивость системы обеспечивается эффективной работой механизмов обеспечения внутрисистемной стабильности — гомеостазиса [3]. Любое возмущение

внутренней или внешней природы может привести к нестабильности системы когнитивных функций двух видов:

– внутрисистемная нестабильность, когда интенсивность флуктуации параметров превышает возможности системы гомеостаза. Необходим временный перевод системы в новое гетеростатическое состояние;

– внешнесистемная — определяется последовательностью переходов жизненного цикла системы от оптимально функционирующей к состоянию хронической нестабильности вплоть до ее гибели. Квазистатическая фаза устойчивости в системе сохраняется до появления в ней режимов с обострением, при этом, находясь в состоянии неустойчивости, система становится чувствительной к резонансным взаимодействиям низкой интенсивности.

Основным видом связей-отношений между подсистемами являются отношения субординации и координации, или отношения по горизонтали и по вертикали. Рассмотрим элементарные каналы  $(w_i)^k \in W \subset CK$ , образующие в совокупности условный системный канал, по которому осуществляется взаимодействие стохастических воздействий с системой когнитивных функций человека. Обозначим параметры входов и выходов между подсистемами на  $k$ -уровне как  $(w_i)_b^k$  и  $(w_i)_a^k$ . В соответствии с установленными связями между подсистемами их  $i$ -параметры выходов и входов тождественны друг другу

$$\begin{aligned} (w_i)_b^k &\equiv (w_i)_a^k \quad \forall (w_i)_b^k \leftrightarrow (w_i)_a^k; \\ (w_i)_b^k &\equiv (w_i)_a^{k+1} \quad \forall (w_i)_b^k \leftrightarrow (w_i)_a^{k+1}. \end{aligned} \quad (6)$$

Первое тождество описывает связи координации, второе — субординации.

Определим свойство (6) подсистем как принцип тождества. Каждый из процессов, протекающий в своем “черном ящике”, функционально предопределен закономерностями, характерными для определенной предметной области. Отношения в системе входов-выходов определяются только системами (6).

Анализ взаимодействия ДХП-генератора и реакцию системы проведем на основании следующих предположений:

– связи-отношения между подсистемами являются локализованными структурами-процессами; параметры выходов-входов между подсистемами подчиняются принципу тождества и взаимно однозначны друг другу (6);

– реакция подсистем открытой и нелинейной выражается флуктуациями интенсивностей и структур-процессов выходных параметров;

– включенный в систему ДХП-генератор изменяет упорядоченность параметров слабо структурируемых элементарных каналов внутри системного канала в системе связей-отношений координации и субординации.

Возьмем два рядом расположенных уровня  $k$  и  $k+1$ , которые содержат подмножества подсистем  $\bigcup_{i=1}^n (P_i)^k$  и надсистем  $(S_j)^{k+1}$ . В процессе функционирования подсистема  $(P_i)^k$  формирует на своем выходе подмножество  $(w_i)_b^k$  выходных параметров, которые, в соответствии с принципом тождества, являются и параметрами подмножества  $(w_i)_a^k$  входов для надсистемы  $(S_j)^{k+1}$ .

Для подсистем  $(P_i)^k$  все выходы  $(w_i)_b^k \in W$  образуют исходную упорядоченную структуру параметров, соответствующую оптимальному функционированию подсистемы и всех подсистем  $(P)_{i+s}^k$ ,  $(S)_j^{k+1}$ , с которой он связан.

При отображении множества

$$(w_i)_b^k \rightarrow (w_i)_a^k, (w_i)_b^k \rightarrow (w_i)_a^{k+1}$$

упорядоченность элементов организует для входов надсистемы  $(S)_j^{k+1}$  некоторый образ  $W^*$ .

На подмножествах рассматриваемых связей-отношений можно выделить  $S$ -подмножество признаков, по которым все параметры выходов-входов могут быть соотнесены в  $\Omega_s$ -подклассы эквивалентности по  $S$ -признаку параметров.

Для каждого из уровней рассматриваемой системы подклассы  $\Omega_s$  образуют кластеры, включающие в себя собственные замыкания  $(\bar{\Omega}_s)^k$ ,  $(\bar{\Omega}_s)^{k+1}$  и нулевой элемент  $\Omega$

$$(\Omega_s) = \bigcup (\Omega_s)^k / (\bar{\Omega}_s)^k \equiv \bigcup (\Omega_s)^{k+1} / (\bar{\Omega}_s)^{k+1}. \quad (7)$$

Это дает возможность рассматривать указанные подклассы как подмножества топологического пространства  $(\Omega_s, w_s)$  с дискретным носителем  $w_s$  топологии.

Каждый подкласс эквивалентности признаков представим в виде объединения подмножеств

$$(\Omega_s)^k = \bigcup_{i=1}^n (\Omega_s)_i^k, \quad (8)$$

при этом объединение (6) может быть тождественным как всему подклассу входов, так и его части

$$(\Omega_s)^k \subset (\Omega_s)^{k+1}. \quad (9)$$

Число элементов  $(w_i)_b^k$  и  $(w_i)_b^{k+1}$ , образующих соответствующие подклассы  $(\Omega_s)^k$  и  $(\Omega_s)^{k+1}$ , конечно, и поэтому они полностью определяют систему конечных связей-отношений с выхода  $(P_i)^k$  подсистемы на вход  $(S_j)^{k+1}$  надсистемы.

Представляет интерес определение границ существования изоморфных преобразований элементов  $(w_i)_b^k$  и  $(w_i)_b^{k+1}$  внутри подклассов эквивалентности  $S$ -признаков при воздействиях ДХП.

Внутри подкласса эквивалентности параметры элементов выходов-входов могут суммироваться,

объединяться, сравниваться, а вне подкласса – нет. Условное поперечное сечение подмножеств  $(w)_i^k$  и  $(w)_i^{k+1}$  топологических областей  $(\Omega_s)^k$  и  $(\Omega_s)^{k+1}$  можно рассматривать как сет параметров выхода с каждой из  $(P_i)^k$  подсистемы  $k$ -уровня и образа системы входов  $S_j$  надсистемы на  $k+1$ -уровне. Такое рассмотрение системы выходов-входов с теоретико-множественной точки зрения возможно вследствие топологической эквивалентности областей

$$(\Omega_s)_{i=1,2,\dots,n}^k \subset (\Omega_s)_j^{k+1}. \quad (10)$$

С этих позиций анализ систем связей выходов и входов между подсистемами и реакции системы на внешние воздействия переводится в плоскость задачи сравнения образов.

Любая флуктуация структуры-процесса, сгенерированная ДХП-генератором, нарушает упорядоченность выходного образа, что свидетельствует об отсутствии взаимной однозначности между  $(\Omega_s)_{i=1,2,\dots,n}^k \subset (\Omega_s)_j^{k+1}$ . Однако при этом области  $(\Omega_s)^k, (\Omega_s)^{k+1}$  гомеоморфны и их размерности (мощности) остаются неизменными при всех топологически подобных преобразованиях.

Будем считать, что все аффинные преобразования как самих  $(w_s)_i^k \subset (w_s)_i^{k+1}$ , так и самих областей  $(\Omega_s)_{i=1,2,\dots,n}^k \subset (\Omega_s)_j^{k+1}$  не изменяют их упорядоченности. Эти свойства могут считаться инвариантами системы связей-отношений.

Проведем реорганизацию элементов  $(w_s)_{i,a}^k \subset (w_s)_{i,a}^{k+1}$  выходов.

Для этого произведем объединение параметров по  $S$ -признакам для всех  $i$ -подсистем

$$(\Omega_{s=1,2,\dots,m})^k = \bigcup_{i=1}^n (\Omega_{s=1,2,\dots,m})^{ki}. \quad (11)$$

В результате такой реорганизации все связи-отношения образуют образ всех входов в виде объединения  $S$ -подклассов

$$(\Omega)_j^{k+1} = \left[ \bigcup (\Omega_s)^k \right] \bigcup (\bar{\Omega}_s)_j^{k+1}, \quad (12)$$

где  $(\bar{\Omega})_j^{k+1}$  – замыкание области образа системы входов для надсистемы  $(S_j)^{k+1}$ .

После реорганизации новые области (12) не меняют топологических инвариантов. Свойство тождественности и упорядоченности параметров  $(w_s)_i^k$  в системе выходов-входов является теоретико-множественной формализацией анализируемых связей-отношений.

### Выводы

Таким образом, рассмотренный подход к моделированию влияния внешних факторов на когнитивные функции модели интеллектуальной деятельности человека дает возможность определить

основные типы преобразований между множествами  $W$  и  $W^*$  системы когнитивных функций человека-оператора.

Предложенный теоретико-множественный подход дает возможность при изучении моделей дискретно-хаотических воздействий на когнитивные функции человека-оператора определить все множества семантик, описывающих потоки исходной информации и построить практические средства повышения устойчивости операторских функций человека как составную часть интеллектуальной человеко-машинной системы.

Такие средства должны основываться на моно-сенсорном визуальном воздействии невербального типа, в котором отсутствуют постоянные, циклически повторяющиеся компоненты изображений, что вызывает у человека-оператора подсознательную заинтересованность и повышает эффективность функционирования комплекса когнитивных функций человека [5, 6].

**Список литературы:** 1. Ерохин, А.Л. О визуальном кодировании информации на основе идентификации когнитивных функций человека-оператора. Сообщение 1 / А.Л. Ерохин // Бионика интеллекта: науч.-техн. журнал. – 2006. – №2(65). – С.19-26. 2. Федоров, Л.И. Эндогенные ритмы организма как факторы модуляции параметров стимуляции / Л.И. Федоров // Биофизика, 1996. – Т. 41, Вып. 3. – С. 718-724. 3. Селье Г. Стресс без дистресса. – Рига: Виеда, 1992. – 109 с. 4. Бурцев, В.М. Застосування теорії груп підстановок для моделювання детермінованих хаотичних процесів / В.М. Бурцев, А.Л. Ерохин // Системи обробки інформації. Збірник наукових праць. – Харків: НАНУ, ПАНМ, ХВУ, 2001. – Вип. 6 (16). – с.47-51. 5. Solso R.L. Cognition and Visual Arts, Cambridge, MA: The MIT Press, 1994. 6. Ruggieri V., Morelli J. Chromatic perception in relation to an hypothesized cerebral dominance // Percept and Mot. Skills. – 1985. – V.60. – P.583-589.

Поступила в редколлегию 09.10.2012

УДК 004.89

**Моделювання взаємодії зовнішніх факторів з системою когнітивних функцій людини-оператора / А.Л. Єрохін // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 41–44.**

В статті на основі теоретико-множинного підходу розроблено модель взаємодії дискретно-хаотичного процесу з системою когнітивних функцій людини в інтелектуальній людино-машинній системі.

Бібліогр.: 6 найм.

UDC 004.89

**Modeling of the external factors interaction with the cognitive functions of the human operator / A.L.Yerokhin // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 41–44.**

The paper proposes a set-theoretic approach to interaction of model discrete chaotic process with the human cognitive functions in intelligent human-machine system.

Ref.: 6 items.

УДК 004.93

А.Д. Дрюк<sup>1</sup>, Е.И. Кучеренко<sup>2</sup><sup>1</sup> ХНУРЭ, г. Харьков, Украина, sanya40@ukr.net<sup>2</sup> ХНУРЭ, г. Харьков, Украина, тел. 7021337

## СТОХАСТИЧЕСКИЕ МОДЕЛИ И МЕТОДЫ СУБОПТИМАЛЬНОЙ МАРШРУТИЗАЦИИ СЛОЖНЫХ ОБЪЕКТОВ

Разработана стохастическая модель перемещения группы сложных объектов по территории производственного помещения. Сформулирована задача перемещения грузов сложными объектами за фиксированное время. Предложен метод, который за полиномиальное время находит субоптимальные траектории объектов, позволяющие решить данную задачу с вероятностью не менее пороговой. Приведена UML-диаграмма классов приложения, реализующего предложенный метод.

МОБИЛЬНЫЙ ОБЪЕКТ, МАРШРУТИЗАЦИЯ, РАЗБИЕНИЕ НА ЗАДАНИЯ, СУБОПТИМАЛЬНАЯ ТРАЕКТОРИЯ, АСИМПТОТИЧЕСКАЯ СЛОЖНОСТЬ

### Введение

Проблемы построения систем управления едва ли не самые трудные при построении гибких производственных систем (ГПС). Однако практика создания, освоения и эксплуатация сначала простых, а затем более сложных систем в промышленности, а также опыт теоретических исследований специфических для ГПС проблем управления позволяют создать необходимую теоретическую и инструментальную базу для быстрого последующего развития инженерной практики в области комплексной автоматизации производства на основе использования наиболее прогрессивных форм его организации, применения самой передовой техники и технологий [1].

Одной из подсистем системы управления ГПС является система управления транспортным и погрузочно-разгрузочным процессом. Этот процесс входит во все фазы производства, так как связан с распределением предметов труда и без принципиально новой транспортной техники и эффективной системы транспортирования, реализующей новую транспортную технологию, невозможна высококоразвитая промышленность.

Наиболее эффективным средством транспортировки грузов являются сложные объекты, к которым, в первую очередь, следует отнести мобильные объекты. В работе [2] описан один из методов оптимизации перемещения группы мобильных объектов, основанный на использовании нейронных сетей. Однако время работы этого метода очень сложно оценить, поскольку время обучения нейронной сети существенно зависит от ее входных параметров.

В данной работе предлагается другой подход, основанный на стохастической модели перемещения мобильных объектов по территории производственного помещения с заданной инфраструктурой. Метод оптимизации маршрутов мобильных объектов, предлагаемый авторами работы, позволяет находить субоптимальное решение поставленной задачи за полиномиальное время.

Одним из преимуществ автоматизации транспортных средств является сокращение времени ожидания грузов на рабочих местах и сокращение затраты энергии за счет эффективного планирования маршрутов движения. Поэтому в системе управления мобильными транспортными объектами одно из основных мест занимает планирование их перемещения, что является важным и актуальным.

Автоматизация процесса планирования перемещения, при минимизации затрат времени на подготовительно-заключительные операции и ускорении процесса переключения транспортного мобильного объекта с одного производственного задания на другое, является основой для организации гибкого производства [3].

Целью данной работы является оптимизация управления мобильными объектами на производстве, что позволит сократить потребление энергии и увеличить срок эксплуатации мобильных объектов.

### 1. Постановка задачи исследования

Пусть территория производственного помещения представляет собой прямоугольное поле, разбитое на единичные клетки. На этой территории находится  $L$  грузов и  $K$  мобильных объектов, способных погрузить, разгрузить и перемещать грузы. Пусть  $G = \{g_1, g_2, \dots, g_L\}$  – множество грузов, а  $R = \{r_1, r_2, \dots, r_K\}$  – множество мобильных объектов.

В связи с завозом новой партии грузов требуется переместить некоторое подмножество грузов  $G_m \subseteq G$  в другие позиции, используя мобильные объекты из множества  $R$ . При этом все время выполнения этого задания не должно превышать предельного времени  $\tau$ , которое осталось до прибытия новых грузов. Если же переместить все грузы за время  $\tau$  невозможно, необходимо модифицировать задачу, чтобы задание было выполнено.

### 2. Разработка подходов к перемещению мобильных объектов по территории производства

Пусть территория производственного помещения представляет собой прямоугольное поле

размером  $N \times M$ , где  $N = 4k + 2$  и  $M = 3l + 1$  ( $k, l \in \mathbb{N}$ ).

Территория разбита на единичные клетки. В некоторых из этих клеток находятся грузы и мобильные объекты, которые могут перемещать грузы. При этом в одной клетке может находиться не более одного груза или мобильного объекта.

Определим координату клетки как пару натуральных чисел  $(x, y)$ , где  $x$  – номер клетки по счету сверху, а  $y$  – номер клетки по счету слева. Так, верхняя левая клетка имеет координаты  $(1, 1)$ , а правая нижняя – координаты  $(N, M)$ .

Пусть  $S$  – множество клеток территории помещения. Разобьем множество  $S$  на 2 класса – *дороги*  $D$  и *хранилища*  $H$  – так, что:

$$H = \{(x, y) \in S : ((x = 3k - 1) \vee (x = 3k)) \wedge ((y = 4l - 1) \vee (y = 4l)) | k, l \in \mathbb{N}\}, \quad (1)$$

$$D = \{(x, y) \in S : (x = 3k - 2) \wedge ((y = 4l - 3) \vee (y = 4l - 2)) | k, l \in \mathbb{N}\}. \quad (2)$$

Очевидно, что  $H \cap D = \emptyset$  и  $H \cup D = S$ .

На рис. 1 приведена схема территории производственного помещения при значениях  $k = 2$  и  $l = 3$  ( $N = M = 10$ ), где серым цветом обозначены клетки хранилищ.

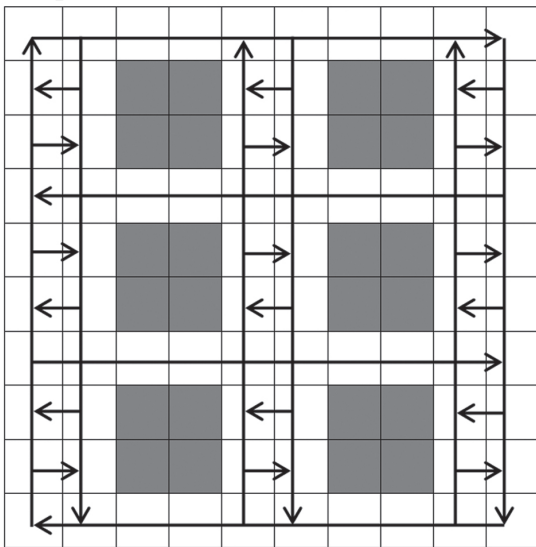


Рис. 1. Схема территории производственного помещения при  $N = M = 10$

Пусть выполняются следующие правила:

1) Груз, не перемещаемый мобильным объектом, может находиться только в хранилище.

2) Мобильный объект может находиться в клетке хранилища в том и только том случае, когда он забирает груз, расположенный в этой клетке. Во всех остальных случаях грузы перемещаются только по дорогам.

3) Мобильные объекты могут перемещаться по дорогам только в определенных направлениях. Пусть  $(x, y)$  – текущая координата объекта. Тогда:

– если  $y = 4k - 3$ , где  $k \in \mathbb{N}$ , то объекту запрещается двигаться вниз. Если при этом  $x$  нечетно, то объекту также запрещено двигаться вправо;

– если  $y = 4k - 2$ , где  $k \in \mathbb{N}$ , то объекту запрещается двигаться вверх. Если при этом  $x$  четно, то объекту также запрещено двигаться влево;

– если  $x = 6l - 5$ , где  $l \in \mathbb{N}$ , то объекту запрещается двигаться влево;

– если  $x = 6l - 2$ , где  $l \in \mathbb{N}$ , то объекту запрещается двигаться вправо.

Кроме того, мобильным объектам запрещается покидать территорию помещения и въезжать на территорию хранилища без погрузки.

На рис. 1 стрелочками показаны разрешенные направления движения мобильных объектов по дорогам.

4) Все объекты движутся с одинаковой скоростью независимо от того, перемещают они груз или нет. Время, за которое объект достигает соседней клетки, обозначим через  $t_0$ .

5) В случае, если два объекта одновременно хотят попасть в одну и ту же клетку, приоритет имеет тот, который движется в вертикальном направлении (вверх или вниз).

6) Время погрузки и разгрузки всех объектов одинаково и равно  $t_1$ , причем  $t_1$  делится нацело на  $t_0$ .

7) В каждый момент времени мобильный объект может перемещать не более одного груза.

8) Время от времени объекты ломаются. Если мобильный объект сломался, он не может двигаться и должен ждать на одном месте, пока его отремонтируют. Время ремонта всех объектов одинаково и равно  $t_r$ , причем  $t_r$  делится нацело на  $t_0$ .

9) Если мобильный объект сломался между двумя клетками, перед началом ремонта его передвигают в центр ближайшей клетки. В таком случае будем говорить, что объект сломался *в окрестности* этой клетки. При этом время передвижения сломанного объекта включено во время его ремонта.

10) Расстояние, которое проходит объект между двумя последовательными поломками, – случайная величина  $\omega$ , имеющая экспоненциальное распределение с параметром  $\lambda$ , одинаковым для всех объектов:  $\omega \sim \text{Exp}(\lambda)$ . Предполагаем также, что изначально все объекты отремонтированы.

11) В начальный момент времени все объекты отремонтированы и после ремонта еще не передвигались.

12) Вероятность  $P(\omega \leq t_0)$  того, что интервал между двумя поломками не превысит  $t_0$ , пренебрежимо мала.

### 3. Разработка модели

Пусть на территории помещения находится  $K$  мобильных объектов  $r_1, r_2, \dots, r_K$  и  $L$  грузов  $g_1, g_2, \dots, g_L$ . Обозначим начальные координаты

$i$ -го объекта через  $(x_{r_i}^{(0)}, y_{r_i}^{(0)})$ , а начальные координаты  $j$ -го груза через  $(x_{g_j}^{(0)}, y_{g_j}^{(0)})$ . Будем предполагать, что начальное расположение мобильных объектов и грузов удовлетворяет правилам:

$$(x_{r_i}^{(0)}, y_{r_i}^{(0)}) \in D \quad \forall i = \overline{1, K}; \quad (3)$$

$$(x_{g_j}^{(0)}, y_{g_j}^{(0)}) \in H \quad \forall j = \overline{1, L}. \quad (4)$$

Разобьем множество объектов  $R = \{r_1, r_2, \dots, r_K\}$  и множество грузов  $G = \{g_1, g_2, \dots, g_L\}$  на  $T$  непересекающихся классов:

$$R = R_1 \cup R_2 \cup \dots \cup R_T; \quad R_i \cap R_j = \emptyset \quad \forall i, j: i \neq j, \quad (5)$$

$$G = G_1 \cup G_2 \cup \dots \cup G_T; \quad G_i \cap G_j = \emptyset \quad \forall i, j: i \neq j. \quad (6)$$

Будем считать, что мобильный объект  $r_i$ , принадлежащий классу  $R_q$ , может перемещать груз  $g_j$  тогда и только тогда, когда  $g_j \in G_q$ , то есть объект и груз принадлежат одному и тому же классу. При этом гарантируется, что для каждого груза найдется мобильный объект, который может его перемещать.

Согласно заданию, требуется переместить  $|G_m| = P \leq L$  грузов  $g_1^m, g_2^m, \dots, g_P^m$  из начальных позиций

$$\begin{aligned} & (x_{g_1^m}^{(0)}, y_{g_1^m}^{(0)}), (x_{g_2^m}^{(0)}, y_{g_2^m}^{(0)}), \dots, (x_{g_P^m}^{(0)}, y_{g_P^m}^{(0)}) \text{ в позиции} \\ & (x_{g_1^m}^{(e)}, y_{g_1^m}^{(e)}), (x_{g_2^m}^{(e)}, y_{g_2^m}^{(e)}), \dots, (x_{g_P^m}^{(e)}, y_{g_P^m}^{(e)}) \text{ соответственно,} \end{aligned}$$

так чтобы общее время выполнения задания было минимально. При этом гарантируется, что все  $(x_{g_i^m}^{(e)}, y_{g_i^m}^{(e)})$  различны и не заняты «неподвижными» грузами (то есть теми, которые перемещать не требуется).

Чтобы формализовать эту задачу, сначала введем несколько определений.

**Определение 1.** *Заданием* мобильного объекта  $r_i$  назовем упорядоченное множество грузов  $G_{r_i} = \{g_i^1, g_i^2, \dots, g_i^k\} \subseteq G_m$ , для которого выполняются следующие условия:

1) Грузы перечислены в том порядке, в котором их нужно переместить на конечные позиции;

2) Классы всех грузов совпадают с классом мобильного объекта:

$$r_i \in R_q \Rightarrow (\forall g_j \in G_{r_i} : g_j \in G_q). \quad (7)$$

3) Каждый подвижный груз принадлежит ровно одному заданию:

$$\forall i, j = \overline{1, K} : (i \neq j \Rightarrow G_{r_i} \cap G_{r_j} = \emptyset); \quad (8)$$

$$\bigcup_{i=1}^K G_{r_i} = G_m. \quad (9)$$

Заметим, что для некоторых мобильных объектов задание может быть пустым.

**Определение 2.** *Распределением заданий между мобильными объектами*  $z(G_m)$  назовем такое разбиение множества  $G_m$  на непересекающиеся упорядоченные множества  $G_{r_1}, G_{r_2}, \dots, G_{r_K}$ , что  $G_{r_i}$  является заданием объекта  $r_i$  для всех  $i = \overline{1, K}$ .

Множество различных распределений заданий между мобильными объектами обозначим через  $Z(G_m)$ .

**Определение 3.** *Траекторией* мобильного объекта  $r_i$  согласно заданию  $G_{r_i}$  назовем вектор-функцию  $tr(G_{r_i}) = \{(x_i^{(0)}, y_i^{(0)}), (x_i^{(1)}, y_i^{(1)}), \dots, (x_i^{(e)}, y_i^{(e)})\}$ , удовлетворяющую следующим условиям:

1) Если  $G_{r_i} = \emptyset$ , то  $tr(G_{r_i}) = \{(x_{r_i}^{(0)}, y_{r_i}^{(0)})\}$ .

2) Если  $G_{r_i} \neq \emptyset$ , то траектория объекта содержит последовательность начальных и конечных координат грузов из его задания в порядке их перечисления:

$$\begin{aligned} & \{(x_{g_i^1}^{(0)}, y_{g_i^1}^{(0)}), (x_{g_i^1}^{(e)}, y_{g_i^1}^{(e)}), (x_{g_i^2}^{(0)}, y_{g_i^2}^{(0)}), (x_{g_i^2}^{(e)}, y_{g_i^2}^{(e)}), \dots, \\ & (x_{g_i^k}^{(0)}, y_{g_i^k}^{(0)}), (x_{g_i^k}^{(e)}, y_{g_i^k}^{(e)})\} \subseteq tr(G_{r_i}), \end{aligned} \quad (10)$$

причем  $(x_i^{(0)}, y_i^{(0)}) = (x_{r_i}^{(0)}, y_{r_i}^{(0)})$  и  $(x_i^{(e)}, y_i^{(e)}) = (x_{g_i^k}^{(e)}, y_{g_i^k}^{(e)})$ .

3) Расстояние между любыми двумя соседними координатами из  $tr(G_{r_i})$  равно единице:

$$|x_i^{(j)} - x_i^{(j-1)}| + |y_i^{(j)} - y_i^{(j-1)}| = 1 \quad \forall j = \overline{1, e}. \quad (11)$$

4) Все координаты из траектории, за исключением начальных и конечных координат грузов, принадлежат множеству дорог  $D$ .

5) При движении по траектории мобильный объект никогда не движется в запрещенном направлении.

Множество всех возможных траекторий  $tr(G_{r_i})$  мобильного объекта  $r_i$  согласно заданию  $G_{r_i}$  обозначим через  $TR(G_{r_i})$ .

**Определение 4.** *Вектором поломок*

$$b(tr(G_{r_i})) = \{b_i^{(0)}, b_i^{(1)}, \dots, b_i^{(e)}\}$$

мобильного объекта  $r_i$ , который движется по траектории  $tr(G_{r_i})$  согласно заданию  $G_{r_i}$ , назовем вектор, обладающий следующими свойствами:

1) Размерность вектора  $b(tr(G_{r_i}))$  совпадает с длиной траектории  $tr(G_{r_i})$  мобильного объекта  $r_i$ :

$$|b(tr(G_{r_i}))| = |tr(G_{r_i})|. \quad (12)$$

2)  $b_i^{(j)} \in \{0, 1\}$ , причем  $b_i^{(j)} = 1$  в том и только том случае, когда объект  $r_i$  сломался в окрестности клетки  $(x_i^{(j)}, y_i^{(j)})$ .

Множество возможных векторов поломок объекта  $r_i$ , движущегося по траектории  $tr(G_{r_i})$  согласно заданию  $G_{r_i}$ , обозначим через  $B(tr(G_{r_i}))$ .

**Определение 5.** Путем без ожиданий мобильного объекта  $r_i$  с вектором поломок  $b(tr(G_{r_i}))$  назовем вектор  $w_{n_i} = w_n(b(tr(G_{r_i})))$ , который можно получить по следующему алгоритму:

- 1) Изначально вектор  $w_{n_i}$  пуст, индекс  $j=1$ .
- 2) Добавляем в вектор  $w_{n_i}$  координату  $(x_i^{(j)}, y_i^{(j)})$ .
- 3) Если  $j$ -й элемент вектора поломок  $b(tr(G_{r_i}))$  равен 1, добавляем в вектор  $w_{n_i}$  координату  $(x_i^{(j)}, y_i^{(j)})$  еще  $t_r$  раз.
- 4) Если  $(x_i^{(j)}, y_i^{(j)}) \in H$ , добавляем в вектор  $w_{n_i}$  координату  $(x_i^{(j)}, y_i^{(j)})$  еще  $t_i$  раз.
- 5) Если  $j=e$ , прекращаем выполнение алгоритма, иначе увеличиваем  $j$  на единицу и переходим к пункту 2.

**Определение 6.** Пусть для всех мобильных объектов  $r_1, r_2, \dots, r_K$  известны их пути без ожиданий  $w_{n_1}, w_{n_2}, \dots, w_{n_K}$ . Пути с ожиданиями  $w_{s_1}, w_{s_2}, \dots, w_{s_K}$  объектов  $r_1, r_2, \dots, r_K$  соответственно назовем совокупность векторов, которую можно получить согласно следующему алгоритму:

- 1) Изначально векторы  $w_{s_1}, w_{s_2}, \dots, w_{s_K}$  пусты, индексы  $j_1 = j_2 = \dots = j_K = 1$ .
- 2) Полагаем индекс  $i=1$ , вводим переменные  $f_1 = f_2 = \dots = f_K = 0$  и выполняем следующие шаги:
  - 2.1. Если  $i > K$ , переходим к шагу 3.
  - 2.2. Сравниваем элемент  $w_{n_i}[j_i]$  со всеми элементами  $w_{n_p}[j_p]$ , где  $p = \overline{1, K}$ ,  $p \neq i$ :
    - если он совпал с каким-либо из этих элементов и  $x$ -координаты (абсциссы) элементов  $w_{n_i}[j_i]$  и  $w_{n_p}[j_p]$  совпадают, добавляем в вектор  $w_{s_i}$  элемент  $w_{n_i}[j_i]$ ;
    - иначе добавляем в вектор  $w_{s_i}$  элемент  $w_{n_i}[j_i]$  и присваиваем  $f_i = 1$ .
  - 2.3. Увеличиваем значение  $i$  на единицу и переходим к подпункту 2.1.
- 3) Если  $j_i = |w_{n_i}|$  для всех  $i = \overline{1, K}$ , то мы уже сформировали все пути с ожиданиями и алгоритм завершается. В противном случае изменяем значения индексов  $j_1, j_2, \dots, j_K$ : если  $j_i \neq |w_{n_i}|$ , присваиваем  $j_i = j_i + f_i$ , иначе  $j_i$  остается прежним. Затем переходим к шагу 2.

Заметим, что поскольку векторы поломок случайны, мы не можем заранее посчитать время выполнения задания. В связи с этим предлагается ввести *пороговую вероятность выполнения задания*  $\gamma$ . При этом задание будет считаться выполнимым тогда и только тогда, когда существуют такие траектории мобильных объектов, что объекты, двигаясь по этим траекториям, справятся с заданием за время, не превышающее  $\tau$ , с вероятностью не менее  $\gamma$ . Тогда формальная модель будет иметь вид:

$$P\left(\max_{i=\overline{1, K}} |w_{s_i}(w_{n_1}, w_{n_2}, \dots, w_{n_K})| \leq \tau\right) \rightarrow \max_{(tr_1, tr_2, \dots, tr_K) \in \Omega}, \quad (13)$$

где  $\Omega = \left\{ (tr_1, \dots, tr_K) : P\left(\max_{i=\overline{1, K}} |w_{s_i}(w_{n_1}, \dots, w_{n_K})| \leq \tau\right) > \gamma \right\}$ ;  
 $w_{n_i} = w_n(b(tr(G_{r_i})))$  – путь без ожиданий  $i$ -го мобильного объекта;  $b(tr(G_{r_i})) \in B(tr(G_{r_i}))$  – случайный вектор поломок  $i$ -го объекта;  $tr(G_{r_i}) \in TR(G_{r_i})$  – траектория  $i$ -го объекта;  $G_{r_i}$  – задание  $i$ -го объекта, полученное при разбиении на задания  $z: G_m \rightarrow \{G_{r_1}, G_{r_2}, \dots, G_{r_K}\}$ , где  $z \in Z$ .

#### 4. Разработка методов решения задачи

Решение задачи (13) можно разбить на 3 этапа:

*1 этап* – нахождение субоптимального разбиения на задания  $z^*: G_m \rightarrow \{G_{r_1}^*, G_{r_2}^*, \dots, G_{r_K}^*\}$ ;

*2 этап* – нахождение субоптимальных траекторий  $(tr_1^*(G_{r_1}^*), tr_2^*(G_{r_2}^*), \dots, tr_K^*(G_{r_K}^*))$  согласно заданиям  $G_{r_1}^*, G_{r_2}^*, \dots, G_{r_K}^*$  соответственно;

*3 этап* – проверка того, что мобильные объекты, двигаясь по субоптимальным траекториям, выполняя задание на время  $\tau$  с вероятностью не менее  $\gamma$ .

Рассмотрим каждый из этапов подробнее.

4.1. Нахождение субоптимального разбиения на задания

*Утверждение 1.* Мобильные объекты и грузы каждого из  $T$  классов можно рассматривать независимо друг от друга.

Действительно, поскольку мобильные объекты могут перемещать только грузы того же класса, данную задачу можно рассматривать как совокупность независимых задач нахождения субоптимальных разбиений на задания  $z_q^*: G_m^q \rightarrow \{G_{r_1^q}^*, \dots, G_{r_u^q}^*\}$ , где  $G_m^q$  – множество подвижных грузов класса  $q$ , а  $r_1^q, \dots, r_u^q$  – мобильные объекты класса  $q$ .

Таким образом, субоптимальное разбиение на задания  $z^* = \bigcup_{q=1}^T z_q^*$ , и для его нахождения необходимо распределить множества  $G_m^1, G_m^2, \dots, G_m^T$  подвижных грузов между мобильными объектами из множеств  $R_1, R_2, \dots, R_T$  соответственно, так чтобы задание каждого объекта соответствовало его субоптимальной траектории.

Для нахождения *оптимального* разбиения на задания необходимо:

- 1) Перебрать все возможные разбиения на задания между мобильными объектами одного класса;
- 2) Для каждого из разбиений найти оптимальную траекторию и затем выбрать из них лучшую.

*Утверждение 2.* Перебор всех разбиений на задания имеет экспоненциальную сложность.

Действительно, пусть в некотором разбиении на задания мобильному объекту  $r_i^q$  класса  $q$

соответствуют грузы  $g_1^q, \dots, g_p^q$ . Тогда перебор всех разбиений на задания должен включать перебор всех перестановок грузов  $g_1^q, \dots, g_p^q$  (поскольку под заданием понимается упорядоченное множество грузов), то есть  $p!$  вариантов. Сложность этого перебора уже экспоненциальна [4].

Таким образом, уже при небольших значениях количества мобильных объектов  $K$ , количества грузов  $L$  и количества классов  $T$  оптимальное разбиение на задания не может быть найдено за разумное время.

Метод, предлагаемый авторами статьи, позволяет находить субоптимальное разбиение на задания за полиномиальное время. Прежде чем описать этот метод, следует ввести несколько утверждений.

*Утверждение 3.* С увеличением длин траекторий  $tr(G_{r_i})$  мобильных объектов  $r_i (i = \overline{1, K})$  ожидаемое время выполнения задания увеличивается.

Действительно, чем длиннее траектория мобильного объекта, тем большее расстояние он проходит, и, следовательно, тем больше ожидаемое количество его поломок в пути. Кроме того, с увеличением длины траектории увеличивается также время мобильного объекта в пути.

Таким образом, разбиение на задания  $z: G_m \rightarrow \{G_{r_1}, \dots, G_{r_K}\}$  будет субоптимальным, если существуют такие траектории  $(tr_1(G_{r_1}), \dots, tr_K(G_{r_K}))$ , что  $W = \max_{i=\overline{1, K}} |tr_i(G_{r_i})|$  минимально среди всех  $z \in Z$ .

*Утверждение 4.* Длина траектории  $tr(G_{r_i})$  мобильного объекта  $r_i$  согласно заданию  $G_{r_i} = \{g_i^1, \dots, g_i^k\}$  равна сумме длин траекторий  $tr(g_i^j), j = \overline{1, K}$ :

$$|tr(G_{r_i})| = \sum_{j=1}^K |tr(g_i^j)|, \quad (14)$$

где  $|tr(g_i^j)|$  – длина траектории из клетки с координатами  $(x_{r_i}^{(j)}, y_{r_i}^{(j)})$  до клетки  $(x_{g_i^j}^{(e)}, y_{g_i^j}^{(e)})$  через клетку  $(x_{g_i^j}^{(0)}, y_{g_i^j}^{(0)})$ ;  $(x_{r_i}^{(1)}, y_{r_i}^{(1)}) = (x_{r_i}^{(0)}, y_{r_i}^{(0)})$  и  $(x_{r_i}^{(j)}, y_{r_i}^{(j)}) = (x_{g_i^{j-1}}^{(e)}, y_{g_i^{j-1}}^{(e)})$  при  $j > 1$ .

Доказательство утверждения следует из непрерывности траектории мобильного объекта.

*Утверждение 5.* Длина кратчайшей траектории между любыми двумя клетками не меняется со временем и может быть вычислена за константное время.

Действительно, поскольку территория производственного помещения и правила движения по этой территории не изменяются со временем, длина кратчайшей траектории также остается неизменной. Заметим, что территория состоит из повторяющихся фрагментов размером  $6 \times 8$  клеток,

содержащих 2 вертикальных и 3 горизонтальных участка дороги (рис. 2). Клетки, в которых пересекаются вертикальная и горизонтальная дорога, назовем *перекрестками* (на рис. 2 перекрестки заштрихованы).

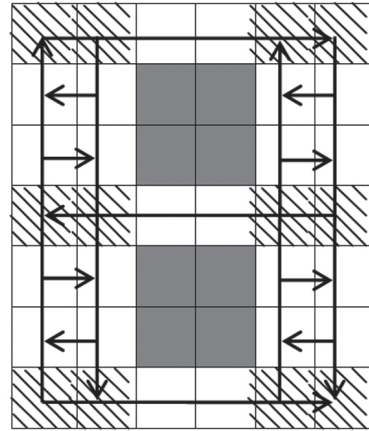


Рис. 2. Повторяющийся фрагмент территории производственного помещения

Расстояние между всеми парами клеток, принадлежащих одному фрагменту, можно найти заранее с помощью алгоритма Флойда-Уоршелла [5] за асимптотику  $O(V^3)$ , где  $V$  – количество вершин в графе [5]. В данном случае  $V = 48$  – количество клеток во фрагменте. Если же клетки принадлежат разным фрагментам, путь между ними можно разбить на 3 этапа: путь от 1-й клетки до 1-го перекрестка, путь между перекрестками и путь от последнего перекрестка до 2-й клетки. На 1-м и на 3-м этапах пути клетки, между которыми ищется расстояние, находятся в одном фрагменте. Длину же кратчайшей траектории между перекрестками можно найти за константное время, заметив, что расстояние от любой пары соседних перекрестков до любой другой пары равно манхэттенскому расстоянию между перекрестками из этих пар. Таким образом, заранее просчитав расстояния между всеми парами точек одного фрагмента, кратчайшее расстояние от любой клетки помещения до любой другой можно найти за константное время.

Теперь опишем этапы метода нахождения субоптимального разбиения на задания.

*1 этап.* Инициализируем  $q = 1$  – текущий класс мобильных объектов, для которых будет найдено субоптимальное разбиение на задания  $z_q$ . Вводим в рассмотрение массив векторов  $mas_z$ , где  $mas_z[i]$  – текущее задание мобильного объекта  $r_i$  (то есть последовательность индексов грузов в том порядке, в котором их нужно забрать этому объекту), и массив целых чисел  $state$ , где  $state[i] = 0$ , если мобильный объект  $r_i$  в данный момент не выполняет задание, и  $state[i] = 1$ , если выполняет. Вводим также массив  $M_r$ , такой что  $M_r[i]$  – минимальное время, за которое мобильный объект  $r_i$  может завершить задание  $mas_z[i]$ , и список  $S_{g_m}$ ,

содержащий индексы всех подвижных грузов, которые еще не были перемещены. Изначально массив  $mas_z$  пуст, массивы  $state$  и  $M_r$  заполнены нулями, а список  $S_{g_m}$  содержит индексы всех грузов из  $G_m$ . Пусть  $curtime$  – время, прошедшее с начала выполнения заданий мобильными объектами. Оно также изначально равно 0.

**2 этап.** Если  $g_{m_i} \notin G_q \forall j \in S_{g_m}$ , увеличиваем  $q$  на единицу. Далее для всех мобильных объектов  $r_i \in R_q$ , выполняющих задание, находим длину кратчайшей траектории  $l_i$ , по которой должен двигаться этот мобильный объект для завершения текущего задания (то есть для перемещения текущего объекта в конечную точку). Для всех «свободных» мобильных объектов  $r_j \in R_q$  находим длину кратчайшей траектории  $p_j$ , по которой должен двигаться этот мобильный объект для выполнения какого-либо задания (то есть наименьшее расстояние, которое нужно пройти объекту, чтобы переместить в конечную точку один из незадействованных до сих пор грузов). Пусть  $\bar{p} = \min_j p_j$ .

**3 этап.** Строим двудольный граф, вершинами левой доли которого являются мобильные объекты  $r_i \in R_q$ , для которых выполняется либо  $state[i]=0$ , либо  $state[i]=1$  и  $l_i < \bar{p}$ , а вершинами правой доли – все грузы класса  $q$  из списка  $S_{g_m}$ . Длина ребра между  $i$ -й вершиной левой доли и  $j$ -й вершиной правой доли вычисляется как длина кратчайшей траектории, по которой должен двигаться мобильный объект  $r_i$  для того чтобы переместить груз  $g_{m_j}$  в его конечную точку. При этом если  $state[i]=1$ , в эту траекторию также включается траектория  $l_i$  (то есть мобильный объект сначала должен завершить текущее задание, а потом начинать следующее).

**4 этап.** С помощью венгерского алгоритма решения задачи о назначениях [6] находим такое распределение грузов между мобильными объектами  $(r_{a_1} \rightarrow g_{b_1}, r_{a_2} \rightarrow g_{b_2}, \dots, r_{a_i} \rightarrow g_{b_i})$ , что сумма кратчайших траекторий, необходимых этим объектам для перемещения соответствующих объектов в необходимые клетки, минимальна по всем возможным распределениям. Добавляем в конец каждого из векторов  $mas_z[a_i]$  соответствующий индекс  $b_i$ , присваиваем  $state[a_i]=1$ , удаляем индексы  $b_1, b_2, \dots, b_i$  из списка  $S_{g_m}$  и прибавляем к каждому из элементов  $M_r[a_i]$  значения  $t(r_{a_i}, g_{b_i})$ , где  $t(r_i, g_j)$  – минимальное время, необходимое мобильному объекту  $r_i$  для перемещения груза  $g_j$  после окончания перемещения предыдущего груза из его задания.

**5 этап.** Если список  $S_{g_m}$  пуст, завершаем работу метода. Иначе присваиваем  $curtime = \min_{i: r_i \in R_q} M_r[i]$ , а затем для тех мобильных объектов  $r_i$ , у которых  $M_r[i] = curtime$ , присваиваем  $state[i]=0$  и переходим ко 2-му этапу.

В результате работы данного метода массив векторов  $mas_z$  будет содержать субоптимальное разбиение на задания. Оценим сложность работы данного метода. Легко показать, что наибольшую асимптотику работы имеет венгерский алгоритм, который выполняется на 4-м этапе, так что выполнение остальных этапов метода не влияет на общую сложность. Рассмотрим  $j$ -ю итерацию метода для класса  $q$ . Пусть в двудольном графе, построенном на 3-м этапе, количество вершин в левой доле равно  $K_{qj}$ , а в правой –  $L_{qj}$ . Случай  $K_{qj} \geq L_{qj}$  возможен только на последней итерации для данного класса, так как в этом случае каждый из оставшихся грузов будет перемещен каким-то мобильным объектом; при этом сложность работы  $j$ -й итерации 2-го этапа метода, согласно асимптотике венгерского алгоритма, будет равна  $O(K_{qj} \cdot L_{qj}^2)$ . Поскольку  $\sum_{q=1}^T K_{qj} \cdot L_{qj}^2 \leq \sum_{q=1}^T K_q \cdot L_q^2 \leq \sum_{q=1}^T K_q \cdot \left( \sum_{q=1}^T L_q \right)^2 = KL^2$ , где  $K_q$  и  $L_q$  – соответственно количество мобильных объектов и грузов класса  $q$ , то сложность работы этой части метода равна  $O(KL^2)$ .

Во всех остальных случаях  $K_{qj} < L_{qj}$ , поэтому сложность  $j$ -й итерации будет равна  $O(K_{qj}^2 \cdot L_{qj})$ . Заметим, что поскольку на каждой итерации перемещенные грузы удаляются из списка, то  $\sum_{j=1}^{d_q} L_{qj} \leq L_q$ , где  $d_q$  – количество итераций для класса  $q$ , для которых  $K_{qj} < L_{qj}$ . Так как

$$\sum_{j=1}^{d_q} K_{qj}^2 \cdot L_{qj} < \sum_{j=1}^{d_q} L_{qj}^3 \leq \left( \sum_{j=1}^{d_q} L_{qj} \right)^3 \leq L_q^3 \quad (15)$$

и

$$\sum_{q=1}^T L_q^3 \leq \left( \sum_{q=1}^T L_q \right)^3 = L^3, \quad (16)$$

то сложность выполнения всех итераций метода, для которых  $K_{qj} < L_{qj}$ , равна  $O(L^3)$ , и общая сложность метода равна  $O(KL^2 + L^3) = O(\max(K, L)^3)$ .

#### 4.2. Нахождение субоптимальных траекторий

Итак, в результате 1-го этапа работы метода для каждого мобильного объекта  $r_i$  был получен список  $S = \{(x_{1i}, y_{1i}), (x_{2i}, y_{2i}), \dots, (x_{qi}, y_{qi})\}$  клеток, которые он должен посетить. Согласно утверждениям 3 и 4, чтобы траектория была субоптимальной, объект  $r_i$  между любыми двумя соседними клетками из списка  $S$  должен идти по кратчайшему пути.

Таким образом, нахождение субоптимальных траекторий сводится к нахождению одного из кратчайших путей между двумя произвольными клетками  $(x_1, y_1)$  и  $(x_2, y_2)$ .

Построим ориентированный граф, в котором вершинами будут клетки, а ребрами – разрешенные переходы между соседними клетками. Тогда кратчайший путь между любыми двумя клетками

можно найти с помощью алгоритма поиска в ширину с запоминанием пути. Применяя этот алгоритм для каждого мобильного объекта и для всех участков пути, получим траектории, которые будут кратчайшими и, следовательно, субоптимальными.

Оценим сложность алгоритма. Очевидно, количество вершин в графе будет равно  $V = NM$ . Поскольку из каждой вершины выходит не более 4 ребер, то количество ребер  $E \leq 4V = 4NM$ . Сложность алгоритма поиска в ширину –  $O(E)$  [5], в данном случае –  $O(NM)$ . Заметим, что каждые два вызова алгоритма соответствуют перемещению одного груза в нужную клетку; соответственно, общее количество вызовов алгоритма –  $O(L)$ . Итого общая сложность нахождения траекторий для всех мобильных объектов –  $O(NML)$ .

#### 4.3. Проверка выполнимости задания.

После того, как были получены траектории, по которым должны двигаться мобильные объекты, следует проверить, удовлетворяют ли они ограничениям задачи. Поскольку перебрать все возможные варианты движения и полочки мобильных объектов за разумное время не представляется возможным, вероятность того, что задание будет выполнено, можно оценить с помощью метода Монте-Карло [7].

Сначала выберем число испытаний  $Q$ . С одной стороны, оно должно быть весьма велико, чтобы оценка вероятности была достаточно точной, а с другой – слишком большое значение  $Q$  приведет к тому, что проверка будет выполняться очень долго. Рекомендуется выбрать  $Q$  в пределах от 1000 до 10000.

Теперь необходимо  $Q$  раз промоделировать процесс движения мобильных объектов по траекториям согласно правилам, рассмотренным в предыдущем разделе. При этом вместо случайной величины  $\omega$  интервала между полочками для каждого мобильного объекта генерируются ее реализации, а роль перемещения в ближайшую клетку при полочке играет округление текущих координат мобильного объекта до ближайшего целого.

Когда движение мобильных объектов по траекториям окончено, необходимо сравнить время  $T$ , которое прошло с момента начала выполнения задания, с требуемым временем  $\tau$ . Если  $T \leq \tau$ , задание успешно выполнено и количество успешных испытаний необходимо увеличить на единицу; иначе задание считается невыполненным.

Пусть в результате моделирования процесса из  $Q$  испытаний успехом окончилось  $U$ . Тогда если  $\frac{U}{Q} \geq \gamma$ , то задача решена – мобильные объекты, двигаясь по траекториям, найденным на предыдущем этапе, с вероятностью  $\gamma$  выполнят задание вовремя. Если же  $\frac{U}{Q} < \gamma$ , считаем, что задача не

может быть решена при заданных ограничениях, и сообщаем, что задачу следует модифицировать.

Оценим сложность выполнения данного этапа. Заметим, что длина траектории мобильного объекта между двумя клетками, которые ему необходимо посетить, не превышает  $N + M$ . Тогда длина пути, который необходимо пройти, чтобы переместить груз в нужную клетку, не превышает  $2(N + M)$ , и суммарная длина всех траекторий не больше  $2L(N + M)$ . Таким образом, при достаточно высокой надежности мобильных объектов (т.е. количество их поломок в пути невелико) и относительно небольших значениях отношений  $t_l/t_0$  и  $t_r/t_0$ , моделирование одного испытания имеет сложность  $O(L(N + M)) = O(L \cdot \max(N, M))$ , а моделирование всех испытаний – соответственно  $O(QL \cdot \max(N, M))$ .

Таким образом, общая сложность метода субоптимальной маршрутизации мобильных объектов равна  $O(\max(K, L)^3 + NML + QL \cdot \max(M, N))$ .

### 5. Программная реализация метода

Описанный метод программно реализован на языке Java; в среде IntelliJ Idea разработано графическое приложение, иллюстрирующее движение мобильных объектов по найденным траекториям.

UML-диаграмма классов приложения приведена на рис. 3. Опишем подробнее каждый из приведенных классов.

*InputParametersFrame* – класс диалогового окна, в которое вводятся входные параметры модели: размеры поля  $N$  и  $M$ , количество мобильных объектов  $K$ , их координаты и классы; количество грузов, их координаты и классы; новые координаты грузов, которые необходимо переместить; максимальное время выполнения задания  $T$ , пороговая вероятность  $\gamma$  и параметр показательного распределения  $\lambda$ . Метод *CheckParameters* проверяет введенные параметры на корректность.

*ParametersContainer* – класс, в котором хранятся входные параметры, введенные пользователем. К этим данным предоставляется доступ другим классам.

*FieldFrame* – диалоговое окно, содержащее кнопки меню «Выход», «Изменить параметры» и «Пауза/продолжить», позволяющие управлять соответственно: завершить моделирование перемещения мобильных объектов и выйти из программы; завершить моделирование и изменить входные параметры; прервать или возобновить моделирование.

*FieldPanel* – поле, расположенное внутри диалогового окна *FieldFrame*, в котором моделируется перемещение мобильных объектов по территории производственного помещения.

*GeneralObject* – обобщенный класс, содержащий поля и методы, общие для мобильного объекта и

груза – текущие координаты, цвет и размер на экране, номер класса, а также метод draw(), рисующий данный объект на экране, и метод move(), перемещающий его.

*MobileObject* – класс мобильного объекта, который наследует класс *GeneralObject*. В нем перегружаются методы draw() и move(), а также вводятся такие поля, как траектория мобильного объекта и его состояние (движется, загружается, разгружается, ожидает или сломался).

*Load* – класс груза, который также наследует класс *GeneralObject*. В нем также перегружены методы draw() и move(), и, кроме того, введены поля, показывающие, является ли данный груз подвижным и перемещается ли он в данный момент.

*TrajectoryGenerator* – класс, в котором реализован вышеописанный метод нахождения субоптимальных траекторий мобильных объектов. В методе findDistance определяется кратчайший путь между двумя клетками, как описано в утверждении 5; методы getSchedule, getTrajectories и checkTrajectories реализуются соответственно: 1-й, 2-й и 3-й этапы метода – получение разбиения на задания, получение траекторий мобильных объектов и проверка полученных траекторий. В результате, если решение существует, инициализируются поля trajectory всех мобильных объектов и внутри поля FieldPanel моделируется их перемещение согласно полученным траекториям. Если же решения не существует, выводится сообщение об этом и пользователю предлагается модифицировать входные параметры.

Поскольку предложенный метод имеет кубическую сложность, на входные параметры были введены следующие ограничения: длина и ширина поля  $N, M \leq 100$ ; количество мобильных объектов и грузов  $K, L \leq 200$ .

Программная реализация метода подтвердила его эффективность.

### Выводы

Усовершенствована модель перемещения группы сложных объектов по территории производственного помещения, которая, в отличие от существующих, минимизирует взаимодействие этих объектов, что позволяет повысить качество управления.

Получил дальнейшее развитие метод маршрутизации группы сложных объектов, который, в отличие от существующих, находит их субоптимальные траектории за время, полиномиальное относительно входных параметров, что позволяет минимизировать вычислительную сложность системы.

Усовершенствован венгерский алгоритм решения задачи о назначениях для случая, когда каждый процесс может выполнять несколько видов работ. При этом полученное решение может оказаться неоптимальным, однако асимптотика метода остается кубической.

Программно реализован метод субоптимальной маршрутизации сложных объектов, который работает за кубическую асимптотику.

Перспективным направлением дальнейших исследований является использование нечеткой логики и знание-ориентированных технологий.

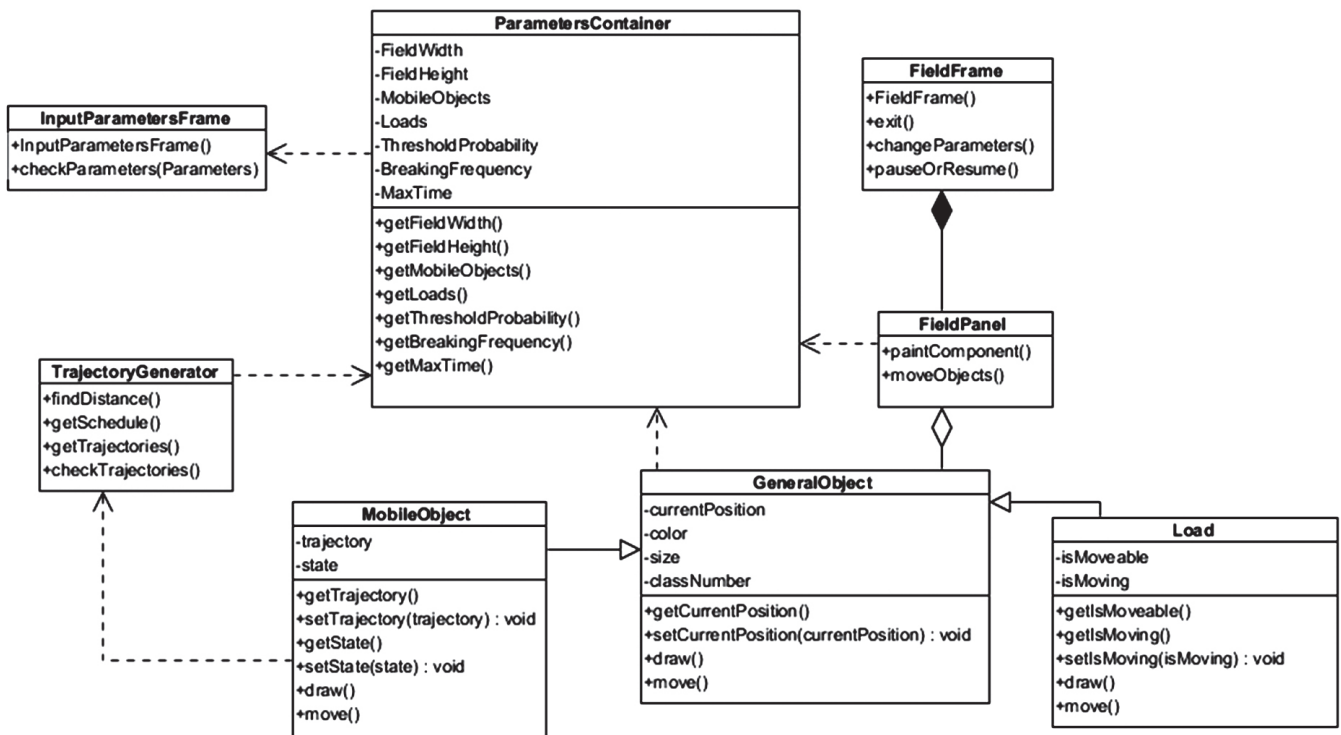


Рис. 3. UML-диаграмма классов приложения

**Список литературы:** 1. Управление ГПС: Модели и алгоритмы [Текст] / Под общ. ред. акад. С. В. Емельянова. – М.: Машиностроение. 1987. – 368 с., ил. 2. Даринцев, О. В. Нейросетевой алгоритм планирования траектории для группы мобильных роботов [Текст] / О. В. Даринцев, А. Б. Мигранов, Б. С. Юдинцев // Искусственный интеллект. – 2011. – №1. – С. 154–160. 3. Транспортные роботы для гибких производственных систем [Текст] / Н. М. Довбня, А. А. Халфен, И. В. Яковлев. Л.: ЛДНТП, 1988 г. – 233 с.: ил. 4. Кузюрин, Н. Н. Эффективные алгоритмы и сложность вычислений [Текст]: учеб. пособие для вузов / Н. Н. Кузюрин, С. А. Фомин. – М., 2008. – 317 с. 5. Алгоритмы: построение и анализ [Текст]: пер. с англ. / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. – М.: Вильямс, 2006. – 1296 с. 6. Ahuja, R. K. Network flows: theory, algorithms, and applications. / R. K. Ahuja, T. L. Magnanti, J. B. Orlin. – New Jersey, Prentice-Hall, 1993. 7. Войтишек А. В. Основы метода Монте-Карло [Текст]: Учеб. пособие / А. В. Войтишек. – Новосибирск: Новосиб. Гос. ун-т, 2010. – 108 с.

*Поступила в редколлегию 26.10.2012*

УДК 004.93

**Стохастичні моделі і методи субоптимальної маршрутизації складних об'єктів** / А.Д. Дрюк, Є.І. Кучеренко // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 45-53.

У статті запропоновано розроблену авторами модель переміщення групи мобільних об'єктів згідно з заданими правилами. Для даної моделі розглянуто метод субоптимальної маршрутизації мобільних об'єктів. Стверджується, що цей метод працює за поліноміальний час і дозволяє знаходити рішення задачі переміщення грузів по виробничій території за фіксований час.

Л. 3. Бібліогр.: 7 найм.

UDK 004.93

**Stochastic models and methods of suboptimal complex objects routing** / O.D. Driuk, Ye.I. Kucherenko // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 45-53.

In article the model of moving of mobile objects according to definite rules developed by authors of article is proposed. For given model the method of suboptimal mobile objects routing is considered. This method is approved to work for polynomial time and to find the solution of problem of moving loads through production area for fixed period of time.

Fig. 3. Ref.: 7 items.

УДК 044.03



М.А. Кернос

ХНУРЭ, г. Харьков, Украина, iust@list.ru

## МАТЕМАТИЧЕСКАЯ МОДЕЛЬ КОНТЕНТА ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ СИСТЕМЫ

В работе рассмотрены элементы контента информационно-аналитической системы, проведен анализ существующих моделей и предложена математическая модель контента системы, объединяющая данные и электронные документы как информационные единицы контента.

КОНТЕНТ, CMS, ИАС, ФАКТОГРАФИЧЕСКИЕ ДАННЫЕ, ЭЛЕКТРОННЫЙ ДОКУМЕНТ, БАЗА ДАННЫХ, ХРАНИЛИЩЕ ДАННЫХ

### Введение

В настоящее время эффективность деятельности предприятия во многом определяется качеством и актуальностью информации, характеризующей его бизнес-процессы (БП), а также оперативностью её обработки. Поэтому одним из приоритетных направлений развития современных предприятий является внедрение в их БП современных информационно-аналитических систем (ИАС) и систем поддержки принятия решений.

Информация, связанная с БП предприятия, очень разнообразна. Она отличается по источникам, формам представления, периодичности возникновения, степени важности, способам хранения и т.д. Во многих случаях носителями информации являются электронные документы. Поэтому на многих предприятиях автоматизация БП предполагает автоматизацию формирования и обработки документов, которые наравне с данными в таблицах базы данных (БД) являются элементами контента ИАС. Уменьшение временных затрат на поиск данных и документов, требующихся сотрудникам, способствует принятию на их основе своевременных и оптимальных управленческих решений [1, 2].

### 1. Задача управления контентом

В 2001 году международной ассоциацией управления информацией и изображениями АИМ (The Association for Information and Image Management) была предложена концепция управления контентом ЕСМ (Enterprise Content Management), выдвигающая набор различных требований к управлению обработкой и распространением корпоративного содержимого или информационными ресурсами предприятия. Целью данной концепции является обеспечение своевременного выполнения системой запросов всех потребителей информации на предприятии и предоставление им актуальных данных о текущем состоянии его БП.

Согласно определению, данному АИМ, управление информационными ресурсами предприятия – это набор информационных технологий, инструментов и методов, применяемых для сбора,

управления, накопления, хранения и доставки информации всем потребителям внутри организации. Реализация данной концепции предполагает развитие ИАС предприятия в системы управления корпоративным содержимым, часть функций которых реализуют современные системы управления контентом CMS (Content Management System), ориентированные на среду web.

Однако ИАС данного класса ориентированы в первую очередь на публикацию информации, при этом автоматизация обработки данных, содержащихся в документах, затруднена вследствие слабой степени формализации документов и сложности её осуществления. В то же время документация является носителем оперативной информации, учитываемой в ИАС, поддерживает и сопровождает БП предприятия, а при формировании документов могут быть использованы данные, полученные из системы. Поэтому в БП многих предприятий электронные документы используются совместно с базами данных (БД) и являются неотъемлемой частью контента ИАС предприятия, а задача управления всеми элементами контента системы является актуальной [3,4].

### 2. Исследование существующих моделей контента

Для формализованного описания контента ИАС в настоящее время применяются модели, основанные на теории множеств и отношений, теории графов, теории конечных автоматов и другие математические модели.

Теоретико-множественные модели контента ИАС представлены тремя видами. Первый из них, применяемый в экономических системах, основан на положении о том, что каждая подсистема ИАС представляет собой модель определённой предметной области (ПрО), которая задаётся в форме понятийной модели. В данном случае на концептуальном уровне контент ИАС ( $IS$ ) представляется выражением следующего вида:

$$IS = \langle A, M, B \rangle,$$

где  $A$  – множество агентов (подсистем ИАС);  $M$  – множество моделей ПрО, зафиксированных в форме

информационного ресурса, каждая из которых соответствует одному агенту ИАС ( $A_i \leftrightarrow M_i, i = \overline{1, N}$ );  $N$  – количество агентов в ИАС;  $B$  – множество связей между агентами. Связь  $B_{ij}$  описывает характер информационного взаимодействия агентов  $A_i$  и  $A_j$  ( $i \neq j, i = \overline{1, N}, j = \overline{1, N}$ ) [5].

Второй вид теоретико-множественных моделей контента ИАС подразумевает его представление в виде многомерного пространства, измерениями которого являются фактографические данные, электронные документы и регламенты работы с ними.

Данные измерения выделены в соответствии с целями создания ИАС. Такие модели используются для экспертной оценки эффективности системы, а координаты в данном пространстве характеризуют уровень организации хранения фактографической информации, автоматизации работы с документами, внедрения единых регламентов работы с данными:

$$IS = \langle F, D, R \rangle,$$

где  $F$  – упорядоченное множество возможных значений числовой характеристики уровня организации хранения фактографической информации в ИАС;  $D$  – упорядоченное множество возможных значений числовой характеристики уровня организации работы с электронными документами в ИАС, их применение для обеспечения взаимодействия автоматизируемых БП;  $R$  – упорядоченное множество возможных значений числовой характеристики уровня регламентации процессов внесения, формирования и обработки данных и электронных документов, описания данных процедур.

Точка в данном пространстве определяет состояние системы. Положение данной точки зависит от уровня развития и стадии внедрения ИАС на предприятии, а также от специфики его БП.

Третий вид теоретико-множественных моделей основан на применении терминов многомерной модели данных, предложенной Э.Ф. Коддом для анализа больших массивов данных с целью поддержки принятия решений. Такие модели применяются в системах оперативного анализа данных OLAP (OnLine Analytical Processing) для анализа исторических данных. Следует отметить, что OLAP-модели, как правило, описывают фрагмент контента ИАС: из всех структур данных, образующих контент ИАС, рассматривается только фиксированное подмножество (документы, как правило, не учитываются).

Основной данной модели являются факты. Для выбора и агрегирования данных о фактах на требуемом уровне детализации используются измерения. Обобщением электронных таблиц являются гиперкубы данных. Набор соответствующих гиперкубов составляет многомерную БД или хранилище данных (ХД). Комбинации значений измерений определяют ячейки куба. В зависимости от

конкретного приложения ячейки в кубе могут располагаться как разрозненно, так и плотно. Кубы, как правило, становятся разрозненными по мере увеличения числа размерностей и степени детализации значений измерений. Алгебра многомерной модели данных включает операции среза (slice), вращения (rotation), консолидации и детализации (drill up/down) [6,7].

Большинство математических моделей OLAP-кубов является закрытым вследствие использования в коммерческих СУБД. Во многих случаях применение многомерной модели данных приводит к разделению оперативных (оперативные базы данных) и аналитических данных ИАС (хранилища данных) [8].

Графовые модели в большинстве случаев применяются в системах электронного документооборота и описывают формальную модель документооборота – в виде множеств его участников, операций над документами и их состояний. Вершинам графа ставятся в соответствие операции, а ребрам – действия над документами. Такие модели позволяют моделировать логику последовательного изменения состояний документов и действий, связанных с этими изменениями. Графовые модели также применяются для агрегации данных в OLAP-кубах (применяются графы измерений, позволяющие регулировать их детализацию) [9].

Автоматные модели также применяются для моделирования документооборота, а не всего контента ИАС, и представлены в виде нескольких множеств: состояний документов (начальных и заключительных), входных данных для системы документооборота (алфавит автомата) и функций переходов документов в новые состояния на основании их текущего состояния и входных данных. Моделирование документооборота с применением данных моделей предполагает принятие решений о переходе от одного состояния документа к другому [9].

Имитационные модели контента ИАС используются редко. Основным их недостатком является ориентация на специфику БП конкретного объекта управления (ОУ) и сложность адаптации к другим объектам. В то же время интенсивно развивается такое междисциплинарное научное направление, как синергетика, применяющая математический аппарат динамической теории информации, который может быть использован для создания имитационных моделей ИАС [2,9].

Таким образом, существующие математические модели контента ИАС фокусируются на его отдельных элементах, в связи с чем описание многообразного контента ИАС требует одновременного применения нескольких моделей. Вследствие этого разработка единой математической модели контента ИАС является актуальной.

### 3. Разработка модели контента системы

Для создания математической модели, описывающей несколько видов контента ИАС необходимо исследование совокупности составляющих его элементов, которыми являются данные и документы, представленные в виде структур данных, содержащих фактографическую информацию. В практическом смысле, применительно к ИАС, информация является объектом хранения, передачи и преобразования во время осуществления БП ОУ [10].

Данное утверждение близко к определению информации, данному К. Шенноном, – под информацией понимаются сигналы, полезные для получателя. При этом знаки или сигналы заключают в себе информацию не потому, что они повторяют реальные объекты, а по общественной договоренности об однозначной связи знаков и объектов (например, предметы связаны со словами, применяемыми для их обозначения). Такое определение позволяет рассматривать вопросы хранения и доставки информации как совокупности знаков или сигналов, т.е. в виде сообщений [11].

Для открытых систем, которыми являются современные ИАС, предпочтительно другое, также предложенное К. Шенноном, определение информации: информация выражается разностью безусловной и условной энтропий. Такие системы обладают свойством детерминированности: состояние системы в начальный момент времени определяет набор возможных состояний системы в следующий момент [1, 11].

Поскольку решение всех ФЗ рассматриваемого класса ИАС связано с использованием или модификацией данных, хранимых в БД или ХД, для систем такого класса применимо следующее определение информации: это запомненный выбор одного варианта или нескольких из многих возможных и равноправных. Равноправными считаются варианты, принадлежащие к одному множеству. Например, равноправными являются множества допустимых значений (домен) отдельного атрибута таблицы БД [1].

Рассмотренные термины, используемые при работе с элементами контента ИАС, позволяют сформулировать одну из основных целей разработки ИАС и реализации задач эксплуатации системы как поддержание актуальности данных о текущем состоянии автоматизируемых БП, сопровождаемых документооборотом, и ведение истории осуществления БП предприятия.

Актуальность информации определяет её ценность. Ценной может считаться информация, позволяющая сэкономить время или средства для достижения поставленной цели. Применительно к ИАС ценной является информация, которая может быть использована для поддержки БП, решения функциональных задач учета, анализа,

прогнозирования и принятия управленческих решений, т.е. ценной может быть только фактографическая информация [2].

Правильно спроектированная и нормализованная схема данных ИАС (в большинстве современных ИАС применяются объектно-реляционные СУБД), как правило, не содержит избыточных данных, которые не используются в системе. В ИАС может использоваться распределенная БД, которая допускает дублирование данных и документов в локальных БД компонентов такой системы (например, в случае реализации сервис-ориентированной архитектуры ИАС или создания ИАС на базе нескольких систем различных классов). Соответственно, на этапе разработки распределённой ИАС должны быть реализованы механизмы синхронизации данных между компонентами системы, управление которыми является одной из задач сопровождения ИАС с целью выполнения основного положения концепции ЕСМ: данные, добавленные в БД ИАС одним из модулей системы, должны быть доступны другим модулям. При этом в качестве контента распределённой ИАС рассматривается объединение множеств данных и документов её компонентов при условии, что осуществлена синхронизация данных в локальных БД:

$$C = \bigcup_{i=1}^n d(Cp_i),$$

где  $C$  – контент распределённой ИАС;  $d(Cp_i)$  – множество данных и документов компонента распределённой ИАС  $Cp_i$ ;  $n$  – количество компонентов распределённой ИАС.

Следовательно, возможно допущение о том, что информация в ИАС является фактографической, а её дублирование минимально.

Однако при реализации различных целей (решения различных ФЗ) не все фактографические данные системы обладают ценностью. Например, в ИАС управления вузом при решении задачи учета контингента студентов информация о семестровых оценках студентов не используется (достаточной является информация о том, сдал студент сессию или нет), поэтому её ценность для достижения данной цели равна нулю. С другой стороны, при учете успеваемости студентов (кондуита) эта информация становится ценной, т.к. её наличие обеспечивает достижение цели, например, формирование рейтинга студентов [10].

Совокупность фактографических данных ИАС применительно к любой её ФЗ представляется в следующем виде:

$$d = d_i^c \cup d_i^o, \forall i, i = \overline{1, n},$$

где  $d$  – множество данных в ИАС;  $d_i^c$  – множество ценных данных для достижения  $i$ -й цели;  $d_i^o$  – множество данных, не являющихся ценными для

достижения  $i$ -й цели;  $n$  – количество целей или функциональных задач ИАС.

Таким образом, контент ИАС образуется множеством фактографических данных о текущем состоянии БП и истории его изменений, принадлежащей различным модулям системы, ценность которых зависит от рассматриваемой цели системы (решаемой функциональной задачи) [10].

Информация, образующая контент ИАС, имеет иерархическую структуру. В большинстве случаев иерархия имеет временную природу, т.е. без наличия первичной информации невозможна рецепция новой. Например, без знания алфавита невозможно интерпретация слов. В реляционных схемах данных эта зависимость отражается в виде механизмов первичных и внешних ключей, идентифицирующих связей и т.д. Информация на одном уровне иерархии составляет множество равноправных элементов [10].

Следует отметить, что между информацией на различных уровнях имеются качественные отличия (например, между словом и символом). Применительно к БД «словарные» (вспомогательные) таблицы и таблицы с оперативными данными также содержат различную по области применения информацию. Классифицируя данные по области применения, в соответствии с современными тенденциями развития технологий БД выделяются следующие уровни контента ИАС:

- а) оперативные данные и документы, отражающие текущее состояние БП;
- б) аналитические данные и документы, отражающие историю изменений в БП;
- в) вспомогательные, служебные или словарные данные и документы [10].

Уровень оперативных данных, как правило, содержит фиксированное множество данных, применяемых для решения основных ФЗ ИАС (соответствует транзакционной БД) [8].

Аналитический уровень регулярно пополняется данными, формирующими историю функционирования БП и представляющими собой состояния оперативного уровня в фиксированные моменты времени, и применяется для решения аналитических задач (соответствует ХД).

Вспомогательный уровень наиболее статичен и, как правило, содержит небольшой массив служебных данных, который очень редко модифицируется и дополняется (например, названия населённых пунктов). Обычно данные таблицы имеют одинаковую структуру как в оперативных БД, так и в ХД.

Согласно теории информации на верхнем уровне иерархической структуры информации могут существовать альтернативные версии. Например, одно и то же изучаемое явление может быть одновременно описано несколькими различными терминами. В результате дальнейшего изучения

этого явления из множества терминов остается только один термин, описывающий это явление. Впоследствии этот термин используется для описания новых явлений (формируется новый слой информации), а альтернативные термины перестают употребляться [1].

Применительно к ИАС альтернативные версии информации могут существовать только на уровне оперативных данных (на верхнем уровне иерархии). Это выражается наличием в системе искаженных данных или рассогласований между данными в ИАС и состоянием БП вследствие временных задержек в актуализации данных. Поэтому оперативные данные о текущем состоянии БП, как правило, не в полной мере соответствуют реальности. В процессе функционирования системы эти данные обновляются, уточняются и корректируются, в результате чего с момента переноса их в хранилище аналитических данных они считаются достоверными, и возможность существования «альтернативных вариантов» (ошибок в учетных данных или искажения такой информации) в ИАС исключается.

ИАС наполняется вследствие как ручного ввода данных в систему, так и формирования и обработки электронных документов, за счет чего ИАС пополняет и модифицирует учетные данные различных её модулей. Расширение контента ИАС осуществляется на уровне оперативных и, как следствие, – аналитических данных.

Следует отметить, что наполнение ХД, как правило, осуществляется в заранее определённые моменты времени, т.е. расширение аналитического уровня контента ИАС осуществляется итеративно [10].

В ходе модификации и разработки новых модулей ИАС также происходит расширение контента системы, поскольку каждой ФЗ ИАС соответствует фрагмент БД.

Графически взаимосвязь оперативного и аналитического уровней информации в ИАС можно представить в виде, показанном на рис. 1, который отражает характеристики контента ИАС и итеративность изменения его состояний.

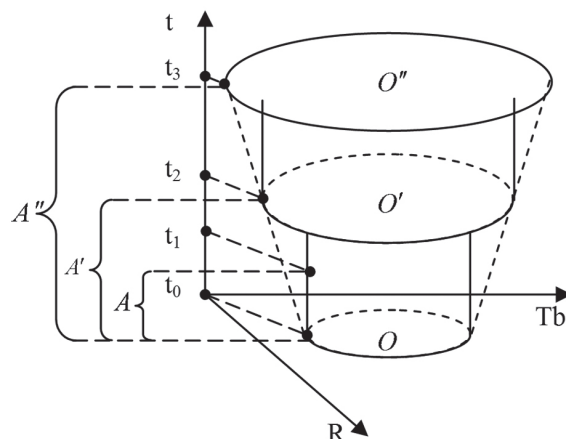


Рис. 1. Взаимосвязь множеств оперативных и аналитических данных и документов контента ИАС

На рис. 1 приняты следующие обозначения: Тб – шкала таблиц в реляционной БД, ХД и хранилище документов ИАС; R – шкала записей и документов в таблицах; t – шкала времени;  $A, A', A''$  – уровень аналитических данных ( $A \subset A' \subset A''$ ) на соответствующих этапах итерации;  $O, O', O''$  – уровень оперативных данных на соответствующих этапах итерации, при этом

$$O'' = \{FZ_1'', \dots, FZ_h'', \dots, FZ_{q_{i1}}'', \dots, FZ_{q_{i2}}', \dots, FZ_{q_{i3}}'\},$$

$$O' = \{FZ_1', \dots, FZ_h', \dots, FZ_{q_{i1}}', \dots, FZ_{q_{i2}}'\},$$

$$O = \{FZ_1, \dots, FZ_h, \dots, FZ_{q_{i1}}\};$$

$t_1, t_2, t_3$  – моменты модификации или добавления данных и документов, поддерживающих функциональные задачи ИАС;  $t_1$  – момент времени в интервале между итерациями модификации или добавления данных и документов;  $FZ_h, FZ_h', FZ_h''$  – данные и документы, поддерживающие  $h$ -ю ФЗ ИАС на соответствующем этапе итерации,  $h = \overline{1, q_{ii}}, i = \overline{1, 3}$ ;  $q_{i1}, q_{i2}, q_{i3}$  – количество реализованных ФЗ ИАС на соответствующем этапе итерации.

Поскольку информационными единицами контента ИАС являются записи в таблицах БД и электронные документы, которые также являются структурами данных, соответственно, формируемые электронные документы относятся к уровню оперативных данных, а утверждённые и обработанные – аналитических. Документы могут храниться в БД ИАС в качестве значений атрибутов отношений, имеющих специализированные типы данных, например, «большой символьный объект» CLOB (Character Large Object), XML Type и т.д. Электронные документы могут также храниться в файловой системе вне БД ИАС. Для унификации работы с электронными документами и структурами данных как информационными единицами контента хранилище документов представляется как таблица, а документы – как записи в такой таблице.

В соответствии с рис. 1, множество аналитических данных  $A$  представляет собой совокупность состояний множества оперативных данных  $O$  на протяжении всего времени функционирования ИАС или с момента начала ведения истории:

$$A(t_p) = \bigcup_{t_{vn}}^{t_p} O(t_j),$$

где  $A(t_p)$  – состояние уровня аналитических данных в текущий момент времени;  $t_p$  – текущий момент времени;  $t_{vn}$  – момент внедрения ИАС в эксплуатацию или начала сбора исторических данных;  $O(t_j)$  – состояние уровня оперативных данных в момент времени  $t_j$ ;  $t_j$  – моменты времени, в которые модифицируются или добавляются данные и документы, поддерживающие функциональные задачи ИАС ( $t_j \in [t_{vn}, t_p]$ ) [10].

Формально структура контента ИАС ( $C$ ) выглядит следующим образом:

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

где  $O, A$  и  $S$  – множества оперативных, аналитических и вспомогательных данных и документов.

При этом каждый из видов контента определяется множеством данных и документов, которые имеют соответствующую уровню область применения и необходимы для решения ФЗ ИАС:

$$L = \bigcup_{h=1}^q Fz_s^L,$$

где  $L$  – структурный уровень контента ИАС (множество оперативных  $O$ , аналитических  $A$  или вспомогательных данных и документов  $S$ );  $q$  – количество функциональных задач ИАС;  $Fz_s^L$  – данные и документы области применения  $L$ , поддерживающие  $h$ -ю функциональную задачу ИАС, которые описываются выражением:

$$Fz_s^L = \bigcup_{i=1}^{n_L} R_{ih}^L(Tb_i),$$

где  $n_L$  – количество таблиц в БД или ХД, содержащих данные и электронные документы области применения  $L$ , необходимые для решения  $h$ -й ФЗ;  $R_{ih}^L(Tb_i)$  – множество записей (информационных единиц) элементов контента ИАС: данных и электронных документов, семантику которых описывают схемы отношений – кортежи, фреймы или экземпляры документов) в таблице  $Tb_i$  БД ИАС, применяемых для решения  $h$ -й ФЗ и соответствующих области применения  $L$  [10].

Описание составляющих каждого из уровней контента ИАС позволяет представить его математическую модель ( $M_C$ ) в следующем виде:

$$M_C = \left\langle \bigcup_{h=1}^q \bigcup_{i=1}^{n_O} R_{ih}^O(Tb_i), \bigcup_{h=1}^q \bigcup_{i=1}^{n_A} R_{ih}^A(Tb_i), \bigcup_{h=1}^q \bigcup_{i=1}^{n_S} R_{ih}^S(Tb_i) \right\rangle,$$

где  $q$  – количество функциональных задач ИАС;  $n_O, n_A, n_S$  – количество таблиц в БД или ХД, содержащих данные и электронные документы областей применения  $O, A$  и  $S$ , необходимые для решения  $h$ -й ФЗ;

С учетом взаимосвязи аналитического и оперативного уровней данная модель модифицируется следующим образом:

$$M_C = \left\langle \left( \bigcup_{h=1}^q \bigcup_{i=1}^{n_O} R_{ih}^O(Tb_i) \right)_{t_p}, \bigcup_{t_{vn}}^{t_p} \left( \bigcup_{h=1}^q \bigcup_{i=1}^{n_O} R_{ih}^O(Tb_i) \right)_{t_i}, \left( \bigcup_{h=1}^q \bigcup_{i=1}^{n_S} R_{ih}^S(Tb_i) \right)_{t_p} \right\rangle,$$

где  $\left( \bigcup_{h=1}^q \bigcup_{i=1}^{n_O} R_{ih}^O(Tb_i) \right)_{t_p}, \left( \bigcup_{h=1}^q \bigcup_{i=1}^{n_S} R_{ih}^S(Tb_i) \right)_{t_p}$  – состояния оперативного и служебного уровней контента ИАС на момент  $t_p$ ;  $\bigcup_{t_{vn}}^{t_p} \left( \bigcup_{h=1}^q \bigcup_{i=1}^{n_O} R_{ih}^O(Tb_i) \right)_{t_i}$  – аналитический

уровень контента ИАС на момент  $t_p$  (содержит историю состояний оперативного уровня ИАС с момента  $t_{vn}$  до  $t_p$ );  $t_{vn}$  – момент внедрения ИАС в эксплуатацию (начала сбора исторических данных);  $t_{ii}$  – момент фиксации состояний оперативного уровня контента ИАС в период от  $t_{vn}$  до  $t_p$ .

Таким образом, с применением терминов многомерной модели данных в контенте ИАС можно выделить пять основных измерений:

- таблицы в реляционной БД, ХД и хранилище документов ИАС;
- записи и документы в таблицах (их идентификаторы);
- функциональные задачи, реализованные в ИАС;
- время;
- области применения данных и документов (структурные уровни контента).

Поскольку в таблицах содержится различное количество данных и электронных документов, которое с течением времени изменяется, как изменяется и набор поддерживаемых ими ФЗ ИАС, их области применения и т.д., контент ИАС представляет собой разреженный многомерный куб данных, который изображен на рис. 2 (измерения времени и уровней контента ИАС на рисунке фиксированы).

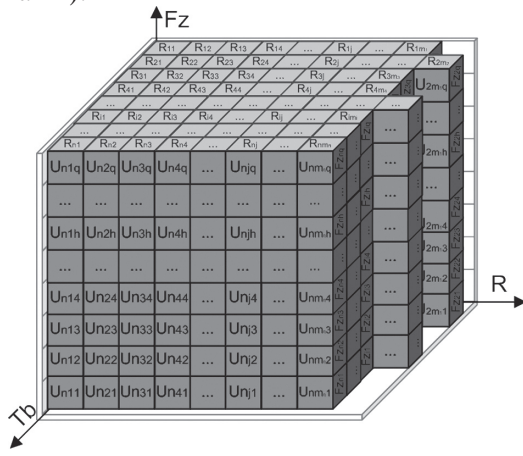


Рис. 2. Визуальное представление контента ИАС

На рис. 2 приняты следующие обозначения:  $Tb$  – упорядоченное множество таблиц БД ИАС ( $Tb_i$  – таблица БД ИАС);  $n$  – количество таблиц в БД ИАС;  $R$  – упорядоченное множество записей в таблицах БД ( $R_{ij}$  – запись в  $i$ -й таблице);  $m_i$  – количество записей в таблице  $Tb_i$ ;  $Fz$  – упорядоченное множество ФЗ ИАС ( $Fz_h$  – ФЗ);  $q$  – количество ФЗ ИАС;  $U_{ijh}$  – элемент гиперкуба с координатами  $(i, j, h)$ , значение которого  $|U_{ijh}|=1$ , если запись  $R_{ij}$  таблицы  $Tb_j$  используется для решения ФЗ  $Fz_h$ , и  $|U_{ijh}|=0$  в противном случае;  $|R_{ij}|$  – значение элемента  $R_{ij}$  среза контента ИАС по осям  $R$  и  $Tb$ , равное количеству ФЗ ИАС, для решения которых используется запись  $R_{ij}$  таблицы  $Tb_i$ ;  $|Fz_{jh}|$  – значение элемента  $Fz_{jh}$  среза контента ИАС по осям

$Fz$  и  $Tb$ , равное количеству записей таблицы  $Tb_i$ , используемых для решения ФЗ  $Fz_h$ ;  $U_{jh}$  – элемент среза контента ИАС по осям  $Fz$  и  $R$  для фиксированной таблицы, значение которого  $|U_{jh}|=1$ , если запись  $R_j$  используется для решения ФЗ  $Fz_h$ , и  $|U_{jh}|=0$  в противном случае.

Из рис. 2 видно, что фиксация измерения ФЗ ИАС ( $Fz$ ) позволяет получить контент функциональных задач ИАС, объединение которых образует контент ИАС.

Измерение времени позволяет осуществлять сравнение состояний контента ИАС (или ФЗ ИАС) в различные моменты времени и отслеживать изменения контента применительно к различным его срезам (например, частоту модификаций записей в таблицах, изменение количества записей в таблицах, частоту обращений к ним; осуществлять сравнение наборов данных, образующих контент ФЗ и т.д.). Это позволяет решать такие задачи управления контентом ИАС, как оптимизация выполнения запросов пользователей и хранения данных. В остальных случаях контент ИАС, как правило, рассматривается в фиксированный момент времени (наиболее часто – в настоящий момент времени, когда ИАС должна отражать текущее состояние БП ОУ).

Разработанная модель контента ИАС обладает следующими характеристиками:

- контент ИАС описан с применением технологии многомерных баз данных;
- контент ИАС описан как многомерный разреженный гиперкуб данных (измерения «Таблицы», «Записи», «Функциональные задачи ИАС», измерения «Время» и «Уровни контента» введены для решения задач управления контентом), который может быть трансформирован в OLAP-куб;
- к модели контента ИАС применимы операции, определённые на многомерной модели данных (срез, вращение, консолидация и детализация);
- контент ИАС включает контент всех её ФЗ (контент ФЗ ИАС может быть получен при фиксации измерения «Функциональные задачи ИАС»).

#### 4. Формирование экземпляра модели контента системы

Формирование экземпляра такой математической модели контента ИАС предполагает отображение схемы данных ИАС в измерения предложенного гиперкуба и зависит от моделей данных, применяемых в БД и ХД системы. Для реляционной модели данных оно включает следующие этапы:

Этап 1. Формирование физической схемы данных

$$DS = \langle Tb_p, Vw, Rl \rangle,$$

где  $DS$  – физическая схема данных ИАС, реализующей множество ФЗ  $Fz$ ;  $Tb_p$  – множество отношений (таблиц) схемы данных, каждое из которых

характеризуется схемой отношения (множеством атрибутов отношения  $\langle A_1, A_2, \dots, A_n \rangle$ ), а также временем создания и именем пользователя-владельца;  $W$  – множество представлений реализованной схемы данных;  $RI$  – множество связей между таблицами и представлениями схемы данных.

Этап 2. Формирование множества таблиц  $Tb$  модели контента ИАС

$$f: Tb_p \rightarrow Tb,$$

где  $Tb$  – упорядоченное множество, на котором задано отношение строгого порядка  ${}_{Tb_a} R_{Tb_b}$ , такое что момент создания  $Tb_a$  наступил раньше, чем  $Tb_b$ ;  $f = \{(Tb_p^1, Tb^1), (Tb_p^2, Tb^2), \dots, (Tb_p^n, Tb^n)\}$  – биективное отображение множества таблиц физической схемы данных  $Tb_p$  в таблицы  $Tb$  модели контента ИАС, т.е.  $|Tb_p| = |Tb|$  (область определения отображения  $D_f = Tb_p$ , область значений  $\mathcal{R}_f = Tb$ );  $n$  – количество таблиц в схеме данных БД ИАС.

Этап 3. Отображение множества хранилищ документов в таблицы  $Tb$  модели контента ИАС (расширение множества  $Tb$ ).

Этап 4. Формирование множества ФЗ  $Fz$ .

Этап 5. Отображение множества таблиц  $Tb$  модели контента ИАС на множества ФЗ  $Fz$  (задаётся разработчиком ИАС):  ${}_{Tb} G_{Fz}$  – нефункциональное отношение  $\forall a \in Tb \exists b \in Fz : (a, b) \in {}_{Tb} G_{Fz}$ ;

Этап 6. Формирование множеств записей (кортежей и документов) для каждой таблицы контента ИАС (задаётся с использованием словаря данных БД):  ${}_{Tb} G_R$ , где  $R = \{R_1 \dots R_n\}$  – множество множеств записей в таблицах  $Tb$  модели контента ИАС.

Для объектно-ориентированной модели данных процедура формирования математической модели контента ИАС остаётся неизменной для всех этапов, кроме первого и второго.

На этих этапах осуществляется формирование таблиц  $Tb$  модели контента ИАС посредством отображения множества классов  $Cl_p$  аналогично отображению таблиц из реляционной схемы данных. Классы описываются фреймами  $[\langle ns_1, vs_1 \rangle, \langle ns_2, vs_2 \rangle, \dots, \langle ns_n, vs_n \rangle]$ , где  $ns$  – имя слота,  $vs$  – значение слота.

Функция биективного отображения  $f: Cl_p \rightarrow Tb$  предполагает отображение множества имен слотов каждого фрейма  $\langle ns_1, ns_2, \dots, ns_n \rangle$  во множество атрибутов  $\langle A_1, A_2, \dots, A_n \rangle$  соответствующей таблицы:  $f: \forall cl \in Cl_p \Rightarrow ns_i \rightarrow A_i, i = \overline{1, n}$ .

Элемент контента ИАС, представленного в виде гиперкуба данных из 5 измерений, имеет координаты  $(i, j, k, h, t)$ , где  $i$  – номер таблицы БД,  $j$  – номер записи в таблице БД,  $k$  – номер уровня контента, к которому относится данная запись,  $h$  – номер ФЗ ИАС,  $t$  – порядковый номер момента модификации контента.

На физическом уровне контент ИАС в фиксированный момент времени может рассматриваться

как гиперкуб, образуемый таблицами в БД и записями в них, которые используются для решения определённых ФЗ ИАС и характеризуются областью применения (вспомогательные, аналитические и оперативные данные). Это влияет на организацию их хранения и поддержание контента ИАС в актуальном состоянии. Запись в таблице БД ИАС соответствует информационной единице контента ИАС – структуре данных или электронному документу.

## 5. Управление контентом

Поскольку контент ИАС непосредственно связан с БП ОУ, это даёт возможность определить условие совместного использования пакетов данных для решения задач управления контентом ИАС, многие из которых предполагают оперирование массивами данных, а не отдельными записями. В данном условии должна быть учтена разница между наборами функциональных задач, для решения которых они используются. Данная разница может быть определена в виде мощности симметрической разности множеств, решаемых ФЗ ИАС. Должны учитываться различия в координатах элементов гиперкуба контента: уровнях контента, к которым они относятся; таблицах, в которых хранятся данные и документы; значениях первичных ключей.

Для обеспечения соответствия контента ИАС бизнес-процессам ОУ эти характеристики не являются существенными. Они характеризуют особенности применения данных и документов (измерение уровней контента), порядок их внесения в ИАС (в случае использования последовательностей для генерации значений первичных ключей), позволяют определить наборы взаимосвязанных данных и документов, которые используются совместно при решении выбранных ФЗ ИАС и т.д. Поэтому разницу координат записей можно представить в виде поправочного коэффициента, который суммируется с расстоянием между множествами ФЗ ИАС, поддерживаемых сравниваемыми записями.

Таким образом, в фиксированный момент времени для рассматриваемого среза контента ИАС условие совместного использования пакетов данных для решения задач управления контентом ИАС, которые включают элементы с координатами  $(i, j, k)$ , где  $i$  – номер таблицы БД,  $j$  – номер записи в таблице БД,  $k$  – структурный уровень, к которому относится данная запись, определяется следующим образом:

$$\forall \rho(a, b) < \infty : \rho(a_{i_1, j_1, k_1}, b_{i_2, j_2, k_2}) = \begin{cases} |R_{i_1, j_1} \Delta R_{i_2, j_2}| + cf, & \text{где } cf = eq(i_1, i_2) + eq(j_1, j_2) + eq(k_1, k_2) \\ \infty, & \text{если } (cf \neq 0) \wedge \\ & \wedge [ (|R_{i_1, j_1}| = 0) \vee (|R_{i_2, j_2}| = 0) \vee (|R_{i_1} \cap R_{i_2}| = 0) ], \end{cases}$$

где  $\rho(a_{i_1, j_1, k_1}, b_{i_2, j_2, k_2})$  – коэффициент, отражающий общность ситуаций использования элементов  $a$  и  $b$  контента ИАС;  $a_{i_1, j_1, k_1}$  и  $b_{i_2, j_2, k_2}$  – элементы контента ИАС (записи – структуры данных и документы);  $|R_j|$  – количество ФЗ ИАС, для решения которых используется запись  $R_j$  таблицы  $Tb_i$ ;  $|R_{i_1 j_1} \Delta R_{i_2 j_2}|$  – расстояние между множествами поддерживаемых ФЗ (считающая мера);  $cf$  – коэффициент, отражающий различия в координатах записей (т.е. в значениях первичных ключей записей, в таблицах, где эти записи хранятся, в их области применения);

$eq(x_1, x_2) = \begin{cases} 1, & \text{если } x_1 \neq x_2 \\ 0, & \text{в противном случае} \end{cases}$  – функция сравнения координат  $x_1$  и  $x_2$  элементов контента.

Условие совместного использования пакетов данных считается невыполненным ( $\rho(a, b) = \infty$ ) в следующих случаях:

а) если хотя бы одна из сравниваемых записей не используется для решения ФЗ ИАС (случай наличия избыточных данных, которые вследствие модификации ИАС больше не используются);

б) сравниваемые записи применяются для решения различных множеств задач, в которых ни одна задача не совпадает.

Вариант б) возникает в случае, когда сравниваемые записи поддерживают различные бизнес-процессы.

### Выводы

Таким образом, полученная модель описывает контент ИАС и его физический смысл в виде совокупности таблиц и записей в БД и ХД, учитывает взаимосвязь контента ИАС с БП ОУ, позволяет описать контент функциональной задачи ИАС. В комплексе с условием совместного использования пакетов данных данная модель может быть применена для решения таких задач управления контентом ИАС, как администрирование данных (включая задачу организации физического хранения данных), администрирование электронных документов и администрирование запросов к контенту ИАС.

Визуально контент ИАС интерпретируется как разреженный гиперкуб, в котором не для всех координат могут быть определены его элементы.

При этом следует выделить уровень оперативных данных: с течением времени на данном уровне появляются отдельные фрагменты, содержащие устаревшие данные, количество и размеры которых будут увеличиваться, т.е. на уровне оперативных данных с течением времени наблюдается рост энтропии. Одна из задач ИАС – поддержание данных о БП в актуальном состоянии. С течением времени энтропия увеличивается и повышается вероятность наличия в БД или ХД рассогласованных данных ( $P_\psi$ ):  $\lim_{t \rightarrow \infty} P_\psi = 1$ , где  $t$  – время отсутствия обновлений. Поскольку документы для ИАС являются источниками данных, обработка этих

документов позволяет актуализировать оперативные данные ( $\lim_{t \rightarrow 0} P_\psi = 0$ ). Поэтому предложенная модель контента ИАС в качестве информационных единиц, используемых для решения ФЗ ИАС, рассматривает как данные, так и электронные документы, которые в формализованном виде представлены структурами данных. Это позволяет обеспечить типизацию функций управления данными и электронными документами в ИАС.

**Список литературы:** 1. Павлов А.А. Информационные технологии и алгоритмизация в управлении / А.А. Павлов, С.Ф. Теленик. – К.: Техника, 2002. – 344 с. 2. Чернавский Д.С. Синергетика и информация (динамическая теория информации) / Д.С. Чернавский. – М.: Едиториал УРСС, 2004. – 288 с. 3. O'Brien J. Management Information Systems – Managing Information Technology in the Interneted Enterprise / J. O'Brien: 6th ed. – Boston: McGraw-Hill/Irwin, 2003. – 656 p. 4. Sarsfield S. The Data Governance Imperative / S. Sarsfield. – London: IT Governance Publishing, 2009. – 161 p. 5. Калинина А.Э. Развитие информационного пространства региональной хозяйственной системы / А.Э. Калинина. – Волгоград: Изд-во ВолГУ, 2005. – 360 с. 6. Codd E.F. Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate / E.F. Codd, S.B. Codd, C.T. Salley. – Hyperion Solutions Corporation. – 1993. – 24 p. 7. Kudryavtsev Yu. A Mathematical Model of the OLAP Cubes / Yu. Kudryavtsev, S. Kuznetsov // Programming and Computer Software, 2009. – Vol. 35. – No. 5. – P. 257-265. 8. Harizopoulos S. OLTP Through the Looking Glass, and What We Found There / S. Harizopoulos, D.J. Abadi, S. Madden, M. Stonebraker // Proceedings of the ACM SIGMOD International Conference on Management of Data, 2008. – P. 981-992. 9. Круковский М.Ю. Автоматно-графовая модель композитного документооборота / М.Ю. Круковский // Математичні машини і системи. – 2006. – № 2. – С. 45-53. 10. Керносов М. А. Разработка модели информационного пространства информационной системы / М. А. Керносов // Новые информационные технологии в научных исследованиях и образовании: XII Всерос. науч.-техн. конф. студентов, молодых ученых и специалистов, 14-16 мая 2008 г. – Рязань: РГРУ. – 2008. – Часть I. – С. 3-5. 11. Шеннон К.Э. Работы по теории информации и кибернетике / К.Э. Шеннон: под ред. Р.Л. Добрушина, О.Б. Лупанова – М.: Иностранная литература, 1963. – 832 с.

Поступила в редколлегию 06.11.2012

УДК 044.03

**Математична модель контенту інформаційно-аналітичної системи** / М.А. Керносов // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 54-61.

В статті розглянуто проблему моделювання контенту інформаційно-аналітичної системи. Стверджується, що існуючі моделі не задовольняють усіх потреб управління контентом, та пропонується вдосконалена модель контенту системи.

Л. 2. Бібліогр.: 11 найм.

UDC 044.03

**Mathematical model of content for information-analytic system** / M.A. Kernosov // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 54-61.

In article the problem of modeling content for information and analytical system is considered. Author states that the existing models do not satisfy all the needs of content management and proposes an improved model of content.

Fig. 2. Ref.: 11 items.

УДК519.6



О.О. Литвин

Українська інженерно-педагогічна академія, м. Харків, Україна, loo71@bk.ru

## ІНТЕГРАЛЬНЕ ПРЕДСТАВЛЕННЯ ЗАЛИШКУ НАБЛИЖЕННЯ ДИФЕРЕНЦІЙОВНИХ ФУНКЦІЙ ІНТЕРПОЛЯЦІЙНИМИ КУБІЧНИМИ ПОЛІНОМАМИ ЗЛАМАЛА-ЖЕНІШЕКА НА ТРИКУТНИКУ

В даній роботі метод побудови явних інтегральних представлень залишкових членів вперше узагальнюється на випадок наближення диференційовних функцій інтерполяційними кубічними поліномами Зламала-Женішека.

ІНТЕРПОЛЯЦІЯ ЗЛАМАЛА-ЖЕНІШЕКА НА ТРИКУТНИКУ, ЗАЛИШОК НАБЛИЖЕННЯ,  
ІНТЕГРАЛЬНА ФОРМА ЗАЛИШКУ НАБЛИЖЕННЯ

### Вступ

Питання інтегрального представлення залишкових членів формул наближення диференційовних функцій було, є і буде одним з найскладніших в обчислювальній математиці. Інтегральні представлення цінні тим, що вони дають точну формулу для залишку наближення диференційовних функцій тим або іншим оператором наближення. В противагу сказаному слід відмітити, що відомі оцінки похибки наближення, які можуть бути отримані з інтегральних представлень, використовують деякі параметри, які задані не точно, а належать деякому інтервалу. Наприклад, оцінка наближення диференційовної функції  $f(x) \in C^{n+1}[a, b]$  поліномом Тейлора степеня  $n$  може бути отримана з інтегрального представлення залишкового члена похибки [див. приклад 1] у формі Лагранжа, Коші, Пеано, Шлемільха-Роше [1]. Нижче наведемо деякі відомі формули інтегрального представлення залишкових членів, які широко використовуються на практиці. Викладений загальний метод їх побудови використаний в даній роботі для побудови інтегрального представлення залишку наближення диференційовних функцій інтерполяційними кубічними поліномами Зламала-Женішека на довільному трикутнику.

### 1. Аналіз літературних джерел

В працях [1-14] сформульовані та доведені теорема існування та єдиності поліномів заданого степеня, які інтерполюють функцію  $f(x, y)$  та її частинні похідні до заданого порядку у вершинах трикутника та у деяких точках на сторонах або всередині трикутника. В працях Ю.Н. Субботіна [9], [12], Байдакової Н.В. [13], Матвеевої Ю.В. [14] досліджувалися оцінки наближення диференційовних функцій такими поліномами. Слід відмітити, що ці оцінки не використовували явних інтегральних представлень залишку наближення.

### 2. Відомі формули інтегрального представлення залишкових членів

1. Інтегральна формула для залишку наближення функції  $f(x) \in C^n[a, b]$ ,  $n \geq 1$  поліномом Тейлора

$$T_{n-1}f(x) = \sum_{s=0}^{n-1} D^s f(x_0) \frac{(x-x_0)^s}{s!},$$

$$D^0 f(x_0) = f(x_0), D^s f(x_0) = \left. \frac{d^s f(x)}{dx^s} \right|_{x=x_0}, s = 1, 2, \dots,$$

$$D^j T_{n-1}f(x) \Big|_{x=x_0} = D^j f(x) \Big|_{x=x_0}, j = \overline{0, n-1}$$

за степенями має вигляд [14]

$$\begin{aligned} R_n f(x) &= f(x) - T_{n-1}f(x) = \int_{x_0}^x D^n f(t) \frac{(x-t)^{n-1}}{(n-1)!} dt = \\ &= (x-x_0)^n \int_0^1 D^n f(x_0 + t(x-x_0)) \frac{(1-t)^{n-1}}{(n-1)!} dt. \end{aligned}$$

Якщо в інтегралі зробити заміну

$$\begin{aligned} t &= x_0 + (x-x_0)u \Rightarrow dt = (x-x_0)du, \\ u &= 0 \Rightarrow t = x_0, u = 1 \Rightarrow t = x. \end{aligned}$$

то отримаємо інший вираз для залишку узагальнення якого будемо використовувати у двовимірному випадку

$$\begin{aligned} R_n f(x) &= (x-x_0)^n \int_0^1 (D^n f)(x_0 + u(x-x_0)) \frac{(1-u)^{n-1}}{(n-1)!} du = \\ &= \int_0^1 \frac{d^n f(x_0 + u(x-x_0))}{du^n} \frac{(1-u)^{n-1}}{(n-1)!} du. \end{aligned}$$

2. Інтегральна формула для залишку наближення функції  $f(x) \in C^r[a, b]$ ,  $1 \leq r \leq M$  інтерполяційним поліномом Лагранжа степеня  $M-1$

$$L_{M-1}f(x) = \sum_{k=1}^M f(x_k) h_k(x), a = x_1 < \dots < x_M = b,$$

$$h_k(x) = \prod_{i=1, i \neq k}^M \frac{(x-x_i)}{(x_k-x_i)},$$

$$L_{M-1}f(x) \Big|_{x=x_i} = f(x) \Big|_{x=x_i}, i = \overline{1, M}$$

має вигляд [16]

$$\begin{aligned} R_M f(x) &= f(x) - L_{M-1}f(x) = \\ &= \sum_{k=1}^M h_k(x) \int_{x_k}^x D^r f(t) \frac{(x_k-t)^{r-1}}{(r-1)!} dt, 1 \leq r \leq M. \end{aligned}$$

3. Інтегральна формула для залишку наближення функції  $f(x) \in C^r[a, b]$ ,  $r \geq p \geq 2$  інтерполяційним

поліномом Ерміта степеня  $n-1$ ,  $n = Mp$

$$E_{n-1}f(x) = \sum_{k=1}^M \sum_{s=0}^{p-1} D^s f(x_k) \frac{(x-x_k)^s}{s!} h_{k,s}(x),$$

$$h_{k,s}(x) = w_k(x) \left\{ w_k(x)^{-1} \right\}_{x_k}^{p-s-1}, w_k(x) = \prod_{i=1, i \neq k}^M (x-x_i)^p,$$

$$\left\{ w_k(x)^{-1} \right\}_{x_k}^{p-s-1} = \sum_{j=0}^{p-s-1} D^j \left[ w_k(x)^{-1} \right]_{x=x_k} \frac{(x-x_k)^j}{j!},$$

$$D^j E_{n-1}f(x) \Big|_{x=x_i} = D^j f(x) \Big|_{x=x_i}, i = \overline{1, M}, j = \overline{0, p-1}$$

має вигляд [15]

$$R_M f(x) = f(x) - E_{M-1}f(x) = \sum_{k=1}^M \sum_{s=0}^{p-1} \frac{(x-x_k)^s}{s!} h_{k,s}(x) \int_{x_k}^x D^r f(t) \frac{(x_k-t)^{r-s-1}}{(r-s-1)!} dt, p \leq r \leq n,$$

$$h_{k,s}(x) = \prod_{j=1, j \neq k}^M (x-x_j)^p \left\{ \prod_{j=1, j \neq k}^M (x-x_j)^{-p} \right\}_{x_k}^{(p-s-1)},$$

$$\left\{ g(x) \right\}_a^{(p-s-1)} = \sum_{s=0}^{p-s-1} D^s g(a) \frac{(x-a)^s}{s!}.$$

### 3. Загальний метод побудови вказаних формул

Цей метод можна сформулювати так. Встановлюється, що інтерполяційні формули Лагранжа та Ерміта є точними на поліномах степеня  $n-1$ . Наступним кроком методу є запис кожної величини  $f(x_k)$  та  $D^s f(x_k)$  за допомогою формули Тейлора у вигляді

$$f(x_k) = \sum_{j=0}^{r-1} D^j f(x) \frac{(x_k-x)^j}{j!} + \int_0^1 \frac{d^r f(x_k+t(x-x_k))}{dt^r} \frac{(1-t)^{r-1}}{(r-1)!} dt, k = \overline{1, n}.$$

Підставляючи ці формули у формулу Лагранжа і враховуючи, що вона є точною на поліномах степеня  $n-1$ , приходимо до написаної в прикладі 2 формули для залишку. Аналогічно, при доведенні відповідної формули для залишку наближення поліномами Ерміта, використовуються такі формули Тейлора

$$D^s f(x_k) = \sum_{j=0}^{r-s-1} D^{s+j} f(x) \frac{(x_k-x)^j}{j!} + \int_0^1 \frac{d^r f(x_k+t(x-x_k))}{dt^r} \frac{(1-t)^{r-1}}{(r-1)!} dt, k = \overline{1, M}.$$

### 4. Постановка задачі

Для побудови кубічних інтерполяційних поліномів Зламала-Женішека на трикутнику потрібно знати значення функції у вершинах і в центрі трикутника та значення перших частинних похідних у вершинах трикутника. Тому задача полягає у побудові точної формули для інтегрального представлення залишку наближення функцій вказаними інтерполяційними поліномами, метод побудови

якої дозволив би знайти оцінки похибки для функцій  $r$  раз диференційовних ( $2 \leq r \leq 4$ ). Описаний вище загальний метод побудови явних інтегральних представлень залишкових членів вперше узагальнюється на випадок наближення диференційовних функцій інтерполяційними кубічними поліномами Зламала-Женішека.

### 5. Основні твердження даної роботи

Вважаємо заданими систему точок  $A_k(X_k, Y_k) \in D = [0, 1]^2, k = 1, \dots, M$ , яку ми триангулюємо – розбиваємо на трикутники  $T_{pqr} \subset D$  з вершинами  $A_p, A_q, A_r$ , та значення

$$D^\alpha f(X_k, Y_k), 0 \leq |\alpha| = \alpha_1 + \alpha_2 \leq 1, f(X_{pqr}, Y_{pqr}),$$

$$X_{pqr} = \frac{X_p + X_q + X_r}{3}, Y_{pqr} = \frac{Y_p + Y_q + Y_r}{3}.$$

Наведемо явні формули для базисних поліномів третього степеня поліноміальної кубічної інтерполяції Зламала-Женішека на трикутнику  $T_{pqr} \subset D$  (див. [16])

$$w_{p,q}(x, y) = \begin{vmatrix} x & y & 1 \\ X_p & Y_p & 1 \\ X_q & Y_q & 1 \end{vmatrix} =$$

$$= (y - Y_p)(X_q - X_p) - (Y_q - Y_p)(x - X_p),$$

$$h_{pqr}(x, y) =$$

$$= \frac{w_{p,q}(x, y)}{w_{p,q}(X_{pqr}, Y_{pqr})} \frac{w_{q,r}(x, y)}{w_{q,r}(X_{pqr}, Y_{pqr})} \frac{w_{r,p}(x, y)}{w_{r,p}(X_{pqr}, Y_{pqr})},$$

$$h_{r,\beta}^{pqr}(x, y) =$$

$$= \frac{(x - X_r)^{\beta_1} (y - Y_r)^{\beta_2}}{\beta!} \cdot w_{p,q}^2(x, y) \cdot \left\{ \frac{1}{w_{p,q}^2(x, y)} \right\}_{(X_r, Y_r)}^{1-|\beta|},$$

де

$$\left\{ \frac{1}{u_{p,q}(x, y)} \right\}_{(X_r, Y_r)}^{1-|\beta|} =$$

$$= \sum_{0 \leq |\gamma| \leq 1-|\beta|} \left( D^\gamma \frac{1}{u_{p,q}(x, y)} \right)_{(X_r, Y_r)} \frac{(x - X_r)^{\gamma_1} (y - Y_r)^{\gamma_2}}{\gamma!},$$

$$\gamma! = \gamma_1! \gamma_2!$$

Ці функції є поліномами третього степеня з властивостями

$$h_{pqr}(X_{pqr}, Y_{pqr}) = 1, \quad (1)$$

$$D^\gamma h_{pqr}(X_i, Y_i) = 0, 0 \leq |\gamma| \leq 1, i \in \{p, q, r\}, \quad (2)$$

$$\gamma = (\gamma_1, \gamma_2), |\gamma| = \gamma_1 + \gamma_2, D^\gamma = \frac{\partial^{|\gamma|}}{\partial x^{\gamma_1} \partial y^{\gamma_2}}, D^{0,0} h_{p,q,r} = h_{p,q,r}$$

та

$$h_{k,\beta}^{pqr}(X_\ell, Y_\ell) = \delta_{k,\ell} \delta_{0,|\beta|}; k, \ell \in \{p, q, r\},$$

$$0 \leq |\beta| \leq 1, \beta = (\beta_1, \beta_2), |\beta| = \beta_1 + \beta_2, \quad (3)$$

$$D^\gamma h_{k,\beta}^{pqr}(X_\ell, Y_\ell) = \delta_{k,\ell} \delta_{\gamma_1, \beta_1} \delta_{\gamma_2, \beta_2},$$

$$|\gamma| = 1, \gamma = (\gamma_1, \gamma_2), |\gamma| = \gamma_1 + \gamma_2.$$

Тобто вони є базисними функціями при побудові операторів двовимірної кубічної інтерполяції на кожному трикутнику  $T_{pqr} \subset D$ . Не обмежуючи загальності, будемо вважати, що довільний трикутник має вершини  $A_1, A_2, A_3$ . В цьому випадку оператор [15]

$$\begin{aligned} Of(x,y) = & \sum_{k=1}^3 h_{k,0,0}(x,y)f(X_k, Y_k) + \\ & + \sum_{k=1}^3 h_{k,1,0}(x,y)f^{(1,0)}(X_k, Y_k) + \\ & + \sum_{k=1}^3 h_{k,0,1}(x,y)f^{(0,1)}(X_k, Y_k) + \\ & + [f(X_{pqr}, Y_{pqr}) - \sum_{k=1}^3 h_{k,0,0}(X_{pqr}, Y_{pqr})f(X_k, Y_k) - \\ & - \sum_{k=1}^3 h_{k,1,0}(X_{pqr}, Y_{pqr})f^{(1,0)}(X_k, Y_k) - \\ & - \sum_{k=1}^3 h_{k,0,1}(X_{pqr}, Y_{pqr})f^{(0,1)}(X_k, Y_k)] h(x,y) \end{aligned}$$

є оператором поліноміальної кубічної інтерполяції Зламала-Женішека на трикутнику з властивостями

$$\begin{aligned} D^\alpha Of(X_p, Y_p) &= D^\alpha f(X_p, Y_p), p=1,2,3; \\ 0 \leq |\alpha| = \alpha_1 + \alpha_2 \leq 1, Of(X_{pqr}, Y_{pqr}) &= f(X_{pqr}, Y_{pqr}), \quad (4) \\ X_{pqr} &= \frac{X_1 + X_2 + X_3}{3}, Y_{pqr} = \frac{Y_1 + Y_2 + Y_3}{3} \end{aligned}$$

Згідно з теорією [2-5] ці оператори є точними на кубічних поліномах 3-го степеня

$$Of(x,y) = f(x,y) \quad \forall f(x,y) = \sum_{0 \leq |\beta| \leq 3} a_\beta x^{\beta_1} y^{\beta_2}. \quad (5)$$

Для доведення цього достатньо відмітити, що поліном  $\sum_{0 \leq |\beta| \leq 3} a_\beta x^{\beta_1} y^{\beta_2}$  є лінійною комбінацією десяти лінійно-незалежних функцій  $x^{\beta_1} y^{\beta_2}, 0 \leq |\beta_1 + \beta_2| \leq 3$  і система лінійних алгебраїчних рівнянь (4) для знаходження невідомих коефіцієнтів  $a_\beta$  має розв'язок і цей розв'язок єдиний. Існування цього розв'язку впливає з того, що кубічний поліном  $Of(x,y)$  задовольняє умови (4). Крім того, з припущення, що існує два таких інтерполяційних поліноми, які задовольняють умови (4), впливає, що їх різниця задовольняє однорідним інтерполяційним умовам, тобто обидва такі поліноми повинні бути однаковими. Таким чином, з тотожності (5) можна написати такі властивості вказаних операторів.

Властивість 1.  $Oc = c \quad \forall c \in R$ , тобто оператор  $Of$  є точним на сталих.

Наслідок 1.  $\sum_{k=1}^3 h_{k,0,0}(x,y) \equiv 1, (x,y) \in R^2$ .

Наслідок 2.  $\sum_{k=1}^3 h_{k,0,0}(x,y)f(x,y) + [f(x,y) - \sum_{k=1}^3 h_{k,0,0}(X_{pqr}, Y_{pqr})f(x,y)]h(x,y) = f(x,y)$ .

Властивість 2. Якщо  $f(x,y) = x^m, 1 \leq m \leq 3$ , то

$$\begin{aligned} & \sum_{k=1}^3 [h_{k,0,0}(x,y)X_k^m + h_{k,1,0}(x,y)mX_k^{m-1}] + \{X_{pqr}^m - \\ & - \sum_{k=1}^3 [h_{k,0,0}(X_{pqr}, Y_{pqr})X_k^m + h_{k,1,0}(X_{pqr}, Y_{pqr})mX_k^{m-1}]\} \times \\ & \times h(x,y) = x^m. \end{aligned}$$

Властивість 3. Якщо  $f(x,y) = y^n, 1 \leq n \leq 3$ , то

$$\begin{aligned} & \sum_{k=1}^3 [h_{k,0,0}(x,y)Y_k^n + h_{k,0,1}(x,y)nY_k^{n-1}] + \{Y_{pqr}^n - \\ & - \sum_{k=1}^3 [h_{k,0,0}(X_{pqr}, Y_{pqr})Y_k^n + h_{k,0,1}(X_{pqr}, Y_{pqr})nY_k^{n-1}]\} \times \\ & \times h(x,y) = y^n. \end{aligned}$$

Властивість 4. Якщо

$$f(x,y) = x^m y^n, 1 \leq m+n \leq 3, m \geq 1, n \geq 1, \text{ то}$$

$$\begin{aligned} & \sum_{k=1}^3 [h_{k,0,0}(x,y)X_k^m Y_k^n + h_{k,1,0}(x,y)mX_k^{m-1} Y_k^n + \\ & + h_{k,0,1}(x,y)X_k^m nY_k^{n-1}] + \\ & + \left\{ X_{pqr}^m Y_{pqr}^n - \sum_{k=1}^3 [h_{k,0,0}(X_{pqr}, Y_{pqr})X_k^m Y_k^n + \right. \\ & \left. + h_{k,1,0}(X_{pqr}, Y_{pqr})mX_k^{m-1} Y_k^n + \right. \\ & \left. + h_{k,0,1}(X_{pqr}, Y_{pqr})X_k^m nY_k^{n-1} \right\} \times h(x,y) \equiv x^m y^n, \\ & O(x-u)^{\alpha_1} (y-v)^{\alpha_2} \Big|_{u=x, v=y} = 0, \quad 1 \leq \alpha_1 + \alpha_2 \leq 3. \end{aligned}$$

Ці властивості дозволяють довести наступну теорему.

**Теорема 1.** Для залишку  $Rf(x,y) = (I-O)f(x,y)$  наближення функції  $f(x,y) \in C^4(\bar{T}_{pqr})$  справедливе таке інтегральне представлення

$$R(x,y) = R00(x,y) + R10(x,y) + R01(x,y) + Rsr(x,y),$$

де  $R00f(x,y) = \sum_{v=1}^3 h(x,y,p,q,r,0,0,v) \times$

$$\times \int_0^1 \frac{\partial^4 f((x,y) + t(x_v - x, y_v - y))}{\partial t^4} \frac{(1-t)^3}{3!} dt,$$

$$\begin{aligned} R10f(x,y) &= \sum_{v=1}^3 h(x,y,p,q,r,1,0,v) \times \\ & \times \int_0^1 \frac{\partial^3 f^{(1,0)}((x,y) + t(x_v - x, y_v - y))}{\partial t^3} \frac{(1-t)^2}{2!} dt, \end{aligned}$$

$$R01f(x,y) = \sum_{v=1}^3 h(x,y,p,q,r,0,1,v) \times$$

$$\times \int_0^1 \frac{\partial^3 f^{(0,1)}((x,y) + t(x_v - x, y_v - y))}{\partial t^3} \frac{(1-t)^2}{2!} dt,$$

$$\begin{aligned} Rsr f(x,y) &= h(x,y) \left[ \int_0^1 \frac{\partial^4 f((x,y) + t(X_{sr} - x, Y_{sr} - y))}{\partial t^4} \frac{(1-t)^3}{3!} dt - \right. \\ & \left. - \bar{R}00f(x,y) - \bar{R}10f(x,y) - \bar{R}01f(x,y) \right], \end{aligned}$$

де

$$\bar{R}00f(x, y) = \sum_{v=1}^3 h(X_{sr}, Y_{sr}, p, q, r, 0, 0, v) \times$$

$$\times \int_0^1 \frac{\partial^3 f((x, y) + t(x_v - x, y_v - y)) (1-t)^3}{\partial t^3} \frac{dt}{3!}$$

$$\bar{R}10f(x, y) = \sum_{v=1}^3 h(X_{sr}, Y_{sr}, p, q, r, 1, 0, v) \times$$

$$\times \int_0^1 \frac{\partial^3 f^{(1,0)}((x, y) + t(x_v - x, y_v - y)) (1-t)^2}{\partial t^3} \frac{dt}{2!}$$

$$\bar{R}01f(x, y) = \sum_{v=1}^3 h(X_{sr}, Y_{sr}, p, q, r, 0, 1, v) \times$$

$$\times \int_0^1 \frac{\partial^3 f^{(0,1)}((x, y) + t(x_v - x, y_v - y)) (1-t)^2}{\partial t^3} \frac{dt}{2!}$$

*Доведення.* Замінімо значення функції  $f(x, y) \in C^4(\bar{T}_{pqr})$  та значення її частинних похідних першого порядку у кутових точках і в центрі трикутника відповідними формулами Тейлора

$$f(X_k, Y_k) = \sum_{0 \leq s+t \leq r} D^{s,t} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} +$$

$$+ \int_0^1 \frac{\partial^{r+1}}{\partial t^{r+1}} f(x + t(X_k - x), y + t(Y_k - y)) \frac{(1-t)^r}{r!} dt,$$

$$1 \leq r \leq 4,$$

$$f(X_k, Y_k) = \sum_{0 \leq s+t \leq 3} D^{s,t} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} +$$

$$+ \int_0^1 \frac{\partial^4}{\partial t^4} f(x + t(X_k - x), y + t(Y_k - y)) \frac{(1-t)^3}{3!} dt,$$

$$f^{(1,0)}(X_k, Y_k) = \sum_{0 \leq s+t \leq 2} D^{1+s,t} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} +$$

$$+ \int_0^1 \frac{\partial^3}{\partial t^3} f^{(1,0)}(x + t(X_k - x), y + t(Y_k - y)) \times \frac{(1-t)^2}{2!} dt,$$

$$f^{(0,1)}(X_k, Y_k) = \sum_{0 \leq s+t \leq 2} D^{s,t+1} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} +$$

$$\int_0^1 \frac{\partial^3}{\partial t^3} f^{(0,1)}(x + t(X_k - x), y + t(Y_k - y)) \times \frac{(1-t)^2}{2!} dt,$$

$$f(X_{pqr}, Y_{pqr}) =$$

$$= \sum_{0 \leq s+t \leq 3} D^{s,t} f(x, y) \frac{(X_{pqr} - x)^s (Y_{pqr} - y)^t}{s!t!} +$$

$$+ \int_0^1 \frac{\partial^4}{\partial u^4} f(x + u(X_{pqr} - x), y + u(Y_{pqr} - y)) \frac{(1-t)^3}{3!} dt.$$

В результаті отримаємо

$$\begin{aligned} Of(x, y) = & \sum_{k=1}^3 \left\{ \sum_{0 \leq \alpha + \beta \leq 1} [h_{k,0,0}(x, y) - \right. \\ & \left. - h_{k,0,0}(X_{pqr}, Y_{pqr})h(x, y)] \times \right. \\ & \times \left[ \sum_{0 \leq s+t \leq 3} D^{s,t} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} + \right. \\ & \left. + \int_0^1 \frac{\partial^4}{\partial t^4} f(x + t(X_k - x), y + t(Y_k - y)) \frac{(1-t)^3}{3!} dt \right] \Big\} + \\ & + \sum_{k=1}^3 \left\{ \sum_{0 \leq \alpha + \beta \leq 1} [h_{k,1,0}(x, y) - h_{k,1,0}(X_{pqr}, Y_{pqr})h(x, y)] \times \right. \\ & \times \left[ \sum_{0 \leq s+t \leq 2} D^{s,t} f^{(1,0)}(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} + \right. \\ & \left. + \int_0^1 \frac{\partial^3}{\partial t^3} f^{(1,0)}(x + t(X_k - x), y + t(Y_k - y)) \frac{(1-t)^2}{2!} dt \right] \Big\} + \\ & + \sum_{k=1}^3 \left\{ \sum_{0 \leq \alpha + \beta \leq 1} [h_{k,1,0}(x, y) - h_{k,1,0}(X_{pqr}, Y_{pqr})h(x, y)] \times \right. \\ & \times \left[ \sum_{0 \leq s+t \leq 2} D^{s,t} f^{(0,1)}(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} + \right. \\ & \left. + \int_0^1 \frac{\partial^3}{\partial t^3} f^{(0,1)}(x + t(X_k - x), y + t(Y_k - y)) \frac{(1-t)^2}{2!} dt \right] \Big\} + \\ & + h(x, y) \left\{ \sum_{0 \leq s+t \leq 3} D^{s,t} f(x, y) \frac{(X_{pqr} - x)^s (Y_{pqr} - y)^t}{s!t!} + \right. \\ & \left. + \int_0^1 \frac{\partial^4}{\partial t^4} f(x + t(X_{pqr} - x), y + t(Y_{pqr} - y)) \frac{(1-t)^3}{3!} dt \right\}. \end{aligned}$$

Перепишемо цю формулу у вигляді

$$\begin{aligned} Of(x, y) = & \sum_{k=1}^3 \left\{ f(x, y) + \right. \\ & + \sum_{1 \leq s+t \leq 3} D^{s,t} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} + \\ & \left. + \int_0^1 \frac{\partial^4}{\partial t^4} f(x + t(X_k - x), y + t(Y_k - y)) \frac{(1-t)^3}{3!} dt \right\} \times \\ & \times [h_{k,0,0}(x, y) - h_{k,0,0}(X_{pqr}, Y_{pqr})h(x, y)] + \\ & + \sum_{k=1}^3 \left\{ \sum_{0 \leq s+t \leq 2} D^{1+s,t} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} + \right. \\ & \left. + \int_0^1 \frac{\partial^3}{\partial t^3} f^{(1,0)}(x + t(X_k - x), y + t(Y_k - y)) \frac{(1-t)^2}{2!} dt \right\} \times \\ & \times [h_{k,1,0}(x, y) - h_{k,1,0}(X_{pqr}, Y_{pqr})h(x, y)] + \\ & + \sum_{k=1}^3 \left\{ \sum_{0 \leq s+t \leq 2} D^{s,t+1} f(x, y) \frac{(X_k - x)^s (Y_k - y)^t}{s!t!} + \right. \end{aligned}$$

$$\begin{aligned} & \left. + \int_0^1 \frac{\partial^3}{\partial t^3} f^{(0,1)}(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^2}{2!} dt \right\} \times \\ & \times \left[ h_{k,0,1}(x,y) - h_{k,0,1}(X_{pqr}, Y_{pqr}) h(x,y) \right] + h(x,y) \times \\ & \times \left\{ f(x,y) + \sum_{1 \leq s+t \leq 3} D^{s,t} f(x,y) \frac{(X_{pqr}-x)^s (Y_{pqr}-y)^t}{s!t!} + \right. \\ & \left. + \int_0^1 \frac{\partial^4}{\partial t^4} f(x+t(X_{pqr}-x), y+t(Y_{pqr}-y)) \frac{(1-t)^3}{3!} dt \right\}. \end{aligned}$$

В результаті для залишку наближення отримаємо формулу

$$\begin{aligned} f(x,y) - Of(x,y) &= f(x,y) - f(x,y) \times \\ & \times \sum_{k=1}^3 \left[ h_{k,0,0}(x,y) - h_{k,0,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] - \\ & - \sum_{1 \leq s+t \leq 3} D^{s,t} f(x,y) \sum_{k=1}^3 \left\{ \frac{(X_k-x)^s (Y_k-y)^t}{s!t!} \times \right. \\ & \times \left[ h_{k,0,0}(x,y) - h_{k,0,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ & \quad + D^{1,0} \frac{(X_k-x)^s (Y_k-y)^t}{s!t!} \times \\ & \times \left[ h_{k,1,0}(x,y) - h_{k,1,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ & \quad + D^{0,1} \frac{(X_k-x)^s (Y_k-y)^t}{s!t!} \times \\ & \times \left[ h_{k,0,1}(x,y) - h_{k,0,1}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ & \quad \left. + \frac{(X_{pqr}-x)^s (Y_{pqr}-y)^t}{s!t!} h(x,y) \right\} + \\ & + \sum_{k=1}^3 \int_0^1 \frac{\partial^4}{\partial t^4} f(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^3}{3!} dt \times \\ & \times \left[ h_{k,0,0}(x,y) - h_{k,0,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ & + \sum_{k=1}^3 \int_0^1 \frac{\partial^3}{\partial t^3} f^{(1,0)}(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^2}{2!} dt \times \\ & \times \left[ h_{k,1,0}(x,y) - h_{k,1,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ & + \sum_{k=1}^3 \int_0^1 \frac{\partial^3}{\partial t^3} f^{(0,1)}(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^2}{2!} dt \times \\ & \times \left[ h_{k,0,1}(x,y) - h_{k,0,1}(X_{pqr}, Y_{pqr}) h(x,y) \right] + h(x,y) \times \\ & \times \int_0^1 \frac{\partial^4}{\partial t^4} f(x+t(X_{pqr}-x), y+t(Y_{pqr}-y)) \frac{(1-t)^3}{3!} dt. \end{aligned}$$

Враховуючи, що

$$f(x,y) -$$

$$-f(x,y) \sum_{k=1}^3 \left[ h_{k,0,0}(x,y) - h_{k,0,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] = 0,$$

та групуючи доданки в останній формулі, отримаємо

$$\begin{aligned} f(x,y) - Of(x,y) &= \\ &= \sum_{1 \leq s+t \leq 3} D^{s,t} f(x,y) O\left( (x-u)^s (y-v)^t \right) \Big|_{u=x, v=y} + \\ &+ \sum_{k=1}^3 \int_0^1 \frac{\partial^4}{\partial t^4} f(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^3}{3!} dt \times \\ & \times \left[ h_{k,0,0}(x,y) - h_{k,0,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ &+ \sum_{k=1}^3 \int_0^1 \frac{\partial^3}{\partial t^3} f^{(1,0)}(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^2}{2!} dt \times \\ & \times \left[ h_{k,1,0}(x,y) - h_{k,1,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ &+ \sum_{k=1}^3 \int_0^1 \frac{\partial^3}{\partial t^3} f^{(0,1)}(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^2}{2!} dt \times \\ & \times \left[ h_{k,0,1}(x,y) - h_{k,0,1}(X_{pqr}, Y_{pqr}) h(x,y) \right] + h(x,y) \times \\ & \times \int_0^1 \frac{\partial^4}{\partial t^4} f(x+t(X_{pqr}-x), y+t(Y_{pqr}-y)) \frac{(1-t)^3}{3!} dt. \end{aligned}$$

Таким чином, для залишку наближення отримаємо

$$\begin{aligned} f(x,y) - Of(x,y) &= \\ &= \sum_{k=1}^3 \int_0^1 \frac{\partial^4}{\partial t^4} f(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^3}{3!} dt \times \\ & \times \left[ h_{k,0,0}(x,y) - h_{k,0,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ &+ \sum_{k=1}^3 \int_0^1 \frac{\partial^3}{\partial t^3} f^{(1,0)}(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^2}{2!} dt \times \\ & \times \left[ h_{k,1,0}(x,y) - h_{k,1,0}(X_{pqr}, Y_{pqr}) h(x,y) \right] + \\ &+ \sum_{k=1}^3 \int_0^1 \frac{\partial^3}{\partial t^3} f^{(0,1)}(x+t(X_k-x), y+t(Y_k-y)) \frac{(1-t)^2}{2!} dt \times \\ & \times \left[ h_{k,0,1}(x,y) - h_{k,0,1}(X_{pqr}, Y_{pqr}) h(x,y) \right] + h(x,y) \times \\ & \times \int_0^1 \frac{\partial^4}{\partial t^4} f(x+t(X_{pqr}-x), y+t(Y_{pqr}-y)) \frac{(1-t)^3}{3!} dt. \end{aligned}$$

Теорему 1 доведено.

### Висновки

Тут у всіх випадках під знаком інтегралів приймають участь похідні  $f^{(\alpha)}$ ,  $|\alpha|=r=4$ . Але і для  $|\alpha|=r=2$  або  $|\alpha|=r=3$  таким методом теж можна отримати інтегральне зображення залишку.

В подальшому планується знайти оцінку похибки для випадку, коли  $f(x,y) \in C^m(\bar{T}_{pqr})$ ,  $m=2,3,4$  та порівняти отримані результати з оцінками, отриманими іншими авторами без використання інтегральних формул для залишку.

Відмітимо, що для  $|\alpha|=r=2$  та для  $|\alpha|=r=3$  таких оцінок взагалі нема.

**Список літератури: 1.** Математическая энциклопедия: Гл.ред. И.М.Виноградов, [Текст] / т.5. Слу-Я –М., “Советская энциклопедия”, 1984. –1248 стб., ил. **2.** Zlamal, M.

A finite element procedure of the second order accuracy [Текст] / M. Zlamal. // Numer. Math., 14 (1970), 394-402. **3. Zenisek, A.** Interpolation polynomials on the triangle [Текст] / A. Zenisek. // Numer. Math. 1970. Vol. 15. S. 283-296. **4. Zlamal, M.** Mathematical aspect of the finite element method [Текст] / M. Zlamal., A. Zenisek. // Technical, physical and mathematical principles of the finite element method (V. Kolar et. al. eds.) Praha: Acad. VED. 1971. P. 15-39. **5. Варга, Р.** Функциональный анализ и теория аппроксимации в численном анализе: Пер. с англ. [Текст] / Р. Варга. – М.: Мир, 1974. – 126 с. **6. Babushka, I.** On the angle condition in the finite element method [Текст] / I. Babushka, A.K. Aziz. // SIAM J. Numer. Anal. 1976. V. 13. №2. P. 214-226. **7. Bramble, J.H.** Triangular elements in the finite element method [Текст] / J.H. Bramble, M. Zlamal. // Math. Comp. 1970. v. 24. P. 809-820. **8. Субботин, Ю.Н.** Зависимость оценок многомерной кусочно полиномиальной аппроксимации от геометрических характеристик триангуляции [Текст] / Ю.Н. Субботин // Труды МИАН СССР, 1989. – Т. 189. – С. 117-137. **9. Латыпова, Н.В.** Погрешность кусочно-кубической интерполяции на треугольнике [Текст] / Н.В. Латыпова // Вестн. Удмрт. ун-та. Сер. Математика. – 2003. – С. 3-10. **10. Zenisek, A.** Maximum-angle condition and triangular finite elements of hermite type [Текст] / A. Zenisek // Math. Comp. 1995. V. 64. №211 P. 929-941. **11. Субботин, Ю.Н.** Новый кубический элемент в МКЭ [Текст] / Ю.Н. Субботин // Труды Института математики и механики. Теория функций: Сб. науч. трудов. Екатеринбург: УрО РАН. – 2005. – V. 11, №2. – С. 120-130. **12. Байдакова, Н.В.** Об одном способе эрмитовой интерполяции многочленами третьей степени на треугольнике [Текст] / Н.В. Байдакова // Труды Института математики и механики. Теории функций: Сб. науч. трудов. Екатеринбург: УрО РАН. 2005. V. 11. №2. P. 47-52. **13. Матвеева, Ю.В.** Об интерполяции кубическими многочленами третьей степени на

треугольнике с использованием смешанных производных [Текст] / Ю.В. Матвеева // Известия Саратовского университета. Новая серия. Сер. математика. Механика. Информатика. – 2007. – Т. 7, Вып. 1. – С. 28-32. **14. Никольский, С.М.** Курс математического анализа. [Текст] / С.М. Никольский // М.: Наука, 1973. – Том 1. – 431 с. **15. Литвин, О.М.** Побудова 2D кубічних інтерполяційних сплайнів класу  $C^1(D)$  [Текст] / О.М. Литвин, О.О. Литвин, О.І. Денисова // Вісник Запорізького університету, 2011. – №1. – С. 66-74. **16. Литвин, О.М.** Методи обчислень. Додаткові розділи. [Текст] / О.М. Литвин – К.: Наукова думка, 2005. – 333 с.

Поступила до редколегії 12.11.2012

УДК 519.6

**Интегральное представление остатка приближения дифференцируемых функций интерполяционными кубическими полиномами Зламала-Женишека** / О.О. Литвин // Бионика интеллекта: науч.-техн. журнал. – 2013. – № 1 (80). – С. 62-67.

Впервые обобщается метод построения явных интегральных представлений остаточных членов на случай приближения дифференцируемых функций интерполяционными кубическими полиномами Зламала-Женишека.

Библиогр.: 16 назв.

UDC 519.6

**Integral representation of the balance approximation of differentiable functions by interpolating cubic polynomials Zlamals-Zhenisheks** / O.O. Lytvyn // Bionics of Intelligence: Sci. Mag. – 2013. – № 1 (80). – P. 62-67.

First generalized method of constructing explicit integral representations of the remainder terms in the case of approximation of differentiable functions by interpolating cubic polynomials Zlamala-Zhenisheka.

Ref.: 16 items.

UDK 004.932.2

S.V. Mashtalir<sup>1</sup>, O.D. Mikhnova<sup>2</sup><sup>1</sup> KNURE, Kharkiv, Ukraine, mashtalir\_s@kture.kharkov.ua;<sup>2</sup> KNURE, Kharkiv, Ukraine, elena\_mikhnova@ukr.net

## STABILIZATION OF KEY FRAME DESCRIPTIONS WITH HIGHER ORDER VORONOI DIAGRAM

Video summary is one of currently developing areas of video mining. Static summary is composed of key frames extracted from video, which fully depict its content. While extracting key frames with the help of Voronoi tessellation comparison, it has been proposed to detail frame content with higher order Voronoi diagrams. This step has led to simplification of computational procedure compared with increasing the number of initial generator points. Key frames extracted with Voronoi diagrams have been checked for precision and recall and compared with three existing extraction techniques based on optical flow, cluster analysis and curve simplification.

VIDEO SUMMARY, KEY FRAME EXTRACTION, GENERATOR POINT, VORONOI TESSELLATION, HIGHER ORDER VORONOI DIAGRAMS, FINITE SET OF POINTS

### Introduction

Last decade is marked by great progress in machine vision based on artificial intelligence techniques. It is mostly due to increase in hardware capacity that made possible to store huge amount of data and process multimedia in a high speed. Just 10 years ago scientists dealt with image processing, on the contrast, contemporary researchers process video, as hardware possibilities are finally opened up. Despite of these favorable conditions, video processing still lacks in high quality uniform methods that might be applicable for a number of subject matters. Aside from variety of application domains, one of the main problems that arises is shooting conditions (for example light, camera move, zoom, etc.). Moreover, video quality also influences on the results. For instance, video shot even at AVCHD lite resolution 1280×720 and high definition video 1920×1080 may show different results for the same processing algorithm. That is why the gap between low level video features (like color and texture) and high level semantics (features that lead to understanding of video content) is so big.

Zhang D., Eakins J.P., Su Z., Hyvonen E., Smith J.R., etc. [1, 2] tried to shorten this gap by incorporating of high level features, but the truth is that the only thing can be done is mid level presentation obtained from interpretation, transformation or filtering of low level data. After transition from image to video retrieval, researchers also continue to analyze the whole video as a sequence of images (consecutive frames), transferring approaches of spatial segmentation. The only thing that changed is added temporal segmentation of video into shots, scenes and subscenes, which group closely related content.

From this point, search for similar content in videos is much more advantageous compared with standalone images. Our research is performed in this context of video recognition. By summarizing video content we propose to extract representative frames that depict information from several similar frames. These similar frames are not obviously consecutive but they must have

identical content to be presented by a frame, or this representative frame corresponds to a great change in the scene (outlier).

Key frame extraction may be implemented by different methods, including simple extraction of first/last frame of each scene, motion analysis, clustering analysis, matrix factorization, curve simplification and their modifications observed in [3]. Let's take a look at three most popular groups of methods.

By analyzing optical flow we can only extract frames with significant changes in motion. Moreover, this kind of methods possesses very slow performance especially for high quality data. The results may happen to be very poor for typical motion with homogeneous exposition and scenes. Changes in light may also influence great motion whereas there is no motion at all. Different threshold values should be set for videos with lots of moving objects and poor motion videos. And the main issue for key frame extraction is that optical flow does not reveal significance of motion detected.

Extracting key frames with cluster analyses forces to set the number of clusters a priori as it is impossible to unify cluster assignment procedure for different video types. Similarly to the previous group of methods, cluster analysis badly deals with homogeneous content. In addition, objects are often assigned to wrong clusters. Frames with identical texture and color scheme, but different content, may be treated the same way and not extracted. But unlike the previous group of methods, clustering enables to find centroid which defines frame importance most accurately.

Analyzing key frames extracted by curve simplification, it is notable that many frames extracted are alike. Additional procedures are needed to decline similar frames obtained. Some frames with really different content are not extracted.

For all the mentioned above techniques, the trick is that some users (experts or respondents) do not like to watch wrong frames extracted. They want to get only relevant frames, no matter how small their number will be. These users look for high precision. Others want to

get high recall. They agree to look through wrong frames extracted, they just want to obtain full information and do not miss any true key frame. For this reason it is very hard to estimate the above methods, but a good method assumes that precision gets lower with increase in the amount of key frames extracted.

To find better solution, we propose extracting key frames with the help of Voronoi diagrams and higher order Voronoi diagrams for detailing video content. In the next section we observe formal definitions for Voronoi tessellation and tessellation of higher order from a frame point of view. Section two shows the results of detailing video content with Voronoi diagrams of higher order, and the last section provides numerical estimation for key frame extraction using Voronoi diagrams. Conclusion is given at the end of the paper.

### 1. Formal Definitions

Initially designed for geodesy, Voronoi diagrams gained large popularity during the past years in computer graphics, especially for 3-D modeling [4, 5]. We propose a novel application for Voronoi diagrams of order- $k$ . The novelty lies in application of Voronoi tessellations as segments for comparison in sequential video to find frames with significant content and extract key frames from video to obtain short static summary. Order- $k$  Voronoi diagrams are to be used for detailing of video frame content.

A simple Voronoi diagram or an order-1 Voronoi diagram  $V$  is built on generator points  $\{p_1, p_2, \dots, p_n\}$  set a priori. The diagram corresponds to decomposition of a set (an image, in our case) into Voronoi tessellations  $\{v(p_1), v(p_2), \dots, v(p_n)\}$  according to the following rule:

$$v(p_i) = \{z \in \mathbb{R}^2 : d(z, p_i) \leq d(z, p_j) \forall i \neq j\} \quad (1)$$

where  $d(\circ, \circ)$  – planar Euclidean metric, Voronoi tessellation  $v(p_i)$ , corresponding to the generator point  $p_i$ , includes all the points  $z$ , distance to  $p_i$  from which is less than distance to the other generator points  $p_i$  with index  $j$  different from index  $i$  [6, 7].

Voronoi diagrams of higher order have been studied by many scientists since 1970. Among the most prominent are the works of Miles, Shamos, Aurenhammer, Agarwal etc. In context of Voronoi diagrams, the term “order” means the amount of generator points that form Voronoi tessellation, and “higher order” or “order- $k$ ” assumes that there are more than one generator point (in contrast to simple Voronoi diagrams where one point form a single Voronoi tessellation). Here, “higher order” does not have any relation to dimensionality of space [8], as all the video frames (images) lie in XY plane.

Voronoi diagram of order  $k$ ,  $V^{(k)}$ , that is built on  $n$  generator points in 2 dimensional space, is a division of a plane into convex polygons, such that points  $z$  of each Voronoi tessellation  $v_i^{(k)}$  have the same number of nearest generator points  $p_i$ , equal to  $k$ . The previously

mentioned simple Voronoi diagram is a particular case of order- $k$  Voronoi diagram with  $k=1$ .

To provide a formal definition for Voronoi tessellation of order  $k$ , assume that  $\{p_1, p_2, \dots, p_n\}$  is a set of generator points, and  $\{\{p_{1,1}, \dots, p_{1,k}\}, \dots, \{p_{l,1}, \dots, p_{l,k}\}\}$  is a set of subsets with  $k$  nearest generator points, then a convex order- $k$  Voronoi polygon  $v_i^{(k)}$ , formed by generator points  $\{p_{i,1}, \dots, p_{i,k}\}$ , can be written the following way:

$$v_i^{(k)} = \{z \in \mathbb{R}^2 : \max\{d(z, p_{i,h}), p_{i,h} \in v_i^{(k)}\} \leq \min\{d(z, p_{i,j}), p_{i,j} \in V^{(k)} \setminus v_i^{(k)}\}\} \quad (2)$$

In other words, the distance between the farthest point of one Voronoi tessellation to its corresponding generator points is closer or equals to the distance to any nearest generator point of another tessellation. Arbitrary Voronoi tessellation of order  $k$  may contain from 0 to  $k$  generator points, i.e. in a Voronoi tessellation of order  $k$  there may be no generator points at all [8, 9].

To compare consecutive video frames presented by Voronoi diagrams, we offer using partition metric  $\rho(V', V'')$  introduced in [10]. It provides understanding of how different Voronoi segments in consecutive frames  $B'(z)$  and  $B''(z)$  (with generator points  $\{p'_1, p'_2, \dots, p'_n\}$  and  $\{p''_1, p''_2, \dots, p''_m\}$  respectively) are.

$$\rho^*(B'(z), B''(z)) \approx \sum_{i=1}^n \sum_{j=1}^m \text{card}(v(p'_i) \Delta v(p''_j)) \times \text{card}(v(p'_i) \cap v(p''_j)) = \rho(V', V''), \quad (3)$$

where

$$v(p'_i) \Delta v(p''_j) = (v(p''_j) \setminus v(p'_i)) \cup (v(p'_i) \setminus v(p''_j)).$$

At the first stage of key frame extraction procedure we extract key frames  $B_r^*(z) \in S_l(i, j)$  (without limitation in number) within each shot  $S_l(i, j) = [B_i(z), B_j(z)]$ ,  $l=1, 2, \dots$ ,  $i, j \in L_l$ ,  $\sum L_l = Q$  ( $q=1, 2, \dots, Q$  is a discrete time of video lasting) according to the rule:

$$r = \arg \min_{r \in L_l} \left( \sum_{t \in L_l, r \neq t} \rho(B_r(z), B_t(z)) \right). \quad (4)$$

At the next stage, key frames from each shot are compared with each other to eliminate repeats or identical key frames.

Shots are detected by adaptive multidimensional time series model described in [11, 12]. Generator points are proposed to be set automatically with one of existing methods mentioned in [13]. During our experiments we have used Harris corner detector to obtain generator points. While performing key frame extraction procedure, area, location and shape have been used as spatial features. Also traditional colour and texture features have been used.

### 2. Experiments on Detailing Video Content

The increase of order  $k$  influences growth in number of Voronoi tessellations, i.e. detailing enhances. Experiments proved that at some step of detailing, it

becomes weaker under gradual increase of order  $k$  (see fig. 1). Moreover, detailing starts to fade not when the value of order  $k$  reaches its maximum (it reaches maximum when  $k = n - 1$ ), but much earlier.

Under  $k = n - 1$ , i.e. when Voronoi diagram order is the highest possible and equals to the total number of generator points minus one, the diagram is called furthest-site Voronoi diagram [9]. It is important to note that the threshold value for detailing reduction is different for different number of generator points.

Fig. 1 shows a frame from Trecvid video sampling collection with order- $k$  Voronoi diagrams for 15 generator points. The order is indicated at the top left of each diagram.



Fig.1. Order- $k$  Voronoi diagrams for 15 generator points

Fig. 2 shows averaged graphic that illustrates the dependence of Voronoi tessellation amount on the order of a diagram with 15 generator points. Though the amount of Voronoi tessellations depends on location of generator points in a plane, experiments held on Trecvid test collection proved that the amount of Voronoi tessellations varies in the interval of 20-30 % for diagrams of the same order with the same number of generator points.

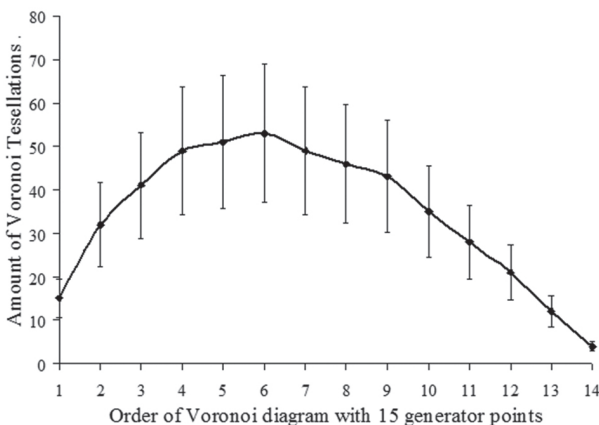


Fig. 2. The dependence of Voronoi tessellation amount on the order of a diagram with 15 generator points

It is important to note that the decrease in detailing usually happens much earlier than a threshold is reached, but, when it is reached, the number of points becomes less in comparison with not only the previous order but also with diagrams of any order including the first order (see fig. 2). In any case, detailing changes under parabola. First, it increases, and then fades smoothly reaching its threshold value near the highest order. Table 1 shows threshold values for 5-35 generator points with 5 point interval. Though, it has been experimentally proved irrational to have more than 20 generator points to construct Voronoi diagrams for further key frame extraction.

Table 1

Threshold values for Voronoi tessellation detailing reduction

Amount of generator points	Order $k$ under which threshold value for detailing reduction is reached
5	4
10	9
15	13
20	18
25	22
30	28
35	32

Thus, Voronoi diagrams of order  $k$  bring an excellent tool for detailing of images when they are to be compared with each other. Another variant is to select more generator points initially, but this influences much more computational complexity compared with construction of order- $k$  diagrams. This fact is easily explainable, as generator points are refined and Voronoi diagrams are rebuilt before comparison. The refinement procedure and building of simple order-1 Voronoi diagrams may not converge for a large number of points (sometimes just because of lack in memory resources). Moreover, it is not rational to have so many tessellations for all kind of images and to incorporate detailing in a usual procedure to be performed over time. The more points, the higher order, the greater computational complexity is.

### 3. Estimation of Key Frame Extraction Procedure

Estimation of results obtained after key frame extraction can be done under subjective opinion of respondents only, which gives an overview of their satisfaction level from video summary they have seen. Such kind of polling results is traditionally studied by the measures of precision and recall. Denote precision by  $P$  ( $P \in [0;1]$ ) and recall by  $R$  ( $R \in [0;1]$ ). Precision is the amount of found key frames that turned out relevant, and recall is the number of found relevant key frames from all the relevant key frames.

$$P = \frac{tp}{tp + fp}; R = \frac{tp}{tp + fn}, \tag{5}$$

where  $tp$ ,  $fp$ ,  $tn$  and  $fn$  can be clearly understood from the following table proposed by [14] while describing textual information search:

Table 2

Notations used for relevant and irrelevant key frames while calculating precision and recall

	Relevant (key frame)	Irrelevant (not a key frame)
Found ( $p$ )	Extracted key frame (true positive, $tp$ )	Not a key frame extracted (fault positive, $fp$ )
Not found ( $n$ )	Not extracted key frame (fault negative, $fn$ )	— (true negative, $tn$ )

The advantage in usage of two measures simultaneously is obvious. As mentioned above, sometimes it is more important to get small number of relevant frames only, but sometimes it is better to have a huge amount of even fault detected key frames. Recall does not get lower when the number of key frames detected arises.

In total, the task is to reach optimal balance for recall with satisfactory level of fault positive key frames. A measure that finds such a balance is called F-measure. If we put equal weights for precision and recall, we obtain balanced F-measure denoted by  $F_1$  ( $F_1 \in [0;1]$ ) that equals to Dice coefficient [14].

$$F_1 = \frac{2PR}{P + R} \quad (6)$$

Consequently, Dice coefficient can be used in order to estimate key frame extraction techniques. Fig. 3 illustrates key frames extracted by 4 different methods from Chinese commercial about Mercedes automobiles. For the purpose of estimation, optical flow has been calculated by Horn-Schunck method which analyses changes in motion field energy. It combines optimal balance between good quality and performance according to general rating of optical flow algorithms from Middleburry database [15].  $K$ -means has been chosen as clustering method. As for curve simplification method, we have used the same features for it (as for clustering method), but due to difference in procedures the results turned out also different.

Dice coefficient calculated for key frames extracted using Voronoi diagrams from Trecvid sampling collection, from several commercials and self-made high definition test samples (shot at the city centre), equals 0.92 at the average, which is better than Dice coefficient for optical flow method (0.65), and even better than Dice coefficient for clustering (0.78) and curve simplification (0.83) methods. The estimation was performed by 10 independent respondents who did not know the name of key frame extraction method they examined. The highest recall was obtained for optical flow method as too many frames were extracted. The highest precision was obtained for curve simplification method, but some key frames were omitted.

**Conclusion**

Tasks of video mining are in demand nowadays. Especially it concerns content based retrieval. And it is easily explained, as most of existing methods do not

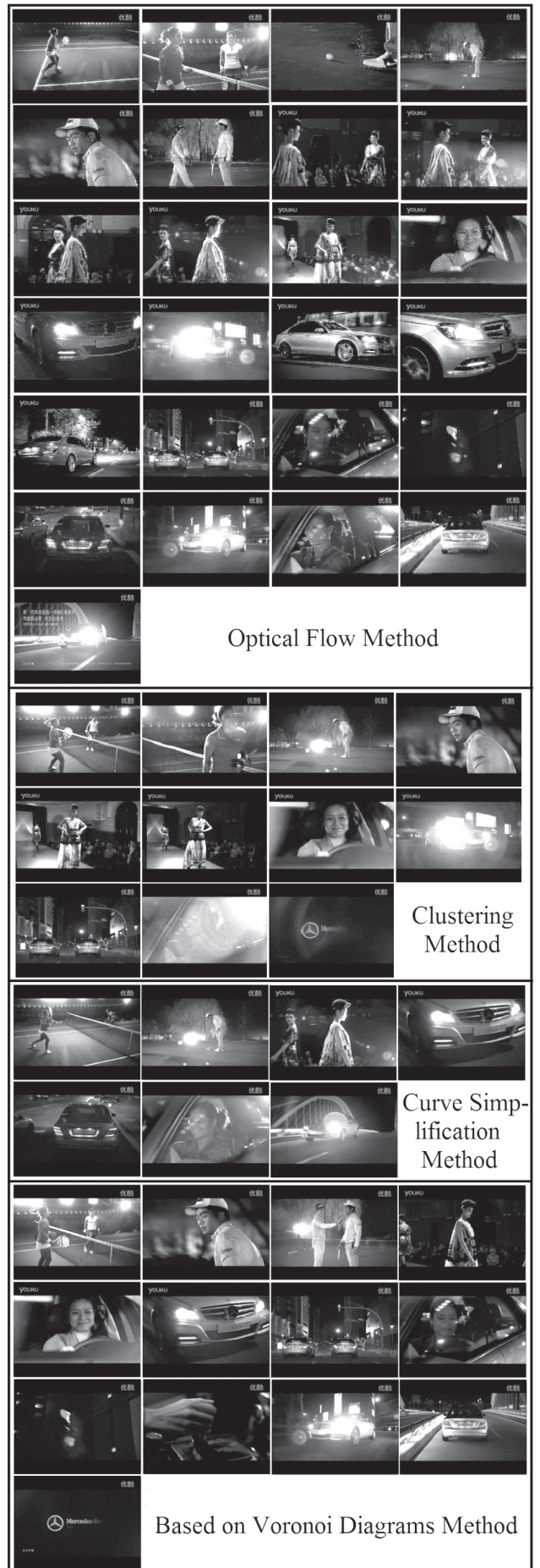


Fig.3. Key frames extracted by 4 different methods

show man-like processing results. This is true for video summarization. In this paper we propose a novel implementation of higher order Voronoi diagrams, found for geodesy, in static summarization of video. Video frame presentation with Voronoi diagrams and further comparison of them enabled to get machine interpretation of how objects are moved in space and time. Similar content in sequential frames turned out to have identical diagrams. Voronoi diagrams of higher order simplify detailing key frame content compared with increasing the number of initial generator points. Estimation of the proposed key frame extraction procedure shows high precision and recall.

**Bibliography:** **1.** Zhang D., Liu Y., Hou J. Digital Image Retrieval Using Intermediate Semantic Features and Multistep Search. In: Digital Image Computing: Techniques and Applications. – 2008. – pp. 513-518. **2.** Jiang Y.-G., Ngo C.-W., Yang J. Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval. In: 6<sup>th</sup> ACM International Conference on Image and Video Retrieval. – 2007. – pp. 494-501. **3.** Mikhnova O. A template-based approach to key frame extraction from video. In: International scientific and technical Internet conf. Computer Graphics and Image Recognition. – Vinnytsia: VNTU. – 2012. – pp. 120-127. **4.** Ledoux H., Gold C.M. The 3D Voronoi Diagram: A Tool for the Modelling of Geoscientific Datasets. In: ГійоCongris. – 2007. – 13 p. **5.** Carotte V. et al. Modelling and visualisation of fish aggregations using 3D Delaunay triangulation and alpha shapes. In: 8-th International Symposium on GIS and Computer Mapping for Coastal Zone Management. – 2007. – pp. 403-413. **6.** Du Q., Faber V., Gunzburger M. Centroidal Voronoi tessellations: Applications and algorithms. In: Society for Industrial and Applied Mathematics Review. – 1999. – Vol. 41, No. 4. – pp. 637-676. **7.** Hurtado F. et al. The weighted farthest color Voronoi diagram on trees and graphs. In: Computational Geometry: Theory and Applications. – 2004. – Vol. 27, No. 1. – pp. 13-26. **8.** Okabe A. et al. Spatial tessellations: Concepts and applications of Voronoi diagrams. – 2-nd ed. – Chichester: Wiley, 2000. – 671 p. **9.** Gavrilova M. L. Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence. In: Studies in Computational Intelligence. – Berlin: Springer. – 2008. – Vol. 158. – 304 p. **10.** Mashtalir V. et al. A novel metric on partitions for image segmentation. In: IEEE International Conference on Video and Signal Based Surveil-

lance. – 2006 – 6 p. **11.** Bodyanskiy Y. et al. Adaptive Video Segmentation via Non-stationary Multidimensional Time Series Analysis. In: International Conference on Applied and Theoretical Information Systems Research. – 2012. – 14 p. **12.** Bodyanskiy Y. et al. On-line video segmentation using methods of fault detection in multidimensional time sequences. In: International Journal of Electronic Commerce Studies. – 2012. – Vol. 3, No. 1. – pp. 1-20. **13.** Sebe N., Lew M.S. Comparing salient point detectors. In: Pattern Recognition Letters. – 2003. – Vol. 24, No. 1-3. – pp. 89–96. **14.** Manning C.D., Raghavan P., Schutze H. Introduction to Information Retrieval. – Cambridge: Cambridge University Press, 2008. – 496 p. **15.** Baker S. et al. A Database and Evaluation Methodology for Optical Flow. In: International Journal of Computer Vision. – 2011. – Vol. 92, No. 1. – P. 1–31.

Поступила до редколегії 04.12.2012

УДК 004.932.2

**Стабілізація описів ключових кадрів з допомогою діаграм Вороного більш високих порядків / С.В. Машталір, Е.Д. Михнова // Біоніка інтелекту: науч.-техн. журнал. – 2013. – № 1 (80). – С. 68-72.**

В статті розглядається актуальне напрямлення розпізнавання відео з урахуванням змісту. Реферування відео шляхом вилучення значимих статичних зображень, що відображають суть всього матеріалу, є темою досліджень. Автори здійснили спробу реалізувати процедуру пошуку ключових кадрів з допомогою діаграм Вороного. Діаграми Вороного більш високих порядків пропонується використовувати при деталізації змісту відеокадрів.

Ил. 3. Бібліогр.: 15 назв.

УДК 004.932.2

**Стабілізація описів ключових кадрів за допомогою діаграм Вороного більш високих порядків / С.В. Машталір, Е.Д. Михнова // Біоніка інтелекту: науч.-техн. журнал. – 2013. – № 1 (80). – С. 68-72.**

У статті розглядається актуальний напрям розпізнавання відео з урахуванням змісту. Реферування відео шляхом вилучення значимих статичних зображень, що відображають суть всього матеріалу, є темою досліджень. Автори зробили спробу реалізувати процедуру пошуку ключових кадрів за допомогою діаграм Вороного. Діаграми Вороного більш високих порядків пропонується використовувати при деталізації змісту відеокадрів.

Ил. 3. Бібліогр.: 15 найм.

УДК 518.81



О.А. Писклакова

НУЦЗУ, м. Харків, Україна, pisklakova@ukr.net;

## МОДЕЛИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В СИЛЬНО ЦЕНТРАЛИЗОВАННЫХ СИСТЕМАХ

Проведен анализ особенностей решения задач многокритериального принятия решений в условиях неопределенности. Предложены модели выбора компромиссного решения в условиях многокритериальности и различных типов интервальной неопределенности при решении задач распределительного типа в системах с сильной степенью централизации.

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ, МНОГОКРИТЕРИАЛЬНОСТЬ, ИНТЕРВАЛЬНАЯ НЕОПРЕДЕЛЕННОСТЬ, ОПТИМИЗАЦИЯ, РЕСУРС, ЦЕНТРАЛИЗАЦИЯ СИСТЕМЫ

### Введение

Одной из наиболее важных проблем, возникающих в различных областях человеческой деятельности (технической, экономической, организационной и др.), является проблема совершенствования управления. Многие функциональные задачи организационного управления являются задачами распределения ресурсов. Следовательно, в инструментарии управления должны входить методы эффективного решения таких задач, проблемно-ориентированных на особенности объекта управления. Это означает, что необходимо разработать единое математическое описание и методы решения широкого класса задач распределения ресурсов, в частности, с учетом нелинейности производственных функций элементов системы, различного вида неопределенностей, многокритериальности, разной степени централизации системы.

Экстремальные задачи распределения ресурсов возникают в связи с тем, что объемы ресурсов являются ограниченными, и это приводит к конфликтным ситуациям.

При этом особенностью задач распределения ресурсов также является то, что во многих случаях их приходится решать в условиях неполноты исходной информации (неопределенности) и многокритериальности целевых функций. Неучет этих особенностей приводит к некорректным решениям, не имеющим практической ценности. В статье рассматриваются подходы к решению задачи в указанной постановке.

### 1. Особенности задачи распределения ресурсов

Задача, в которой требуется наилучшим образом, в смысле выбранного критерия оптимальности, распределить ограниченные ресурсы по нескольким видам деятельности (потребителям) называется задачей распределительного типа [1].

В дальнейшем будем полагать, что в качестве распределяемого рассматривается моноресурс. Это не снижает общности полученных результатов, так как если решается задача распределения комплекса взаимосвязанных ресурсов, то для него можно

или сформировать скалярную многофакторную оценку (например, в денежном выражении) или выделить критический (наиболее дефицитный) ресурс и по нему принимать решение.

Получив ресурс  $d$ ,  $i$ -й потребитель производит некоторый эффект  $E_i$

$$E_i = F_i(d_i), \quad (1)$$

при этом оператор  $F_i$ , устанавливающий связь между входом ( $d_i$ ) и выходом ( $E_i$ ), называют производственной функцией.

В целом система получает суммарный эффект  $E_{\Sigma}$ , который стремится максимизировать производственную функцию

$$E_{\Sigma} = \max_{r_i \in R} Q[F_i(r_i)] \quad (2)$$

при ограничении

$$\sum d_i \leq D, \quad (3)$$

где  $D$  — общее качество распределяемого ресурса.

В общем случае, эффекты  $E$  генерируемые как подсистемами (потребителями ресурсов), так и системой в целом, являются набором разнородных эффектов  $j$ -го вида, т.е.

$$E_i = \langle E_{ij} \rangle, \quad i = \overline{1, n}; \quad j = \overline{1, m}. \quad (4)$$

Еще одна особенность заключается в том, что распределительные задачи в большинстве случаев приходится решать в условиях интервальной неопределенности.

Основными источниками неопределенности являются:

— неполнота знаний о виде и параметрах производственных функций  $F_i$  и  $Q$ ;

— неопределенности, возникающие при формулировании скалярных многофакторных оценок [2] эффекта;

— неточности задания системы ограничений;

Общими особенностями задач распределения ресурсов является наличие в них целевой функции (критерия оптимальности) (2), а также ограничений на объем ресурсов (3) и на значения оптимизируемых переменных.

Для решения задач распределения ресурсов известные аналитические методы математического

программирования, основанные на исследовании производных целевой функции, часто оказываются непригодными в силу наличия сильных ограничений на переменные и область изменения целевой функции. Метод полного перебора всех возможных вариантов решения задач распределения ресурсов также находит ограниченное применение в силу большой размерности практически важных задач.

Пусть задана иерархическая двухуровневая система центр – комплекс подсистем  $A = \{A_v\}$ ,  $v = \overline{1, N}$ . В текущий момент времени центр располагает некоторым количеством ресурса  $D(t)$ . Каждой подсистеме для нормального функционирования требуется количество ресурса  $d_v(t)_{\min}$ , а для экстремального по заданному критерию –  $d_v(t)_{\max}$ . При этом

$$\sum_{v=1}^N d_v(t)_{\min} < D(t) < \sum_{v=1}^N d_v(t)_{\max}. \quad (5)$$

Это означает, что

$$\Delta D(t) = D(t) - \sum_{v=1}^N d_v(t)_{\min} \quad (6)$$

являются ресурсами, которые можно инвестировать в развитие производства.

Получая ресурс, каждая подсистема производит набор разнокачественных эффектов  $\Theta_v$ , компонентами которого являются, например, экономический, социальный, экологический эффекты, каждый из которых в свою очередь определяется набором характеристик. Предположим существование такой оценки  $\overline{\Theta}_v$ , что

$$\overline{\Theta}_v(t) = H_v(\Delta d_v, t), \quad (7)$$

а система в целом получает набор разнокачественных эффектов

$$\overline{\Theta}_v(t) = \langle \overline{\Theta}_v(t) \rangle, \quad v = \overline{1, N}. \quad (8)$$

В общем случае эффект центра  $\Theta_c$  не совпадает с эффектом системы и в зависимости от степени централизации осуществляется выбор оператора  $H_c(t)$

$$\Theta_c(A) = H_c[\Theta(t)]. \quad (9)$$

Ресурс между подсистемами распределяется по фиксированному правилу

$$d_v(t) = F_v[\Theta_v(t), t] \quad (10)$$

при условиях

$$\sum_{v=1}^N d_v(t) \leq \Delta D(t); \quad (11)$$

$$d_v(t)_{\min} \leq d_v(t) \leq d_v(t)_{\max}. \quad (12)$$

Цель центра – максимизация его эффекта

$$\Theta_c(t) \rightarrow \max_{d_v}. \quad (13)$$

Постановка рассмотренной формальной задачи не отличается принципиально от классической задачи распределения ресурсов. Ее конкретные

особенности, определяющие степень универсальности и метод решения зависят от вида операторов  $H_v(t)$ ,  $H_c(t)$  и правила (10). В статье рассматриваются эти особенности без учета зависимости процесса от времени.

В экономике зависимость выпуска от затрат ресурсов вида (1), (2) называется производственной функцией. Обычно предполагают, что такие функции – линейные или неубывающие выпуклые вверх или вниз зависимости, одинаковые для всех элементов. Такое допущение существенно упрощает вычисления, но не отражает действительности. В общем случае можно считать, что на интервале  $d = [0, \infty]$  производственная функция (1) имеет вид монотонно возрастающей S-образной кривой, показанной на рис.1. Последняя включает в себя вогнутый (1), линейный (2) и выпуклый (3) участки.

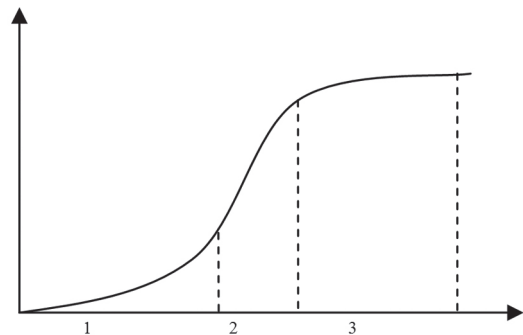


Рис. 1. Производственная функция

Чтобы облегчить вычисления, желательно выбрать универсальную форму производственной функции, позволяющую изменением одного параметра реализовать кривые всех трех указанных видов. С этой целью введем вспомогательную функцию вида

$$\varphi(d) = \left( \frac{d - d_{\min}}{d_{\max} - d_{\min}} \right)^\alpha, \quad (14)$$

где  $\alpha$  – параметр нелинейности. В этом случае при  $0 < \alpha < 1$  реализуются выпуклые зависимости, при  $\alpha = 1$  – линейная и при  $\alpha > 1$  – вогнутые. Функция (14) безразмерна и изменяется в пределах от нуля до единицы, а  $d$  может изменяться непрерывно и дискретно. Выходной эффект  $v$ -го элемента представляется так:

$$\Theta_v = \Theta_{v \min} + b_v \varphi_v(d_v). \quad (15)$$

Здесь  $\Theta_{v \min} = H_v(d_{v \min})$  – эффект, получаемый подсистемой при предоставлении ей ресурса в количестве  $d_{v \min}$ ;  $b_v$  – масштабный коэффициент, трансформирующий безразмерную функцию (14) в реальный эффект.

Эффект центра равен  $\Theta_A = \{\Theta_{cv}\}$ ,  $v = \overline{1, N}$ , где  $\Theta_{cv}$  – в общем случае разнокачественные эффекты, которые монотонно зависят от агрегатов. С учетом равенства (7)

$$\Theta_{cv} = H'_{cv}[\Theta_{v \min} + b_v \varphi_v(d_v)] = H_{cv}(d_v). \quad (16)$$

Так как функции  $H'_{cv}$  и  $\varphi_v(d_v)$  монотонные, возрастающие, выпуклые вверх или вниз, зависимость  $H_{cv}(d_v)$  согласно свойствам суперпозиции является монотонной, возрастающей и в соответствии с видом исходных функций может быть линейной, выпуклой или вогнутой. Поэтому, по аналогии с эффектом агрегата разнокачественный эффект можно представить в виде:

$$\Theta_{cv} = \Theta_{cv\min} + c_v \varphi_{cv}(d_v). \quad (17)$$

Здесь  $c_v$  — масштабный коэффициент;  $\varphi_{cv}(d_v)$  — функция вида (14).

Поскольку  $\Theta_{cv}$  — разнокачественные эффекты, задача максимизации эффекта центра представляет собой типичную задачу многокритериальной оптимизации, причем роль локальных критериев играют  $\Theta_{cv}$ , а  $\varphi_{cv}(d_v)$  — функции полезности этих критериев. Тогда в рамках аддитивной теории полезности обобщенная полезность эффекта центра выражается следующим образом:

$$\bar{\Theta}_c = \sum_{v=1}^N g_v \Theta_{cv} = \sum_{v=1}^N g_v \Theta_{cv\min} + \sum_{v=1}^N g_v c_v \varphi_{cv}(d_v), \quad (18)$$

где первое слагаемое — гарантированный минимальный доход центра, а управлению поддается только часть эффекта, составляющая которого представлена в виде:

$$\Delta \bar{\Theta}_c = \Delta \bar{\Theta} - \bar{\Theta}_{c\min} = \sum_{v=1}^N g_v b_v \varphi_{cv}(d_v). \quad (19)$$

В этих формулах  $g_v$  — коэффициент изоморфизма, учитывающий относительный вес (значимость) разнокачественных эффектов.

Правило распределения ресурсов зависит от особенностей системы. В основе классификации таких особенностей лежит степень централизации и согласованности целей элементов системы.

В классе централизованных систем возможны следующие ситуации: подсистемы не целеустремленные; подсистемы с сильной централизацией; подсистемы со слабой централизацией.

Для рассматриваемого класса организационных систем характерна централизованная структура. Следовательно, как видно из проведенного анализа, все ситуации можно свести к двум случаям [3]:

- распределению ресурсов по критерию  $\Delta \bar{\Theta}_c \rightarrow \max_{d_v}$  для систем с сильной централизацией;
- определению поведения центра, максимизирующего критерий  $\Delta \bar{\Theta}_c \rightarrow \max_{d_v}$  при фиксированном правиле распределения ресурсов для систем со слабой централизацией.

Рассмотрим возможные модели распределения ресурсов в сильно централизованных системах.

## 2. Постановка задачи

При распределении ресурсов в сильно централизованных системах необходимо с учетом

заданных ограничений получить максимальный эффект от распределения некоторого количества ресурса  $d_v(t)$ :

$$\sum_{v=1}^N g_v c_v \varphi_{cv}(d_v) \rightarrow \max_{d_v}, \quad (20)$$

причем, согласно выражению (14)

$$\varphi_{cv}(d_v) = \left( \frac{d_v - d_{v\min}}{d_{v\max} - d_{v\min}} \right)^{\alpha_{cv}} \quad (21)$$

при ограничениях (11), (12).

В общем случае для этой задачи характерны такие особенности: нелинейность функционала (20), обусловленная нелинейностью зависимости (21); неопределенность (неточность задания) функционала по значениям коэффициентов  $g_v, c_v$  и виду функции  $\varphi_{cv}$ , т.е. по значениям параметра  $\alpha_{cv}$  из формулы (21).

Первый вид неопределенности обусловлен неточностью идентификации весовых коэффициентов  $g_v, c_v$  при формировании скалярной многофакторной оценки обобщенного эффекта, а второй — неполнотой знаний о виде производственных функций, что приводит к неточности задания параметра  $\alpha_{cv}$  функции (21).

Если первая особенность не принципиальна и здесь задача заключается в выборе эффективного по затратам вычислительных ресурсов и точности численного метода отыскания глобального экстремума функционала (20), то преодоление трудностей, обусловленных неопределенностью такого функционала, связано с обоснованием принципов принятия решений и построения соответствующих математических моделей и алгоритмов.

## 3. Определение эффективного решения в условиях неопределенности

Особенность задачи заключается в том, что вид производственной функции центра задан неточно, с большей или меньшей интервальной неопределенностью  $\Delta \alpha_{cv}, \Delta c_v, \Delta g_v$ . Это приводит к тому, что в зависимости от конкретного сочетания значений параметров  $\alpha_{cv} \in [\alpha_{cv}^{\max}, \alpha_{cv}^{\min}]$ ,  $c_v \in [c_v^{\max}, c_v^{\min}]$ ,  $g_v \in [g_v^{\max}, g_v^{\min}]$  на множестве допустимых решений, определяемых ограничениями (11), (12), получаем некоторое подмножество экстремальных по критерию (20) решений  $\bar{D}_i^0 = \{d_{vi}^0\}$ ;  $v = \overline{1, N}$ ;  $i = \overline{1, n}$ , где  $n$  — число возможных комбинаций значений параметров  $\alpha_{cv}, c_v, g_v$ . Необходимо выбрать из этого подмножества решений единственное  $\bar{D}^0$ .

Эффективное решение определяется с помощью двухэтапной процедуры: сначала выделяется подобласть допустимого множества решений, соответствующая вариациям параметров оптимизируемого функционала  $\bar{D}^0$ ,  $i = \overline{1, n}$ , а затем из этого подмножества выбирается единственное решение. Для выбора единственного решения наиболее

приемлем минимаксный критерий, который принимает вид [4]

$$K = \min_D \left\{ \sum_{i=1}^n [a_i \varphi_i(\bar{D}_i^0)]^\beta \right\}^{1/\beta}, \quad i = \overline{1, n}, \quad \beta > 1, \quad (22)$$

где  $n$  — число локальных экстремальных решений;  $a_i$  — коэффициент, оценивающий вес  $i$ -го решения, его значение определяется в каждом конкретном случае по информации о предпочтительности значений параметров функционала;  $\varphi_i(\bar{D}_i^0)$  — функция полезности  $i$ -го решения.

Распределения ресурсов в условиях полной неопределенности вида производственных функций агрегатов.

В данном случае функционал (20) не задан. Ни одной точки допустимого множества решений, определяемого ограничениями (11), (12), нельзя отдать предпочтение, они равноценны. Но объективно существуют конкретные производственные функции, а следовательно, есть эффективное решение, причем оно может лежать в любой точке допустимого пространства. Поэтому рационально в качестве решения выбрать точку, минимизирующую максимально возможные потери эффективности вследствие несовпадения выбранного решения с объективно эффективным. Таким решением является точка, минимально удаленная от границ допустимой области [5]. С учетом сказанного в функционале (22)  $a_i = 1$ , а значения функций  $\varphi_i(\bar{D}_i^0)$  имеют смысл нормированного расстояния от границ допустимого множества решений по каждой переменной  $d_v$ :

$$\varphi_i(d_v) = \frac{d_v^0 - d'_{v \min}}{d'_{v \max} - d'_{v \min}}, \quad i = \overline{1, N}, \quad v = \overline{1, N}. \quad (23)$$

В общем случае  $d'_{v \min} \geq d_{v \min}$ ;  $d'_{v \max} \leq d_{v \max}$ . Определение значений  $d'_{v \min(\max)}$  даже в многомерном пространстве не представляет труда:

$$d'_{v \max} = \begin{cases} d_{v \max} \setminus d_{v \max} \leq D - \sum_{i=1}^N d_{i \min}; \\ D - \sum_{i=1}^N d_{i \min} \setminus d_{v \max} > D - \sum_{i=1}^N d_{i \min}, \\ v = \overline{1, N}; j = \overline{1, N}; j \neq v; \end{cases} \quad (24)$$

$$d'_{v \min} = \begin{cases} d_{v \min} \setminus \sum_{i=1}^N d_{j \max} \geq D - d_{v \min}; \\ D - \sum_{i=1}^N d_{i \max} \setminus \sum_{i=1}^N d_{j \max} < D - d_{v \min}, \\ v = \overline{1, N}; j = \overline{1, N}; j \neq v; \end{cases} \quad (25)$$

Ограничения принимают вид

$$d'_{v \min} \leq d_v \leq d'_{v \max}; \quad \sum_{v=1}^N d_v = D. \quad (26)$$

При такой высокой неопределенности не нужна большая «жестокость» нахождения минимаксного решения, поэтому в формуле (21) примем  $\beta = 2$ , что обеспечивает достаточно высокую точность выравнивания значений переменных.

### Выводы

В статье предложены модели распределения ресурсов между подсистемами предприятия с учетом многокритериальности эффектов и неопределенности производственных функций с учетом степени централизации системы, т.е. степени согласованности целей подсистем и центра.

**Список литературы:** 1. *Автоматизированные системы управления городским хозяйством* [Текст] / И.В. Кузьмин, Э.Г. Петров, И.А. Алферов, В.В. Евсеев, Л.В. Мигунова. — Киев, «Будівельник», 1978. — 144 с. 2. *Пискалова О.А.* Анализ особенностей решения задачи многокритериальной оптимизации в условиях неопределенности [Текст] / О.А. Пискалова, Н.А. Брынза, Д.И. Филипская // Системные технологии. Региональный межвузовский сборник научных работ — Выпуск 3 (56). — Днепропетровск, 2008. — №01. — С. 147-157. 3. *Крянев А.В.* Основы финансового анализа и портфельного инвестирования в рыночной экономике [Текст] / А.В. Крянев. — М.: МИФИ, 2001. — 54 с. 4. *Вентцель Е.С.* Теория вероятностей и ее инженерные приложения [Текст] / Е.С. Вентцель, Л.А. Овчаров. — М.: Высшая шк., 2000. — 480 с. 5. *Бурков В.Н.* Теория активных систем: состояние и перспективы [Текст] / В.Н. Бурков, Д.А. Новиков. — М.: Синтег, 1999. — 128 с.

Поступила в редколлегию 18.12.2012

УДК 518.81

**Моделі розподілу ресурсів у сильно централізованих системах** / О.О. Пискалова // Біоніка інтелекту: наук.-техн. журнал. — 2013. — № 1 (80). — С. 73-76.

У статті обґрунтовані моделі рішення задач з розподілу ресурсів в умовах багатокритеріальності й невизначеності вхідних даних. Затверджується що у запропонованих моделях враховується багатокритеріальність ефектів та невизначеність виробничих функцій.

Л. 1. Бібліогр.: 5 найм.

UDK 518.81

**The models of resource allocation in highly centralized systems** / O.A. Pisklakova / Bionics of Intelligense: Sci. Mag. — 2013. — № 1 (80). — P. 73-76.

In the article based models problem solving with resource allocation in bahatokryterialnosti and uncertainty of input data. Approved that the proposed model takes into account the uncertainty bahatokryterialnist effects and production functions.

Fig. 1. Ref.: 5 items.

УДК 004(4'242+053)



С.И. Чайников<sup>1</sup>, А.С. Солодовников<sup>2</sup>

<sup>1</sup>ХНУРЭ, г. Харьков, Украина;

<sup>2</sup>ХНМУ, г. Харьков, Украина

## ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ГРАФ-МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ

Рассмотрено формализованное описание граф-модели предметной области и элементов диалоговой системы. Описаны алгоритмы обработки граф-модели, приводящие граф к форме, необходимой для обеспечения функций контроля и управления работой диалоговой системы.

ДИАЛОГОВЫЕ СИСТЕМЫ, ГРАФ-МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ, ГРАФ ГЕРЦА, КОНЕЧНЫЙ АВТОМАТ

### Введение

Большинство современных информационных систем (ИС) разрабатывается с учетом модульного принципа, что дает ряд преимуществ, например, в выборе языка программирования или локализации мест ошибок при отладке программного средства (ПС). Учитывая модульный принцип, структура ПС легко описывается с использованием теории графов, а функциональные особенности – на основе теории автоматов (как показывает анализ литературы). Также графы применимы при описании предметных областей для ИС. Тем не менее, всегда остается проблема оптимального построения ПС и моделей предметных областей (ПрО). Такое описание должно обеспечивать возможность контроля над выполнением вычислений, позволять видоизменять саму структуру программы и осуществлять сквозное управление данными в ИС. В связи с этим целью работы является формализованное описание граф-модели ПрО и элементов диалоговой системы (ДС), описанной в [1].

### 1. Формальное описание модели диалоговой системы

Структура ДС описывается с помощью ориентированного графа, приведенного к ярусно-параллельной форме (ЯПФ) [2]. Такая форма позволяет осуществить параллельную организацию вычислений и распределить вычислительные процессы на компьютерных кластерах, к примеру, гетерогенных кластерах, или многопроцессорных системах, относящихся к классам SMP и NUMA. Однако описание структуры программы с помощью канонической ЯПФ приводит к появлению такой проблемы, как одновременная рассылка одних и тех же данных по многим процессорам (или рабочим станциям), что влечет за собой увеличение времени работы ПС. Также появляется проблема организации многопользовательского доступа к общим массивам данных. Прежде чем приступить к решению указанных проблем, необходимо выполнить формальное описание модели ДС на базе орграфа.

Пусть задан граф (рис. 1):

$$G = (V, X), \tag{1}$$

где  $V$  – множество вершин  $v_i$ ;  $X$  – множество дуг  $x_{ij} = (v_i, v_j)$  графа  $G$ , соединяющих вершины  $v_i$  и  $v_j$  между собой, для которых задано направление ( $i, j = \overline{1, n}$ , где  $n$  – количество вершин графа  $G$ ).

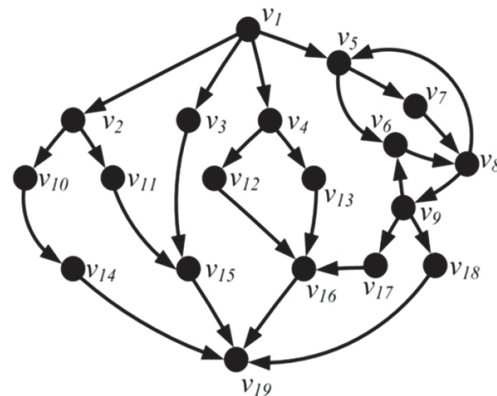


Рис. 1. Граф-модель ИС

Множеству вершин  $V$  сопоставляется множество макроопераций  $M$ , которые выполняются определенными программными модулями. Множеству дуг сопоставляется множество потоков данных  $D$  между модулями. При этом каждая дуга  $x_{i,j} \in X$  характеризуется парой  $(d_{ij}, t_{ij})$ , где  $d_{i,j} \in D$  – пересылаемые данные между  $v_i$  и  $v_j$  вершинами, а  $t_{ij}$  – время, необходимое на пересылку этих данных. Каждый программный модуль  $m_k \in M, \forall k = \overline{1, n}$ , где  $n$  – общее количество модулей, в свою очередь, характеризуется парой  $(n_k, s_k)$ , где  $n_k$  – имя модуля,  $s_k$  – его размер. Размер модуля определяется количеством элементарных операций, входящих в его состав. На орграфе задается входная вершина  $v_1$ , для которой  $deg^+(x_0) = 0$  и выходная вершина  $v_n$ , для которой  $deg^-(x_n) = 0$ . При этом для указанных вершин выполняются условия, такие что каждая вершина  $v_i \in V, \forall i = \overline{1, n}$  достижима из входа  $v_1$  и каждая вершина  $v_i \in V, \forall i = \overline{1, n}$  достигает выхода  $v_n$ . К орграфу (1) предъявляется требование ацикличности, отсутствия парных ребер и петель. Эти требования обеспечиваются заданием уровня детализации граф-модели. Формально это выглядит

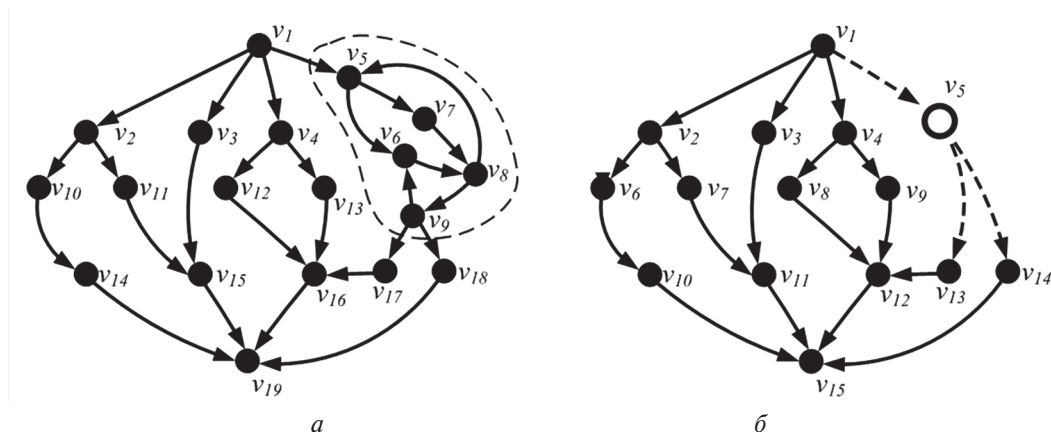


Рис. 2. Макропроектирование структуры ИС

следующим образом. На заданном графе (рис. 2, а) выделяются области с циклами  $\mu [v_b, v_e]$ , где  $v_b$  – вершина начала цикла,  $v_e$  – его конец. Далее все вершины данного цикла исключаются из графа и заменяются одной вершиной  $v_p$ , для которой индекс  $p$  (рис. 2, б) определяется как минимальный номер вершины, которая входит в цикл. У всех остальных вершин индекс заменяется по правилу: если  $ind > p$ , тогда  $Nind = ind - p + 1$ , где  $ind$  – текущий индекс вершины,  $Nind$  – новый индекс вершины,  $p$  – номер вершины, соответствующей исключенному из графа циклу. По этому принципу первоначальный граф, описывающий предметную область, преобразуется в граф конденсации, называемый графом Герца [3]. Могут возникать случаи, когда цикл не начинается вовсе или исполнение цикла прерывается по каким-либо причинам и переход по связи происходит в другую вершину, которая не принадлежит множеству вершин заданного цикла. В первом случае  $v_b$  будет совпадать с  $v_e$ . Во втором случае в процессе конденсации необходима дополнительная дуга, учитывающая возможность прерывания цикла. После того, как граф приведен к ациклической форме, на нем выполняется алгоритм приведения к ЯПФ (рис. 3):

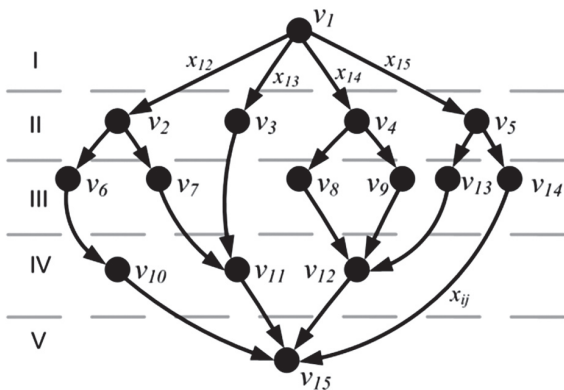


Рис. 3. Ярусно-параллельная форма графа

1) просматривается матрица смежности графа и в очередной ярус выбираются вершины с нулевой полустепенью захода, соответствующие нулевым столбцам;

- 2) строки, соответствующие выбранным на предыдущем шаге вершинам, обнуляются;
- 3) осуществляется возврат к шагу 1, до полного исчерпания матрицы смежности;
- 4) прорисовываются дуги.

Учитывая тот факт, что между модулями существуют связи, и один модуль может использовать переменные, константы и программный код другого модуля, при компиляции модулей в единую ИС, необходимо формально задать требования к структуре каждого программного модуля в виде множества спецификаций. Спецификацией программы будем называть набор требований к ней. Пусть  $P$  – программа на некотором языке программирования, а  $Sp$  – спецификация. Задача верификации программы  $P$  относительно спецификации  $Sp$  состоит в определении того, что  $P$  соответствует всем требованиям, содержащимся в  $Sp$ . Другими словами, для  $i$ -го программного модуля определяется спецификация  $Sp_i \in SPC, \forall i = \overline{1, n}$ , где  $SPC$  – множество спецификаций, вида:

$$Sp_i = \{D^{in}, D^{out}, T^{in}, T^{out}, R^{in}, R^{out}, F\}, \quad (2)$$

где  $D^{in}$  – множество входных массивов данных;  $D^{out}$  – множество выходных массивов данных;  $T^{in}$  – множество типов входных данных;  $T^{out}$  – множество типов выходных данных;  $R^{in}$  – множество правил ввода входных данных модуля;  $R^{out}$  – множество правил вывода выходных данных модуля;  $F$  – множество управляющих воздействий  $i$ -го модуля.

Следующим этапом является анализ ПрО. Выделяются основные элементы и связи между ними. Такой анализ приводит к построению граф-модели ПрО вида

$$Pr = (Vpr, Xpr), \quad (3)$$

где  $Vpr$  – множество вершин графа ПрО, которым соответствуют подсистемы объекта автоматизации (функциональная подсистема), а  $Xpr$  – множество ориентированных дуг, соединяющих вершины, которым соответствуют потоки данных между подсистемами. Так как топологии при заданных графах (1) и (3) должны соответствовать друг другу, выполняется функтор отображения вида:

$$\vartheta : Pr \rightarrow G . \quad (4)$$

Для обеспечения хранения информации о выполненных вычислениях, текущем состоянии ИС и предоставления этой информации по запросам как пользователя, так и некоторого программного модуля, используются кортежи данных.

Кортеж данных  $C$  для каждой вершины  $v_i$ , который в дальнейшем будет базой для организации хранения информации о работе ИС в базе данных, определяется следующим образом:

$$C_i = \{User\_ID, Time\_Op, Num\_V, Data\_Res, Data\_Inp, PCall\_Par\} , \quad (5)$$

где  $User\_ID$  – идентификатор пользователя, активировавшего процедуру;  $Time\_Op$  – время активации процедуры;  $Num\_V$  – номер вершины, инициализирующей вычислительный процесс;  $Data\_Res$  – ссылка на массив данных, полученных в результате вычислений  $i$ -ой процедуры;  $Data\_Inp$  – ссылка на входные массивы данных для  $i$ -ой процедуры;  $PCall\_Par$  – параметры вызова  $i$ -ой процедуры. Входные и выходные данные процедуры записываются в виде векторов данных:

$$Data\_Res = \{d_q^{RES}\}, \forall q = \overline{1, Q} \quad (6)$$

$$Data\_Inp = \{d_p^{INP}\}, \forall p = \overline{1, P} , \quad (7)$$

где  $Q, P$  – количество всех выходных и входных элементов данных. Сформированное формальное описание ИС и ПрО используется далее для построения детальных графовых моделей (таких как информационный граф или граф управления). В дальнейшем по полученным моделям производится построение и отладка программного средства.

## 2. Графовые модели

Выделяют четыре основные графовые модели: граф управления, информационный граф, операционно-логическая история и история реализации [2]. Первые две модели не зависят от входных данных и строятся непосредственно по тексту программы. Две последние модели для своего построения формально требуют слежения за выполнением всех операндов. Сложность построения модели возрастает в порядке указанного перечисления. Достоинством всех моделей является то, что они существуют для всех программ.

Если задать вектор исходных данных (7) и проследить за выполнением программы, зафиксировав каждое срабатывание оператора отдельной вершиной, можно получить ориентированный граф, называемый историей реализации программы. Объединяя такие графы на различных векторах входных данных (7), получаем информационный граф. Такой граф представляет взаимодействие данных и команд. Информационный граф описывает последовательность выполнения операций и взаимную зависимость между различными

макрооперациями или блоками операций. Узлами информационного графа являются макрооперации, а однонаправленными дугами – каналы обмена данными [4]. Принцип организации вычислений в ИС подразумевает поиск зависимостей от данных других программных модулей для текущей, активной, процедуры, которая запущена пользователем. Для реализации данного принципа требуется наличие информационного графа (рис. 4, а), на котором выделяется подграф (рис. 4, б), позволяющий описать зависимость программных модулей.

Формально этот алгоритм записывается следующим образом. Пусть задан граф

$$G' = (V', X') , \quad (8)$$

где  $V' \subset V, X' \subset X$ ;  $V$  – множество вершин;  $X$  – множество дуг графа (1), а матрицы  $A = \|a_{ij}\|$  и являются матрицами смежности для графа (1) и (8) соответственно. Тогда для выделения подграфа (8) на графе (1) необходимо:

- 1) запомнить индекс текущей вершины  $k$ , для которой определяются зависимости;
- 2) найти все элементы  $a_{ij} \neq 0$  матрицы  $A$  при  $i = k, \forall j = \overline{1, n}$ , а затем переписать их в матрицу  $A'$  при соответствующих  $i$  и  $j$ ;
- 3) запомнить индексы  $j$  для всех найденных вершин и, перебирая значения  $k = j$  повторить шаг 2;
- 4) выполнять шаги пока не будет найден индекс для вершины со значением  $deg^+(x_0) = 0$ .

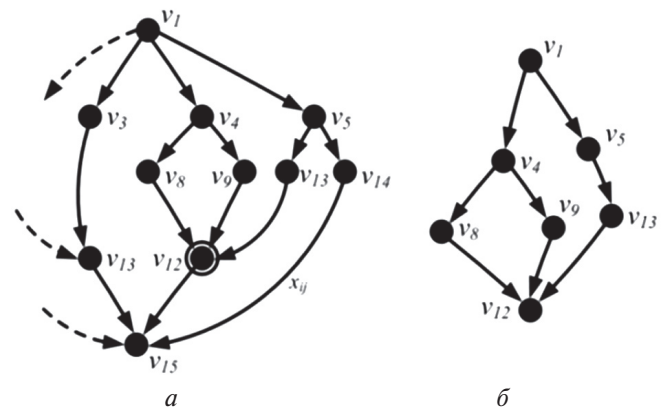


Рис. 4. Выделение подграфа для активной вершины

По окончании алгоритма будет сформирован подграф (8) с матрицей смежности, позволяющий в дальнейшем сформировать граф истории реализации программы (рис. 4, б).

Полученные графовые модели ИС позволяют определить тип модели параллельного программирования, используемой при разработке ИС.

## 3. Модели параллельного программирования

В зависимости от архитектур вычислительных систем и инструментальных средств, применяемых для разработки программных средств, выделяют различные модели параллельного программирования [2]:

Модель передачи сообщений. Программа порождает несколько задач, каждой задаче присваивается свой уникальный идентификатор, взаимодействие осуществляется посредством отправки и приема сообщений, новые задачи могут создаваться во время выполнения параллельной программы, несколько задач могут выполняться на одном процессоре.

Модель параллелизма данных. Одна операция применяется ко множеству элементов структуры данных, программа содержит последовательность таких операций, зернистость вычислений мала. Программист в этом случае указывает транслятору, как данные следует распределять между задачами.

Модель общей памяти. Задачи обращаются к общей памяти, имея общее адресное пространство и выполняя операции считывания/записи. Управление доступом к памяти осуществляется с помощью разных механизмов, таких, например, как семафоры. В рамках этой модели не требуется описывать обмен данными между задачами в явном виде, что упрощает программирование.

Основываясь на сформированных спецификациях программных модулей (2) и учитывая древовидную структуру ИС, можно сделать вывод, что наиболее оптимальной моделью программирования будет являться модель передачи сообщений, которая минимизирует нагрузку на каналы передачи данных, организовывая работу зависимых модулей ИС посредством только управляющих воздействий.

В качестве основного элемента, обеспечивающего управление ИС, берется так называемый семафор — специальный тип данных, принимающих значения из множества  $\{0, 1\}$ , на котором определены две операции [5]. Эти операции обозначаются  $P$  и  $V$ . Если  $s$  — семафор, то  $P(s)$  имеет результатом  $s-1$ , но выполняется только если  $s > 0$ .  $V(s)$  имеет результатом  $s+1$ . Подобный элемент управляет состояниями ИС, в которые она переходит под воздействием вводимых данных. Поэтому такой процесс удобнее всего будет описать с использованием теории автоматов.

#### 4. Конечный автомат в качестве модели ИС

Конечный автомат (КА) удобно использовать для структурирования приложения, организации и сопровождения логики пользовательского интерфейса. Он используется для управления набором ресурсов, которые должны удерживаться в памяти в любой момент времени. При этом состояние приложения определяется как совокупность значений всех его переменных. Значения меняются под воздействием внешнего события, а для определения состояния КА возможно использовать переменную перечисления, которая сохраняется в памяти. Каждый конкретный набор данных всегда

будет переводить КА в одно и то же множество состояний, которое называется предысторией КА. В случае эквивалентных предысторий, если они одинаковым образом влияют на дальнейшее поведение автомата, нет необходимости запоминать все входные истории, а хранить в памяти их класс эквивалентности [5], что обеспечивает оптимизацию памяти ИС. Кроме того, КА позволяет осуществлять централизованное явное управление ПС [6]. Определяется КА следующим образом:

$$A = \{S, X, Y, s_0, \delta, \lambda\}, \quad (10)$$

где  $S$  — конечное непустое множество состояний;  $X$  — конечное непустое множество входных сигналов (входной алфавит);  $Y$  — конечное непустое множество выходных сигналов (выходной алфавит);  $s_0 \in S$  — начальное состояние;  $\delta: S \times X \rightarrow S$  — функция переходов;  $\lambda: S \times X \rightarrow Y$  — функция выходов. Определим КА информационной системы для заданной модели ПрО в виде графа переходов (рис. 5). Для такого графа задается вершина  $s_0$  — начальное состояние ИС — ожидание ввода данных. В качестве данных выступает номер модуля, который должен быть запущен в заданный момент времени или для которого должен быть выполнен откат системы до заданного состояния. Режиму ожидания соответствует переход  $a_{00}$ . Переход  $a_{01}$  — активация вычислительного процесса; переход  $a_{05}$  — активация процесса отката системы. Вершина  $s_1$  определяет подграф зависимостей между модулями ИС и текущим заданным модулем.

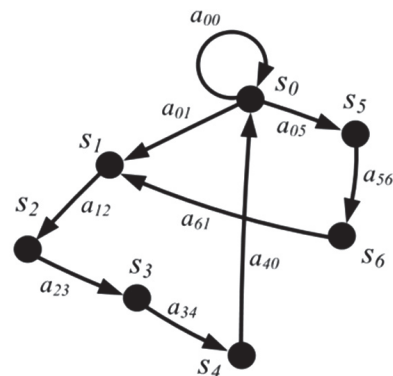


Рис. 5. КА состояний ИС при заданной модели ПрО

Переход  $a_{12}$  — передача управления функции поиска при успешном выделении подграфа зависимостей. Вершине  $s_2$  соответствует поиск сохраненной предыстории вычислений и восстановление из памяти массивов данных. Переход  $a_{23}$  — передача управления текущему модулю ИС. Вершине  $s_3$  соответствует активация вычислительного процесса заданного модуля и дополнение предыстории вычислений с последующим сохранением (вершина  $s_4$ ). Переход  $a_{40}$  — означает возвращение в режим ожидания. Вершина  $s_5$  — ввод данных отката системы до заданного состояния,  $s_6$  — выделение части

подграфа, необходимого для пересчета, то есть тех модулей, результаты работы которых будут обновляться. Переход  $a_{61}$  – активация функции выделения подграфа зависимостей при заданном подграфе, определяющем массивы данных для обновления.

### Выводы

Формальное описание структуры ПС позволяет выделить такие основные элементы как 1) вершина граф-модели ИС, которая соответствует программному модулю и являет собой описание макропроцесса заданной ПрО; 2) ориентированная дуга граф-модели, связывающая на информационном уровне модули ПС и являющаяся каналом передачи данных; 3) спецификация модуля, которая задает стандарты взаимодействия между модулями ПС и компиляции модулей в единую структуру ИС; 4) кортеж данных, представляющий собой базис, необходимый для обеспечения хранения и восстановления информации о результатах работы программы; 5) подграф, позволяющий сохранять историю реализации ИС.

Сформированный КА информационной системы описывает ее макросостояния и позволяет выполнить централизованное управление ИС, что дает преимущество при разработке и проектировании ПС для заданных моделей ПрО.

Описанные и структурированные элементы граф-модели ПрО позволяют детально разработать алгоритмы функционирования исполнительской системы [1] и механизмы управления вычислительными процессами и обмена данными между ними, что является перспективой данного исследования. А также построить математическую модель, описывающую ИС как динамическую информационную систему, которая дает возможность оценить характеристики надежности ИС.

**Список литературы:** 1. Чайников С.И. Принципы организации вычислений на базе граф-модели предметной области [Текст] / С.И. Чайников, А.С. Солодовников // Бионика интеллекта: науч.-техн. журнал. – 2012. – №2 (79). – С. 72-75. 2. Воеводин В.В. Параллельные вычисления [Текст] / В.В. Воеводин, Вл. В. Воеводин. – СПб.: БВХ-Петербург, 2002. – 608 с. 3. Евстигнеев В.А. Применение теории графов в программировании [Текст] / В.А. Евстигнеев. – М.: Наука, 1985. – 352 с. 4. Немнюгин С.А. Модели и средства программирования для многопроцессорных вычислительных систем [Текст] / С.А. Немнюгин. – СПб.: С.-Петербургский ГУ, 2010. – 100 с. 5. Карпов Ю. Г. Теория автоматов [Текст] / Ю. Г. Карпов. – СПб.: Питер, 2003. – 208 с. 6. Салмре И. Программирование мобильных устройств на платформе .NETCompactFramework / Иво Салмре, пер. с англ. – М.: Изд. дом «Вильямс», 2006. – 736 с.

Поступила в редколлегию 20.12.2012

УДК 004(4'242+053)

**Формализований опис ГРАФ-модели предметной галузі** / С.І. Чайніков, А.С. Солодовніков // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 77-81.

Пропонується формализований опис елементів діалогової системи та предметної галузі. Розглянуто механізми організації даних, необхідних для виконання обчислень та відновлення збереженої інформації у діалоговій системі. Запропоновані механізми конденсації граф-моделі предметної галузі та укрупнений опис стану інформаційної системи на базі теорії автоматів.

Л. 5. Бібліогр.: 6. найм.

UDC 004(4'242+053)

**The formalized description of problem domain graph-model** / S.I. Chaaynikov, A.S. Solodovnikov // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 77-81.

The formalized description of main elements of a dialog system and problem domain is proposed. The mechanisms of data organization that computing processes need at and problems of data storing and restoration are considered. The principles of graph-model condensation and aggregated description of information system states based on automata theory are proposed.

Fig. 5. Ref.: 6 items.



С.С. Таянский

ХНУРЭ, г. Харьков, Украина, tanyansky\_ss@yahoo.com

## ФОРМАЛИЗАЦИЯ СРЕДСТВ ДОСТУПА К БАЗАМ ДАННЫХ ПРОИЗВОЛЬНОЙ СТРУКТУРЫ

Рассмотрен основанный на логике подход к формированию запросов в среде неоднородных баз данных. Определен алфавит с расширенной сигнатурой, позволяющий выполнять операции над данными нереляционной структуры. Сформулированы условия, обеспечивающие целостность данных при переходе состояний базы данных. Предложен алгоритм проверки соответствия текущего состояния данных заданным ограничениям.

БАЗА ДАННЫХ, ИНФОРМАЦИОННЫЙ ОБЪЕКТ, СЕМАНТИКА ДАННЫХ, ЛОГИЧЕСКОЕ ПРАВИЛО, ЗАПРОС

### Введение

Языки запросов к базам данных (БД) можно разбить на три класса: алгебраические языки, позволяющие выражать запросы средствами специализированных операторов; языки исчисления предикатов, где запросы описывают требуемое множество значений путем спецификации предикатов, которому должны удовлетворять эти значения; логические языки, в которых запрос реализует построение цепочек логических выводов, комбинирующих правила и фактическую информацию для доказательства истинности или отрицания справедливости первоначального утверждения [1, 2].

В системах БД запрос, выраженный с помощью специализированного языка манипулирования данными, должен обрабатываться при использовании наиболее эффективных путей доступа к большим объемам данных для извлечения релевантной информации. Более того, представление запросов и ограничений целостности необходимо представлять в одном формализме и их проверку осуществлять одними и теми же механизмами, что должно позволить обрабатывать более сложные структуры данных.

Запросы на обновления данных могут задаваться либо явно, с помощью соответствующей инструкции языка SQL или средствами алгоритмических алгебр, обобщающих SQL [3], либо неявно посредством формул некоторого логического языка [4]. При этом наличие ограничений целостности, изменения состояния БД может отличаться от изменений, задаваемых самим запросом.

Принципиальный характер этих проблем отмечался различными коллективами исследователей и разработчиков. По мере развития информационных систем (ИС) их качество повышается в силу высокоуровневых языковых средств, используемых системами управления базой данных (СУБД) и разнотипностью используемых в них моделей данных, что отмечено в работах Л.В. Городней [5] и И. Грэхема [6]. С другой стороны, чрезмерный рост перепрограммирования имеющихся прикладных программ вызывается их жесткой зависимостью от типа применяемой СУБД. Таким

проблемам посвящены работы Р. Стивенса [7] и Б. Карвина [8].

На практике большая часть данных, совместную работу с которыми необходимо автоматизировать, использует реляционные СУБД. При этом внешняя организация табличного документа не всегда соответствует реляционному отношению. Тогда необходимо использовать дополнительные программные средства для обеспечения возможности корректного доступа к данным из независимых приложений.

Таким образом, определение запросов в виде выражения исчисления с переменными на доменах, а также определение их свойств и интерпретации является актуальной в задачах совместной обработки распределенных данных.

Основной проблемой при формировании запросов неоднородных данных является обобщенное представление объектов БД. В статье используется синтаксис правил логического существования информационных объектов предметной области (ПрО) [9], на основе которых формируется запрос и рассмотрены средства декларативного описания данных.

Целью работы является формализованное представление средств описания языка БД и определение семантики данных, используя введенные языковые конструкции, основанные на логических выражениях предикатов первого порядка. Для обеспечения корректности выполнения запросов предложен алгоритм, реализующий вычисление состояния БД, удовлетворяющего заданному множеству ограничений.

### 1. Расширения языка исчисления с переменными на доменах

Используя результаты исследования литературных источников [10, 11] и реализованных практических задач организации доступа к данным средствами логического программирования, выражения исчисления предикатов, положенные в основу языка запросов, определим следующим образом.

Пусть  $O = \{o_1, \dots, o_n, o_{n+1}, \dots\}$  — произвольное множество символов информационных объектов

(в терминах исчисления – символов). Необходимо отметить, что согласно обозначениям, введенным в [9], элементы множества  $O$  будем записывать в виде  $\{l_{o_1}, \dots, l_{o_n}, l_{o_{n+1}}, \dots\}$  или  $\{l_{o_1, \dots, o_n, o_{n+1}, \dots}\}$ . Зададим алфавит  $\mathcal{A}$  как объединение множеств вида:

$$\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5 \cup \mathcal{A}_6, \quad (1)$$

где  $\mathcal{A}_1 = \{l_{o_1}^i, \dots, l_{o_n}^i, l_{o_{n+1}}^i, \dots\}$  – предметные переменные (информационные объекты рассматриваемой ПрО) и  $\{i \in I \mid I = \overline{1, \infty}\}$ ;  $\mathcal{A}_2 = \{a^i, b^i, c^i, a_2^i, b_2^i, c_2^i, \dots\}$  – предметные константы (параметры поиска данных) и  $\{i \in I \mid I = \overline{1, \infty}\}$ ;  $\mathcal{A}_3 = \{\mathcal{P}^{1, 2, \dots, n}\}$  –  $n$ -местные предикатные символы и  $\{i_1, i_2, \dots, i_{n_j} \in I, j \in J \mid J = \overline{1, \infty}\}$ ;  $\mathcal{A}_4 = \{‘=’, ‘\neq’, ‘<’, ‘>’, ‘\leq’, ‘\geq’\}$  – символы сравнения;  $\mathcal{A}_5 = \{‘\wedge’, ‘\vee’, ‘\neg’, ‘\exists’, ‘\forall’, ‘\perp’, ‘F’\}$  – логические символы;  $\mathcal{A}_6 = \{‘,’, ‘(’, ‘)’\}$  – вспомогательные символы.

В дальнейших выкладках некоторые индексы будут опущены, если в них не будет явной необходимости. Объединение первых трех множеств, в обозначении  $\mathcal{G} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ , будем называть сигнатурой формального языка на схеме БД. Используя заданную сигнатуру, определим атомы формул в виде многочлена следующим образом:

1. Выражения вида  $l_o^i \Theta a_i$  или  $l_o^i \Theta l_o^k$ , где  $\Theta$  – символ сравнения, есть условие.
2. Атомом является условие или многочлен, взятый в скобки.
3. Атом  $\alpha$  является многочленом.
4. Если  $\alpha$  есть многочлен, а  $\alpha_1$  является атомом, то выражения  $\alpha \wedge \alpha_1, \alpha \vee \alpha_1, \alpha \wedge \neg \alpha_1, \alpha \vee \neg \alpha_1$  также являются многочленами.

Все входящие в многочлен переменные считаются свободными (здесь следует отметить, что термин “свободные” взят из теории исчисления и не эквивалентен такому же термину, используемому в разделах 3 и 4, для обозначения информационных объектов из множества  $O$ ). Формулы определим индуктивно следующим образом:

1. Предикат  $\mathcal{P}^{1, 2, \dots, n} (l_{o_1}^1, l_{o_2}^1, \dots, l_{o_n}^1)$  есть формула, при этом все переменные в ней свободные.
2. Если  $\beta$  – формула со свободными переменными, а  $\alpha$  – многочлен, свободные переменные которого входят в множество свободных переменных  $\beta$ , то  $\alpha \wedge \beta$  также формула с теми же свободными переменными, что и  $\beta$ .
3. Если  $\beta_1$  и  $\beta_2$  две формулы, имеющие одинаковые свободные переменные, то выражения  $(\beta_1 \vee \beta_2)$  и  $(\beta_1 \wedge \neg \beta_2)$  также формулы с тем же множеством свободных переменных.
4. Если  $\beta_1$  и  $\beta_2$  две формулы, не имеющие свободных переменных или со свободными переменными, пересечение которых не пусто, то выражения  $(\beta_1 \wedge \beta_2)$  также формулы со свободными переменными, соответствующими  $\beta_1$  и  $\beta_2$ .
5. Пусть  $\varepsilon = l_{o_1}^1, l_{o_2}^1, \dots, l_{o_n}^1$  – свободные переменные формул  $\beta_1$  и  $\beta_2$ , причем в  $\beta_1$  других переменных

нет, а в  $\beta_2$  могут присутствовать и другие свободные переменные, отличные от  $\varepsilon$ . Тогда выражения вида  $\exists \varepsilon(\beta_1), \exists \varepsilon(\beta_2), \forall \varepsilon(\beta_1 \wedge \beta_2)$  и  $\forall \varepsilon(\beta_2 \wedge \beta_1)$  есть формулы, где переменные из  $\varepsilon$  называются связными, а остальные переменные, которые входят как связано, так и свободно в  $\beta_1$  и  $\beta_2$ , остаются такими же во вновь образованных формулах.

6. Пусть  $\alpha$  – многочлен,  $\beta$  – формула, а  $\varepsilon = l_{o_1}^1, l_{o_2}^1, \dots, l_{o_n}^1$  – свободные переменные из  $\alpha$ , причем элементы из  $\varepsilon$  включаются в множество свободных переменных  $\beta$ . Исходя из этого, выражения вида  $\forall \varepsilon(\alpha \wedge \beta)$  и  $\forall \varepsilon(\beta \wedge \alpha)$  есть формулы, у которых переменные из  $\varepsilon$  связные, а остальные свободные или связные переменные из  $\beta$  остаются такими же во вновь образованных формулах.

7. Если  $l_{o_j}^i$  – свободная переменная формулы  $\beta$ ,  $a_j^i$  – константа,  $\Theta$  – символ сравнения, то выражение вида  $\mathcal{F}(l_{o_j}^i(\beta)) \Theta a^i$  является формулой, у которой переменная  $l_{o_j}^i$  связная, а остальные свободные или связные переменные из  $\beta$  остаются такими же во вновь образованной формуле.

Содержательная интерпретация формул исчисления предикатов при интеграции БД определяет сигнатуру  $\mathcal{G}$  следующим образом:

1. Множество  $\mathcal{A}_1$  включает все информационные объекты  $\{l_{o_j}^i\}$  рассматриваемой ПрО.
2. В  $\mathcal{A}_2$  входят множества значений из множества  $O$ , которые используются как параметры поиска значений, представленных в БД.
3. Каждому значению от 1 до  $n$  множества информационных объектов  $O$  ставится в соответствии  $\sum_{i=1}^n C_n^i = 2^{n-1}$  (символ “ $C$ ” – обозначает сочетания из  $n$  по  $i$ ) символов предикатов. При этом, если данные не имеют реляционной структуры, то каждому информационному объекту ставится в соответствие  $\sum_{j=1}^m (2^{n_j-1})$  информационных объектов, где  $m$  – число различных последовательностей вывода возможных информационных объектов из  $O$ ,  $n_j$  – число информационных объектов, принадлежащих некоторой последовательности вывода.

Множество всех правильных формул, которые можно построить с помощью алфавита  $\mathcal{A}$ , обозначим  $\beta_{\mathcal{A}}$ . Формулы из  $\beta_{\mathcal{A}}$  будем разделять на замкнутые и открытые. Замкнутыми будем называть формулы, не содержащие свободных переменных, а открытыми – формулы у которых не все переменные связаны кванторами ‘ $\exists$ ’, ‘ $\forall$ ’, ‘ $\mathcal{F}$ ’ ( $\mathcal{F}$  – функциональный квантор, определяющий какую-либо агрегатную функцию). Замкнутые формулы представляют собой высказывания, которые являются истинными или ложными на множестве  $O$ . Открытым формулам соответствует множество истинности формулы в обозначении  $\mathcal{L}(\beta)$ .

При введенной интерпретации смысл правильных формул можно представить следующим образом:

1. Если  $\beta_1$  и  $\beta_2$  – формулы с одинаковыми свободными переменными, тогда выполняются тождества:

- a.  $\mathcal{L}(\beta_1 \vee \beta_2) = \mathcal{L}(\beta_1) \cup \mathcal{L}(\beta_2)$ ;
- b.  $\mathcal{L}(\beta_1 \wedge \neg\beta_2) = \mathcal{L}(\beta_1) - \mathcal{L}(\beta_2)$ ;
- c.  $\mathcal{L}(\beta_1 \wedge \beta_2) = \mathcal{L}(\beta_1) \cap \mathcal{L}(\beta_2)$ .

2. Если  $\beta_1$  – формула, содержащая свободные переменные  $\mathcal{V}_1$ ,  $\beta_2$  – формула, содержащая свободные переменные  $\mathcal{V}_2$  и  $\mathcal{V}_1 \subseteq \mathcal{V}_2$ , то  $\mathcal{L}(\beta_1 \wedge \beta_2) = \mathcal{L}(\beta_1) \times \{I_o^j\} \cap \mathcal{L}(\beta_2)$ , где  $i \in I \mid I = \overline{1, \infty}, j = \overline{1, k}$  и  $\{I_o^j\} = V_2 - V_1$ . Этот факт говорит о том, что  $\mathcal{L}(\beta_1 \wedge \beta_2)$  будет содержать информационные объекты из  $\mathcal{L}(\beta_1)$ , которые также присутствуют и в  $\mathcal{L}(\beta_2)$ .

3. Если  $\beta$  – некоторая формула,  $\alpha$  – многочлен, в котором свободные переменные  $\mathcal{V}$  входят в  $\beta$ , тогда  $\mathcal{L}(\beta \wedge \alpha)$  включает те информационные объекты из  $\mathcal{L}(\beta) \in \mathcal{V}$ , для которых многочлен  $\alpha$  принимает значение истинно.

4. Множество истинных формул вида  $\exists \epsilon(\beta)$ , использующих квантор существования, является проекцией множества истинности  $\beta$  на несвязанный квантором  $\exists$  предметные переменные.

5. Квантор всеобщности  $\forall$  в формулах может иметь двоякий смысл:

a. Ограниченный квантор всеобщности вида  $\forall \epsilon \in \beta_1(\beta_2)$  в эквивалентной записи  $\forall \epsilon(\beta_2 \downarrow \beta_1)$ . Истинностью таких формул является подмножество значений  $v = \mathcal{L}(\beta_2)$  множества несвязанных квантором свободных переменных  $\beta_2$ . Аналогичный смысл имеют также формулы вида  $\forall \epsilon(\beta \downarrow \alpha)$ .

b. Ограниченный квантор всеобщности в формулах с квантором существования вида  $\forall \epsilon \in \exists v(\beta_2)(\beta_1 \downarrow \beta_2)$  или  $\forall \epsilon(\beta_1 \downarrow \beta_2)$ . Истинностью таких формул является подмножество значений  $\mathcal{L}(\beta_2)$  множества  $v$  несвязанных квантором свободных переменных  $\beta_2$ . Аналогичный смысл имеют также формулы вида  $\forall \epsilon(\alpha \downarrow \beta)$ .

Использование ограниченного квантора всеобщности обеспечивает определение соответствующего формуле множества истинности.

6. В качестве функционального квантора  $\mathcal{F}$  используются какие-либо функции (как стандартные, например  $\min$ ,  $\max$ ,  $\text{sum}$  и др., так и сформулированные для индивидуальных вычислений). Функциональный квантор связывает аргумент функции  $\mathcal{F}$  в формулах  $\mathcal{F}(I_o^j(\beta)) \Theta a^i$ . Если  $\beta$  содержит более одной свободной переменной, то множество истинности с функциональным квантором представляет собой подмножество  $\mathcal{L}(\beta)$  множества несвязанных квантором  $\mathcal{F}$  переменных.

7. Множество истинных формул вида  $\mathcal{F}(I_o^j(\alpha \downarrow \beta))$ , использующих функциональный квантор. Истинностью таких формул является новый многочлен  $\alpha$ , состоящий из несвязанных квантором свободных переменных в формуле  $\beta$ .

Использование исчисления предикатов позволяет создать гибкую схему формирования сложных

запросов к БД, обеспечивающую построение посредством логических связок, кванторов и скобок многоуровневых условий, в которых формулы выражают условия идентификации данных интегрированной БД. Особенностью таких запросов является возможность проведения формального контроля правильности запроса, основанного на соответствии используемых в формулах переменных и констант определенным свойствам схемы БД.

Определенные выше описания формул могут быть применены не только к данным реляционной структуры, но для работы с данными иерархической или сетевой структуры необходимо ввести дополнительные атомы и фиктивные переменные, связанные иерархическими зависимостями.

При обработке данных нереляционной структуры необходимо обеспечить их целостность. Очевидно, что традиционными средствами реляционной модели обеспечить корректность таких данных не представляется возможным. С целью поддержки целостности и согласованности интегрированных БД введем дополнительные ограничения на семантику данных.

## 2. Поддержка логической целостности в таблицах иерархической и сетевой структур

Определим характеристики компонентов ограничений целостности для формирования формул исчисления. Ограничения, которые определяют допустимые расширения БД и объектов БД, называются статическими. Ограничения, которые регламентируют доступные переходы БД между допустимыми состояниями, называются динамическими [10]. Динамические спецификации, как правило, операционно-ориентированные, то есть обосновывают выполнение той или иной конкретной операции.

Простым является ограничение, которое строится с использованием одноместных предикатов арифметического сравнения:

$$\mathcal{P}(I_{o_1}^i, \dots, I_{o_n}^i) \supset I_{o_1}^i \Theta \alpha_1 \wedge \dots \wedge I_{o_n}^i \Theta \alpha_n, \quad (2)$$

где  $\mathcal{P}(I_{o_1}^i, \dots, I_{o_n}^i)$  –  $n$ -местный предикат;  $I_{o_i}^i$  – предметные переменные;  $a_i$  – предметные константы;  $\Theta = \{‘=’, ‘\neq’, ‘<’, ‘>’, ‘\leq’, ‘\geq’\}$  – символы сравнения, символ ‘ $\supset$ ’ – является знаком импликации.

Более сложные виды ограничений могут быть выражены обязательной поддержкой ключей, первичного и внешнего. Первичный ключ определим как предикат  $\mathcal{PK}(R, I_{o_i}^i)$ , где  $R$  – имя таблицы, уникально определяющее множество предметных переменных  $I_{o_1}^i, \dots, I_{o_n}^i$ , соответствующих атрибутам  $R$ . Выражение, определяющее первичный ключ, имеет вид:

$$\begin{aligned} \mathcal{PK}(R, I_{o_i}^i) \equiv & \forall I_{o_1}^1 \vee I_{o_1}^2 \vee I_{o_1}^3 (R(I_{o_1}^1, I_{o_2}^1, \dots, I_{o_n}^1, \\ & I_{o_1}^2, I_{o_2}^2, \dots, I_{o_n}^2) \wedge (I_{o_1}^1, I_{o_2}^1, \dots, I_{o_n}^1, I_{o_1}^3, I_{o_2}^3, \dots, \\ & I_{o_n}^3) \supset (I_{o_1}^2 = I_{o_1}^3) \wedge (I_{o_2}^2 = I_{o_2}^3) \wedge \dots \wedge \\ & \wedge (I_{o_n}^2 = I_{o_n}^3) \wedge (I_{o_1}^1 \neq a_1) \wedge (I_{o_2}^1 \neq a_2) \wedge \dots \wedge (I_{o_n}^1 \neq a_n)). \end{aligned} \quad (3)$$

Ограничение на внешний ключ определяется как совпадение значений связанных атрибутов двух отношений. Представим внешний ключ в виде предиката  $\mathcal{FK}(R_1, R_2, l_{o_i}^i)$ , где переменные  $l_{o_1}^1, l_{o_2}^2, \dots, l_{o_n}^n$  таблицы  $R_1$  ссылаются на атрибуты отношения  $R_2$ , то есть совокупность этих переменных в отношении  $R_2$  является внешним ключом. Выражение для описания внешнего ключа имеет вид:

$$\begin{aligned} \mathcal{FK}(R_1, R_2, l_{o_i}^i) \equiv \forall l_{o_1}^1 \exists l_{o_2}^2 (R_1(l_{o_1}^1, l_{o_2}^2, \dots, l_{o_n}^n) \supset \\ \supset ((R_2(l_{o_1}^1, l_{o_2}^2, \dots, l_{o_n}^2) \wedge (l_{o_1}^1 = l_{o_1}^2) \wedge (l_{o_2}^1 = l_{o_2}^2) \wedge \\ \wedge \dots \wedge (l_{o_n}^1 = l_{o_n}^2) \wedge \mathcal{PK}(R_2, l_{o_1}^2, l_{o_2}^2, \dots, l_{o_n}^2))) \vee \\ \vee ((l_{o_1}^1 = a_1) \wedge (l_{o_2}^1 = a_2) \wedge \dots \wedge (l_{o_n}^1 = a_n))). \end{aligned} \quad (4)$$

Рассмотренные ограничения являются структурными, то есть определяемыми при проектировании БД, и зависят от множества заданных зависимостей (для реляционной модели – функциональных и многозначных зависимостей). Под явными ограничениями целостности будем понимать ограничения, заданные пользователем. Среди таких ограничений можно выделить ограничение на домен (явно неподдерживаемое в реляционной модели данных). К ним относится, например ограничение, которое устанавливает значение, определенное по умолчанию, в случае если ни одно из вводимых значений не соответствует ограничению. Соответствующее выражение имеет вид:

$$\begin{aligned} \mathcal{NH}(R, l_{o_i}^i, a_j) \equiv \forall l_{o_i}^i (R(l_{o_i}^i) \wedge a_j \notin D_i \supset \\ \supset R(\langle l_{o_i}^i, D_i(a_j) \rangle)), \end{aligned} \quad (5)$$

где  $a_j$  – предметная константа (вносимое значение);  $\langle l_{o_i}^i, D_i(a_j) \rangle$  – значение  $a_j \in D_i$  информационного объекта  $l_{o_i}^i$ , устанавливаемое по умолчанию, заметим, что  $O = \bigcup_{i=1}^n D_i$  и  $l_{o_i}^i \in O$ .

Необходимо отметить, что частным случаем такого ограничения является внесение пустого значения в  $l_{o_i}^i$  (такой прием относится к пользовательским допущениям).

Ограничения идентификаторов объектов формулируются в терминах зависимостей атрибутов. В терминах введенных информационных объектов ограничение на зависимость обладает следующим свойством: для информационного объекта  $l_{o_i}^1$  не существует информационного объекта  $l_{o_i}^2$ , такого что  $\langle l_{o_i}^1, D_1(a_i) \rangle = \langle l_{o_i}^2, D_2(b_i) \rangle$ ,  $\langle l_{o_i}^2, D_1(a_i) \rangle \neq \langle l_{o_i}^2, D_2(b_i) \rangle$ .

Предикат для описания зависимости обозначим как  $\mathcal{FD}(R, l_{o_i}^1, l_{o_i}^2)$ , для которого выражение имеет вид:

$$\begin{aligned} \mathcal{FD}(R, l_{o_i}^1, l_{o_i}^2) \equiv \forall l_{o_i}^1 (R(l_{o_i}^1) \supset \neg \exists l_{o_i}^2 (R(l_{o_i}^2) \wedge \\ \wedge \langle l_{o_i}^1, D_1(a_i) \rangle = \langle l_{o_i}^2, D_2(b_i) \rangle \wedge \langle l_{o_i}^2, \\ D_1(a_i) \rangle \neq \langle l_{o_i}^2, D_2(b_i) \rangle)). \end{aligned} \quad (6)$$

Рассмотрим два основных типа преобразования атрибутов: 1:1 и 1:m. Кроме этого можно рассматривать типы m:1 и m:m [12]. Преобразование 1:1 имеет место, когда атрибуту целевой таблицы строго соответствует атрибут таблицы источника. Преобразование 1:m, m:1 возникает тогда, когда существует функциональная зависимость между атрибутом целевой таблицы и несколькими атрибутами таблицы источника или нескольких атрибутов целевой таблицы и атрибутом таблицы источника соответственно. Преобразование m:m возникает при необходимости одновременного получения нескольких атрибутов целевой таблицы из нескольких атрибутов таблицы источника.

Использование информации об ограничениях, которые накладываются на отношения, позволяет получить дополнительную информацию для преобразования БД. Информация о первичном ключе дает возможность определить перечень полей для определения операций добавления, удаления или модификации целевой таблицы. Информация об ограничениях ссылочной целостности позволяет корректно реализовывать соответствие между несколькими таблицами-источниками и целевой таблицей. Использование ограничений по умолчанию дает возможность определить несформированные значения атрибутов первичного ключа.

Определим предикат  $\mathcal{JN}(R_1, R_2, l_{o_i}^i)$  как последовательность ссылочных ограничений между таблицами вида:

$$\begin{aligned} \mathcal{JN}(R_1, R_2, l_{o_i}^i) \equiv \exists R_1, \exists R_2, \forall l_{o_i}^i (R_1, \\ \mathcal{FK}(R_1, R_2, l_{o_i}^i), R_2) \supset \mathcal{JN}(R_1, R_2, l_{o_i}^i), \end{aligned} \quad (7)$$

где  $R_1$  и  $R_2$  – таблицы;  $l_{o_i}^i$  – связанный информационный объект.

Пусть необходимо преобразовать схему БД, то есть построить переход типа  $R_1 \leftarrow R_2$ , представленный на рис. 1(а) в схему, представленную на рис. 1(б).

$R_1$			$R_2$				
$l_{o_1}^1$	$l_{o_2}^1$	$l_{o_3}^1$	$l_{o_1}^2$	$l_{o_4}^2$	$l_{o_5}^2$	$l_{o_6}^2$	...
$a_1^1$	$a_1^2$	$a_1^3$	$a_1^1$	$a_1^3$			...
$a_2^1$	$a_2^2$	$a_2^3$	$a_2^1$		$a_2^3$		...

а) таблица реляционной структуры      б) таблица нереляционной структуры

Рис. 1. Схемы БД, для которых необходимо построить переход  $R_1 \leftarrow R_2$

Отметим, что выделены атрибуты первичного ключа. Для перехода  $R_1 \leftarrow R_2$  необходимо построить соответствие “информационный объект”  $\leftarrow$  “значение” и “значение”  $\leftarrow$  “информационный объект”  $l_{o_3}^1 \leftarrow a_1^3, l_{o_3}^1 \leftarrow a_2^3$  и соответствия атрибутов  $a_1^2 \leftarrow l_{o_4}^2, a_2^2 \leftarrow l_{o_5}^2$ . Для обеспечения целостности при таком переходе необходимо воспользоваться свойством предиката  $\mathcal{PK}$ .

Таким образом, предикатное выражение для преобразования схем  $R_2$  в  $R_1$  будет иметь вид:

$$\begin{aligned} & \exists R_1 \exists R_2 \forall l_{o_i}^1 (R_1(l_{o_i}^1, l_{o_i}^2) \supset R_2(l_{o_i}^1) \wedge (l_{o_2}^1 = l_{o_4}^2) \wedge \\ & \wedge (l_{o_2}^1 = l_{o_5}^2) \wedge (l_{o_2}^1 = l_{o_6}^2) \wedge \dots \wedge (l_{o_2}^1 = l_{o_n}^2) \wedge \\ & \wedge (a_q^2 \neq a_j^2)); i = \overline{1, n}; q, j = \overline{1, m}. \end{aligned} \quad (8)$$

При формировании целевой таблицы выполняется объединение переменных, определяющих атрибуты таблиц - источников. С другой стороны, при формировании значений ключевых атрибутов целевой таблицы необходимо использовать дополнительную информацию, которая извлекается из ограничений, накладываемых на таблицы - источник. В рассматриваемом примере атому  $(a_q^2 \neq a_j^2)$  всегда будет соответствовать значение *True*. Это определяется тем, что атрибуты в одной таблице должны иметь различные имена и, следовательно, при преобразовании  $a_1^2 \leftarrow l_{o_4}^2$  и  $a_2^2 \leftarrow l_{o_5}^2$  пара информационных объектов  $l_{o_4}^2, l_{o_5}^2$  в качестве значений информационного объекта  $l_{o_2}^1$  в таблице  $R_1$  повторяться не будет.

Решение задачи реализации перехода  $R_1 \leftarrow R_2$  можно осуществить с помощью написания запроса на каком-либо языке программирования либо записать в терминах реляционного исчисления. Второй подход является более эффективным, так как носит описательно-формализованный характер и гарантирует корректность результатов запросов.

Для примера, представленного на рис. 1, сформируем запрос на выборку для каждой таблицы средствами реляционного исчисления с переменными на доменах. Для таблицы  $R_1$  выражение имеет вид (9), а для таблицы  $R_2$  - вид (10):

$$\begin{aligned} & \{ p^{l_{o_i}^1} \mid (\exists l_{o_1}^1, l_{o_2}^1, l_{o_3}^1)(R_1(l_{o_1}^1, l_{o_2}^1, l_{o_3}^1) \wedge \\ & \wedge p^{l_{o_1}^1} = l_{o_1}^1 \wedge p^{l_{o_2}^1} = l_{o_2}^1 \wedge p^{l_{o_3}^1} = l_{o_3}^1) \}, \\ & \{ p^{l_{o_i}^2} \mid (\exists l_{o_1}^2, l_{o_4}^2, l_{o_5}^2, l_{o_6}^2, \dots, l_{o_n}^2) \\ & (R_2(l_{o_1}^2, l_{o_4}^2, l_{o_5}^2, l_{o_6}^2, \dots, l_{o_n}^2) \wedge p^{l_{o_1}^2} = l_{o_1}^2 \wedge p^{l_{o_4}^2} = \\ & = l_{o_4}^2 \wedge p^{l_{o_5}^2} = l_{o_5}^2 \wedge p^{l_{o_6}^2} = l_{o_6}^2 \wedge \dots \wedge p^{l_{o_n}^2} = l_{o_n}^2) \}, \end{aligned} \quad (9)$$

где  $p^{l_{o_i}^1}$  - свободная переменная, определяющая результат запроса.

Для одновременной обработки данных объединим формулы, описывающие доступ к данным таблиц  $R_1$  и  $R_2$ , с учетом свойств предиката (9) и правила преобразования (10). Получаем выражение:

$$\begin{aligned} & \{ p^{l_{o_i}^1} \mid (\exists l_{o_1}^1, l_{o_2}^1, l_{o_3}^1) ((R_1(\exists l_{o_1}^1, l_{o_2}^1, l_{o_3}^1) \wedge \\ & \wedge (\exists l_{o_1}^2, l_{o_4}^2, l_{o_5}^2, l_{o_6}^2, \dots, l_{o_n}^2) (R_2(l_{o_1}^2, l_{o_4}^2, l_{o_5}^2, l_{o_6}^2, \dots, l_{o_n}^2) \wedge \\ & \wedge (p^{l_{o_1}^1} = l_{o_1}^1 \wedge p^{l_{o_2}^1} = l_{o_2}^1 \wedge p^{l_{o_3}^1} = l_{o_3}^1) \wedge (p^{l_{o_1}^2} = l_{o_1}^2 \wedge \\ & \wedge p^{l_{o_4}^2} = l_{o_4}^2 \wedge p^{l_{o_5}^2} = l_{o_5}^2 \wedge p^{l_{o_6}^2} = l_{o_6}^2 \wedge \dots \wedge p^{l_{o_n}^2} = l_{o_n}^2) \wedge \\ & \wedge (l_{o_2}^1 = l_{o_4}^2) \wedge (l_{o_2}^1 = l_{o_5}^2) \wedge (l_{o_2}^1 = l_{o_6}^2) \wedge \dots \\ & \wedge (l_{o_2}^1 = l_{o_n}^2) \wedge (a_q^2 \neq a_j^2)); i = \overline{1, n}; q, j = \overline{1, m}. \end{aligned} \quad (11)$$

Необходимо отметить, что в  $R_2$  могут присутствовать неопределенные значения. Для поддержки корректного преобразования данных воспользуемся предикатом  $\mathcal{CH}$  (5).

В рассматриваемом примере модифицируем предикат, исключив кортежи с неопределенными значениями.

Поддержка ограничений целостности на зависимости, как отмечалось выше, заключается в интерпретации предиката  $\mathcal{FD}$ , то есть в вычислении его истинного значения. Пусть  $R = (Sch, F)$  - таблица БД с количеством строк  $m$ , где  $Sch$  - схема таблицы и  $Sch \subseteq O$ ,  $F$  - множество функциональных зависимостей вида  $F = \{l_{o_i}^1 \rightarrow l_{o_j}^2\}$ ,  $i, j = \overline{1, n}$ ,  $l_{o_i}^1, l_{o_j}^2 \subseteq O$ . Необходимо проверить, удовлетворяет ли состояние  $R$  некоторой зависимости  $l_{o_k}^1 \rightarrow l_{o_k}^2 \in F$ ,  $k = \overline{1, p}$ .

Таким образом, выражение (11) примет вид:

$$\begin{aligned} & p^{l_{o_i}^1} \mid (\exists l_{o_1}^1, l_{o_2}^1, l_{o_3}^1) ((R_1(\exists l_{o_1}^1, l_{o_2}^1, l_{o_3}^1) \wedge \\ & \wedge (\exists l_{o_1}^2, l_{o_4}^2, l_{o_5}^2, l_{o_6}^2, \dots, l_{o_n}^2) (R_2(l_{o_1}^2, l_{o_4}^2, l_{o_5}^2, l_{o_6}^2, \dots, l_{o_n}^2) \wedge \\ & (p^{l_{o_1}^1} = l_{o_1}^1 \wedge p^{l_{o_2}^1} = l_{o_2}^1 \wedge p^{l_{o_3}^1} = l_{o_3}^1) \wedge (p^{l_{o_1}^2} = \\ & = l_{o_1}^2 \wedge p^{l_{o_4}^2} = l_{o_4}^2 \wedge p^{l_{o_5}^2} = l_{o_5}^2 \wedge p^{l_{o_6}^2} = l_{o_6}^2 \wedge \dots \wedge p^{l_{o_n}^2} = \\ & = l_{o_n}^2) \wedge \wedge (l_{o_2}^1 = l_{o_4}^2) \wedge (l_{o_2}^1 = l_{o_5}^2) \wedge (l_{o_2}^1 = l_{o_6}^2) \wedge \dots \wedge \\ & \wedge (l_{o_2}^1 = l_{o_n}^2) \wedge (a_q^2 \neq a_j^2) \wedge l_{o_2}^1 \neq \emptyset); \\ & i = \overline{1, n}; q, j = \overline{1, m}. \end{aligned} \quad (12)$$

АЛГОРИТМ. Проверка состояния БД.

ВХОД.  $R$  - таблица базы данных, функциональная зависимость  $l_{o_k}^1 \rightarrow l_{o_k}^2 \in F$ .

ВЫХОД. Множество строк  $\{Str_i\} (i = \overline{1, n} \mid n \leq m)$ , удовлетворяющих  $l_{o_k}^1 \rightarrow l_{o_k}^2$  (при этом значение логической переменной  $\alpha$  соответствует *True*).

МЕТОД.

ШАГ 1. Устанавливаем значение логической переменной  $\alpha = True$  и просматриваем строки  $\{Str_i\}$ ,  $i = \overline{1, m}$ .

ШАГ 2. Для каждого значения левого элемента  $Str_i^1(l_{o_k}^1)$  просматриваем значения правого элемента строки  $Str_j^2(l_{o_k}^2)$ ,  $j = \overline{1, m-1}$ .

ШАГ 3. Если  $Str_i^1(l_{o_k}^1) = Str_j^2(l_{o_k}^2)$ , то выполняется проверка  $Str_i^1(l_{o_k}^1) = Str_j^2(l_{o_k}^2)$ .

ШАГ 4. Если условие шага 3 выполняется, то удаляем  $Str_j$  из области перебора и продолжаем просмотр.

ШАГ 5. Если условие шага 3 не выполняется, то выходное значение логической переменной определяется как  $\alpha = False$  и шаг 7.

ШАГ 6. При успешном окончании просмотра  $Str_i$ , то есть когда  $i = m$ , выходное значение устанавливается как  $\alpha = True$  и шаг 7.

ШАГ 7. Алгоритм закончен.

Как видно, в наихудшем случае придется выполнить  $m+(m-1)+\dots+1 = m(m+1)/2$  обращений. Для проверки всех зависимостей из  $F$  алгоритм необходимо выполнить  $p$  раз. Таким образом, вычислительную сложность поддержки множества

зависимостей, в среднем, можно оценить как  $p \cdot m \cdot (m+1)/2$  обращений.

Таким образом, наиболее эффективным путем разрешения проблем преобразования данных является оснащение систем управления БД специальными языковыми средствами поддержки ограничений, способными реализовать требования пользователей, определенные в концептуальной схеме БД.

### Выводы

При использовании различных источников информации одни и те же сущности реального мира могут моделироваться с помощью различных структур и типов данных. При попытке интеграции разнородных источников необходимо либо приводить данные к общей структуре, либо в результате интеграции данные будут иметь неправильную структуру. Приведение к общей структуре большого количества источников может быть невозможным, или обобщенная структура будет очень сложной, а ее использование неэффективным.

На практике большая часть данных, совместную работу с которыми необходимо автоматизировать, использует реляционные СУБД. При этом внешняя организация табличного документа не всегда соответствует реляционному отношению. Тогда необходимо использовать дополнительные средства для обеспечения возможности корректного доступа к данным из независимых приложений.

Средства, определяющие корректность модификации данных выражаются в виде ограничений, накладываемых на значения, допустимые в некотором состоянии БД. При модификации структуры данных необходимо сохранить начальные ограничения или так их изменить, чтобы не противоречить семантике данных, то есть ограничения исходного и модифицированного состояния должны быть эквивалентны.

Исследован класс ограничений целостности, который определяет допустимые транзакции, то есть переход из одного состояния БД в другое. К классу рассматриваемых модификаций добавлены операторы замены одних фактов другими, не сводимые непосредственно к операторам добавления и удаления.

Формально определен язык запросов как выражения исчисления с переменными на доменах. Определены свойства и интерпретация формул языка. Показано, что предложенная формулировка языка может быть применена не только к данным реляционной структуры.

Определены характеристики компонентов ограничений целостности для формирования формул исчисления. Заданы ограничения на допустимые значения и ключи, а также введены дополнительные ограничения, явно не поддерживаемые в реляционных моделях, ограничения по ссылке и ограничение на домен. Разработан алгоритм проверки соответствия состояния БД заданным ограничениям при модификации данных.

Исследована практическая значимость полученных результатов, определены критерии

использования неоднородных (не соответствующих реляционным требованиям) отношений локальных БД. Использование рассмотренных способов построения запроса позволяет средствами традиционных реляционных СУБД организовать доступ к неоднородным данным.

Дальнейшее исследование в области неоднородных БД следует направить на определения ограничений для корректного перехода состояний БД при интеграции логически независимых информационных систем.

**Список литературы:** 1. Ульман Дж. Основы систем баз данных [Текст] / Дж. Ульман. – М.: Финансы и статистика, 1983. – 334 с. 2. Черри С. Логическое программирование и базы данных [Текст] / С. Черри, Г. Готлоб, Л. Танка. – М.: Мир, 1992. – 352 с. 3. Abiteboul S. Updates a new Frontier [Text] / S. Abiteboul S. // Proc. of the Second International Conference on the Theory of Databases. – ICDT'88. – 1988. – № 326. – P.1 – 18. 4. Katsuno H. Propositional knowledge base revision and minimal change [Text] / H. Katsuno, A.O. Mendelson. // Artificial Intelligence. – 1991. – V. 52. – P. 253 – 294. 5. Городняя Л.В. Основы функционального программирования. [Текст] / Л.В. Городняя - М.: Изд-во "Интернет - университет информационных технологий – INTUIT.ru", 2004. – 280 с. 6. Грэхем И. Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. [Текст] / И. Грэхем – 3-е изд. – М.: «Вильямс», 2004. – 880 с. 7. Стивенс Р. Программирование баз данных [Текст] / Р. Стивенс. – Изд-во «Бином-Пресс», 2003. – 384 с. 8. Карвин Б. Программирование баз данных SQL. Типичные ошибки и их устранение [Текст] / Б. Карвин. – Изд-во «Рид Групп», 2012. – 336 с. 9. Таянский, С.С. Характеристические свойства объектов информационных систем [Текст] / С.С. Таянский. // "Штучний інтелект" науковий журнал – 2007. – № 1. – С. 78-89. 10. Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных [Текст] / Л.А. Калиниченко. – М.: Наука, 1983. – 424 с. 11. Маркова Л.И. Метод построения блочно-ациклической схемы реляционной базы данных [Текст] / Л.И. Маркова, С.С. Таянский, Д.А. Руденко // Глобальные информационные системы. Проблемы и тенденции развития: материалы I междунар. научн. конф. – Харьков – Туапсе, ХНУРЭ, 2006. – С.101 – 102. 12. Дейт К.Дж. Введение в системы баз данных [Текст] / К.Дж. Дейт. – М.: Диалектика, 1998. – 784 с.

Поступила в редколлегию 20.11.2012

УДК 004.047:681.3.01

**Формалізація засобів доступу до баз даних довільної структури** / С.С. Таянський // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 82-87.

Досліджено й запропоновано формальні засоби формування запитів до даних нереляційної структури. Визначено основні властивості і характеристики станів бази даних, що інтегруються. Запропоновано алгоритм перевірки відповідності бази даних заданим обмеженням.

Л. 1. Бібліогр.: 12 найм.

UDK 004.047:681.3.01

**Formalization means of access to the database of any structure** / S. Tanyansky // Bionics of Intelligence: Sci. Mag. – 2013. – № 1 (80). – P. 82-87.

Explore and propose formal means querying relational data, not structure. The basic properties and characteristics of the states integrable database. The algorithm for checking the conformity database defined constraints.

Fig. 1. Ref.: 12 items.

УДК 004.912

Л.Е. Чала<sup>1</sup>, П.Ю. Попаденко<sup>2</sup><sup>1, 2</sup> ХНУРЕ, м. Харків, Україна, aspirantura@kture.kharkov.ua

## МЕТОД ВИЯВЛЕННЯ НЕЧІТКИХ ДУБЛІКАТІВ ЕЛЕКТРОННИХ ТЕКСТОВИХ ДОКУМЕНТІВ

В роботі здійснено аналіз методів виявлення нечітких дублікатів текстових файлів. Показано, що існуючі методи виявлення нечітких дублікатів спрямовані на мінімізацію обчислювальної складності з одночасним збільшенням повноти і точності алгоритму. У даній роботі запропоновано та протестовано гібридний метод, який дозволяє істотно поліпшити якість виявлення нечітких дублікатів при незначному збільшенні обчислювальної складності базового алгоритму.

НЕЧІТКІ ДУБЛІКАТИ, ЧАСТОТНИЙ АЛГОРИТМ, ШИНГЛУВАННЯ, ГІБРИДНИЙ МЕТОД

### Вступ

Пошук дублікатів текстових та медіа-файлів завжди був одним з ключових завдань для пошукових інформаційних систем. Як правило, такі системи аналізують всі дані, які надаються пошуковими роботами. При цьому пошукові роботи автоматично індексують будь-який сайт, на який вони переходять за посиланнями, а також кожен новий сайт, зареєстрований в базі даних пошукової системи. Найчастіше роботи можуть індексувати одні й ті ж інформаційні ресурси з певною (наперед заданою) періодичністю, внаслідок чого в базу даних пошукових систем потрапляє надлишкова інформація.

Зберігання надлишкового обсягу даних, пов'язане з дублюванням даних, вимагає додаткових часових і обчислювальних витрат. Крім того, повне або часткове дублювання документів позначається на їх порядкових номерах в пошуковій видачі, тому що впливає на ранжування інформаційних блоків, які аналізуються.

Таким чином, вдосконалення методів виявлення нечітких дублікатів є актуальним завданням, успішне вирішення якого дозволяє підвищити ефективність пошукових систем.

Основною перешкодою для успішного виявлення нечітких дублікатів є гігантський обсяг інформації, яка зберігається в сучасних базах даних, що робить практично неможливим безпосереднє виявлення нечітких дублікатів шляхом попарного порівняння текстів документів. У зв'язку з цим останнім часом значна увага приділяється розробці методів зниження обчислювальної складності алгоритмів пошуку за рахунок застосування різних евристик (наприклад, хешування певного фіксованого набору «значущих» слів або речень документу, семплування набору підрядків тексту, використання дактилограм тощо) [1].

Існуючі методи виявлення нечітких дублікатів спрямовані на мінімізацію обчислювальної складності з одночасним збільшенням повноти і точності алгоритму.

У даній роботі представлений метод, який дозволяє істотно поліпшити якість виявлення нечітких дублікатів при незначному збільшенні обчислювальної складності базового алгоритму.

### 1. Загальна характеристика проблеми виявлення нечітких дублікатів

Нечіткий дублікат документа — це документ, частково змінений у змістовній частині і / або в частині форматування.

Методи виявлення нечітких дублікатів дозволяють визначити, чи є два документи нечіткими дублікатами один одного. В даному випадку дуже важливо не тільки виявити всі нечіткі дублікати, але і помилково не ідентифікувати як дублікати документи, близькі за змістом. Таким чином, вдосконалення методів виявлення нечітких дублікатів електронних документів безпосередньо пов'язане зі зниженням ймовірності виникнення помилок першого і другого роду.

Існуючі на даний час методи аналізу електронних документів, що дозволяють виявляти дублікати, можна розділити на три основні групи:

— частотні методи, для яких основним критерієм вибору значущих слів з необробленого тексту є частота;

— вагові методи, де для кожної мовної одиниці за певними формулами розраховується ваговий коефіцієнт, який є умовною мірою значущості слова в аналізованому тексті;

— методи, в основі яких лежить шинглування [2].

Усі ці методи є наближеними, оскільки дозволяють уникнути повного попарного порівняння аналізованих документів і отримати прийнятні результати при допустимій обчислювальній складності.

При застосуванні наближених методів спостерігається зменшення (іноді вельми значне) показника повноти виявлення дублів.

Одним з ключових вимог, що висуваються до якості алгоритмів детектування нечітких дублікатів, є їх стійкість до невеликих змін вихідних документів і можливість обробляти короткі документи. Для виконання цих вимог вважається доцільним розробити гібридний алгоритм, що дозволяє об'єднати позитивні властивості існуючих методів виявлення дублікатів електронних документів.

Метою роботи є розробка та програмна реалізація гібридного методу виявлення нечітких дублікатів, який би мав кращі показники точності та повноти, ніж відомі алгоритми.

Згідно з цією метою необхідно вирішити наступні задачі:

- здійснити порівняльний аналіз існуючих методів з метою виявлення їх недоліків та переваг;
- розробити гібридний метод пошуку нечітких дублікатів, який забезпечив би необхідні повноту та точність за умови прийнятних обчислювальних витрат;
- здійснити програмну реалізацію та тестування розробленого алгоритму в середовищі Visual Studio за допомогою мови програмування C#.

## 2. Порівняльний аналіз методів виявлення нечітких дублікатів

Розглянемо особливості застосування методів виявлення нечітких дублікатів [2-5]. Слід зазначити, що перед оцінкою тексту (незалежно від типу методу, що застосовується) його попередньо очищують від розділових знаків, букв різного регістра і слів, які не несуть смислового навантаження (часток, спілок, прийменників, слів-паразитів). В основі вагових алгоритмів лежить ідея морфологічного (або якогось іншого) аналізу тексту з метою виявлення найбільш «важливих» або «важких» слів (в даній роботі під словом будемо розуміти деяку мовну одиницю). Очевидно, що не всі слова в тексті мають однакоє смислове навантаження, тому в процесі аналізу складається так званий словник документа, де виділені слова йдуть в порядку убивання їх «ваги». Далі вважається, що при визначенні нечіткого дублікату достатньо оцінювати не весь текст повністю, а частину отриманого словника, яка містить найбільш «важкі» слова документа.

Частотні алгоритми (зокрема, TF, TF\*IDF, TF\*RIDF) в певному сенсі є окремим випадком вагових алгоритмів, де «вага» слова відповідає частоті його появи в документі (tf). Далі (за аналогією з ваговими алгоритмами) складається словник, у якому слова розташовані за спаданням значень tf, і проводиться порівняння документів із застосуванням такого словника. Порівняння документів за їх словниками може проводитися різними способами. Зазвичай обирається обмежена (до 10) кількість найбільш важких слів із словників, для якого підраховуються контрольні суми (CRC32), а далі ці суми порівнюються.

Алгоритми, засновані на шинглюванні (зокрема, Log Shingles, Megashingles), відрізняються своєю простотою і ефективністю, тому саме їх використовують пошукові системи для виявлення нечітких дублікатів. Після очищення тексту тут відбувається виділення шинглів (послідовних фрагментів тексту), які обираються «внахлест» один за одним. Далі визначаються контрольні суми кожного з шинглів і здійснюється їх порівняння. Слід зазначити, що тут відсутні попередня оцінка документа і складання словника. Це збільшує швидкодію алгоритму, а розташування шинглів «внахлест» дозволяє

підвищити його точність, так як контрольні суми таких шинглів дуже чутливі до найменших змін у наданому наборі слів.

Особливий інтерес становлять алгоритми, засновані на шинглюванні, що вперше були викладені в [1], а саме, метод декомпозиції і метод «3+5». Розглянемо ці методи докладніше.

У методі декомпозиції використовуються повні набори шинглів і методи скорочення квадратичної залежності, яка виникає, коли загальну ознаку має велика кількість документів. Ідея заснована на тому спостереженні, що розмір документа в словах має значну поділяючу властивість, а також на знаходженні загальних списків документів, які мають однакову ознаку.

Робота алгоритму полягає в наступному:

- для кожного документа обчислюємо неповторювані шингли (можна не всі, а вибірково) і зберігаємо у файлі в форматі <shingle, doc\_id, doc\_len>;
- для однакових шинглів будемо ланцюжки в форматі <doc\_id1, doc\_len1> <doc\_id2, doc\_len2> ..., упорядковані за зростанням doc\_len;
- поділяємо ланцюжки на більш дрібні, якщо у сусідніх довжин відношення більшої довжини до меншої перевищує деякий поріг, який визначається мінімальним рівнем подібності для дублікатів (наприклад, для рівня подібності 0,85 можна практично без втрати повноти використовувати поріг 1,15);
- видаляємо дублі ланцюжків і ланцюжки, які цілком входять в інші. У результаті число ланцюжків скорочується в сотні разів, а решта ланцюжків в переважній більшості є достатньо короткими (2 – 10 елементів);
- документи всередині ланцюжка порівнюємо попарно (наприклад, шляхом використання функції Perl Similarity або ж за допомогою якихось додаткових числових характеристик документів). При цьому порівняння здійснюється не за всім ланцюжком, а тільки в межах невеликої локальної околиці, обумовленої порогом відношення довжин, тому загальне число реальних порівнянь невелике;
- для виключення дублів перевірок в різних ланцюжках списки вже оброблених пар зберігаємо в хеші.

Основним недоліком алгоритму, що знижує його продуктивність, є необхідність обчислення шинглів для документів і використання функції Similarity. Хоча, як зазначалося раніше, можна обчислювати не всі шингли, а тільки деякі, у відповідності з якою-небудь евристикою, і замінити Similarity більш простими засобами.

У методі «3+5» використовуються лише ознаки, що вимагають мінімальних обчислювальних ресурсів, і не визначаються контрольні суми всіх підрядків тексту. Крім того, беруться до уваги лише «локальні» властивості текстів, тобто не використовуються глобальні параметри колекції.

Робота алгоритму полягає в наступному:

– для кожного документа колекції формується не більше трьох записів наступного вигляду:

<ss1,id,len,num,ss1,ss2,ss3,<ws1..5>>

<ss2,id,len,num,ss1,ss2,ss3,<ws1..5>>

<ss3,id,len,num,ss1,ss2,ss3,<ws1..5>>

де

<ss1,ss2,ss3> – сигнатури трьох найбільш довгих речень документа, впорядковані за спаданням довжини пропозицій (а при рівності довжин – за сигнатурами); <id> – ідентифікатор документа; <len> – довжина документа (кількість слів завдовжки три і більше букв); <num> – число пропозицій в документі; <ws1..5> = <ws1,ws2,ws3,ws4,ws5> – сигнатури п'яти самих довгих слів документу, впорядковані за спаданням довжини слів (а при рівності довжин – за сигнатурами);

– для коротких документів, що складаються з одного або двох речень, створюються одна або два записи виду:

<sis1,id,len,ss1,0,0,<ws1..5>>, або

<sis1,id,len,ss1,ss2,0,<ws1..5>>

<sis2,id,len,ss1,ss2,0,<ws1..5>>;

– отриманий файл сигнатур сортується, та для записів зі співпадаючим першим полем формується ланцюжка вигляду (найперше поле відкидається):

<id1,len1,num1,ss11,ss21,ss31,<ws11..51>>

<id2,len2,num2,ss12,ss22,ss32,<ws12..52>> ...;

– впорядковані за зростанням поля <len> – довжини документа (а при рівності довжин – по <id>). При цьому відношення довжин двох сусідніх елементів ланцюжка не повинно перевищувати деякого порогу, що визначається заданим мінімальним коефіцієнтом схожості документів (наприклад, для коефіцієнта 0,85 оптимальне значення порогу довівнює 1,15);

– при перевищенні порогу формування поточного ланцюжка закінчується і починається формування нового, незважаючи на те, що перше поле може залишатися колишнім. У результаті вихідний файл сигнатур перетворюється у множину порівняно коротких ланцюжків, що містять компактні (з невеликим розкидом) послідовності довжин документів, які монотонно спадають;

– файл ланцюжків сортується і з нього виключаються дублі та ланцюжки, які цілком входять в інші, довші. В результаті такої нормалізації виходить невеликого розміру файл ланцюжків, в якому коефіцієнт надмірності (входження елементів у різні ланцюжки) становить не більше 10%;

– оскільки всередині ланцюжка елементи впорядковані за зростанням довжин документів, для кожного елемента існує локальна околиця відносно невеликого розміру, визначеного тим же пороговим значенням відношення довжин;

– переглядаємо файл ланцюжків і для кожного елемента ланцюжка знаходимо дублікати за наступними правилами:

– шукаємо тільки в межах локальної околиці;

– порівнюємо пари, тільки якщо відношення числа пропозицій в них не перевищує деякого порогу (1,20) та з п'яти сигнатур слів <ws1..5> збігається не менше двох (не важливо в якому порядку);

– два документи вважаємо дублікатами, якщо у них збігаються сигнатури найдовших речень <ss1> або (для документів, що містять більше 5 речень, а таких - переважна більшість) з трьох сигнатур <ss1, ss2, ss3> збігаються дві (неважливо в якому порядку).

Для створення гібридного методу необхідно визначити сильні і слабкі сторони розглянутих алгоритмів, щоб вибрати основу для алгоритму і спосіб її модифікації. Результати оцінки наведених вище алгоритмів наведені в таблиці 1 [1].

Таблиця 1

Результати оцінки алгоритмів нечітких дублікатів

Метод	Повнота	Точність	F-міра
«3+5»	0,96	0,95	0,95
Long Sent	0,84	0,80	0,82
TF	0,60	0,94	0,73
Opt Freq	0,59	0,94	0,73
TF*RIDF	0,59	0,95	0,73
Heavy Sent	0,62	0,86	0,72
TF*IDF	0,54	0,96	0,69
Lex Rand	0,50	0,97	0,66
Descr Words	0,44	0,77	0,56
Log Shingles	0,39	0,97	0,56
Megashingles	0,36	0,91	0,51

У таблиці наведені три основні характеристики оцінки алгоритмів: точність, повнота і F-міра. Результати впорядковані за спаданням F-міри як результуючої характеристики, пов'язаної з першими двома параметрами. Очевидно, що найбільш ефективним є алгоритм «3+5», що має дуже високі показники за повнотою та точністю. У той же час деякі алгоритми (Log Shingles, Lex Rand і TF \* IDF) мають більш високу точність. У двох з них використовуються частотні характеристики слів тексту.

Оцінюючи далі результати з табл. 1, виділяємо п'ять алгоритмів, що мають найбільш високі показники повноти результатів: «3+5», Long Sent, TF, Opt Freq, TF\*RIDF. При цьому в п'ятірку найбільш точних алгоритмів входять Log Shingles, Lex Rand, TF \* IDF, «3 +5» і TF \* RIDF.

Таким чином, можна констатувати, що використання частотних характеристик дозволяє підвищити точність алгоритмів, а оцінка за довжиною слів і речень в кінцевому підсумку збільшує їх повноту. З таблиці також випливає, що алгоритми з хорошими показниками точності зазвичай мають низьку повноту, тому що вони відкидають занадто багато документів, що є нечіткими дублікатами, при цьому пропускаючи лише малу частину документів. Найбільшу повноту дає алгоритм «3 + 5» (різниця між показниками повноти алгоритму «3 +5» та інших алгоритмів становить 12%).

Як наслідок, доцільно взяти за основу гібридного методу саме алгоритм «3 + 5», доповнивши і модифікувавши його частотними параметрами з метою істотного підвищення точності (при незначному зменшенні повноти).

### 3. Розробка гібридного методу

Пропонований гібридний метод виявлення нечітких дублікатів заснований на аналізі як частотних характеристик, так і довжини слів. За аналогією з методом «3 + 5», назвемо гібридний метод методом «3 + 2», оскільки тут будуть враховуватися частота і довжина в співвідношенні 3/2. Відзначимо, що методи «3 + 5» і «3 + 2» багато в чому близькі, однак істотно розрізняються за процедурами вибору пропозицій і слів у них.

Для оцінки можливого дублювання в методі «3 + 2» пропонується враховувати по 5 пропозицій текстового документа і по 5 слів з цих пропозицій.

Наведемо опис етапів реалізації методу «3 + 2».

Етап 1: Створення частотного словника документа.

Цей етап відповідає початковим етапам частотних алгоритмів. Для кожного значущого слова підраховується ступінь його зустрічальності (як і в звичайних частотних алгоритмах – наприклад, в алгоритмі TF). Далі обираються 10 слів, які найбільш часто зустрічаються в документі, вони упорядковуються за частотою входження в документ, а при рівній кількості – за довжиною (якщо і довжини дорівнюють – в алфавітному порядку).

Етап 2: Вибір 5 речень, що описують документ.

У відповідності з принципом «3 + 2» спочатку обираються три речення, в які входить найбільше число слів з частотного словника документа. Далі обираються два найбільш довгих речення. Якщо речення повторюється, воно опускається і шукається наступне за довжиною речення. Перші три речення упорядковуються за кількістю входження слів з частотного словника документа, а при рівності слів – за довжиною. Останні два речення упорядковуються за довжиною.

Етап 3: Вибір 5 слів, що описують речення.

Тут також використовується принцип 3 + 2: спочатку обираються слова з частотного словника документа (не більше трьох), які доповнюються найдовшими словами документа до загальної кількості слів, рівній 5. Якщо слово повторюється, воно відкидається і береться наступне за довжиною слово. Перші слова (обрані з частотного словника документа) розташовуються в порядку, у якому вони розташовані в частотному словнику. Решта упорядковуються по довжині.

Етап 4: Визначення сигнатур слів і речень.

Сигнатура – символ або ряд символів, що утворюють унікальний ідентифікатор об'єкта, предмета чи документа.

В якості сигнатур тексту використовуємо контрольні суми, що визначаються за алгоритмом CRC32, який найбільш широко застосовується у всіх розглянутих вище методах.

Етап 5: Складання ланцюжків за принципом алгоритму «3 + 5», зі зміненою кількістю ланцюжків для 1 документа (5 замість 3). При цьому ланцюжки будуть мати наступну структуру:

```
<ss1,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss2,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss3,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss4,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss5,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
```

Якщо речень менше п'яти, то створюється кількість ланцюжків, рівна кількості речень, з таким же порядком, як і в алгоритмі «3 + 5», а частотні характеристики опускаються.

Файл ланцюжків складається так само, як і в алгоритмі «3 + 5».

Пошук здійснюється згідно з принципами алгоритму «3 + 5»: пошук проводиться тільки в межах локальної околиці; пари порівнюються, тільки якщо відношення числа пропозицій в них не перевищує деякого порогу (1.20) і з п'яти сигнатур слів <ws1..5> збігається не менше двох (неважливо в якому порядку); два документи вважаються дублікатами, якщо у них збігаються сигнатури <ss1> або (для документів, що містять більше 5 речень) з п'яти сигнатур <ss1, ss2, ss3, ss4, ss5,> співпадають N сигнатур, де N = 2 у випадку пріоритетності повноти над точністю, або N = 3 у разі пріоритетності точності над повнотою (неважливо в якому порядку).

### 4. Вхідні та вихідні дані

Метод проходив тестування на матеріалах, наданих електронною бібліотекою ХНУРЕ – а саме, методичних вказівках до лабораторних і курсових робіт. Виходячи зі специфіки даних матеріалів, подібні документи перевидаються практично щорічно, з незначними змінами. Ці документи були оцінені експертами і в них були виділені пари і множини нечітких дублікатів щодо кожного документа.

Вихідними даними методу є отримані в результаті роботи множини нечітких дублікатів щодо кожного документа.

### 5. Оцінка результатів роботи

Так як для визначення дубліката було вибрано два варіанти алгоритму (для досягнення більшої точності й для більшої повноти), нижче наведені результати його роботи в двох цих випадках.

Повнота визначається як відношення числа знайдених нечітких дублікатів до загального числа нечітких дублікатів документів в базі:

$$Recall = \frac{|D_{dub} \cap D_{retr}|}{|D_{dub}|},$$

де  $D_{dub}$  — множина нечітких дублікатів документів

в базі, а  $D_{retr}$  — множина документів, знайдених системою.

Точність визначається як відношення числа нечітких дублікатів, знайдених методом, до загально-го числа знайдених документів:

$$Precision = \frac{|D_{dub} \cap D_{retr}|}{|D_{retr}|},$$

де  $D_{dub}$  — множина нечітких дублікатів документів в базі, а  $D_{retr}$  — множина документів, знайдених системою.

F-мера при однаковій вазі точності і повноти:

$$F = \frac{2PR}{R+P},$$

де  $R$  — повнота, а  $P$  — точність.

Таблиця 2

Результати роботи методу

Метод	Повнота	Точність	F-міра
«3+5» (оригінальний)	0,96	0,95	0,954
«3+2»; N=3 (орієнтованість на точність)	0,927	0,978	0,952
«3+2»; N=2 (орієнтованість на мінімальну втрату повноти)	0,956	0,963	0,959

З табл. 2 видно, що точність роботи методу «3 + 5» зростає у двох випадках, проте втрати повноти залишаються неминучими. Тим не менш, у другому випадку втрата повноти складає всього 0,4%, тоді як точність підвищується на 1,3%. Це відповідає збільшенню інтегрованого показника F-міри, а значить, можна говорити про загальне підвищення ефективності роботи алгоритму. Таким чином, результати тестування підтверджують доцільність застосування запропонованого гібридного методу для автоматичного виявлення нечітких дублікатів в електронних текстах. Можлива його подальша модернізація з перевизначенням деяких параметрів (наприклад, відношення частотної складової до складових, обраних на основі довжин), зменшення кількості ланцюжків, що складаються при обробці тексту тощо. Перспективним видається проведення досліджень методу «3 + 2», спрямованих на збільшення показника повноти без зменшення точності

### Висновки

В роботі наведено результати роботи гібридного методу пошуку нечітких дублікатів, розробленого методом модифікації алгоритму «3+5» з роботи [1] з введенням частотних характеристик слів з документа. Цей метод має на меті покращення показників точності виявлення дублікатів з найменшими можливими втратами у повноті. Роботу запропонованого гібридного методу було протестовано на виборці електронних документів

(методичних вказівок, наданих електронною бібліотекою ХНУРЕ).

В ході тестування розроблений метод виявлення нечітких дублікатів показав суттєве покращення результатів за показниками точності з невеликою втратою у показниках повноти.

Перспективною здається подальша модернізація алгоритму за допомогою удосконалення процедур вибору параметрів створення ланцюгів та інших ключових параметрів.

Перспективним також є знаходження дублікатів медіа-файлів (аудіо-, відео файлів, файлів зображень) з певною модифікацією цього алгоритму та відповідною попередньою обробкою.

**Список літератури:** 1. *Зеленков Ю.Г.* Сравнительный анализ методов определения нечетких дубликатов для Web-документов [Текст] / Ю.Г. Зеленков, И.В. Сегалович // RCDL'2007: Сб. работ участников конкурса: Переславль-Залесский, Россия, 2007. — Том 1. — С. 166-174. 2. *Чала, Л. Э.* Определение нечетких дубликатов медиафайлов [Текст] / Л.Э.Чала, П.Ю.Попаденко // Международная научно-техническая конференция, посвященная 75-летию В.В. Свиридова «Информационные системы и технологии» ИСТ-2012: материалы першої міжнар. наук.-техн. конф., 22–29 вересня 2012 р. — С.74. 3. *S. Robertson.* Okapi at trec-3 [Text] / S. Robertson, S. Walker S. Jones, M. Hancock-Beaulieu, M. Gatford // The Third Text REtrieval Conference (TREC-3), 1995 — P.109-127. 4. *K. Church.* Poisson mcxtures [Text] / K. Church, W. Gale. // Natural Language Engineering, 1995, P. 163–190. 5. *A. Broder.* On the resemblance and containment of documents, Compression and Complexity of Sequences [Electronic resource] SEQUENCES'97, IEEE Computer Society, 1998, — P. 21-29.

Надійшла до редколегії 07.12.2012

УДК 004.912

**Метод обнаружения нечетких дубликатов электронных текстовых документов** / Л.Э. Чала, П.Ю. Попаденко // Бионика интеллекта: науч.-техн. журнал. — 2013. — № 1 (80). — С. 88-92.

В работе проведен анализ методов обнаружения нечетких дубликатов текстовых файлов. Показано, что существующие методы обнаружения нечетких дубликатов направлены на минимизацию вычислительной сложности с одновременным увеличением полноты и точности алгоритма. В данной работе предложен и протестирован гибридный метод, который позволяет существенно улучшить качество обнаружения нечетких дубликатов при незначительном увеличении вычислительной сложности базового алгоритма.

Табл. 2. Библиогр.: 5 назв.

УДК 004.912

**Method of near-duplicate detection for electronic textual documents** / L.E. Chalaya, P.Yu.Popadenko // Bionics of Intelligense: Sci. Mag. — 2013. — № 1 (80). — P. 88-92.

This paper analyzes the methods of near-duplicate detection in text files. It is shown that the existing methods of near-duplicates detection directed on minimization the computational complexity while increasing the completeness and accuracy of the algorithm. In this paper the hybrid method was proposed and tested, which can significantly improve the quality of near-duplicate detection slightly increasing the computational complexity of the basic algorithm.

Tab. 2. Ref.: 5 items.

УДК 004.932.2:004.93'1



Н.В. Власенко

ХНУРЭ, г. Харьков, Украина, gorohovatskaja@gmail.com

## ПОСТРОЕНИЕ ИНФОРМАТИВНЫХ КОМПАКТНЫХ ОПИСАНИЙ И КЛАССИФИКАЦИЯ ОБЪЕКТОВ ПУТЕМ ПРЕДСТАВЛЕНИЯ В ОРТОГОНАЛЬНОМ БАЗИСЕ

Рассматривается применение ортогонального базиса для построения сжатых информативных описаний изображений. Предложен эффективный метод построения компактных описаний на основе применения базиса функций Уолша. Приведены результаты сравнительных экспериментов предложенного метода с известными подходами.

СТРУКТУРНОЕ ОПИСАНИЕ ВИДЕО-ОБЕКТА, ФУНКЦИИ УОЛША, ОРТОГОНАЛЬНЫЙ БАЗИС, СЖАТИЕ, КЛАССИФИКАЦИЯ

### Введение

Классификация – это отнесение данных к определенному классу путем выделения существенных признаков из общей массы данных. Архиважным здесь представляется формирование системы признаков, количество и качество которых необходимо и достаточно для осуществления эффективной классификации. Известные методы выявления точечных особенностей изображений [1, 2] позволяют сформировать описание видео-объекта в виде конечного множества векторов-дескрипторов – характерных признаков (ХП).

Модель сигнала изображения представляют как конечный набор  $Z$  ХП из пространства  $R^n$  векторов фиксированной размерности с действительными компонентами [1-4]. Описание видео-объекта – это подмножество  $Z \subset R^n$ . При этом векторы, элементы описания, полученного, например, с использованием методов SIFT или SURF [1], являются инвариантными к смещениям, масштабу и вращению. Нумерация элементов во множестве  $Z$  получается в процессе формирования описания.

Описание  $Z$  имеет в некотором смысле противоречивый характер с точки зрения требований к классификации. С одной стороны, оно является в некотором смысле однородным (векторы из одного видео-объекта). С другой стороны, возникают трудности из-за того, что векторы из описаний разных объектов могут быть эквивалентными. Кроме того, в практических задачах классификатору часто достаточно ограниченного числа элементов описания  $Z$  для принятия решения с тем же уровнем ошибки. Это порождает возможность постановки и решения проблемы уменьшения избыточности путем достижения компактности описаний для эффективного хранения и обработки данных [3].

Эффективное представление в пространстве векторов может быть связано с преобразованием к некоторому ортогональному пространству. Прикладной интерес по причине достаточно простой программной или аппаратной реализации вызывает применение в этих целях функций Уолша (ФУ) и преобразований, связанных с ними [3-5].

Для компактности опишем  $Z$  в виде матрицы  $Z = \{z_{ij}\}, i = \overline{1, s}, j = \overline{1, n}$ , строки которой содержат  $s$  векторов описания. Поставим задачу путем перехода в другое пространство на основе построения отображения  $\Theta: Z \rightarrow U$  за счет применения ортогонального разложения по системе ФУ сформировать более компактное информативное описание, обладающее лучшими свойствами для классификации.

Цель статьи – разработка эффективного в плане быстродействия метода классификации изображений на основе построения сжатых описаний с использованием наиболее информативных компонентов спектра Уолша.

Задачи исследования включают формализацию и обоснование метода формирования компактного информативного описания и практическое сопоставление синтезированного метода с известными подходами при применении их для обработки изображений из реальных баз видео-информации.

### 2. Метод сжатия описания на основе ортогонального базиса

Применения двумерных преобразований в обработке изображений связаны с подчеркиванием отдельных свойств, кодированием изображений путем отбрасывания малых по величине коэффициентов преобразования, сокращением размерности описания для повышения быстродействия [6].

Рассмотрим преобразование, связанное с разложением по семейству прямоугольных базисных функций, в качестве которых рассмотрим дискретные ФУ, представляющие собой целочисленные векторы конечной размерности, состоящие из 1 и -1. В практических применениях исследуются ФУ с размерностью  $n = 2^k$ . Полный набор ФУ для фиксированного  $n$  образует ортогональную матрицу Адамара [3]. Преобразуем описание путем умножения матриц

$$U = Z * A, \quad (1)$$

где  $A$  – матрица Адамара размера  $n \times n$ , включающая  $n$  векторов ФУ  $w_1, \dots, w_n$ . Например, матрица Адамара размерностью 4 имеет вид:

$$A = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

где коэффициент  $1/2$  служит для обеспечения свойства ортонормальности. Ортонормальность при практическом применении ФУ играет не такую принципиальную роль, как, например, при использовании базиса Фрэя-Чена [2], т.к. нормировочные коэффициенты равны для всех ФУ одного базиса. Выражение (1) можно трактовать как одномерное преобразование Уолша, выполненное для каждого из векторов описания  $Z$ .

Метод синтеза сжатого описания через отображение  $\Theta: Z \rightarrow U$  представим как последовательность шагов.

1. Применяя к исходному описанию преобразование (1), получим матрицу  $U = \{u_{ij}\}$  размером  $s \times n$ . Вычислим квадраты элементов  $U$  и будем анализировать матрицу  $U^2 = \{u_{ij}^2\}$ . Значение  $u_{ij}^2$  соответствует энергии ХП с номером  $i$  из описания  $Z$ , приходящейся на базисный вектор  $w_j$  (с номером  $j$ ). Как известно, при представлении в ортонормированном базисе полная энергия сигнала равна сумме компонентов энергии, соответствующих всем базисным векторам [2].

2. В целях построения информативного кортежа из спектра применим анализ посредством дисперсионного критерия [3,4], вычисляя дисперсии элементов в каждом из  $n$  столбцов матрицы  $U^2$ :

$$\sigma_i^2 = \frac{1}{s-1} \sum_{k=1}^s (u_{ik}^2 - u_i^2)^2,$$

где  $u_i^2 = \frac{1}{s} \sum_{k=1}^s u_{ik}^2$ ,  $i = \overline{1, n}$ . Значения дисперсии характеризуют разброс величин откликов ХП для отдельных ФУ спектра. Далее отсортируем полученный массив дисперсий  $\sigma_1^2, \dots, \sigma_n^2$  с сохранением номеров ФУ, для которых они вычислены.

3. Выберем фиксированное число  $m \ll n$  ФУ с наибольшими дисперсиями. Выбранные ФУ  $w_1, \dots, w_m$  с их исходными номерами  $j_1, \dots, j_m$  будем использовать как наиболее информативную часть базиса для получения сжатого описания. Здесь применен подход [5], где в целях выделения и

фильтрации протяженных объектов используется ограниченный набор ФУ с наиболее подходящими свойствами.

4. Осуществим классификацию векторов описания  $Z$  на  $m$  классов в соответствии с выбранным информативным кортежем ФУ. Для этого проанализируем избранные  $m$  столбцов матрицы  $U^2$  и для каждого из  $s$  векторов исходного описания реализуем оптимальное правило  $R_Z$  отнесения вектора  $z_q \in Z$  к одному из классов  $k$  в диапазоне  $1, \dots, m$ :

$$R_Z: z_q \in k \mid \arg \max_{h=1, \dots, m} u_{qh}^2 = k, k = \overline{1, m}, q = \overline{1, s}. \quad (2)$$

Таким образом, каждый элемент исходного описания будет отнесен к тому из  $m$  классов, для которого квадрат отклика с соответствующей функцией Уолша будет максимальным. Правило  $R_Z$  реализует классификацию на основе оптимальной согласованной фильтрации элементов описания в ортогональном базисе кортежа ФУ. Здесь возможны варианты обработки с ограничениями и отсеиванием, когда вектор ХП не классифицируется ни к одному из классов в силу незначимости отклика.

5. В результате осуществления (2) получим распределение  $s$  элементов по  $m$  классам. Сформируем сжатое описание  $u^*$  в интегрированном виде как вектор целых чисел

$$u^* = (u_1, \dots, u_m), \sum_{j=1}^m u_j = s, \quad (3)$$

элементы  $u_j$  которого – количества векторов ХП, отнесенные к классу  $j$ . Учитывая, что эталоны классов могут содержать в своем составе разное число ХП, в практических задачах часто необходимо осуществить нормировку описания (3):  $u_i = u_i / s$ , что приводит к соотношению  $\sum_{j=1}^m u_j = 1$ . Схема построения метода приведена на рис. 1.

Предложенный метод преобразует исходное описание в виде матрицы размера  $s \times n$  к вектору из  $m$  числовых значений, причем можно выбрать  $m \ll n$ , что существенно снижает вычислительные затраты особенно при больших значениях числа векторов  $s$ , которое в практических задачах достигает 200-400. При этом преобразование эталонных описаний к сжатому виду выполняется на предварительном этапе и не влияет на общее время

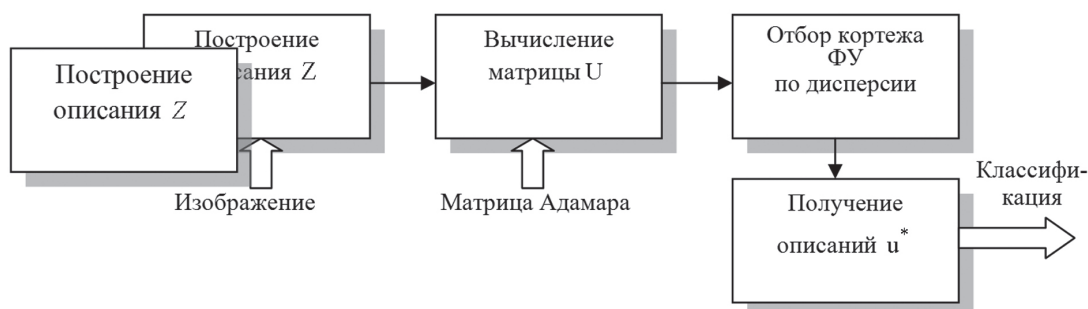


Рис. 1. Схема построения метода

классификации. Отбрасывание части коэффициентов разложения не вызывает существенных ошибок или потери нужных свойств. Размером объема описания (параметр  $m$ ) можно управлять, повышая характеристику достоверности.

Выбор функций Уолша в качестве ортогонального базиса, кроме простоты аппаратной реализации, дополнительно обоснован их высокими фильтрационными свойствами относительно действия аддитивных флуктуационных помех, которые характерны для прикладных задач обработки изображений [5].

Важным частным случаем рассмотренного метода есть вариант  $m = n$ , т.е. все имеющиеся ФУ участвуют в классификации. Для этой ситуации сжатие описания возможно на основе анализа значимости элементов представления (3) путем отбрасывания малых или выделения  $m$  наибольших по величине среди  $u_j$ . Однако наши исследования показывают, что энергия представления в базисе ФУ для векторов ХП на практике распределяется очень неравномерно по спектру. Три ФУ (с номерами 1, 3, 49) в значительной мере доминируют над остальными, что приводит к классификации подавляющего числа ХП к этим трем классам. За счет этого фактора происходит чрезмерное сжатие информации, и в результате ухудшаются показатели качества классификации, особенно при действии помех. Такая картина наблюдается для нескольких исследуемых баз видеоданных, что говорит об объективности отмеченных спектральных свойств и полученных выводов.

## 2. Классификация на основе сжатого информативного описания

Будем использовать сжатое описание (3) в задаче классификации путем поиска эталона с наибольшим значением меры подобия по правилу [8]:

$$c(u) = \arg \underset{q \in \{1, \dots, K_Q\}}{\text{opt}} \vartheta[u, u(q)], \quad (4)$$

где  $c(u)$  – функция классификации описания  $u$ ;  $\vartheta[u, u(q)]$  – мера подобия между векторами в пространстве  $R^m$ ;  $u(q)$  – вектор описания эталона с номером  $q$ ;  $K_Q$  – множество классов. При этом в процессе классификации описание объекта строится с учетом ФУ, синтезированных для видео-базы, что обеспечивает дополнительные согласование и помехозащищенность. В результате представления с учетом сжатия формируется множество  $K_p$  классов признаков, которое включает  $m$  классов и основано на выделенном кортеже ФУ.

За счет представления координатного векторного пространства ХП в ортогональном базисе ФУ получаем новое векторное пространство, переходя от представления объекта в виде множества векторов к одному вектору, отражающему спектральное представление для наиболее значимых (информативных)

элементов базиса. Теперь классификация сводится к представлению объекта в виде вектора  $u$  з заданным для базы множеством классов  $K_p$ .

Мера подобия  $\vartheta$  в (4) может быть вычислена либо путем голосования, либо путем векторного представления объекта и сопоставления описаний как векторов. Понятно, что описание может при этом содержать элементы помех и ложных фоновых объектов, что приводит к искажениям вектора (3). В частности, можно использовать метрику в пространстве мультимножеств:

$$\rho(A, B) = \frac{\sum_i w_i |u_A(i) - u_B(i)|}{\sum_i w_i \max[u_A(i), u_B(i)]}, \quad (5)$$

где  $A, B$  – сопоставляемые векторы;  $u_A(i), u_B(i)$  – значения компонент векторов с номером  $i$ ;  $w_i$  – весовые коэффициенты.

Кроме свойств различимости и помехозащищенности, важным моментом остается оценка выигрыша в быстродействии предложенной обработки по сравнению с традиционным голосованием. Проведем ее путем подсчета числа вычислительных операций. Обозначим  $t_E$  – время вычисления квадрата евклидова расстояния или скалярного произведения между векторами ХП размером  $n$ ,  $t_c$  – время сравнения и суммирования двух чисел при поиске минимума расстояния и инкрементации числа голосов,  $s, s_1$  – мощности описаний эталона и объекта. Тогда оценка времени при вычислении подобия в виде числа голосов выглядит как  $T_v = s s_1 (t_E + t_c)$ . Время для вычисления подобия в соответствии с обработкой (1)-(3) можно оценить как  $T_m = s_1 [m(t_E + t_c) + t_c] + (t_E m) / n$ , где выражение  $(t_E m) / n$  соответствует вычислению подобия между векторами размером  $m$ . Отношение оценок имеет вид

$$\alpha = \frac{T_v}{T_m} = \frac{s s_1 (t_E + t_c)}{s_1 [m(t_E + t_c) + t_c] + (t_E m) / n}. \quad (6)$$

Детальный анализ (6) показывает, что в наибольшей степени величина  $\alpha$  зависит от соотношения  $s / m$ , т.е. пропорционально увеличению  $s$  и уменьшается с увеличением  $m$ . Отсутствие зависимости от  $n$  можно объяснить тем, что оба варианта включают анализ всего описания классифицируемого объекта, включающего  $s_1$  векторов одинаковой размерности  $n$ .

Конкретно при  $m = 8$ ,  $s = 100$  значение  $\alpha \approx 12,5$ , а при  $s = 200$  –  $\alpha \approx 25$ . Как видим, преимущества предложенного метода в быстродействии для практических значений параметров могут достигать 10-20 раз.

## 3. Результаты компьютерного моделирования

Эксперименты проводились для базы из 12-ти эталонных изображений домашних животных (кошки) [8]. Предварительная обработка базы состояла в следующем. Для каждого эталона с

использованием метода SURF получено структурное описание как набор векторов размерностью  $n=64$ , при этом число векторов для эталонов базы колеблется от 66 до 262. Для перехода к спектральному пространству Уолша вычислены скалярное произведение каждого вектора с множеством ФУ и матрица  $U^2$  для всей базы. Значения элементов  $u_{ij}^2$  есть базовыми данными для проведения классификации.

Основой для применения метода сжатия описания является предположение о неравнозначности его элементов, а также предположение о том, что для осуществления классификации с высоким уровнем помехозащищенности достаточно ограниченного числа компонентов в базисе ФУ. Для отбора информативного кортежа ФУ размером  $m=8$  элементов применен дисперсионный критерий. Наибольшие 16 значений дисперсии квадратов откликов представлены в табл. 1.

Как видим из табл. 1, функции 1, 3, 49 дают наиболее значимые отклики, которые существенно выделяются среди откликов остальных ФУ. Использование при классификации этих трех ФУ, как показало моделирование, приводит к тому, что подавляющее число элементов описания будет отнесено к одному из этих трех классов, что в результате значительно ухудшает показатели различимости объектов.

По этой причине исключим из рассмотрения ФУ с номерами 1, 3, 49 и выберем из табл. 1 в качестве информативного сокращенного базиса следующие восемь ФУ, на базе которых построим классификацию.

Выбранные 8 ФУ позволяют классифицировать векторы описания  $Z$  на классы и, как результат, получить для каждого эталона новое описание в виде вектора – кортежа из 8 коэффициентов по правилу (2). В результате получим новое информативное описание в виде векторов, содержащих количества ХП исходного описания, отнесенных к сформированному классу с использованием информативных ФУ.

Синтезированные описания представлены в табл.2. Для обеспечения сравнимости описаний из разных классов выполнена нормировка значений табл. 2 на число векторов в описании изображения/эталона в соответствии с выражением  $u_i = u_i / s$ .

Далее экспериментально исследовалась классификация объектов с использованием компактных описаний при  $m=8$ , в результате получена оценка вероятности правильной классификации в зависимости от уровня аддитивных помех.

В качестве критерия близости преобразованных описаний использована мера (5).

Уровень действия аддитивной помехи, действующей на компоненты векторов, характеризовался дисперсией  $\sigma^2$ , которая напрямую отражает соотношение сигнал-шум. Описания подвергались действию помехи, после чего проводилось сравнение с эталонами по минимуму (5).

На вход системы классификации поступало описание  $Z^*$ , подверженное действию помехи в виде  $Z^* = Z + \xi$ , где  $\xi$  – матрица шума с независимыми нормально распределенными элементами

Таблица 1

Наибольшие 16 значений дисперсии для базы домашних животных

№ п\п	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
№ФУ	3	1	49	33	2	50	4	35	34	36	17	51	13	11	52	19
дисперсия	66,26	51,85	13,7	8,63	7,37	6,82	5,48	5,35	3,18	3,048	2,729	2,207	2,102	1,94	1,699	1,614

Таблица 2

Состав эталонных классов по числу классифицированных ХП

Номер класса ХП/ номер ФУ								Число векторов эталона	Класс эталона
1/2	2/4	3/17	4/33	5/34	6/35	7/36	8/50		
25	11	8	35	13	23	13	18	146	1
3	8	1	14	6	27	1	6	66	2
56	41	8	57	16	57	17	10	262	3
21	2	5	38	12	18	3	18	117	4
35	27	2	41	12	31	15	15	178	5
33	19	9	43	13	35	15	12	179	6
43	31	6	52	9	31	15	9	196	7
31	11	10	36	16	20	8	14	146	8
40	15	4	29	9	25	1	5	128	9
30	12	11	37	12	13	13	24	152	10
46	25	6	42	11	41	13	13	197	11
27	11	2	13	7	9	9	11	89	12

(матожидание – 0, дисперсия –  $\sigma^2$ ). Вероятность правильной классификации вычисляется как отношение числа правильных ответов к общему числу экспериментов. В результате 100 экспериментов получены зависимости вероятности правильной классификации от уровня помехи  $\sigma^2$  (рис. 2).

Проведен сравнительный эксперимент по классификации для базы домашних животных, в котором описание представлено тремя вариантами: в пространстве 8 отобранных по дисперсионному критерию ФУ (с отбрасыванием ФУ 1, 3, 49); в пространстве 8 ФУ с наименьшими дисперсиями; в пространстве полного базиса Уолша (64 ФУ). Для сопоставления проведены также эксперименты для метода голосования путем подсчета числа голосов, порог эквивалентности двух ХП выбран равным 0,16 (1% от максимума значения евклидовой метрики).

Как видим и рис. 2, предложенный в статье метод построения информативного описания имеет несомненное преимущество в плане помехозащищенности перед традиционным голосованием. До уровня  $\sigma^2 = 0,02$  эффективность методов примерно одинакова (вероятность правильного распознавания выше 0,94), далее с увеличением  $\sigma^2$  кривая вероятности для разработанного метода выше, и при  $\sigma^2 = 0,03$  вероятности составляют 0,8 и 0,6 соответственно. Высокая устойчивость к шуму обусловлена прежде всего интегральными свойствами ФУ. В то же время и быстродействие разработанного метода существенно выше: сравнительное время классификации одного изображения составило 0,03 сек и 0,95 сек соответственно, т.е. более чем в 30 раз лучше (компьютер Intel Core2 Duo 2,67 ГГц).

Другие модификации предложенного метода, как видно из рис. 2, имеют несколько худшие показатели помехозащищенности: если для предложенного варианта при  $\sigma^2 = 0,01$  вероятность равна 0,99, то для методов в пространстве 8 ФУ с наименьшими значениями дисперсии и в пространстве

полного базиса Уолша из 64 ФУ вероятности составили 0,53, что неприемлемо на практике.

## Выводы

Предложенный в работе метод построения информативных компактных описаний на основе анализа свойств структурного представления в ортогональном базисе Уолша позволяет проводить классификацию объектов с высоким быстродействием без снижения уровня помехозащищенности. Проведенные вычислительные эксперименты подтвердили предположение о возможности существенного сокращения описания путем отбрасывания малых по величине коэффициентов преобразования.

При сравнении уровня помехозащищенности с традиционным голосованием установлено, что метод построения компактных описаний в пространстве кортежа выделенных ФУ не уступает по помехозащищенности голосованию, что обусловлено интегральными свойствами базиса Уолша. Основным достоинством предложенного подхода является значительный выигрыш в быстродействии: теоретически – в 10-20 раз, практически – в 30 раз по сравнению с голосованием.

В результате исследования разработан новый метод построения сжатых информативных описаний изображений на основе отбора наиболее значимых компонентов спектра преобразования Уолша, обеспечивающий высокое быстродействие в сравнении с традиционным голосованием без снижения показателя помехозащищенности.

Практическая ценность работы состоит в экспериментальном подтверждении эффективности синтезированного метода на прикладной базе видео-данных домашних животных.

Перспективное направление дальнейших исследований состоит в разработке адаптивной процедуры выбора функций Уолша на основе исследования свойств эталонов базы.

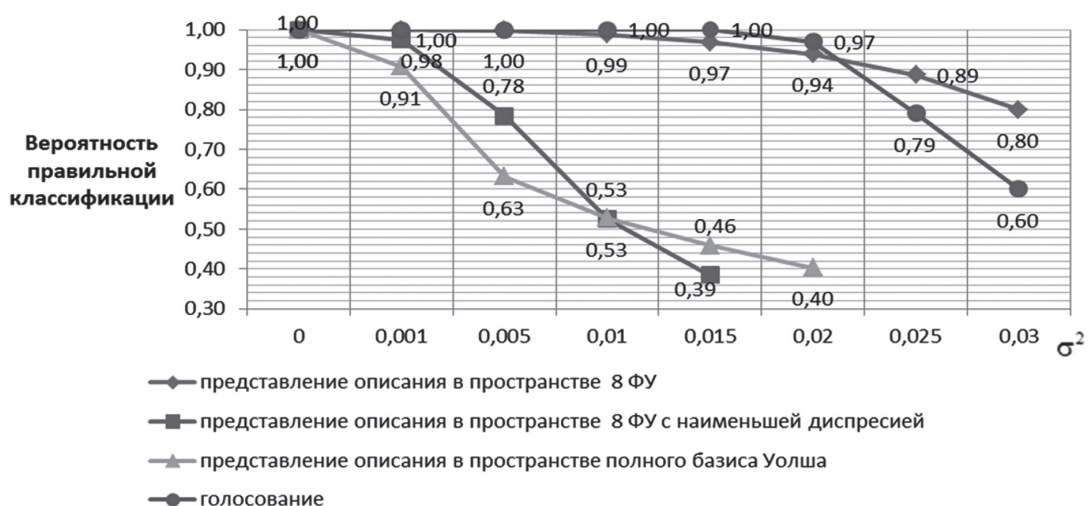


Рис. 2. Зависимость качества классификации от уровня помехи

**Список литературы:** 1. *Lowe D.* Distinctive Image Features from Scale-Invariant Keypoints/ D. Lowe // In International Journal of Computer Vision. – 2004. – vol. 60. – P. 91–110. 2. *Шапиро Л.* Компьютерное зрение [Текст] / Л. Шапиро, Дж. Стокман: пер. с англ.— М.: БИНОМ. Лаборатория знаний, 2006. – 752 с. 3. *Ахмед Н.* Ортогональные преобразования при обработке цифровых сигналов [Текст] / Н. Ахмед, К.Р. Рао: пер. с англ. – М.: Связь, 1980. – 248 с. 4. *Залманзон Л. А.* Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи и других областях [Текст] / Л. А. Залманзон. – М.: Наука, 1989. – 496 с. 5. *Ересько Ю.Н.* Локализация изображений в автоматических визирах / Ю.Н. Ересько. – М.: Компания Спутник+, 2002. – 357 с. 6. *Pratt W. K.* Digital Image Processing / W. K. Pratt. – New York: John Wiley and Sons Inc., 2001. – 723 p. 7. Прикладная статистика: Классификация и снижение размерности: Справ. изд. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин; под ред. С.А. Айвазяна. – М.: Финансы и статистика, 1989. – 607 с. 8. *Власенко Н.В.* Анализ кортежей наиболее значимых соответствий между описаниями объектов в задачах метрического распознавания [Текст] / Н.В. Власенко, В.П. Машталир // Реестрация, зберігання і обробка даних.— 2012. – Т. 14, № 3. – С. 20-32.

*Поступила в редколлегию 16.01.2013*

УДК 004.932.2:004.93'1

**Побудова інформативних компактних описів та класифікація об'єктів шляхом представлення в ортогональному базисі** / Н.В. Власенко // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 93–98.

У роботі запропоновано ефективний метод класифікації зображень на основі побудови стиснених описів із використанням найбільш інформативних компонентів спектру Уолша. Представлено результати експериментального порівняння синтезованого методу з відомими підходами при використанні їх для обробки реальних баз відео-інформації.

Табл. 2. Лл. 2. Бібліогр.: 8 назв.

UDC 004.932.2:004.93'1

**Building a compact informative descriptions and classification of objects by presenting in an orthogonal basis** / N.V. Vlasenko // Bionics of Intelligence: Sci. Mag. – 2013. – № 1 (80). – P. 93–98.

In this paper we propose an effective method for image classification based on constructing a succinct description of using the most informative parts of the spectrum Walsh. The results of an experimental comparison of the synthesized method with known approaches when they are used to process live video data bases are presented.

Fig. 2. Tab. 2 Ref.: 8 items

УДК 004.55

Н.Г. Аксак<sup>1</sup>, С.А. Коргут<sup>2</sup>, Н.Г. Стрельцова<sup>3</sup><sup>1</sup>ХНУРЕ, м. Харків, Україна, ahak@kture.kharkov.ua;<sup>2</sup>ХНУРЕ, м. Харків, Україна, korguts@gmail.com;<sup>3</sup>ХНУРЕ, м. Харків, Україна, lexy@email.ua

## РАСШИРЕННЫЙ BROWSERANK НА ОСНОВЕ УЧЕТА ПОВЕДЕНЧЕСКИХ ФАКТОРОВ ПОЛЬЗОВАТЕЛЕЙ

В работе предлагается модификация метода BrowseRank, отличающаяся от существующего введением качественного коэффициента, учитывающего дополнительную информацию о поведении пользователей, что позволяет более точно определять важность веб-страницы. Приведены результаты экспериментального моделирования, подтверждающие эффективность предложенного подхода.

PAGERANK, BROWSERANK, ДАННЫЕ О ПОВЕДЕНИИ ПОЛЬЗОВАТЕЛЕЙ, ПОВЕДЕНЧЕСКИЙ ГРАФ, ЦЕПИ МАРКОВА, Q-МАТРИЦЫ

### Введение

Вес или важность страницы, отражающая ее «ценность» для посетителей Интернет, является ключевым фактором в алгоритмах веб-поиска, потому что для современных поисковых систем сканирование, индексирование и ранжирование документов во многом зависит именно от этого фактора [1].

Вес страницы в большинстве современных поисковых систем вычисляется при помощи графа переходов PageRank, либо основанном на нем BrowseRank, и известен как процесс анализа веб-ссылок и данных о поведении пользователя [4,10]. Большинство подобных алгоритмов предполагают, что если большое количество значимых страниц ссылается на документ – то он, по всей вероятности, тоже важен, и значимость вычисляется по графу переходов BrowseRank, который учитывает некоторые данные о поведении пользователя.

Из-за стремительного роста объема информации в Интернет, точный расчет веса веб-страницы становится критическим и является серьезной проблемой для поисковых систем [2]. Кроме того, проблемой для поисковых систем является ссылочный спам [7] – создание веб-страниц с целью вводить в заблуждение поисковые системы. Это страницы, в основном созданные в коммерческих целях, использующие различные методы для достижения более высокого рейтинга на страницах поисковых результатов.

Поэтому критичными являются точность определения важности страницы и стойкость метода вычисления к ссылочному спаму. В работе предлагается модификация метода BrowseRank для решения указанных проблем.

### 1. Цель и задачи исследования

Целью работы является анализ методов вычисления важности Web-документа для улучшения работы поисковых систем.

Для постижения поставленной цели необходимо решить следующие задачи:

– исследовать наиболее популярные методы вычисления значимости веб-страниц;

– исследовать влияние доступной информации о поведении пользователя для повышения релевантности документов.

### 2. Обзор существующих методов вычисления важности страницы

PageRank (PR) представляет собой метод вычисления веса страницы путём подсчёта важности ссылок на неё [5, 9, 12, 14], который применяется к коллекции веб-документов, связанных гиперссылками и назначает каждому из них некоторое численное значение, измеряющее его «важность» или «авторитетность» среди остальных документов, и представляется числом от 0 до 10. Чем больше ссылок на страницу, тем она становится «значительнее» [3]. Кроме того, вес страницы A определяется весом ссылки, передаваемой страницей B:

$$W_{PR}(A) = (1-d) + d \left( \frac{W_{PR}(X_1)}{C(X_1)} + \dots + \frac{W_{PR}(X_n)}{C(X_n)} \right),$$

где  $W_{PR}(A)$  – вес PageRank анализируемой страницы A;  $d$  – “damping factor” или коэффициент затухания в диапазоне от 0 до 1, который обычно устанавливают равным 0.85-0.9, выражает вероятность дальнейшего путешествия пользователя по ссылкам;  $W_{PR}(X_i)$  – вес PageRank страницы  $X_i$ , указывающей на страницу A;  $C(X_i)$  – число ссылок со страницы  $X_i$ .

Однако PR это только один из показателей, которые используются современными поисковыми системами для определения релевантности и важности страницы. PR отражает приблизительное качество страницы, но не связан с ее тематической релевантностью и не учитывает поведенческие факторы. Основным недостатком данного метода – его неустойчивость перед спамом (искусственной накруткой).

BrowseRank – метод корпорации Microsoft позволяющий вести учет поведения пользователей при ранжировании документов в результатах поиска [11].

Основное отличие BrowseRank от PageRank заключается в структуре графа. Если в технологии

PageRank узлами графа служат документы HTML, а ребрами – ссылки, то в технологии BrowseRank в качестве ребер выступают «клики» или количество переходов. Помимо этого, сохраняется вся мета-информация о длительности сессии в рамках того или иного ресурса.

Метод BrowseRank использует цепи Маркова на графе переходов для вычисления важности страницы, по сути, имитируя случайный «веб-серфинг» пользователя по ссылкам сайтов с учетом времени посещения страницы и типом перехода (по ссылке, по вводу URL).

Перед PageRank у этого метода есть следующие преимущества: большая устойчивость к ссылочному спаму и определение продолжительности сессии, что позволяет более надежно оценить, насколько документ важен для посетителя.

Данные о поведении пользователя в BrowseRank представлены в тройках, состоящих из  $\langle \text{URL}, \text{Time}, \text{Type} \rangle$ . Здесь URL содержит адрес веб-страницы, которую посещал пользователь, Time означает время посещения страницы, а Type указывает, как пользователь попал на данную страницу путем ввода URL в браузере (INPUT), либо посредством перехода по ссылке (CLICK). Записи сортируются в хронологическом порядке.

На их основе строится поведенческий граф:

$$G = \langle V, W, T, \delta \rangle,$$

где  $V = \{v_i\}$  – множество вершин;  $W = \{w_{ij}\}$  – множество весов;  $T = \{t_i\}$  – время посещения;  $\delta = \{\delta_i\}$  – вероятность распределения;  $(i, j = 1, \dots, N)$  – количество веб-страниц в графе переходов.

**Модель.** В модели BrowseRank для представления случайного веб-серфинга на графе переходов используется Марковская цепь с непрерывным временем [11].

$X_t$  – страница, которую пользователь посетил во время  $t, (t > 0)$ . Процесс  $X = \{X_t, t \geq 0\}$  является процессом Маркова с непрерывным временем.  $P_{ij}(t)$  – это вероятность перехода со страницы  $i$  на  $j$  для интервала времени  $t$  в этом процессе. Существует стационарное распределение вероятностей  $\pi$ , которое не зависит от  $t$  и связано с  $P(t) = [p_{ij}(t)]_{N \times N}$ , такое, что для каждого  $t > 0$

$$\pi = \pi P(t).$$

Каждое  $i$ -е вхождение распределения  $\pi$  обозначает отношение времени нахождения пользователя на  $i$ -ой странице к общему времени, проведенному на всех страницах, т.е. распределение является показателем важности страницы.

**Алгоритм.** Распределение вероятностей, служащее показателем веса страницы, вычисляется при помощи матрицы интенсивностей переходов. Матрица интенсивностей переходов определяется как производная  $P(t)$  при  $t$ , стремящемся к 0, т.е.

$Q = P'(0)$ , матрица  $Q = (Q_{ij})_{N \times N}$  далее –  $Q$ -матрица [6].

При конечном пространстве состояний соответствие между  $Q$ -матрицей и  $P(t)$  является однозначным и  $-\infty < q_{ij} < 0$ ;  $\sum_j q_{ij} \rightarrow 0$ . Соответственно

$Q$  процесс является Марковским процессом с непрерывным временем, то есть процесс перехода по ссылкам  $X = \{X_t, t \geq 0\}$  является  $Q$ -процессом из-за конечности пространства состояний.  $Q$ -процесс относится к модели с вложенной Марковской цепью. Так называемая вложенная Марковская цепь – это Марковский процесс с дискретным временем, представленный матрицей вероятностей переходов с нулевыми значениями во всех диагональных позициях и  $-\frac{q_{ij}}{q_{ii}}$  во всех остальных позициях, где все параметры  $q_{ij}, i, j = 1, \dots, N$  имеют те же значения, как и раньше.

$X$  представлен как процесс, а  $Y$  как вложенная Марковская цепь, полученная из  $Q$ -матрицы.  $\pi = (\pi_1, \dots, \pi_N)$  и  $\bar{\pi} = (\bar{\pi}_1, \dots, \bar{\pi}_N)$  обозначают стационарную вероятность распределений процессов  $X$  и  $Y$ :

$$\bar{\pi}_i = \frac{\pi_i}{\sum_{j=1}^N \frac{\pi_i}{q_{ij}}} \quad (1)$$

**Вычисление  $q_{ii}$ .** Для  $Q$ -процесса время просмотра страницы  $t_i$  для  $i$ -ой вершины графа регулируется экспоненциальным распределением с параметром  $q_{ii}$

$$P(t_i > t) = \exp(q_{ii}t).$$

Это означает, что определяется на основе большого числа значений времени просмотра страниц, взятых из данных о поведении пользователя. Эта задача не является тривиальной, потому что данные о пользователе обычно содержат погрешности, и для решения этой проблемы используется дополнительная модель погрешностей для получения объективных значений параметра.

Для страницы  $i$  используются значения времени посещения  $Z_1, Z_2, \dots, Z_{m_i}$  имеющие такое же распределение, как случайная переменная  $Z$ . Предполагается, что эта  $Z$  является комбинацией реально проведенного на странице времени  $t_i$  и погрешностей, т. е.:

$$Z = u + t_i.$$

Погрешность  $u$  регулируется распределением Хи-квадрат как  $x^{(k)}$ , его среднее значение будет  $k$ , а дисперсия –  $2k$ . Среднее значение и дисперсия  $Z$  равны  $\mu$  и  $\delta^2$  соответственно. Поскольку  $u$  и  $t_i$  не зависят друг от друга, получается:

$$\mu = E(Z) = E(u + t_i) = k - \frac{1}{q_{ij}},$$

$$\delta^2 = \text{Var}(Z) = \text{Var}(u + t_i) = 2k + \frac{1}{q_{ii}^2}.$$

Выборочное среднее  $\bar{Z} = \frac{1}{m_i} \sum_{l=1}^{m_i} Z_l$  и выборочная

дисперсия  $S^2 = \frac{1}{m_i - 1} \sum_{l=1}^{m_i} (Z_l - \bar{Z})^2$  являются объективными оценками для  $\mu$  и  $\delta^2$ .

Далее  $q_{ii}$  вычисляется через решение следующей задачи оптимизации:

$$\min_{q_{ii}} \left( \left( \bar{Z} + \frac{1}{q_{ii}} \right) - \frac{1}{2} \left( S^2 - \frac{1}{q_{ii}^2} \right) \right),$$

$$q_{ii} < 0.$$

К недостаткам метода BrowseRank относится то, что он имеет свои ограничения как модель для определения важности страницы, поскольку учитывает не все доступные данные о поведении пользователя, которые могут послужить надежной информацией для определения важности страницы. К таким данным относятся количество страниц, просмотренных на сайте, общее время посещения сайта и отдельных страниц, возвраты пользователя в результаты поиска после просмотра сайта.

### 3. Расширенный BrowseRank

Несмотря на свои очевидные преимущества, BrowseRank обладает следующим недостатком - он моделирует процесс блуждания по ссылочному графу с учетом данных о поведении пользователя, ограничиваясь только временем посещения страницы. Не учитываются другие доступные данные о поведении пользователя, которые могут улучшить показатель качества и, следовательно, важности веб-страницы.

Для решения этой проблемы предлагается использовать дополнительные данные о поведении пользователя, такие как глубина просмотра и возврат в результаты поиска.

Под глубиной просмотра будем понимать количество просмотренных страниц сайта за сессию. Если посетители сайта просматривают всего 1-2 страницы, это может свидетельствовать о том, что ресурс им неинтересен (они попали на него случайно). Другая возможная причина — сайт имеет слишком сложную и неудобную навигацию, и пользователи просто не могут найти нужную им информацию. В любом случае глубина просмотра отражает степень релевантности/качества ресурса.

Возврат в результаты поиска после посещения страницы, как правило, говорит о ее нерелевантности/некачественности. Соответственно, данный показатель может быть использован как надежный источник информации для вычисления качества страницы.

Как правило, в зависимости от целей, поставленных перед сайтом, и сложности решаемых задач, сайты можно отнести к разным категориям (табл. 1).

Таблица 1

Назначение сайтов и их категории

Категории	Назначение сайтов
1	интернет-магазины, аукционы
2	информационные справочники
3	тематические, городские порталы, сайты новостей
4	блоги, форумы
5	корпоративные сайты, сайты организаций
6	общий доступ к файлам

Таким образом, граф сёрфинга пользователя, сгенерированного из данных о поведении пользователя принимает следующий вид:

$$G = \langle V, W, T, TR, RS, \delta \rangle,$$

где —  $V = \{v_j\}$  множество вершин;  $W = \{w_i\}$  — множество весов;  $T = \{t_i\}$  — время посещения;  $TR = \{tr_i\}$  — глубина просмотра;  $RS = \{rs_i\}$  — возврат в результаты поиска;  $\delta = \{\delta_i\}$  — вероятность распределения;  $(i, j = 1, \dots, N)$  — количество веб-страниц в графе переходов.

Для сбора информации о переходах и времени нахождения на страницах для большого числа пользователей строится граф переходов пользователей (рис. 1, 2). Каждое ребро графа представляет URL из данных о поведении пользователя и связанные с ним метаданные. Каждое направленное ребро представляет переход между двумя вершинами, отражающих количество переходов в качестве веса страницы. Другими словами, граф переходов пользователей является взвешенным графом с ребрами, содержащими метаданные и вершинами, содержащими вес веб-страницы.

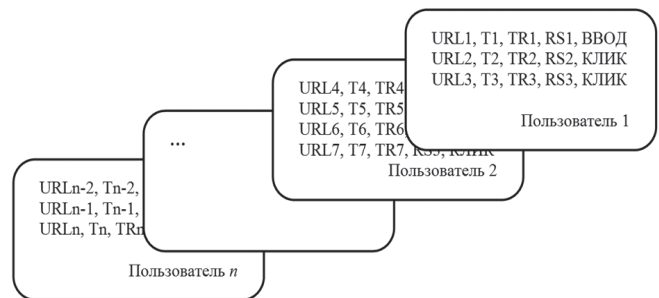


Рис. 1. Информация о поведении пользователя в расширенном BrowseRank

Механизм извлечения данных о переходах пользователя осуществляется следующим образом:

Новая сессия инициируется в случае 30-минутной и более паузы с момента предыдущей активности, либо в случае ввода названия сайта в адресную строку.

В рамках каждой сессии создаются пары url из соседних записей. Пара url означает, что переход был осуществлен при помощи ссылки. На основе этих данных вычисляется количество переходов рамках одного веб-сайта (глубина просмотра TR) и показатель возврата в результаты поиска RS.

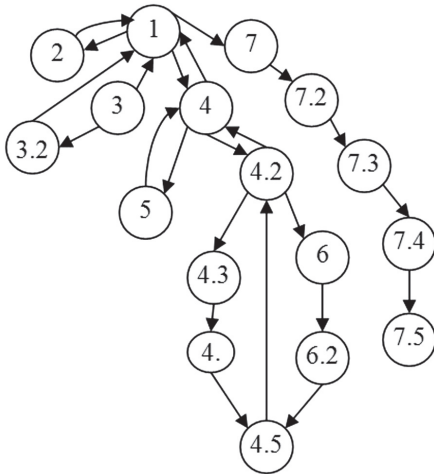


Рис. 2. Граф переходов в расширенном BrowseRank

В каждой сессии, сегментированной по типу перехода, первый url введен непосредственно пользователем. Такие url считаются «доверительными» и называются «зеленым» трафиком. Обработывая данные о поведении пользователей, переходы на указанные url считаются следствием случайного распределения. Нормализация на частоту посещения этих документов дает начальные вероятности посещения соответствующих страниц.

Для каждой пары url продолжительность сессии первого url вычисляется простой разностью дат. Если url был последним в сессии, возможны два варианта. Для сессий, сегментированных по времени, продолжительность просмотра последнего url рассчитывается на основании данных о просмотрах других страниц. Для сессий, сегментированных по типу, время просмотра последнего url рассчитывается исходя из времени начала следующей сессии.

#### 4. Вычисление вероятности переходов для вложенной Марковской цепи

Вероятность переходов во вложенной Марковской цепи описывает «чистые», без погрешностей, переходы пользователя в графе. Их оценка может быть основана на наблюдаемых у пользователя переходах между веб-страницами, на «зеленом» трафике (когда url введен непосредственно пользователем) и на директ-трафике (трафик с закладок). Директ-трафик указывает о том, что пользователи переходят из закладок по памяти, что свидетельствует о качестве сайта. Для объединения этих типов данных при вычислении вероятности переходов используется следующий метод. Вначале берется граф переходов  $G = \langle V, W, T, TR, RS, \delta \rangle$ . Затем к нему добавляется псевдо-вершина  $(N + 1)$  и два вида ребер:

- ребра от последней страницы каждой сессии к псевдо-вершине, связанные с количеством переходов на нее в качестве ее веса, для последнего типа данных (трафик с закладок) вес усиливается введением качественного коэффициента  $\gamma$ ;

- ребра от псевдо вершины к первой странице каждой сессии, связанные с вероятностью случайного распределения. Новый граф обозначается как

$$\bar{G} = \langle \bar{V}, \bar{W}, \bar{T}, \bar{TR}, \bar{RS}, \bar{\delta} \rangle,$$

где  $|\bar{V}| = (N + 1), \bar{\delta} = \langle \bar{\delta}_1, \dots, \bar{\delta}_N, 0 \rangle$ .

Модель рассматривается как случайные переходы по графу  $\bar{G}$ .

Для учета дополнительных данных о поведении пользователя при расчете вероятности переходов в графе серфинга пользователя введем качественный коэффициент  $\gamma$ :

$$a_{ij} = \begin{cases} a \frac{\gamma w_{ij}}{\sum_{k=1}^{N+1} \gamma w_{ik}} + (1-a)\delta_i, & i \in V, j \in \bar{V} \\ \delta_j, & i = N+1, j \in V \end{cases}$$

Другими словами, когда пользователь переходит по ссылкам графа, он переходит по ребрам с вероятностью  $a$  или начинает с новой страницы с вероятностью  $(1-a)$ .

Для расчета качественного коэффициента для категорий сайтов 1-5 применяется формула:

$$\gamma = 1 - 0.5 \times \left( \frac{1}{T + TR} + RS \right), \tag{2}$$

а для категории 6:

$$\gamma = 0.5 \times \left( \frac{1}{T + TR} - RS + 1 \right), \tag{3}$$

где  $T$  – общее время пребывания пользователя на сайте в течение одной сессии ( $T > 0$ );  $TR$  – глубина просмотра выражает количество страниц сайта, которые посетил пользователь в течение одной сессии ( $TR \geq 1$ );  $RS$  – возврат в результаты поиска, принимает значения 0, если возврат не произошел, 1, если возврат имел место быть.

В табл. 2 приведены значения качественного коэффициента  $\gamma$  для примера, изображенного на рис. 2.

Таблица 2

Примеры данных о поведении пользователя с учетом дополнительных параметров

url	Время, мин.	Улубина просмотра	Возврат в результаты поиска	Коэффициент $\gamma(2)$	Коэффициент $\gamma(3)$
site1	2,5	1	0	0,857142	0,214286
site2	0,1	1	1	0,499998	0,250002
site3	0,33	2	1	0,285407	0,571888
site4	8,1	5	1	0,462121	0,306818
site5	0,3	1	1	0,115384	0,826923
site6	1,4	2	1	0,352941	0,470588
site7	20,2	5	0	0,980158	0,029762

На рис. 3-5 приведены диаграммы, которые отображают значение качественного коэффициента в зависимости от различных значений показателей времени, глубины просмотра и возврата в результаты поиска.

Как видно из рисунков, введение дополнительных сведений о поведении пользователя позволяет точнее оценить релевантность web-ресурса.

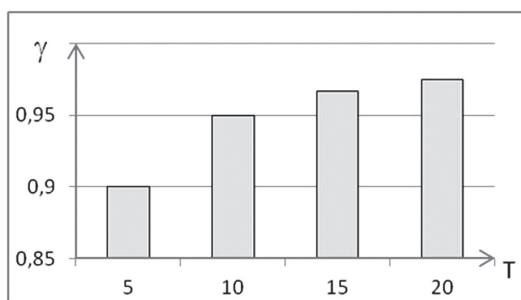


Рис. 3. Значение коэффициента  $\gamma$  в зависимости от значения времени пребывания пользователя на сайте для категорий 1-5

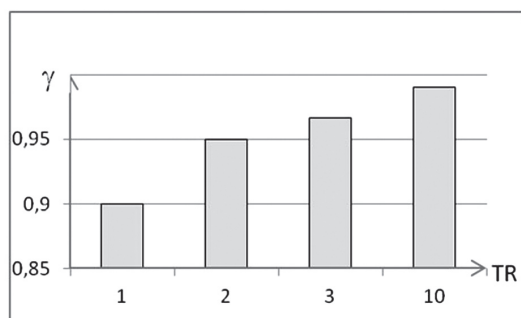


Рис. 4. Значение коэффициента в зависимости от глубины просмотра для категорий 1-5

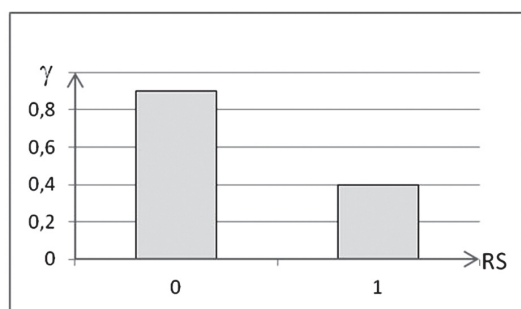


Рис. 5. Значение коэффициента в зависимости от возврата в результаты поиска для категорий 1-5

Таким образом, для реализации модификации метода BrowseRank необходимо следующее.

Входными данными являются время посещения страницы, тип перехода и информация о поведении пользователя, выходные данные — значение важности страницы  $\pi$ .

1. Создать граф переходов пользователя на основе данных о его поведении.
2. Вычислить качественный коэффициент  $\gamma$  по формулам (2) или (3).
3. Вычислить  $q_{ij}$  для всех страниц.
4. Вычислить матрицы вероятностей переходов для вложенной Марковской цепи и получить стационарное распределение вероятностей.
5. Вычислить стационарного распределения вероятностей -процесса по соотношению (1).

### Выводы

В ходе проведенных исследований обнаружено, что наиболее популярные методы вычисления важности страницы имеют недостатки, связанные

с информацией о посещении веб-ресурса, что влечет к недостаточно точному вычислению ее значимости.

Проведен анализ на влияние дополнительных параметров о поведении пользователя (глубина просмотра и показатель возврата в результаты поиска), который показал, что введение качественного коэффициента в метод BrowseRank, позволяет более точно определять важность веб-страницы, эффективно бороться с поисковым спамом и выдавать пользователям поисковых систем более релевантные результаты. Приведены результаты экспериментального моделирования, подтверждающие эффективность предложенного подхода.

**Список литературы:** 1. *B. Amento, L. Terveen, and W. Hill.* Does authority mean quality? Predicting expert quality ratings of web documents. In SIGIR '00. ACM, 2000. 2. *R. Baeza-Yates and B. Ribeiro-Neto.* Modern Information Retrieval. Addison Wesley, May 1999. 3. *M. Bianchini, M. Gori, and F. Scarselli.* Inside pagerank. ACM Trans. Interet Technol., 5(1):92–128, 2005. 4. *P. Boldi, M. Santini, and S. Vigna.* Pagerank as a function of the damping factor. In WWW '05. ACM, 2005. 5. *S. Brin and L. Page.* The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1–7):107–117, 1998. 6. *G. H. Golub and C. F. V. Loan.* Matrix computations (3rd ed.). Johns Hopkins University Press, Baltimore, MD, USA, 1996. 7. *Z. Gyongyi and H. Garcia-Molina.* Web spam taxonomy. In AIRWeb '05. 2005. 8. *Z. Gyongyi, H. Garcia-Molina, and J. Pedersen.* Combating web spam with trustrank. In VLDB '04, pages 576–587. VLDB Endowment, 2004. 9. *T. Haveliwala.* Ecient computation of pageRank. Technical Report, Stanford University, 1999. 10. *T. Haveliwala and S. Kamvar.* The second eigenvalue of the google matrix. Technical Report, Stanford University, 2003. 11. *Yuting Liu, Bin Gao, Tie-Yan Liu.* BrowseRank: Letting Web Users Vote for Page Importance. SIGIR 2008 (Best Student Paper Award). 12. *Sergey Brin, Lawrence Page.* The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Science Department, Stanford University, Stanford, 1996. 13. *Z. Gyungyi, H. Garcia-Molina, J. Pedersen.* Combating Web Spam with TrustRank. 14. *Matthew Richardson, Amit Prakash, Eric Brill.* Beyond PageRank: Machine Learning for Static Ranking. — 2006.

Поступила до редколлегии 25.01.2013

УДК 004.55

**Розширений browserank на основі обліку поведінкових факторів користувачів / Н.Г. Аксак, С.А. Коргут, Н.Г. Стрельцова // Біоніка інтелекту: наук.-техн. журнал. — 2013. — № 1 (80). — С. 99-103.**

У роботі пропонується удосконалений метод BrowseRank, що дозволяє більш точно визначати важливість веб-сторінки за рахунок урахування додаткової інформації про поведінку користувачів.

Л. 5. Табл. 2. Бібліогр.: 14 найм.

UDK 004.55

**Advanced browserank: algorithm accounting behavioral factors of users / N.G. Axak, S.A. Korgut, N.G. Streltsova // Bionics of Intelligense: Sci. Mag. — 2013. — № 1 (80). — P. 99-103.**

The paper proposes an improved method BrowseRank, allowing more accurately determine the importance of web pages by taking into account additional information on user behavior.

Fig. 5. Tab.2. Ref.: 14 items.

УДК 004.9

З.В. Дударь<sup>1</sup>, С.В. Егоров<sup>2</sup><sup>1</sup> ХНУРЭ, г. Харьков, Украина,<sup>2</sup> ХНУРЭ, г. Харьков, Украина, stas.iegorov@gmail.com

## СЕМАНТИЧЕСКОЕ АННОТИРОВАНИЕ В ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМАХ

Проведен анализ структуры информационно-поисковых систем, в основу работы которых положена концепция инвертированного индекса. Рассмотрены наиболее эффективные методы аннотирования текста. Предложен метод поэтапного информационного поиска, позволяющий расширить возможности пользователя и сократить время, затрачиваемое им на поиск необходимой информации.

АННОТАЦИЯ, ЗАПРОС, ИНФОРМАЦИОННО-ПОИСКОВЫЕ СИСТЕМЫ, СЕМАНТИЧЕСКОЕ СЖАТИЕ

### Введение

Стремительное развитие Интернет влечет за собой развитие различных поисковых систем, сервисов и служб.

В рамках информационно-поисковых систем постоянно появляются новые сервисы, расширяется круг возможностей пользователя.

Анализ работы информационно-поисковых систем, принцип действия которых основывается на построении инвертированного индекса, позволяет расширить круг возможностей пользователя посредством предварительного ознакомления его с кратким содержанием документов, предоставляемых системой в ответ на запрос пользователя.

Необходимая реорганизация предполагает изменения в работе поискового сервера и одной из программ модуля индексирования.

Предлагаемый в работе метод поэтапного поиска информации может быть востребован в информационно-поисковых системах и позволит сократить время, затрачиваемое пользователем на поиск интересующей информации.

### 1. Анализ проблемы

Современные информационно-поисковые системы используют метод, в котором по запросу пользователя, состоящего из набора ключевых слов, выдается страница результатов поиска (Search Engine Result Page - SERP), содержащая ссылки на документы, находящиеся в Сети. Список составлен в порядке релевантности документов поисковому запросу. Расположение документов на странице соответствует их индексу цитирования: чем выше индекс цитирования, тем выше вероятность того, что пользователь увидит этот документ на первых страницах поисковой выдачи [1]. Далее пользователь самостоятельно осуществляет поиск нужной ему информации, просматривая каждый документ в отдельности. Пользователь вынужден тратить время на просмотр документа целиком и только по результатам просмотра судить — содержит ли документ нужную информацию или нет. Если же пользователю предлагать документ в сжатом виде, т.е. аннотацию, по которой он будет судить

о присутствии искомой информации в документе, и только при наличии таковой просматривать при необходимости весь документ целиком, то временные затраты существенно сократятся.

### 2. Постановка задачи исследования

Целью исследования является расширение возможностей пользователя в области информационного поиска, а именно: по запросу пользователя предоставить ему возможность получения аннотации, которая полностью отражает смысл исходного текста, после прочтения которой пользователь может судить, содержит ли документ необходимую информацию. В случае позитивной оценки предоставить пользователю возможность, в зависимости от типа документа, либо загрузить полную его версию на компьютер, либо перейти на интересующий ресурс. Таким образом, предлагается использовать семантическое аннотирование текста для предоставления в распоряжение пользователя аннотации и ссылок на ресурсы сети, а в случае необходимости — предоставить полный текст документа. Таким образом, можно существенно снизить время, затрачиваемое пользователем на поиск необходимой информации.

### 3. Реализация

Следует отметить два аспекта: первый случай — когда пользователем корректно составлен поисковый запрос, то, получая аннотацию, он может самостоятельно принимать решение о целесообразности дальнейшего ознакомления с полным текстом исходного документа; второй случай — когда пользователь допустил неточности в формулировке поискового запроса, тогда краткие аннотации позволят гораздо быстрее обнаружить неточности в запросе и скорректировать его.

Рассмотрим последовательно техническую реализацию.

Принцип работы современных информационно-поисковых систем основан на концепции инвертированного индекса, который ставит в соответствие терминам те части документа, в которых они встречаются. Процесс создания инвертированного

индекса включает ряд этапов, важнейшими из которых являются: создание списка нормализованных лексем для каждого документа; сортировка списка, в результате которой термины располагаются в алфавитном порядке; группировка многократно повторяющихся терминов. В результате индекс представляет собой словарь, который хранит термины, расположенные в алфавитном порядке; документную частоту, равную длине списка словопозиций (принадлежность термина документу и координата термина в данном документе), а также номера документов, в которых встречается данный термин. Считается, что такая структура инвертированного индекса является достаточно эффективной для поиска текстовой информации по произвольному запросу. Однако организация инвертированного индекса по аннотациям документов, которая предлагается в данной работе, позволит существенно сократить объем обрабатываемой информации и повысить скорость работы информационно-поисковой системы.

Как известно, основными компонентами информационно-поисковых систем являются: модуль индексирования, база данных и поисковый сервер (рис. 1).

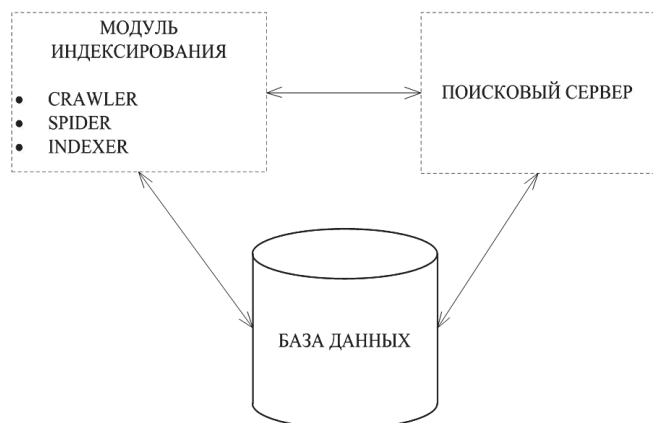


Рис. 1. Структура информационно-поисковой системы

Модуль индексирования, в свою очередь, состоит из трех вспомогательных программ:

– Spider – программа, которая осуществляет скачивание Web-страниц. Скачивается HTML-код каждой страницы;

– Crawler – программа, выделяющая ссылки, присутствующие на странице, и определяющая направление дальнейшего поиска, т.е. осуществляющая поиск новых документов, которых еще нет в системе;

– Indexer – программа, анализирующая скачанные роботами Web-страницы. Прежде всего, эта программа очищает текст индексируемой страницы от всех нетекстовых элементов, таких как: графика, теги разметки HTML. Затем осуществляется просмотр страниц с целью выборки слов,

и удаления всей информации, которая словами не считается (пробелы, знаки препинания и др.). Нужно сказать, что у каждой информационно-поисковой системы есть свое определение слова в тексте. Каждая поисковая машина использует свой алгоритм лингвистической обработки слов, то есть, приведения слов к их начальным грамматическим формам или основам. Этот алгоритм называют машинной морфологией и используют для экономии места в инвертированном индексе. На следующем этапе осуществляется формирование инвертированного индекса, в котором собраны основы всех слов, а также информация об их местоположении.

Согласно предлагаемому методу, необходимо к функциям Indexer'a, таким как: анализ текста, заголовка, специальных HTML-тегов, добавить функцию аннотирования документа (рис. 2).

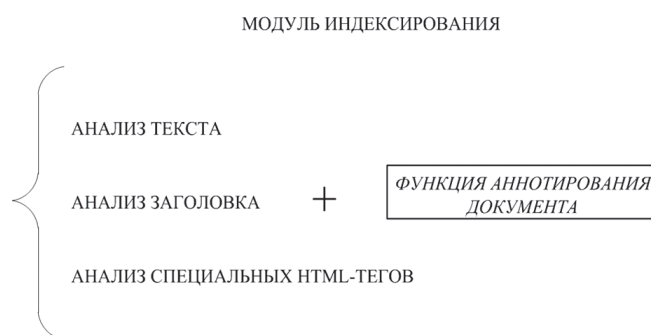


Рис. 2. Функции обновленного модуля индексирования

Алгоритм работы поискового сервера претерпевает незначительные изменения, а именно: пользовательский запрос подвергается морфологическому анализу, который передается в качестве входных параметров модулю ранжирования. Далее генерируется сниппет, в котором предлагается наряду с обычным заголовком двумя-тремя предложениями, содержащими ключевые слова, и ссылкой на сам документ дать пользователю возможность при наведении курсора на ссылку, кроме просмотра сохраненной в кэше копии документа или похожих документов, ознакомиться и с аннотацией.

Таким образом, пользователь получает возможность уже на начальном этапе поиска иметь представление о документах, содержащих искомую информацию, и в случае необходимости просмотреть тексты документов.

Такая техническая реализация позволит также существенно сократить время, затрачиваемое системой на поиск нужной информации.

Важным является тот факт, что предоставляемая пользователю информация должна полностью отражать смысл исходного текста.

На данный момент существует несколько различных подходов к составлению аннотаций: семантическое сжатие, сжатие текста путем замещения терминов их ближайшими аналогами, а также

так называемое «умное кодирование», в основе которого лежит совмещение различных подходов к кодированию разнородной информации. Например, использование различных методик для кодирования текста, чисел, IP и URL-адресов в рамках одного документа.

Поскольку аннотирование само по себе является достаточно сложной задачей, то, как правило, собственно аннотированию предшествуют подготовительные этапы. Одним из них может являться упрощение исходного текста, которое заключается в формировании из синтаксически сложных предложений исходного текста, содержащих различные грамматические обороты, «легкодоступных предложений» (Easy Access Sentences - EAS). Такие предложения содержат только один глагол и имеют максимально простую синтаксическую структуру.

Создание таких «легкодоступных предложений» необходимо с целью упрощения анализа текста системами машинного перевода, извлечения информации, а также – системами аннотирования текстов.

Легкодоступные предложения – это попытка объединения насыщенных информацией естественных языковых данных с приложениями, разработанными для эффективного использования структурированных, хорошо организованных данных, которые, однако, трудно получить без существенного человеческого вмешательства [2].

Самым сложным при формировании EAS является точная передача смысла исходного текста. EAS представляют собой предложения, сформированные путем декомпозиции исходных предложений на более простые части. Поэтому формирование EAS включает в себя такие этапы, как: выявление имен, установление семантической зависимости между именами и соответствующими частями предложения, выделение этих частей из текста, нахождение глаголов и приведение их в соответствующую временную форму, формирование простого утвердительного предложения с использованием имени, глагола и оставшейся части предложения.

Теоретической основой сжатия является такое свойство текстовых сообщений, как повторы частей информации в текстах. Это свойство широко используется в информационных системах при создании поискового образа документа. Частота упоминаний значимых слов говорит об их важности для содержания текста. Задача заключается в отделении значимых слов от простых. Семантическая значимость фрагмента учитывает синтаксические индикаторы, лексические повторы, а также разного рода вспомогательные списки, которые дифференцируют лексику текста. Обычно семантический компонент работает «снизу» с интерпретации синтаксического представления. Система смысловой компрессии интерпретирует

связи между словами и целыми предложениями таким образом, чтобы стало меньше узлов и связей в структуре, но их содержательный вес повысился. Для этой цели разработана словарная классификация лексем по степени информативности. Таким образом, в сложный узел попадут как имя объекта со всеми его атрибутами, так и имя действия с обозначениями модальности, стадии и времени.

На следующем этапе «снизу» осуществляется разбиение текста на другие фразы или простые предложения и построение других объединений, которые являются продолжением или дополнением предиката с большим весом. Таким образом, в тексте намечаются основы ситуаций, а дальнейшее их оформление требует выхода за пределы текстового материала [3].

Поскольку речь идет об аннотировании текста, то особый интерес представляют методы необратимого сжатия, то есть сжатия информации с потерями. Известно, что алгоритмы сжатия информации с потерями делятся на семантически независимые и семантически зависимые. Алгоритмы первого типа в свою очередь представлены адаптивными и статистическими алгоритмами. Адаптивные алгоритмы осуществляют сжатие текста путем его однократного сканирования. Кодирование в этом случае происходит посредством нахождения повторяющихся участков текста и замене их указателями, адресованными к фрагменту текста, где такой участок уже встречался. Следует отметить, что адаптивные алгоритмы сравнительно медленны. Время кодирования у них непосредственно зависит от длины исходного текста, а достижимое сжатие определяется машинной реализацией алгоритма.

Статистические алгоритмы используют словари, представленные либо кодовой таблицей символов алфавита, либо словарем фрагментов переменной длины. В зависимости от избранного вида словаря различают и способы реализации алгоритмов: кодирование фрагментов фиксированной или переменной длины.

Информационно-поисковые системы используют, как правило, способ кодирования фрагментов кодами фиксированной длины, в которых используются двухбайтовые коды. Однако такое кодирование дает небольшой коэффициент сжатия.

Наиболее эффективным методом кодирования фрагментов кодами переменной длины является метод Хаффмана, который для идентификации использует поиск по двоичному дереву [4]. Следует заметить, что коды переменной длины используют для словарей больших размеров, содержащих как буквы алфавита, так и фрагменты текста.

Особый интерес для целей информационного поиска представляют семантически зависимые методы, опирающиеся на грамматику естественного языка. При формировании аннотаций используют

принципы автоматического обобщения текста [5]. Обобщение может представлять уменьшающее преобразование исходного текста путем сжатия содержания с помощью выделения и/или обобщения той информации, которая является важной [6]. Наиболее эффективно использование принципа автоматического обобщения текста для обработки больших объемов документов по схожей тематике, а также в области обработки естественного языка (Data Mining и понимание текста).

Перспективным с точки зрения семантического сжатия текста представляется новый подход, основанный на «принципе количества кода». В основу принципа положена гипотеза о том, что большее количество информации будет закодировано большим количеством кода.

Основной проблемой автоматического обобщения текста является его количественная оценка, которая предполагает сопоставление созданных автоматически обобщений с моделями, разработанными человеком. Одними из последних разработок в области автоматических методологий оценок обобщений является метрика ROUGE и метод Пирамиды.

Обработанный таким образом исходный текст представляет собой набор синтаксически простых предложений, легко поддающихся анализу системами аннотирования текстов.

### Выводы

Проведенное исследование основных принципов работы информационно-поисковых систем позволило внести коррективы в процесс формирования страницы результатов поиска в ответ на запрос пользователя.

В работе предложен метод поэтапного информационного поиска, который поможет расширить возможности пользователя при нахождении нужной информации и существенно сократить время, затрачиваемое на поиск.

Метод предполагает, что при работе пользователя в информационно-поисковой системе в ответ на его запрос необходимо предоставлять аннотацию, полностью отражающую смысл исходного текста, и ссылки на местоположение исходного документа. По результатам обзора аннотации пользователь может самостоятельно принимать решение о необходимости ознакомления с полным текстом документа в случае обнаружения в аннотации интересующей информации.

Предложенный метод предусматривает внесение корректив в работу модуля индексирования. Так, к обычным функциям анализа текста, заголовка и специальных HTML-тегов следует добавить функцию аннотирования документа.

Проведенный в работе анализ методов аннотирования показывает, что для создания аннотаций в методе поэтапного информационного поиска

наиболее эффективными являются семантически-зависимые методы сжатия. Особый интерес представляют методы, основанные на принципах автоматического обобщения текста и использующие новый подход «принципа количества кода».

Предложенный в работе метод поэтапного поиска предназначен для использования в информационно-поисковых системах и позволяет качественно улучшить условия работы пользователя в системе, существенно повысить скорость обработки запроса и предоставить новые возможности в области информационного поиска.

**Список литературы.** 1. Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце. Введение в информационный поиск : Пер. с англ. — М. : ООО «И.Д. Вильямс», 2011. — 528 с. 2. Klebanov B. B., Knight K., Marcu D. Text Simplification for Information-Seeking Applications. — Springer Verlag, 2004. — 13 p. 3. Леонтьева Н.Н. Автоматическое понимание текстов: системы, модели, ресурсы // учеб. пособие для студ. лингв. фак. вузов. — М.: Издательский центр «Академия», 2006. — 304 с. 4. Alonso i Alemany, L., Fuentes Fort, M.: Integrating cohesion and coherence for automatic summarization. In: EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics. (2003) 1–8. 5. Lloret, E., Ferrández, O., Muñoz, R., Palomar, M.: A Text Summarization Approach Under the Influence of Textual Entailment. In: Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 2008) 12-16 June, Barcelona, Spain. (2008) 22–31. 6. Дударь З.В., Егоров С.В. Исследование и оптимизация методов сжатия текстовой информации // Вестник ХНТУ. — Херсон. — 2012. — № 1(44). — 65-68 с.

Поступила в редколлегию 04.02.2013

УДК 004.9

**Семантичне анотування в інформаційно-пошукових системах** / З.В. Дудар, С.В. Єгоров // Біоніка інтелекту: наук.-техн. журнал. — 2013. — № 1 (80). — С. 104-107.

У роботі запропоновано метод поетапного інформаційного пошуку, що допоможе розширити можливості користувача при знаходженні потрібної інформації та істотно скоротити час, що витрачається на пошук.

Метод передбачає, що користувач у відповідь на свій запит отримує анотацію та посилання на місцезнаходження вихідного документу. За результатами огляду анотації користувач може самостійно приймати рішення щодо необхідності ознайомлення із повним текстом документу.

Л. 2. Бібліогр.: 6 найм.

UDC 004.9

**Semantic annotation at information storage and retrieval system** / Z. Dudar, S. Iegorov // Bionics of Intelligense: Sci. Mag. — 2013. — № 1 (80). — P. 104-107.

Method of step-by-step information retrieval which will help to enlarge user possibilities while searching for necessary information and dramatically decrease search time was proposed in the work.

Method provides for giving an annotation and reference to location of source text per user's request. By results of brief review of the annotation user can make decision of necessity of full original text viewing.

Fig.2. Ref.: 6 items.

УДК 004.89

Н.М. Кораблев<sup>1</sup>, Г.С. Иващенко<sup>2</sup><sup>1</sup> ХНУРЭ, г. Харьков, Украина, korablev.nm@gmail.com,<sup>2</sup> gs2005@rambler.ru

## ПРИМЕНЕНИЕ МОДЕЛИ КЛОНАЛЬНОГО ОТБОРА, ИСПОЛЬЗУЮЩЕЙ ВЫВОД ПО ПРЕЦЕДЕНТАМ, ДЛЯ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

В статье предложен комбинированный метод прогнозирования временных рядов при помощи искусственных иммунных систем. Рассмотрена модель прогнозирования на основе модели клонального отбора и метода вывода по прецедентам (СВР). Оценка эффективности модели выполнена путем сравнительного анализа, представлены результаты экспериментальных исследований, демонстрирующие особенности предлагаемого подхода.

ПРОГНОЗИРОВАНИЕ, ВРЕМЕННОЙ РЯД, ВЫБОРКА, ВЫВОД ПО ПРЕЦЕДЕНТАМ, ИСКУССТВЕННАЯ ИММУННАЯ СИСТЕМА, МОДЕЛЬ КЛОНАЛЬНОГО ОТБОРА, АНТИТЕЛО, АНТИГЕН, АФФИННОСТЬ, КЛОНИРОВАНИЕ, МУТАЦИЯ, СУПРЕССИЯ

### Введение

Задача прогнозирования временных рядов играет важную роль в процессе принятия решений в технических и экономических системах. Прогнозирование будущих значений временного ряда позволяет решить актуальную задачу определения будущего состояния на основе анализа уже имеющихся ретроспективных данных. Точность прогноза зависит от многих параметров, таких как объем собранной статистической информации, ее качество и достоверность, требуемая величина горизонта прогнозирования, учет возможности изменения среды, в которой протекает исследуемый процесс, число одновременно учтенных членов ряда, доступность необходимых вычислительных ресурсов.

Существует широкий спектр методов прогнозирования [1], и выбор среди них наиболее подходящего для заданных условий является одним из определяющих факторов получения достоверного прогноза. В настоящее время актуальным направлением в области разработки подходов прогнозирования является создание комбинированных моделей и методов, что позволяет компенсировать недостатки одних моделей при помощи особенностей других. В [2] предлагается применение группы предикторов, моделирующих разные участки временного ряда, что позволяет достичь более высокой точности прогноза по сравнению с применением по отдельности каждого предиктора из заданного набора.

Перспективным подходом является применение метода вывода на основе прецедентов (case based reasoning – СВР), в котором при рассмотрении новой задачи прогнозирования отыскивается подобный прецедент в предыстории в качестве аналога [3–5]. Метод, предложенный в [3], основывается на представлении всего временного ряда как последовательности выборок.

Прогнозирование при помощи СВР основывается на предположении, что путем определения выборки, максимально соответствующей последним известным значениям временного ряда, возможна

оценка будущих значений ряда. Проблема выбора подходящего прецедента среди имеющихся в хранилище прецедентов и его адаптация к текущим условиям является одной из самых актуальных в СВР-системах [4].

В настоящее время активно развиваются подходы к решению задачи краткосрочного прогнозирования на основе методов искусственного интеллекта, таких как искусственные нейронные сети и искусственные иммунные системы (ИИС) [6–9], которые могут интегрироваться с другими подходами и отличаются быстроедействием и адаптационными возможностями.

Существуют различные модели ИИС, основанные на принципах работы иммунной системы: модель клонального отбора, модель иммунной сети, модель отрицательного отбора и другие. В данной работе для решения поставленной задачи предлагается использование модели клонального отбора [6].

### 1. Постановка задачи

Рассматриваются дискретные временные ряды, значения которых получены в моменты времени  $t_1, t_2, t_3, \dots, t_N$ , которые могут быть неравноотстоящими [5]. Временной ряд  $Z = z(t_1), z(t_2), z(t_3), \dots, z(t_N)$  длины  $N$  обозначим  $Z_1^N = z_1, z_2, z_3, \dots, z_N$ . Набор последовательных значений  $Z_t^L = z_t, z_{t+1}, z_{t+2}, \dots, z_{t+L-1}$ , лежащих внутри временного ряда  $Z_1^N$ , назовем выборкой из этого ряда длины  $L$  с моментом начала отсчета  $t$ ,  $L \in [1, N-1]$ ,  $t \in [1, N-L]$ . Задача прогнозирования временного ряда заключается в том, чтобы по его известному участку оценить будущие значения.

Согласно гипотезе, сформулированной в [3], если мера подобия между выборками  $Z_t^L$  и  $Z_{t-k}^L$  имеет значение, близкое к единице, то мера подобия между выборками длины  $P$ , следующими за ними,  $Z_{t+L}^P$  и  $Z_{t-k+L}^P$ , также близка к единице. Тогда путем определения выборки, максимально соответствующей последним известным значениям временного ряда, возможна оценка его будущих значений.

Цель проводимого исследования – разработка иммунного алгоритма краткосрочного прогнозирования временных рядов на основе модели клонального отбора, использующей метод вывода по прецедентам, и проведение экспериментальных исследований.

## 2. Модель клонального отбора, использующая метод вывода по прецедентам

Вывод, основанный на прецедентах, это метод анализа данных, который делает заключения относительно текущей ситуации по результатам поиска аналогий – прецедентов, хранящихся в базе. Данный процесс включает в себя такие этапы, как отбор множества прецедентов из имеющихся в базе на основе заданного отношения подобия, адаптация выбранных прецедентов в связи с текущими условиями, сравнение полученного и реального значений прогнозируемой величины, и сохранение в базе прецедентов для дальнейшего использования принятого решения и текущей ситуации в качестве нового прецедента, или соответствующее изменение ранее выбранного прецедента [4].

Возможность применения CBR для решения задачи прогнозирования определяется на основании предположений о природе временного ряда, предложенных в [2]: будущая динамика временного ряда зависит от его предыстории; существуют участки временного ряда, на которых действуют одинаковые или близкие зависимости (история повторяется).

Алгоритм клонального отбора оперирует строками данных фиксированной длины и часто используется для решения задач классификации, распознавания и оптимизации [6]. Каждая строка данных алгоритма, называемая антителом (в терминах CBR – прецедент), представляет собой набор параметров, описывающих поставленную задачу (набор известных значений ряда) и принятое решение (прогнозируемые значения ряда).

Для построения модели прогнозирования на основе ИИС необходимо сопоставить биологические объекты и процессы с их аналогами из предметной области.

Антитело включает в себя выборку известных значений ряда  $Ab_i = ab_1, ab_2, ab_3, \dots, ab_L, \dots, ab_{L+f}$ , где  $i$  – индекс в популяции,  $L$  – длина выборки известных значений ряда,  $f$  – величина горизонта прогнозирования, длина выборки прогнозируемых значений ряда. Антиген  $Ag_j = ag_1, ag_2, ag_3, \dots, ag_L$ , где  $j$  – индекс в популяции, представляет собой последовательность значений временного ряда, непосредственно предшествующих прогнозируемым.

Аффинность – скалярная величина, оценивающая меру близости между антителом и антигеном, основной критерий отбора антител – определяется как

$$Aff = \frac{\sum_{k=1}^L (1 + |ab_k - ag_k|)^{-1}}{L} \in (0, 1].$$

Антитело также включает в себя выборку длины  $f$  прогнозируемых им значений. Эта часть не участвует в определении аффинности. Число значений временного ряда, которые в нее входят, соответствует величине горизонта прогнозирования.

Таким образом, в терминах подхода CBR антитело исполняет роль прецедента – содержит описание текущей ситуации (последовательность известных значений ряда) и принятое в данный момент решение (соответствующий прогноз). Задача прогнозирования в таком случае сводится к нахождению антитела с наибольшей аффинностью, т.е. к поиску прецедента, наиболее соответствующего последнему участку значений временного ряда. Модель может быть расширена путем включения в антитело и антиген внешних факторов, представленных в виде других временных рядов.

Формально модель ИИС можно представить следующим образом:

$$AIS\_CS = \langle Ab, n', n_c, Ab^C, Ab^M, Ag, D, \sigma_d, S, \sigma_s, \sigma_{age} \rangle,$$

где  $Ab$  – популяция антител размера  $n$ ;  $n'$  – число клеток, отобранных для клонирования и мутации;  $n_c$  – количество клонов, создаваемых одним антителом;  $Ab^C$  – популяция клонов;  $Ab^M$  – популяция клеток памяти,  $Ag$  – популяция антигенов размера  $m$ ;  $D$  – матрица Ab-Ag аффинностей;  $\sigma_d$  – пороговый коэффициент стимуляции клетки;  $S$  – матрица Ab-Ab аффинностей;  $\sigma_s$  – пороговый коэффициент сжатия;  $\sigma_{age}$  – пороговый возраст антител.

Модель ИИС включает в себя популяции антител и антигенов (отдельно выделяется популяция клеток памяти, подмножество популяции антител), параметры, определяющие супрессию и отбор – матрицы аффинностей между антителами и антигенами и антител между собой, а также пороговые значения, и параметры, регулирующие клонирование.

Каждое антитело обладает «возрастом» – числом запусков алгоритма (попыток получить прогноз) с момента появления антитела в популяции. Если «возраст» выше порогового значения, и антитело не входит в число клеток памяти, оно устраняется из популяции в ходе работы алгоритма. Клетка памяти – антитело, которое хотя бы один раз было отобрано для клонирования, т.е. обладавшее аффинностью выше пороговой.

Алгоритм получения прогноза включает в себя следующие этапы:

1. Создание начальной популяции антител. Для создания антител используется часть известных значений временного ряда. Неиспользуемые значения исполняют роль обучающей и контрольной выборок.

2. Формирование антигена, соответствующего текущей задаче, т.е. создание выборки из последовательности известных значений ряда, предшествующих прогнозируемым.

3. Определение Ab-Ag аффинностей и отбор антител, обладающих аффинностью выше порогового значения.

4. В качестве результата принимается прогноз антитела, имеющего наибольшую аффинность.

5. При наличии реального значения прогнозируемой величины определяется ошибка прогноза.

6. Клонирование антител с наибольшей аффинностью (отобранных на этапе 3), в процессе которого выполняется оператор мутации.

7. Определение Ab-Ab аффинностей, применение оператора супрессии с целью устранения избыточности. Оставшиеся антитела из числа ранее отобранных становятся клетками памяти.

8. Применение оператора старения и коррекция популяции.

После получения реального значения прогнозируемой величины (шаги 5-6), отобранные антитела клонируются, в процессе чего выполняется оператор мутации.

Мутация – случайное изменение элементов выборки известных значений и направленное изменение выборки прогнозируемых значений, входящих в состав антитела. Степень мутации определяется аффинностью данного антитела – наиболее подходящий прецедент подвергается более существенной коррекции.

После применения оператора старения возраст антител увеличивается на единицу. Антитела, обладающие возрастом выше порогового значения и не являющиеся клетками памяти, заменяются антителами с наибольшей аффинностью из популяции клонов.

Шаги 2-8 представляют собой обучение ИИС и повторяются для каждого антигена из обучающей выборки заданное число раз. В качестве критерия останова иммунного алгоритма используется останов по достижении определенного числа поколений или останов по достижении некоторого заданного значения ошибки.

Для оценки прогноза используется значение средней абсолютной ошибки (mean absolute error, MAE). На рис.1 показана зависимость ошибки прогноза антител, имеющих различную аффинность, для необученной (график 1) и обученной (график 2) ИИС. Таким образом, в процессе обучения происходит настройка системы на новые значения временного ряда.

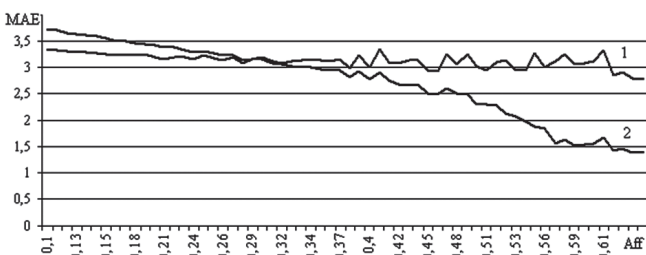


Рис. 1. Зависимость величины ошибки от аффинности

Выбор значений параметров иммунной системы, таких как максимальный размер популяции

антител и параметры отбора, клонирования и супрессии, для заданного ряда осуществляется методами вычислительного эксперимента. Определение длины выборки известных значений ряда, входящих в состав антител, это настройка на текущий ряд, которая выполняется на основе рекомендаций, изложенных в [5]. График на рис.2 демонстрирует влияние величины  $L$  на точность получаемого прогноза для необученной ИИС как построенной на основе модели клонального отбора, так и на основе модели иммунной сети.

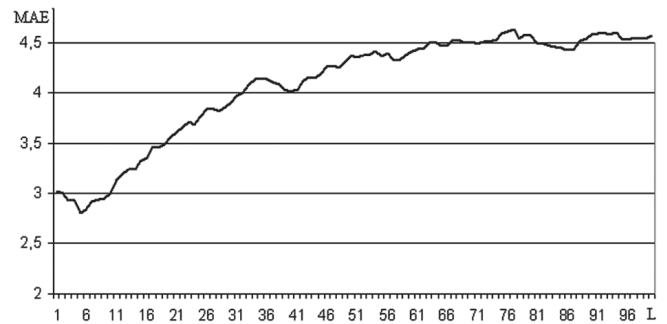


Рис. 2. Зависимость средней абсолютной ошибки прогнозирования от величины антитела

### 3. Результаты сравнительного анализа

В ходе экспериментальных исследований было проведено краткосрочное прогнозирование рядов, используемых в соревновании M3-Competition [10], а также ряда среднесуточных показаний температуры. Средние абсолютные ошибки прогнозирования разными методами приведены в табл. 1. Данные для методов, отличных от рассматриваемого в работе и AINet, получены из результатов соревнования в [10].

Сравнивая результаты предложенного подхода и модели на основе искусственной иммунной сети (AINet), можно отметить, что хотя для ряда температуры результаты близки, использование клонального отбора позволяет лучше прогнозировать ряды с меньшей предысторией, т.е. данный подход менее требователен к количеству значений временного ряда.

Результат прогнозирования ряда среднесуточных показаний температуры (Meteo) подтверждает преимущество использования подходов на основе ИИС – его размеры позволяют создать популяцию, соответствующую обширной базе прецедентов, и провести обучение.

Однако при малом количестве значений временного ряда при получении прогноза вероятно отсутствие в базе антител с высокой аффинностью (нет подходящего прецедента), а величина обучающей выборки не позволит в полной мере настроить (обучить) систему. Вследствие этого на некоторых рядах, представленных в таблице, традиционные методы имеют преимущества перед предложенным подходом.

Таблица 1

Средняя ошибка прогнозирования при использовании различных методов

Метод	N2832 (104)	N2841 (104)	N2856 (92)	N2859 (79)	N2864 (79)	N2868 (79)	N2869 (79)	N2875 (79)	Meteo (21337)
Exp.Smoothing	3148,09	68,70	253,08	132,71	1028,13	775	497,5	350,625	2,92
Holt–Winters	3676,51	52,72	296,47	149,2	1146,32	947,2	580,31	327,585	2,9
Box–Jenkins	2814,27	68,7	260,21	125,09	896,62	1039,33	210,65	1061,77	2,99
RBF	3759,34	37,47	381,63	145,57	1091,78	1227,36	628,15	348	–
ForecastPro	3098,99	68,58	260,26	132,71	1028,13	775	185,79	1161,59	–
SmartFcs	3183,35	67,71	255,99	132,71	1077,06	1831,23	525,2	697,11	–
Automat ANN	1937,68	72,95	300,77	111,76	976,70	985,22	349,1	1072,96	–
AINet	2848,68	151,46	216,50	130,51	1463,12	1372,74	225,76	855,89	2,45
ClonalSelection	2917,45	126,1	216,50	128,3	812,5	1044,67	218,7	502,23	2,44

**Выводы**

В работе впервые предложена модель краткосрочного прогнозирования временных рядов на основе модели клонального отбора, использующей метод вывода по прецедентам. Основные особенности предложенной модели заключаются в следующем:

- использование популяции антител в качестве базы прецедентов;
- применение прямо пропорциональной направленной мутации;
- формирование множества паттернов (популяции клеток памяти) в процессе анализа временного ряда в ходе обучения ИИС;
- целесообразность применения для решения задачи краткосрочного прогнозирования;
- влияние размера антител на точность получаемого прогноза;
- необходимость обучающей выборки значительного объема, но меньшего, чем при использовании модели иммунной сети;
- низкий результат при наличии тренда, желательное постоянство математического ожидания;
- модель может быть расширена с целью решения задачи прогнозирования временного ряда с учетом влияния внешних факторов, представленных в виде других временных рядов.

Представленные в работе результаты подтверждают эффективность использования предложенного подхода для краткосрочного прогнозирования временных рядов, однако использование модели для прогнозирования ряда в условиях недостатка исходной информации требует дополнительных исследований и доработки модели прогнозирования.

**Список литературы 1.** Лукашин, Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов [Текст] : Учебное пособие / Ю. П. Лукашин. – М. : Финансы и статистика, 2003. – 416 с. ISBN 5-279-02740-5. **2.** Батуро, А. П. Финансовые временные ряды: кусочное прогнозирование и проблема обнаружения предвестников существующего изменения закономерности [Текст] / А. П. Батуро, Н. М. Еременко // Банковские технологии. – 2001. – № 12. – С. 70–77. **3.** Singh, S. Pattern Modelling in Time-Series Forecasting [Текст] / S. Singh // Cybernetics and Systems – An International Journal. – 2000. – № 1(31). –

Р. 49–65. **4.** Черный, С. Г. Применение case based reasoning для поддержки принятия решений [Текст] / С. Г. Черный // Вестник ХНТУ. – 2010. – № 2(38). – С. 336–342. **5.** Чучуева, И. А. Модель экстраполяции временных рядов по выборке максимального подобия [Текст] / И. А. Чучуева // Информационные технологии. – 2010. – № 12. – С. 43–47. **6.** Дасгупта, Д. Искусственные иммунные системы и их применение [Текст] : пер. с англ. / под ред. А. А. Романюхи; М.: ФИЗМАТЛИТ, 2006. – 344 с. **7.** Timmis, J. I. An Overview of Artificial Immune Systems [Текст] / J. I. Timmis, T. Knight, L. N. De Castro, E. H. Art // Computation in Cells and Tissues. – Springer. – 2004. – Р. 51–86. **8.** Бидюк, П. И. Алгоритм клонального отбора для прогнозирования нестационарных динамических систем [Текст] / П. И. Бидюк, В. И. Литвиненко, И. В. Баклан, А. А. Фелелов // Искусственный интеллект. – 2004. – № 4. – С. 89–99. **9.** Самигулина, Г. А. Разработка интеллектуальных экспертных систем прогнозирования и управления на основе искусственных иммунных систем [Текст] / Г. А. Самигулина // Проблемы информатики. – Новосибирск, 2010. – № 1. – С. 15–22. **10.** Makridakis, S. The M-3 Competition: Results, Conclusions and Implications [Текст] / S. Makridakis, M. Hibon // International Journal of Forecasting. – 2000. – № 16. – Р. 451–476.

*Поступила в редколлегию 19.02.2013*

УДК 004.89

**Застосування моделі клонального відбору з використанням висновку за прецедентами для прогнозування часових рядів / М. М. Корабльов, Г. С. Івашенко // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 108–111.**

У статті розглядається комбінований метод короткострокового прогнозування часових рядів за допомогою штучних імунних систем. Запропоновано підхід на основі моделі клонального відбору та методу виведення з прецедентів. Був проведений порівняльний аналіз ефективності застосування запропонованої моделі.

Табл. 1. Іл. 2. Бібліогр.: 10 найм.

UDK 004.89

**Application of the clonal selection model using the case based reasoning for time series forecasting. / N. M. Korablyov, G. S. Ivaschenko // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 108–111.**

This paper proposes the combined method of short-term time series forecasting using artificial immune systems. A model of the prediction based on the model of clonal selection and the case based reasoning method. Was performed a comparative analysis of the effectiveness of the proposed model, the approach based on the immune network model and the traditional methods of time series prediction.

Tab. 1. Fig. 2. Ref.: 10 items.

УДК 004.67



Е.В. Мантула

ХНУРЭ, г. Харьков, Украина, elenamantula@gmail.com

## ПРОГНОЗИРУЮЩАЯ НЕЙРОННАЯ СЕТЬ С ПЕРЕМЕННОЙ СТРУКТУРОЙ ДЛЯ КОНТРОЛЯ ПОКАЗАТЕЛЕЙ ЗАГРЯЗНЕНИЯ ОКРУЖАЮЩЕЙ СРЕДЫ

В статье проведен анализ возможности использования в задачах экологического мониторинга для прогнозирования нестационарных временных рядов полиномиальных нейронных сетей, характеризующихся высокой скоростью обучения, и МГУА-сетей, которые имеют переменную структуру с возможностью изменения во время обучения. Предложена нейронная сеть, которая объединяет преимущества многослойного персептрона и МГУА-сети для обучения на основе малой выборки и численного упрощения процесса обучения сети.

ПРОГНОЗИРОВАНИЕ, НЕЙРОННАЯ СЕТЬ, МГУА – НЕЙРОННАЯ СЕТЬ, ПОЛИНОМИАЛЬНАЯ НЕЙРОННАЯ СЕТЬ, ПЕРСЕПТРОН РОЗЕНБЛАТТА

### Введение

Необходимость прогнозирования временных рядов различной природы часто возникает при анализе промышленных, сельскохозяйственных, финансово-экономических, медико-биологических, экологических систем. При этом необходимо анализировать не только интересующие исследователя ряды показателей, но и множество других факторов (экзогенных переменных), определяющих поведение контролируемого объекта. Сложность реальных систем и повышенные требования к точности прогнозирования порождают необходимость в использовании современных, зачастую достаточно сложных подходов, среди которых в первую очередь можно отметить нейросетевую, основанную на использовании искусственных нейронных сетей, обладающих необходимыми аппроксимирующими и экстраполирующими свойствами [1-4].

Понятно, что без учета специфики конкретной задачи невозможно выбрать или синтезировать архитектуру нейронной сети, наилучшим образом приспособленную для данного случая. Примером такой весьма специфической задачи является проблема контроля и прогнозирования показателей загрязнения окружающей среды, описываемых многомерными хаотическими нелинейными нестационарными рядами [5].

При этом участки стационарности рядов контролируемых показателей настолько коротки, что имеющихся данных просто не хватает, чтобы обучить все синаптические веса традиционных широко используемых нейронных сетей. В связи с этим Н. Ниска с соавторами [5] для прогнозирования показателей загрязнения воздуха предложили использовать эволюционирующие нейронные сети. При этом для выбора архитектуры нейронной сети (и набора ее входов) был использован генетический алгоритм. Данный подход весьма громоздок с вычислительной точки зрения, характеризуется низкой скоростью обучения, а базовая архитектура синтезируемой сети ограничивается

многослойным персептроном, обучаемым на основе обратного распространения ошибок. В данной работе предпринимается попытка синтеза гибридной конструкции на основе многослойного персептрона и метода группового учета аргументов [6, 7].

### 1. Полиномиальная нейронная сеть

Следуя А.Г. Ивахненко [6]: «Метод Группового Учета Аргументов (МГУА) является основой математического обеспечения ... для прямого моделирования сложных систем по небольшому числу экспериментальных данных.

Если представлены короткие выборки данных, то наиболее точные ... прогнозы дают математические модели, синтезированные по Методу Группового Учета Аргументов».

И далее: «МГУА представляет собой объединение регрессионного анализа и способов регуляризации».

Задача построения математических моделей экологического мониторинга сводится в общем случае к восстановлению неизвестных, однако, объективно существующих зависимостей между многомерными рядами контролируемых показателей загрязнения  $y(k) = (y_1(k), \dots, y_i(k), \dots, y_n(k))^T$  и многомерными же рядами экзогенных переменных  $x(k) = (x_1(k), \dots, x_p(k), \dots, x_q(k))^T$ , описывающих текущие метеорологические условия. При этом  $y_i(k)$  — значение  $i$ -го контролируемого показателя в момент времени  $k = 1, 2, \dots$ ;  $x_p(k)$  — значение  $p$ -го фактора метеорологических условий, например, скорость и направление ветра, атмосферное давление и т.п. Таким образом, речь идет о синтезе многомерной и нелинейной модели вида

$$\hat{y}(k) = f(x(k)), \quad (1)$$

связывающей с помощью некоторого априори неизвестного преобразования  $f(\cdot)$  экзогенный многомерный ряд  $x(k)$  с прогнозной оценкой  $\hat{y}(k)$  векторного процесса  $y(k)$ .

Учитывая предысторию прогнозируемых и экзогенных переменных, уравнение (1) для  $i$ -го прогнозируемого показателя можно переписать в виде:

$$\hat{y}_i(k) = f_i(y_i(k-1), \dots, y_i(k-n_{A,i}), x_1(k-1), \dots, x_1(k-n_{B,1}), x_2(k-1), \dots, x_2(k-n_{B,2}), \dots, x_p(k-l), \dots, x_q(k-n_{B,q})) \quad (2)$$

(здесь  $n_{A,i}$ ,  $n_{B,q}$  – порядки максимального запаздывания прогнозируемой и экзогенных переменных соответственно) или, переобозначая аргументы:

$$\hat{y}_i(k) = f_i(z_1(k), \dots, z_{n_{A,i}}(k), z_{n_{A,i}+1}(k), \dots, z_{n_{A,i}+n_{B,1}}(k), \dots, z_{n_{A,i}+n_{B,1}+\dots+n_{B,q}}(k)), \quad (3)$$

общее количество которых составляет

$$n_{A,i} + n_{B,1} + \dots + n_{B,q} = \hat{n}.$$

В качестве базовой модели в МГУА наиболее часто используется степенной полином Колмогорова-Габора [6], в общем случае имеющий вид:

$$\hat{y}_i(k) = w_{i0} + \sum_{j_1=1}^{\hat{n}} w_{ij_1} z_{j_1}(k) + \sum_{j_1=1}^{\hat{n}} \sum_{j_2=j_1+1}^{\hat{n}} w_{ij_1 j_2} z_{j_1}(k) z_{j_2}(k) + \dots + \sum_{j_1=1}^{\hat{n}} \sum_{j_2=j_1+1}^{\hat{n}} \dots \sum_{j_l=j_{l-1}+1}^{\hat{n}} w_{ij_1 j_2 \dots j_l} z_{j_1}(k) z_{j_2}(k) \dots z_{j_l}(k), \quad (4)$$

где  $w_{ij}, \dots$  – параметры полинома, подлежащие оцениванию. Опираясь на теорему Стоуна-Вейерштрасса, можно показать, что при некоторых степенях аргументов  $z_j$  может быть достигнута сколь угодно высокая точность аппроксимации.

Описание (4) лежит в основе так называемой полиномиальной нейронной сети [8-11], архитектура которой приведена на рис. 1.

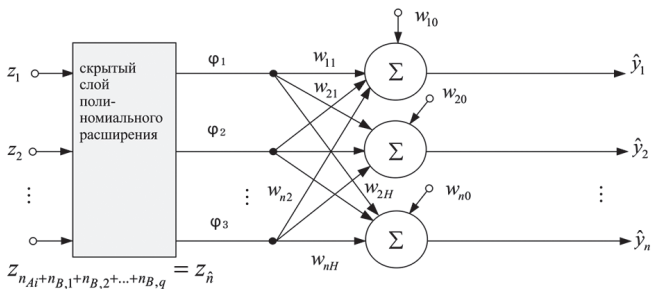


Рис. 1. Полиномиальная нейронная сеть

Первый скрытый слой расширяет входное пространство с помощью полиномиального нелинейного преобразования входного вектора

$$z = (z_1, z_2, \dots, z_{n_{A,i}+n_{B,1}+\dots+n_{B,q}})^T,$$

так что происходит отображение входов в новое пространство повышенной размерности

$$z \in R^{n_{A,i}+n_{B,1}+\dots+n_{B,q}} \rightarrow (\varphi_1(z), \varphi_2(z), \dots, \varphi_{\hat{n}}(z))^T \in R^{\hat{n}}. \quad (5)$$

Выходной же слой образован  $n$  обычными адальнами [4] и реализует отображение расширенного пространства в выходное

$$\varphi(z) \in R^{\hat{n}} \rightarrow \hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)^T \in R^n, \quad (6)$$

так что

$$\hat{y}_i = w_{i0} + \sum_{j=1}^{\hat{n}} w_{ij} \varphi_j(z), \quad i=1, 2, \dots, n. \quad (7)$$

Здесь важно, что каждая из функций  $\varphi_j(z)$  зависит только от входного сигнала  $z$  и используемого полиномиального расширения и не содержит свободных параметров, что выгодно отличает полиномиальные сети от популярных и широко используемых радиально-базисных нейронных сетей.

Полиномиальные сети характеризуются высокой скоростью обучения, благодаря тому, что их входной сигнал линейно зависит от настраиваемых синаптических весов  $w_{ij}$ . Однако они имеют один существенный недостаток: большое количество этих весов, что приводит к так называемому «проклятию размерности» [12]. Несложно показать, что при использовании кубических полиномов и  $\hat{n} = 20$  (что не так уж велико) полиномиальная сеть содержит 1270 настраиваемых синаптических весов [8], что вызывает проблемы со скоростью обучения и требованием достаточно длинной обучающей выборки, получить которую в системах экологического мониторинга весьма сложно.

## 2. МГУА-нейронная сеть на элементарных персептронах Розенблатта

Преодолеть отмеченное выше затруднение можно, используя так называемые МГУА-нейронные сети [3], в основе которых лежат идеи стандартного МГУА А.Г. Ивахненко. На рис. 2 приведен пример такой сети с четырьмя входами  $z_1, z_2, z_3, z_4$  и одним выходом  $\hat{y}_i$ .

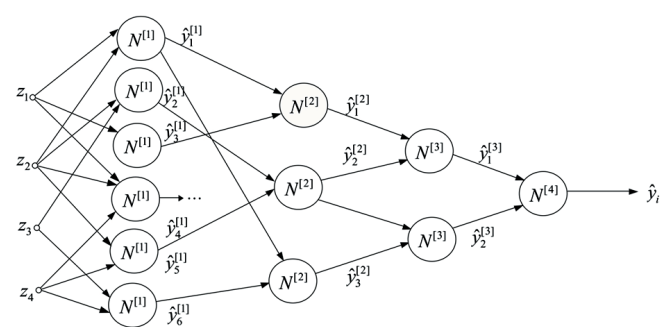


Рис. 2. МГУА-нейронная сеть

В отличие от традиционных нейронных сетей с фиксированной архитектурой типа классического многослойного персептрона МГУА-сеть имеет переменную структуру, которая может изменяться в процессе обучения. В наиболее известной МГУА-нейронной сети [3] в качестве нейронов  $N^{l-1}$  используются нелинейные адальны (N-адальны), которые отличаются от описанных выше обычных адалин тем, что содержат элементарный нелинейный препроцессор, образованный тремя блоками умножения, и вычисляют квадратичную комбинацию двух произвольных входов в виде:

$$\hat{y}_{gh} = f_{gh}(z_g, z_h) = w_{gh0} + w_{gh1}z_g + w_{gh2}z_g^2 + w_{gh3}z_g z_h + w_{gh4}z_h^2 + w_{gh5}z_h^3 = w_{gh}^T z_{gh}, \quad (8)$$

где  $w_{gh} = (w_{gh0}, w_{gh1}, w_{gh2}, w_{gh3}, w_{gh4}, w_{gh5})^T$ ,  
 $z_{gh} = (1, z_g, z_g^2, z_g z_h, z_h^2, z_h^3)^T$ .

При этом, поскольку каждый нейрон содержит два входа, общее количество нейронов первого скрытого слоя определяется числом сочетаний  $c_n^2 = 0,5(\hat{n}(\hat{n}-1))$ .

В случае, когда прогнозируемый сигнал  $y_i(k)$  имеет достаточно сложную внутреннюю структуру, аппроксимирующих возможностей N-адалин может оказаться недостаточно. В этом случае в качестве узлов МГУА-сети могут быть использованы более сложные конструкции [13, 14], которые при этом и обучаются сложно, и требуют больших объемов обучающей выборки.

Преодолеть отмеченное затруднение, на наш взгляд, возможно, объединяя достоинства МГУА-нейронных сетей с универсальными аппроксимирующими возможностями многослойных персептронов [15-17], узлами которых являются элементарные персептроны Розенблатта [1]. На рис. 3 приведена схема такого нейрона с двумя входами  $z_g$  и  $z_h$ .

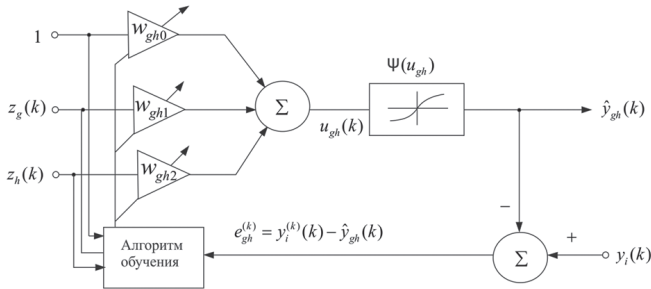


Рис. 3. Элементарный персептрон Розенблатта с двумя входами

Преобразование, реализуемое таким нейроном, может быть записано в виде:

$$\hat{y}_{gh} = f(z_g, z_h) = \psi(w_{gh0} + w_{gh1}z_g + w_{gh2}z_h) = \psi(u_{gh}) = \psi(w_{gh}^T z_{gh}), \quad (9)$$

где  $\psi(u_{gh})$  – нелинейная активационная функция, обеспечивающая нейрону требуемые свойства [15].

В качестве активационных функций обычно используются либо сигмоидальная функция (однополярная)

$$\hat{y}_{gh} = \frac{1}{1 + e^{-\gamma u_{gh}}}, \quad (10)$$

либо биполярная функция гиперболического тангенса

$$\hat{y}_{gh} = \tanh(\gamma u_{gh}) = \frac{1 - e^{-2\gamma u_{gh}}}{1 + e^{-2\gamma u_{gh}}}, \quad (11)$$

где  $\gamma > 0$  – параметр крутизны активационной функции.

Основное отличие между адалиной и элементарным персептроном Розенблатта состоит в алгоритме обучения, поскольку в адалине ошибка обучения

$$e_{gh} = y_i - \hat{y}_{gh} \quad (12)$$

является линейной функцией вектора входов  $z_{gh}$ , в то время как ошибка обучения персептрона

$$e_{gh} = y_i - \hat{y}_{gh} = y_i - \psi(w_{gh}^T z_{gh}) \quad (13)$$

от входов и синаптических весов зависит нелинейно, причем характер этой нелинейности определяется активационной функцией. Данное обстоятельство затрудняет процесс обучения персептрона и требует использования специальных методов обучения, отличных от процедур, основанных на стандартном методе наименьших квадратов.

Процесс обучения будем проводить путем минимизации критерия

$$\begin{aligned} J_i(k) &= \frac{1}{2} e_{gh}^2(k) = \frac{1}{2} (y_i(k) - \hat{y}_{gh}(k))^2 = \\ &= \frac{1}{2} (y_i(k) - \psi(u_{gh}(k)))^2 = \\ &= \frac{1}{2} (y_i(k) - \psi(\sum_{l=0}^5 w_{ghl} z_l(k)))^2 = \\ &= \frac{1}{2} (y_i(k) - \psi(w_{gh}^T z_{gh}))^2 \end{aligned} \quad (14)$$

с помощью рекуррентной процедуры

$$\begin{aligned} w_{ghl}(k+1) &= w_{ghl}(k) - \eta(k) \frac{\partial J_i(k)}{\partial e_{gh}(k)} \frac{\partial e_{gh}(k)}{\partial w_{ghl}} = \\ &= w_{ghl}(k) - \eta(k) e_{gh}(k) \frac{\partial e_{gh}(k)}{\partial w_{ghl}} = \\ &= w_{ghl}(k) - \eta(k) e_{gh}(k) \frac{\partial e_{gh}(k)}{\partial u_{gh}(k)} \frac{\partial u_{gh}(k)}{\partial w_{ghl}} = \\ &= w_{ghl}(k) + \eta(k) e_{gh}(k) \psi'(u_{gh}(k)) z_l(k) = \\ &= w_{ghl}(k) + \eta(k) \delta_{gh}(k) z_l(k), \end{aligned} \quad (15)$$

где

$$\delta_{gh}(k) = e_{gh}(k) \psi'(u_{gh}(k)) = -\frac{\partial J_i(k)}{\partial u_{gh}} \quad (16)$$

– локальная ошибка,  $\eta(k)$  – параметр шага обучения.

В векторной форме алгоритм (15) имеет вид:

$$w_{gh}(k+1) = w_{gh}(k) + \eta(k) \delta_{gh}(k) z_{gh}(k) \quad (17)$$

и известен под названием дельта – правила обучения.

Заметим также, что для активационной функции (10) алгоритм (17) приобретает форму

$$\begin{aligned} w_{gh}(k+1) &= w_{gh}(k) + \eta(k) \gamma e_{gh}(k) \hat{y}_{gh}(k) \\ &\quad (1 - \hat{y}_{gh}(k)) z_{gh}(k), \end{aligned} \quad (18)$$

а для активационной функции (11) –

$$\begin{aligned} w_{gh}(k+1) &= w_{gh}(k) + \eta(k) \gamma e_{gh}(k) \\ &\quad (1 - \hat{y}_{gh}^2(k)) z_{gh}(k). \end{aligned} \quad (19)$$

Качество обучения на основе дельта – правила (12) существенным образом зависит от обоснованного выбора параметра шага  $\eta(k)$ , при этом если имеется достаточно длинная обучающая выборка, неудачный выбор приведет лишь к снижению скорости сходимости. В случае же короткой выборки результаты гораздо более критичны к выбору шага.

В этом случае более целесообразным представляется использование модифицированного алгоритма Чана-Фоллсайда [18], приобретающего в данном случае вид

$$w_{gh}(k+1) = w_{gh}(k) + \eta(k)(\nabla_{w_{gh}} J_i(k) + \beta(k)\Delta w_{gh}(k-1)), \quad (20)$$

где

$$\beta(k) = \begin{cases} \beta_0 \frac{\Delta J_i(k-1)}{\|\Delta w_{gh}(k-1)\|}, & \text{если } \Delta w_{gh}(k-1) \leq \varepsilon, \\ 0 & \text{в противном случае,} \end{cases} \quad (21)$$

$$\eta(k) = \eta(k-1)(1 + a \cos \theta(k)), \quad (22)$$

$$\cos \theta(k) = \frac{\nabla_{w_{gh}} J_i^T(k) \Delta w_{gh}(k-1)}{\|\nabla_{w_{gh}} J_i(k)\| \|\Delta w_{gh}(k-1)\|}, \quad (23)$$

$\varepsilon$  - пороговый параметр, определяющий момент “включения” регуляризующего члена

$$\Delta w_{gh}(k-1);$$

$$\Delta w_{gh}(k-1) = \Delta w_{gh}(k) - \Delta w_{gh}(k-1);$$

$$\Delta J_i(k-1) = J_i(k) - J_i(k-1); 0,1 \leq a \leq 0,5.$$

Одновременно с синаптическими весами, число которых всего три, можно настраивать и четвертый параметр, характеризующий крутизну активационных функций (10), (11) -  $\gamma$ . Для этого можно воспользоваться алгоритмом, введенным в [19] и являющимся обычной градиентной процедурой оптимизации

$$\begin{aligned} \gamma(k+1) &= \gamma(k) - \eta_\gamma(k) \frac{\partial J_i(k)}{\partial \gamma} = \\ &= \gamma(k) - \eta_\gamma(k) e_{gh}(k) \frac{\partial \psi(\gamma(k) u_{gh}(k))}{\partial \gamma}. \end{aligned} \quad (24)$$

При этом для регулирования параметра шага  $\eta_\gamma(k)$  можно воспользоваться соотношением (22).

Таким образом, настройка всех параметров персептрона Розенблатта производится с помощью соотношений (20) – (24).

В целом же МГУА-нейронная сеть обучается следующим образом. Сначала формируется первый скрытый слой из персептронов Розенблатта с двумя входами  $z_g(k)$  и  $z_h(k)$  с четырьмя настраиваемыми параметрами  $w_{gh0}, w_{gh1}, w_{gh2}, \gamma$ . Общее число этих персептронов  $0,5\hat{n}(\hat{n}-1)$ , а обучаются они параллельно и независимо друг от друга с помощью общего обучающего сигнала  $y_i(k)$ . По

исчерпанию обучающей выборки оценивается точность каждого из узлов, после чего в первом слое оставляют только  $\hat{n}$  наилучших. После этого их параметры “замораживаются” и в дальнейшем эти узлы только пропускают через себя входной сигнал на второй скрытый слой. Второй скрытый слой формируют из  $0,5\hat{n}(\hat{n}-1)$  нейронов и обучают точно так же. Входами этого слоя являются выходы  $\hat{n}$  наилучших нейронов первого слоя. По исчерпанию обучающей выборки оценивается точность каждого из персептронов второго слоя, и в слое остаются только нейроны, чья точность выше чем у наилучшего из нейронов первого слоя. Процесс наращивания слоев продолжается до тех пор, пока в последующем слое не останется единственный нейрон, чей выход является и выходом МГУА-сети в целом.

Поскольку любой из нейронов сети обучается независимо от других, многослойная сеть, содержащая произвольное число узлов, может обучаться на основе малой выборки, что характерно в задачах экологического мониторинга.

## Выводы

Рассмотрена задача прогнозирования нестационарных временных рядов, характеризующихся короткой обучающей выборкой, что характерно для задач экологического мониторинга. Рассмотрены возможности применения полиномиальных и МГУА-нейронных сетей и введена нейронная сеть, являющаяся “гибридом” многослойного персептрона и МГУА-сети и обладающая рядом преимуществ перед своими прототипами. Введен численно простой метод обучения этой сети.

**Список литературы:** 1. *Cichocki A.* Neural Networks for Optimization and Signal Processing [Текст] / A. Cichocki, R. Unbehauen // – Stuttgart: Teubner, 1993 – 526 p. 2. *Masters T.* Practical Neural Networks Recipes in C++ [Текст] / T. Masters. – San Diego: Academic Press, Inc., 1993. – 493 p. 3. *Pham D. T.,* Neural Networks for Identification, Prediction and Control [Текст] / D. T. Pham, X. Liu // – London: Springer – Verlag, 1995. – 238 p. 4. *Haykin S.* Neural Networks: A Comprehensive Foundation [Текст] / S. Haykin – Upper Saddle River, N. J.: Prentice-Hall, Inc., 1999. – 842 p. 5. *Niska H.* Evolving the neural network model for forecasting air pollution time series // Engineering Application of Artificial Intelligence [Текст] / H. Niska, T. Hiltunen, A. Karppinen, J. – 2004. – 17. – P. 159–167. 6. *Ивахненко А. Г.* Системы эвристической самоорганизации в технической кибернетике [Текст] / А. Г.Ивахненко – Киев: Техника, 1971. – 372 с. 7. *Ивахненко А. Г.* Помехоустойчивость моделирования [Текст] / А. Г. Ивахненко, В. С. Степашко // Киев: Наук. Думка, 1985. – 216 с. 8. *Billings A.* Extended model set, global data and threshold model identification of severely non-linear systems [Текст] // A. Billings, S. Chen // Int. J. Control. – 1989. – 50. – P. 1897-1923. 9. *Harris C. J.,* Intelligent Control. Aspects of Fuzzy Logic and Neural Nets [Текст] / C. J. Harris, C. G. Moore, M. Brown // Singapore: World Scientific. – 1993. – 380 p. 10. *Brown M.* Neural networks for

modelling and control / Ed. by C. J. Harris "Advances in Intelligent Control" [Текст] / M. Brown, C. J. Harris - London: Taylor and Francis, 1994. - P. 85-112. **11.** Бодянский Е.В. Искусственные нейронные сети: архитектуры, обучение, применения [Текст] / Е. В.Бодянский, О. Г. Руденко // - Харьков. ТЕЛТЕХ, 2004. - 372 с. **12.** Wang H., Advanced Adaptive Control [Текст] / H. Wang, G. P., C. J. Harris, M. Brown - Oxford: Pergamon, 1995. - 262 p. **13.** Bodyanskiy Ye., Hybrid GMDH-neural network of computational intelligence // Proc. Int. Workshop on Inductive Modelling 2009 [Текст] / Ye. Bodyanskiy, O. Vynokurova, I. Pliss // - Krynica, Poland, 2009. - P. 100 - 107. **14.** Bodyanskiy Ye. The neo-fuzzy neural network structure optimization using the GMDH for the solving forecasting and classification problems // Proc. Int. Workshop on Inductive Modelling 2009 [Текст] / Ye. Bodyanskiy, Yu. Zaychenko, E. Pavlikovskaya, M. Samarina, Ye. Viktora. - Krynica, Poland, 2009. - P. 77-89. **15.** Cybenko G. Approximation by superposition of a sigmoidal function // Math Contr. Sign. and Systems [Текст] / G. Cybenko, 1989 -2. P. 303-314. **16.** Hornik K., Multilayer feedforward networks are universal approximators [Текст] / K. Hornik, M. Stinchcombe, H. White // Neural Networks. - 1989. - 2 - P. 359-366. **17.** Hornik K. Approximation capabilities of multilayer feedforward networks [Текст] / K. Hornik // Neural Networks. - 1991. - 4. - P.251-257. **18.** Chan L.W., An adaptive learning algorithm for backpropagation networks [Текст] / L.W. Chan, F. Fallside // Computer Speech and Language. - 1987. - 2. - P. 205-218. **19.** Kruschke J. K., Benefits of gain: Speeded learning and minimal hidden layers in back propagation networks [Текст] / J. K. Kruschke, J.R. Movellan // IEEE Trans. on Syst., Man and Cybern. - 1991. - 21. - P. 67-70.

Поступила в редколлегию 22.02.2013

УДК 004.67

**Прогнозуюча нейронна мережа зі змінною структурою для контролю показників забруднення навколишнього середовища** / О.В. Мантула // Біоніка інтелекту: наук.-техн. журнал. - 2013. - № 1 (80). - С. 112-116.

В статті розглядається задача прогнозування часових рядів, яка є характерною при вирішенні питань екологічного моніторингу. Розглядається застосування поліноміальних мереж, які характеризуються високою швидкістю навчання завдяки тому, що сигнал на виході лінійно залежить від синаптичних ваг. Проте використання цих мереж призводить до проблеми, пов'язаної з експонентним зростанням кількості даних, обумовленим зростанням розмірності вхідного простору та МГУА-нейронних мереж, що дають найточніші прогнози в умовах короткої вибірки. Запропоновано та побудовано на їх основі нейронну мережу, що має ряд переваг і використання якої чисельно спрощує процес навчання мережі.

Л. 3. Бібліогр.: 19 найм.

UDK 004.67

**Predictive neural network with variable structure to control pollution indicators** / O.V. Mantula // Bionics of Intelligense: Sci. Mag. - 2013. - № 1 (80). - P. 112-116.

Article is devoted to the problem of time series forecasting, which is typical problem in environmental monitoring. The application of polynomial networks, which characterized by a high rate of learning due to the fact that the output signal of a linear function depends on synaptic weights, which using leads to a problem with the amount of data exponential increase due to higher dimensional space and the GMDH-neural networks, which give the most accurate forecasts in a conditions of short sample. Proposed neural network, based on them, and has a number of advantages, which using is greatly simplifies network training.

Fig. 3. Ref.: 19 items

УДК 681.3, 004.85, 004.89



К.А. Ручкін

ДонНТУ, м. Донецьк, Україна, c\_ruchkin@mail.ru

## АНАЛІЗ ТА ПРОЕКТУВАННЯ БАГАТОАГЕНТНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ НЕЛІНІЙНОЇ ДИНАМІЧНОЇ СИСТЕМИ У РЕАЛЬНОМУ ЧАСІ

У роботі продовжені дослідження [1], які пов'язані з розробкою багатоагентної системи прогнозування поведінки нелінійної динамічної системи у реальному часі. Складена загальна схема вирішення задачі, описана загальна архітектура багатоагентної системи, розроблена структура бази даних, спроектовані агенти багатоагентної системи. Запропонована багатоагентна система спроектована на основі методу розподіленого штучного інтелекту з використанням даних, одержаних з середовища моделювання Modeler [2] в режимі реального часу. Система складається з сімнадцяти агентів: інтерфейсного агента, інформаційного агента, агента-планувальника, керуючого агента, тактового агента і дванадцяти спеціалізованих агентів-розпізнавачів. Наведено специфікацію багатоагентної системи, її агентів та бази даних у вигляді діаграм UML, а також опис структури, функцій і порядку роботи системи та примітки щодо її програмної реалізації в майбутньому.

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ, БАГАТОАГЕНТНІ СИСТЕМИ, ЗАДАЧА ПРОГНОЗУВАННЯ,  
НЕЛІНІЙНІ ДИНАМІЧНІ СИСТЕМИ

### Вступ

Задача побудови багатоагентної системи (БАС) дослідження та прогнозування поведінки динамічних систем є досить новою та складною задачею. Її суть полягає у наступному. Необхідно дослідити та спрогнозувати поведінку нелінійної автономної багатопараметричної динамічної системи на основі графічної інформації (двовимірних та тривимірних перетинів Пуанкаре), яка отримується у реальному часі. Складність розв'язування цієї задачі полягає в тому що, вона погано формалізується та може бути віднесена до задач штучного інтелекту. Так, наприклад, проведення окремих досліджень двовимірних або тривимірних перетинів Пуанкаре недостатньо для вирішення початкової проблеми. Також для побудови перетинів Пуанкаре нефіксований час, який необхідно для розрахунку перетинів. Малий час – не дає змоги побудувати адекватні, великий час – дозволяє накопичувати помилки. Задача ускладнюється і тим, що динамічна система є багатопараметричною. У залежності від параметрів вона може мати протилежні властивості: бути добре прогнозованою або ні. Для вирішення цих проблем пропонується застосувати багатоагентно-орієнтовану технологію та розробити відповідну багатоагентну систему прогнозування поведінки нелінійної динамічної системи у реальному часі [1].

Як відомо, у сучасних дослідженнях систем штучного інтелекту агентно-орієнтовані технології знаходять все більш широке застосування. Ці технології можуть бути розглянуті як нова парадигма задля аналізу, проектування, розробки та реалізації складних програмних систем інтелектуального напрямку. В агентно-орієнтованих технологіях агенти – це складні комп'ютерні програми (або підпрограми), які можуть діяти автономно (від імені користувача) для вирішення поставлених задач

згідно з обраною стратегією поведінки. Проте все більше задач потребує застосування декількох різноманітних стратегій, також як і декількох агентів. В мультиагентних системах агенти повинні взаємодіяти кооперативно між собою та виробляти більш адекватні кооперативні рішення. Багато-агентні інтелектуальні системи мають більш вагомі переваги перед класичними системами штучного інтелекту та здатні вирішувати більш складні задачі, наприклад, задачі пошуку, розпізнавання та прогнозування. Але їх проектування та моделювання є досить складною задачею та залежить від вибраної методології проектування БАС. Розглянемо сучасний стан напряму багатоагентно-орієнтованого аналізу та проектування і його методи.

### 1. Сучасний стан напряму багатоагентно-орієнтованого аналізу та проектування

Серед останніх досліджень з використанням багатоагентних систем для вирішення різноманітних завдань, пов'язаних з комплексними і розподіленими інтелектуальними обчисленнями, відзначимо біоінформаційний аналіз [3,4], пошук та вилучення інформації [5-9], електронну комерцію та управління [11-13], логістику та імітаційне моделювання динамічних систем та організацій [10, 14], розпізнавання образів [15-19], експертні системи та системи підтримки прийняття рішень [21,22].

Особливий інтерес для цього дослідження представляє собою багатоагентні системи розпізнавання графічних образів та багатоагентні системи реального часу. Розглянемо деякі з них детальніше.

Так А. Аддіс, Г. Арман і Е. Варгю розробили загальну архітектуру багатоагентної системи для пошуку та обробки інформації [5]. Дана архітектура передбачає наявність у системи таких можливостей, як пошук інформації в безлічі різнорідних

джерел (зокрема в мережі Інтернет), динамічна інтеграція нових джерел в процесі пошуку, контроль релевантності результатів, врахування відгуків та уподобань користувача. Отримана архітектура лягла в основу кількох багатоагентних систем, а саме системи керування мультимедіа-ресурсами, системи моніторингу човнів і системи класифікації вмісту Вікіпедії.

Г. Пур провела дослідження зі створення гнучкої, адаптивної, розширюваної і робастної системи для контролю якості в режимі реального часу з метою застосування в сфері електронної комерції [13]. Результатом дослідження стала архітектура і прототип багатоагентної системи, заснованої на принципах автономних комп'ютерних систем (autonomic computing). Дана архітектура передбачає наявність у системи таких можливостей, як моніторинг параметрів якості, оцінка рівня якості відповідно до заданих стандартами, вироблення і виконання планів для регулювання рівня якості.

Д. Рахалем, Ф. Рахалем і М. Чекуном була розроблена SMAST – багатоагентна система для моделювання транспортних систем [14]. Дана БАС призначена для моніторингу, діагностики та підтримки управління трафіком в режимі реального часу. У SMAST кожен суб'єкт дорожнього руху представляється у вигляді незалежного діючого агента з певними задачами і переконаннями. За допомогою додаткових агентів для моніторингу та аналізу переміщення транспорту SMAST дозволяє передбачати дорожні інциденти, моделювати їх протікання і пропонувати можливі шляхи їх вирішення.

Дж. Каселлою, В. Деуфеміей і В. Маскарді була розроблена AgentSketch – багатоагентна онлайн-система для розпізнавання намальованих від руки діаграм в режимі реального часу [15]. Дана БАС призначена для використання в сукупності з планшетними і пір'яними (pen-based) інтерфейсами. AgentSketch використовує багатоагентний підхід для управління багатьма розпізнавачами символів і підвищення ефективності розпізнавання шляхом аналізу контекстної інформації. Передбачається, що отриману систему можна буде застосовувати в різних предметних областях шляхом створення розпізнавачів символів для цих областей. Розроблена БАС була успішно протестована в задачі розпізнавання діаграм UML.

О. Гемпелем, У. Бекером і Дж. Хартманном була розроблена Visent – загальна архітектура багатоагентної системи для паралельного розпізнавання об'єктів [16]. Дана архітектура призначена для створення систем візуального розпізнавання, що діють на основі бази знань. Процес розпізнавання розподіляється між багатьма агентами, кожен з яких намагається розпізнати задану частину потрібованого об'єкта. Тестування отриманого

прототипу багатоагентної системи продемонструвало збільшення швидкості розпізнавання в порівнянні з аналогічним багатопотоковим рішенням за рахунок обміну інформацією між агентами.

Р. Джума, П. Леклерк і Ш. Азар розробили EsqQUIsE – багатоагентну онлайн-систему для захвату та інтерпретації архітектурних креслень, намальованих від руки [17]. Дана система призначена для підтримки ранніх етапів проектування, коли архітектурний проект являє собою намальований від руки ескіз. БАС побудована за багаторівневою схемою з розбивкою комплексної задачі розпізнавання на простіші підзадачі, аж до інтерпретації найпростіших графічних примітивів. Отриманий прототип дозволяє проводити розпізнавання креслень в режимі реального часу, тобто по мірі їх малювання формувати семантичну і тривимірну моделі будівлі, що проектується.

Дж. Ржевським, П. Скобелевим, І. Мінаковим, С. Вольманом був розроблений багатоагентний метод динамічної кластеризації [18]. У даній БАС кожен окремих підлягаючий класифікації об'єкт асоціюється з агентом, а кластери формуються в процесі переговорів і об'єднання цих агентів з метою збільшення оцінки системи (загального показника кластеризації). Головними особливостями розробленої системи є здатність працювати з динамічними даними (нові об'єкти можуть у будь-який момент часу включатися в процес кластеризації) і здатність до еволюції (кластери можуть виникати, зникати і міняти свій склад у процесі досягнення максимальної оцінки системи).

Г. Саламою, А. Фахмі і М. Ельбаром була розроблена багатоагентна модель і прототип системи для розпізнавання осіб [19]. Цю БАС передбачається використовувати для біометричної ідентифікації особистості по мережі. Дана система комбінує кілька методів розпізнавання, розподіляючи завдання ідентифікації між кількома незалежними агентами і виводячи підсумковий результат розпізнавання на основі колективного результату цих агентів. Тестування системи показало як збільшення ефективності, так і зменшення витрат часу на процес розпізнавання.

К. Сас, Г. О'Харе і Р. Рейллі провели дослідження перспективи створення адаптивних віртуальних середовищ (virtual environment) [21]. У процесі дослідження була проаналізована можливість створення інтелектуальної системи підтримки віртуальної навігації та динамічної адаптації віртуального середовища на основі багатоагентної підходу. Був розроблений ранній прототип такої системи, здатної аналізувати і класифікувати траєкторії рухів користувачів у віртуальному просторі.

В. Розколом, А. Левассьєром, О. Чабролом, С. Грузе, Є. Данчіном і П. Понтаротті була розроблена CASSIOPE (Clever Agent System for Synteny

Inheritance and Other Phenomena in Evolution) – багатоагентна біоінформаційна система для пошуку консервативних ланцюжків в геномах в режимі реального часу [22]. Розроблена система дозволяє автоматично ідентифікувати потрібні ланцюжки у великому обсязі динамічних вхідних даних, проводити оцінку значущості отриманих результатів, а також виконувати реверсивний пошук.

Таким чином проведений аналіз показує ефективність багатоагентної технології задля вирішення різноманітних науково-технічних задач. Розглянемо методи побудови багатоагентних систем.

## 2. Дослідження методів агентно-орієнтованого проектування і аналізу

Поряд з дослідженнями, пов'язаними із застосуванням багатоагентних систем та розробкою нових багатоагентних архітектур, варто відзначити роботи по створенню нових методологій [23, 24, 25] і фреймворків [17, 22, 26] для багатоагентного проектування та аналізу.

М. Вулдріджем, Н. Дженнінгсоном і Д. Кінні була розроблена GAIA – універсальна методологія для агентно-орієнтованого аналізу і проектування [25]. Дана методологія базується на уявленні багатоагентної системи у вигляді обчислювальної організації, що складається з різних взаємодіючих ролей. В розробленій системі кожен агент виконує одну або кілька таких ролей, реалізуючи пов'язані з ними функції та властивості. GAIA не враховує предметну область і архітектуру окремого агента, тому вона може бути застосована для розробки широкого кола багатоагентних систем.

К. Берноном, М. Глейзе, С. Пейрукву і Г. Пікардом була розроблена ADELFE – методологія для розробки адаптивних багатоагентних систем [27]. Ця методологія є доповненням до більш загальної методології розробки програмного забезпечення Rational Unified Process. ADELFE призначена для створення систем з високим рівнем стійкості до внутрішніх і зовнішніх змін. Дана методологія приділяє основну увагу середовищу функціонування системи, робить акцент на забезпеченні можливості самоорганізації агента для адаптації до змін у середовищі, а також на розпізнаванні та вирішенні «некооперативних» (виняткових) ситуацій при взаємодії агентів один з одним.

А. Омідіні була розроблена SODA – агентно-орієнтована методологія для розробки та аналізу Інтернет-систем [28]. Дана методологія робить акцент на проектуванні спільнот агентів і розглядає індивідуального агента перш за все як частину такої спільноти, яка має певний набір соціальних функцій і якостей. Використовуючи такі поняття, як завдання, роль, ресурс, сервіс, права і протокол взаємодії SODA дозволяє відобразити соціальну

модель агента на його інфраструктуру. Таким чином, головна риса SODA полягає в тому, що внутрішнє влаштування індивідуального агента виникає з його соціальної ролі і зовнішніх якостей.

На основі зробленого огляду типів агентів, деяких агентних архітектур та моделей поведінки, можна зробити висновок, що найбільш підходящим для розробляємої системи варіантом побудови індивідуального агента є реактивна архітектура, що визначає діяльність агента у вигляді реакцій на зміни у зовнішньому середовищі. Використання цієї архітектури не дозволяє домогтися високого рівня самоврядування, але значно спрощує процес побудови агента та забезпечує надійність і передбачуваність його поведінки.

## 3. Постановка завдання

Метою даної роботи є розробка багатоагентної системи, що заснована на використанні методу розподіленого штучного інтелекту і призначена для прогнозування поведінки деякої нелінійної динамічної системи у реальному часі [1,2].

Задля цього необхідно виконати наступне:

- описати загальну схему вирішення поставленої задачі;
- спроектувати агентів системи, описати їх ролі та функції;
- розробити структуру бази даних системи;
- описати загальну архітектуру та технічні аспекти реалізації розробленої багатоагентної системи;

До багатоагентної системи пред'являються наступні вимоги:

- 1) автономність (робота без втручання користувача);
- 2) функціонування в режимі реального часу (паралельно з роботою Modeler);
- 3) розпізнавання різних типів траєкторій на графічних уявленнях моделі, яка досліджується;
- 4) регулювання часу прорахунку кожної окремої траєкторії;
- 5) визначення початкових точок для розрахунку траєкторій;
- 6) оптимізація процесу розпізнавання шляхом регулювання числа задіяних в ньому компонентів системи;
- 7) мінімізація помилок розпізнавання шляхом використання багатьох автономних розпізнавачів;
- 8) ведення бази даних, яка містить опис виявлених траєкторій динамічної системи.

## 4. Аналіз та проектування загальної схеми БАС

Багатоагентна система (БАС) – це система, що складається з багатьох інтелектуальних агентів, сполучених один з одним (рис. 1). Метою використання багатоагентної системи є вирішення задачі, яку надто складно або неможливо вирішити за

допомогою індивідуального агента або монолітної системи.

Архітектуру багатоагентної системи має сенс розглядати в двох площинах: архітектура складаючих її агентів і організація взаємодії цих агентів між собою.

Агент – це система, що знаходиться у деякому середовищі і здатна виконувати деякі автономні дії для виконання закладених у неї цілей. Агент може виконувати певний набір дій, за допомогою чого він впливає на стан середовища, у якому він діє; список можливих дій агента в деякий момент часу залежить від поточного стану середовища. Основною проблемою в роботі агента є вибір найбільш підходящої для поточних умов дії, яка б дозволила виконати завдання агента найкраще.

Основними моделями організації обчислень за допомогою багатоагентних систем в даний час є розподілений штучний інтелект і штучне життя.

Загальна модель розподіленого штучного інтелекту (РШІ) визначає рішення деякої загальної задачі за допомогою її декомпозиції на декілька підзадач, що вирішуються індивідуальними інтелектуальними агентами, і формування загального результату на основі аналізу локальних результатів, отриманих агентами (рис. 1). Розподілений штучний інтелект передбачає узгодження цілей, інтересів і стратегій окремих агентів та координацію їх діяльності.

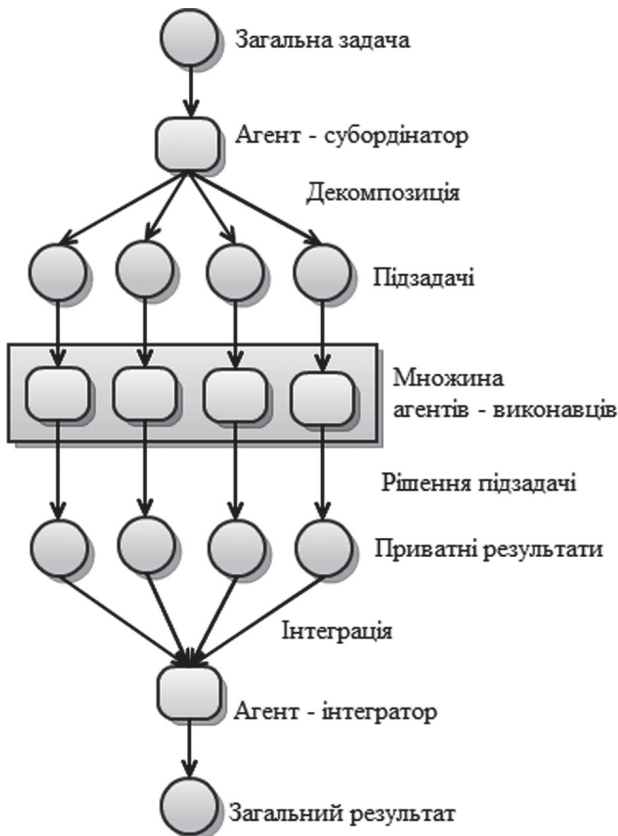


Рис. 1. Схема вирішення задачі методом розподіленого штучного інтелекту

Основною задачею програмної системи є своєчасне розпізнавання та інтерпретація траєкторій, представлених у вигляді сукупностей точок на візуальних відображеннях перетинів Пуанкаре.

Середовище моделювання Modeler надає чотири різних графічних відображення досліджуваної моделі (зовнішня і внутрішня секція, сфера і поверхня), які користувач може використовувати для дослідження таких траєкторій. Саме необхідність використання всіх чотирьох відображень моделі для розуміння загальної картини руху обумовлює застосування багатоагентного підходу і формує вигляд потрібної багатоагентної системи.

Очевидно, що «множинність» і неоднорідність процесу розпізнавання у даному випадку природньо вписується в багатоагентну парадигму – кожне відображення досліджуємої моделі може бути асоційоване з окремим агентом або спільнотою агентів. Це дозволяє розбити комплексний процес дослідження на простіші підзадачі, які розподіляються між спеціалізованими агентами. Дані агенти потім автономно вирішують виділені їм частини загальної задачі і формують локальні (приватні) результати, які використовуються для редукції загального (підсумкового) результату розпізнавання.

Таким чином, це обумовлює використання для дослідження і розпізнавання методу розподіленого штучного інтелекту. Існуючі на даний момент багатоагентні системи розпізнавання, як правило, побудовані схожим чином [5, 15, 16, 17, 19]; єдиним істотним відхиленням від даної схеми можна вважати лише роль агента-інтегратора, яка може бути відсутня, якщо агенти-виконавці досить самостійні для колективної редукції загального результату за допомогою переговорів або редукція просто не передбачена (як в Visent [16]).

Сам процес розпізнавання траєкторії на окремо взятому графічному відображенні моделі також може бути розбитий на підзадачі і, відповідно, виконуватися групою агентів.

Загалом траєкторії руху можна розділити на типи: регулярна, з малою хаотичністю, з великою хаотичністю, хаотична. Кожен з випадків володіє своїми відмінними візуальними характеристиками. Регулярна траєкторія являє собою концентричні замкнуті криві, траєкторія з малою хаотичністю – група неконцентричних замкнутих кривих, хаотична траєкторія – сукупність точок, що утворюють множину.

Таким чином, для розпізнавання кожного випадку може знадобитися окремий метод або навіть група методів (наприклад, різні методи для розпізнавання на площині і на поверхні об'ємної фігури). Це дозволяє виділити спеціалізовані типи агентів (A1, A2, A3, A4) для розпізнавання певних випадків траєкторії. У підсумку отримуємо локальну групу агентів-розпізнавачів (A11, A12, A13, A21 і т.д.) для

кожного з чотирьох графічних відображень моделі і додатковий рівень ієрархії в розподіленому процесі розпізнавання (рис. 2).

Також варто враховувати, що окрім розпізнавання траєкторій існують й інші задачі, які мають вирішувати, як розробляється багатоагентна система: визначення оптимальної початкової точки для прорахунку нової траєкторії, своєчасне припинення прорахунку поточної траєкторії і процесу дослідження моделі в цілому.

Для вирішення цих задач має сенс заснувати блок з одного або кількох агентів для забезпечення інтелектуального регулювання процесу дослідження. В рамках схеми роботи розподіленого штучного інтелекту такий блок або агент управління (АУ) повинен грати роль як агента-субординатора, так і агента-інтегратора, тобто формувати початкові дані та умови для роботи агентів-розпізнавачів та інтерпретувати отримані ними результати (P11, P12, P13 і т.д.).

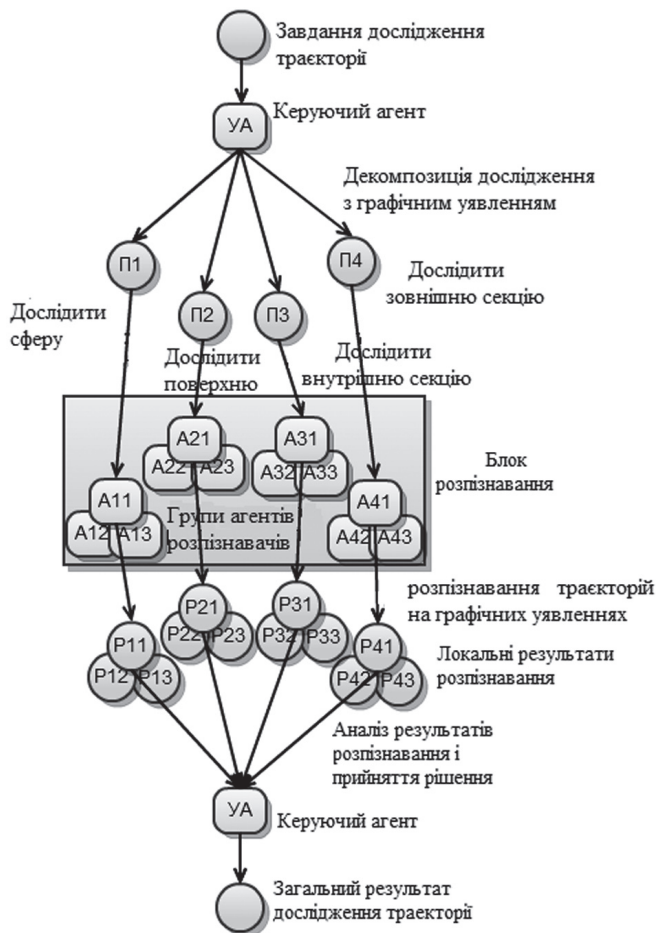


Рис. 2. Схема розпізнавання траєкторії за методом РШІ

В узагальненому вигляді роботу блоку управління можна розглядати як циклічний процес ініціації, моніторингу та регулювання прорахунку і розпізнавання окремої траєкторії.

Наступним аспектом роботи багатоагентної системи є ввід і вивід даних. У даному випадку вхідними даними є графічні відображення досліджуємої

моделі, що генеруються Modeler, а вихідними – опис цієї моделі, отриманий в результаті її дослідження. Для забезпечення модульності і уніфікації вводу/виводу даних має сенс організувати інтерфейсний агент.

Інтерфейсні агенти досить часто зустрічаються в багатоагентних системах [5, 15, 19], відіграючи роль «заглушок» (stubs) між системою і зовнішнім середовищем. Задача такого агента – надання абстрактного інтерфейсу для вводу/виводу даних, що не залежить від конкретного джерела або приймача даних. Це дозволяє організувати уніфікований ввід/вивід, забезпечує адаптивність і розширюваність системи в рамках її зв'язків із зовнішнім середовищем. Більш того, багатоагентна система може містити кілька інтерфейсних агентів, зокрема для отримання інформації з різних джерел.

В розроблюваній системі інтерфейсний агент перш за все може бути використаний для посилки запитів і команд Modeler, отримання від нього даних, перетворення даних в структурований вигляд і обробки можливих помилок комунікації.

Також має сенс виділити в системі окремого агента для передобробки вхідних даних. Таких агентів можна зустріти в деяких багатоагентних системах розпізнавання [15, 19] поряд з інтерфейсними агентами. Як правило, агент передобробки даних служить для перетворення «сирих» даних, отриманих безпосередньо від інтерфейсного агента, в систематизовану та структуровану інформацію з метою подальшого використання іншими компонентами багатоагентної системи.

В розроблюваній системі може знадобитися приводити отримані з Modeler зображення досліджуваної моделі до такого вигляду, в якому їх буде зручно обробляти агентам-розпізнавачам. Зокрема, перед розпізнаванням окремої траєкторії має сенс прибрати з зображення всі «зайві» елементи: сітку та інші статичні елементи, точки знайдених раніше траєкторій та інше. Також залежно від обраних методів розпізнавання може знадобитися масштабувати зображення, змінювати спосіб його колірного кодування, апроксимувати лінії траєкторій і т.п.

Таким чином, усі перераховані задачі можна покласти на спеціалізованого агента передобробки даних або, скоріше, на інформаційного агента. Такий агент буде надавати високорівневий доступ до даних Modeler за допомогою звернення до інтерфейсного агента.

Інформаційний агент повинен виконувати наступне:

- 1) одержувати високорівневі запити від інших агентів системи;
- 2) посилати низькорівневі запити інформаційного агента для збору даних, необхідних для виконання запиту;

3) отримувати та обробляти надіслані інтерфейсним агентом дані;

4) вислати отриману інформацію замовникові в необхідному форматі.

Таким чином, укрупнена структура розроблюваної багатоагентної системи включає в себе наступні компоненти (рис. 3): інтерфейсний агент, інформаційний агент, блок розпізнавання, блок управління.

Призначення перерахованих компонентів наступне:

1) інтерфейсний агент служить для забезпечення взаємодії з Modeler;

2) інформаційний агент використовується для передобробки графічних даних і організації високорівневого доступу до них;

3) блок розпізнавання виконує розпізнавання траєкторій на зображеннях моделі (внутрішньої і зовнішньої секції, сфери і поверхні);

4) блок управління координує загальний процес дослідження.

Вхідними даними БАС є графічні відображення досліджуємої моделі (зображення траєкторій руху), що генеруються в середовище Modeler.

Вихідними даними системи є база даних, що містить аналітичну і статистичну інформацію, отриману в процесі дослідження моделі, зокрема опис знайдених траєкторій.

#### 4. Розробка структури бази даних

В розроблюваній багатоагентній системі база даних містить інформацію щодо дослідження моделі. В процесі роботи БАС база даних є проміжним результатом дослідження, який керуючий агент використовує для прийняття рішення щодо класифікації системи та регулювання інтенсивності процесу розпізнавання. Після закінчення функціонування

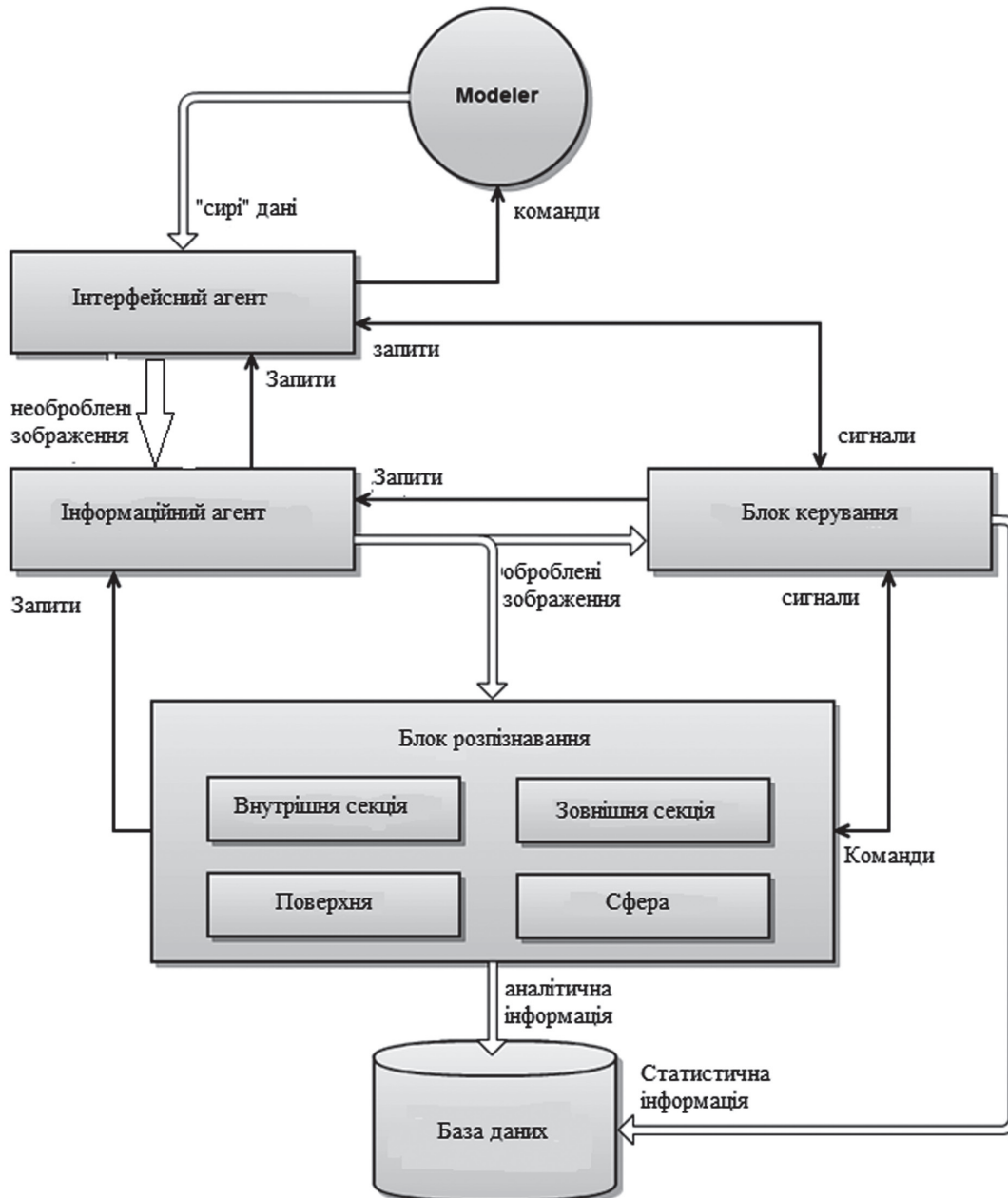


Рис. 3. Загальна схема розроблюємої багатоагентної системи

багатоагентної системи база даних являє собою загальний результат дослідження, що надається користувачеві.

У загальному випадку база даних може містити такі дані про траєкторію:

- тип (регулярна, з малою хаотичністю, хаотична);
- початкова точка розрахунку;
- точність кінцевого результату розпізнавання
- обсяг часу, витрачений на розрахунок і розпізнавання;
- склад активних розпізнавачів на момент припинення розрахунку;
- кінцеві результати всіх задіяних агентів-розпізнавачів;
- зображення кінцевого вигляду траєкторії на різних графічних відображеннях моделі;

– оцінена швидкість приросту точок на початку і наприкінці прорахунку.

Структура бази даних наведена на рис. 4.

### 5. Опис агентів та кооперативна взаємодія між ними

Розроблена система складається з сімнадцяти агентів: інтерфейсного агента, інформаційного агента, агента-планувальника, керуючого агента, тактового агента і дванадцяти спеціалізованих агентів-розпізнавачів.

Інтерфейсний агент є відповідальним за взаємодію системи із середовищем моделювання Modeler. Він надає іншим агентам системи інтерфейс для передачі в Modeler запитів, команд і отримання згенерованих в ньому зображень, а також забезпечує обробку помилок комунікації.

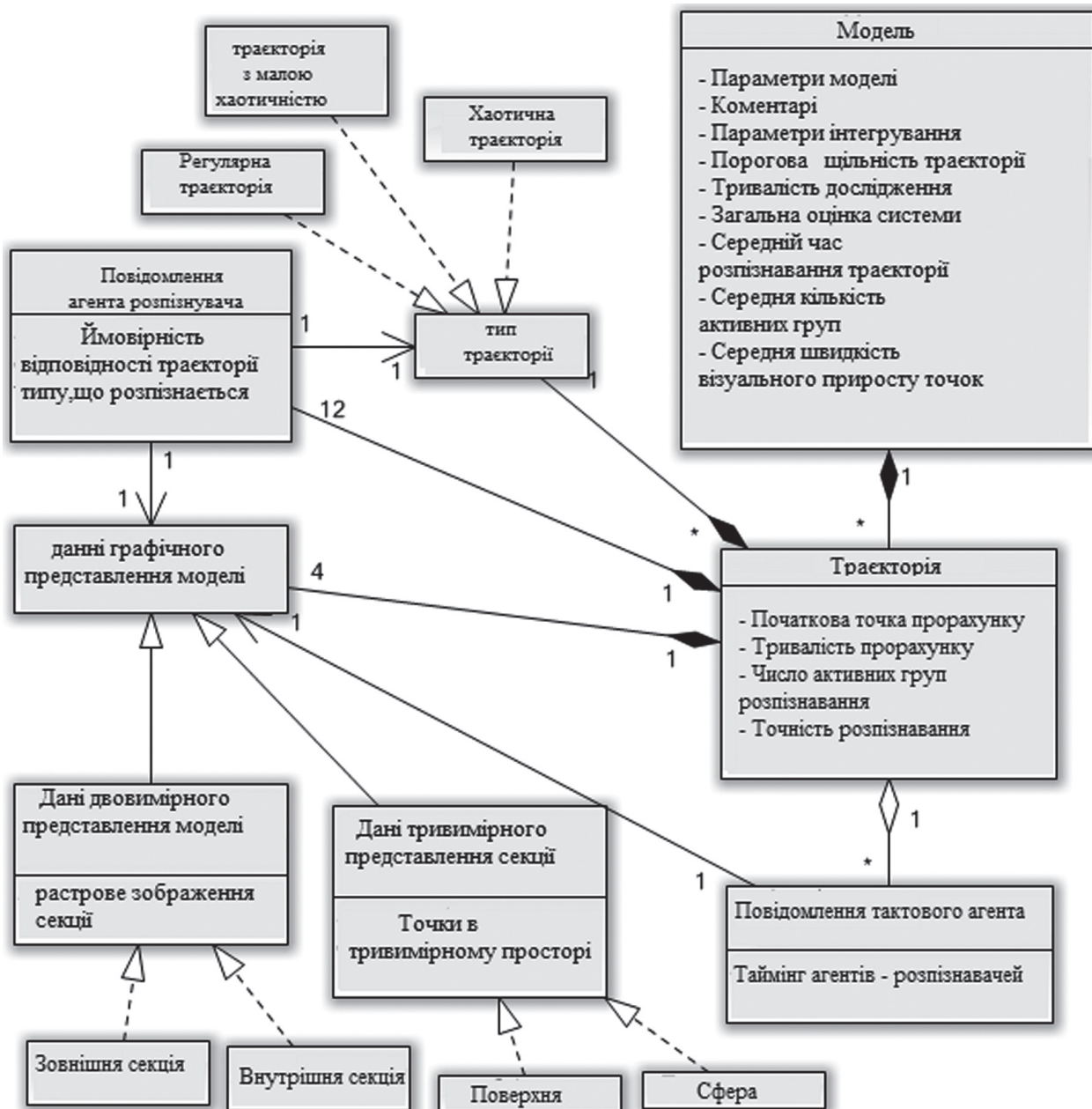


Рис. 4. Схема структури бази даних процесу дослідження

Інформаційний агент надає високорівневий інтерфейс до даних, отриманих з Modeler за допомогою інтерфейсного агента. Він відповідальний за перетворення формату і предобробку графічних даних, фільтрацію і кешування запитуваних зображень.

Агент-планувальник вирішує задачу вибору початкової точки для прорахунку нової траєкторії, базуючись на розташуванні знайдених раніше траєкторій. Також цей агент визначає, коли знайдено досить траєкторій і дослідження моделі можна завершити.

Агент-розпізнавач виконує розпізнавання одного типу траєкторії (регулярна, з малою хаотичністю, хаотична) на певному графічному відображенні моделі (зовнішній чи внутрішній секції, сфері або поверхні). Всі агенти-розпізнавачі організовані в чотири групи, по три агенти на кожне відображення моделі. Результатом роботи агента даного класу є ймовірність відповідності поточної траєкторії типу, що розпізнається.

Тактовий агент регулює частоту періодичного спрацьовування агентів-розпізнавачів в кожній окремій групі розпізнавання в залежності від швидкості приросту точок на підконтрольних групі зображеннях моделі.

Керуючий агент координує роботу агентів системи в цілому. Зокрема, він відповідальний за

регулювання розрахунку і розпізнавання окремої траєкторії, прийняття рішення щодо її класифікації та занесення відомостей про знайдені траєкторії в базу даних системи.

Узагальнена схема взаємодії агентів в системі приведена на рис. 5.

Передбачається, що розроблювана БАС повинна бути релізована у складі окремого програмного забезпечення з графічним інтерфейсом користувача, що взаємодіє з середовищем Modeler за допомогою стандартних комунікаційних засобів операційної системи (сокети, канали, міжпроцесні повідомлення).

Також варто зазначити, що структура спроектованих агентів передбачає, що вони будуть сполучатися за принципом сигналів і слотів, тобто за допомогою асинхронного зворотного виклику обробників подій.

Для звернення до інших агентів (передачі запиту на отримання даних або виконання дії) у кожного агента передбачено набір відповідних методів-сигналів.

У свою чергу, для отримання таких запитів від інших агентів використовується набір методів-слотів, що спрацьовують при виникненні пов'язаних з ними сигналів. Кожен сигнал може бути підключений до необмеженої кількості слотів, і навпаки.

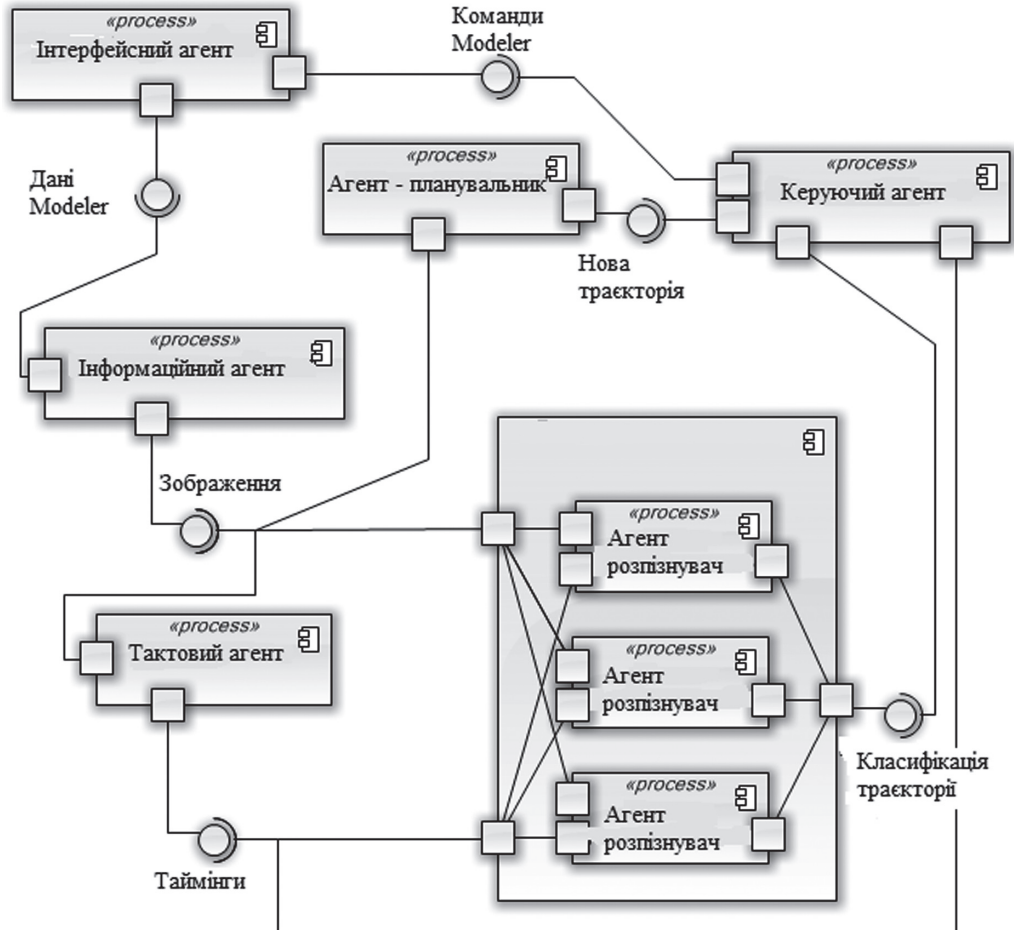


Рис. 5. Діаграма компонентів БАС

Для з'єднання сигналу зі слотом їх сигнатури повинні бути сумісні (слот повинен приймати на вхід тільки параметри, зазначені в сигналі).

Структура і склад класів агентів наведені на рис. 6.

Застосування такої схеми взаємодії дозволяє домогтися гнучкості, простоти і абстракції в організації передачі повідомлень між автономними агентами в багатоагентній системі.

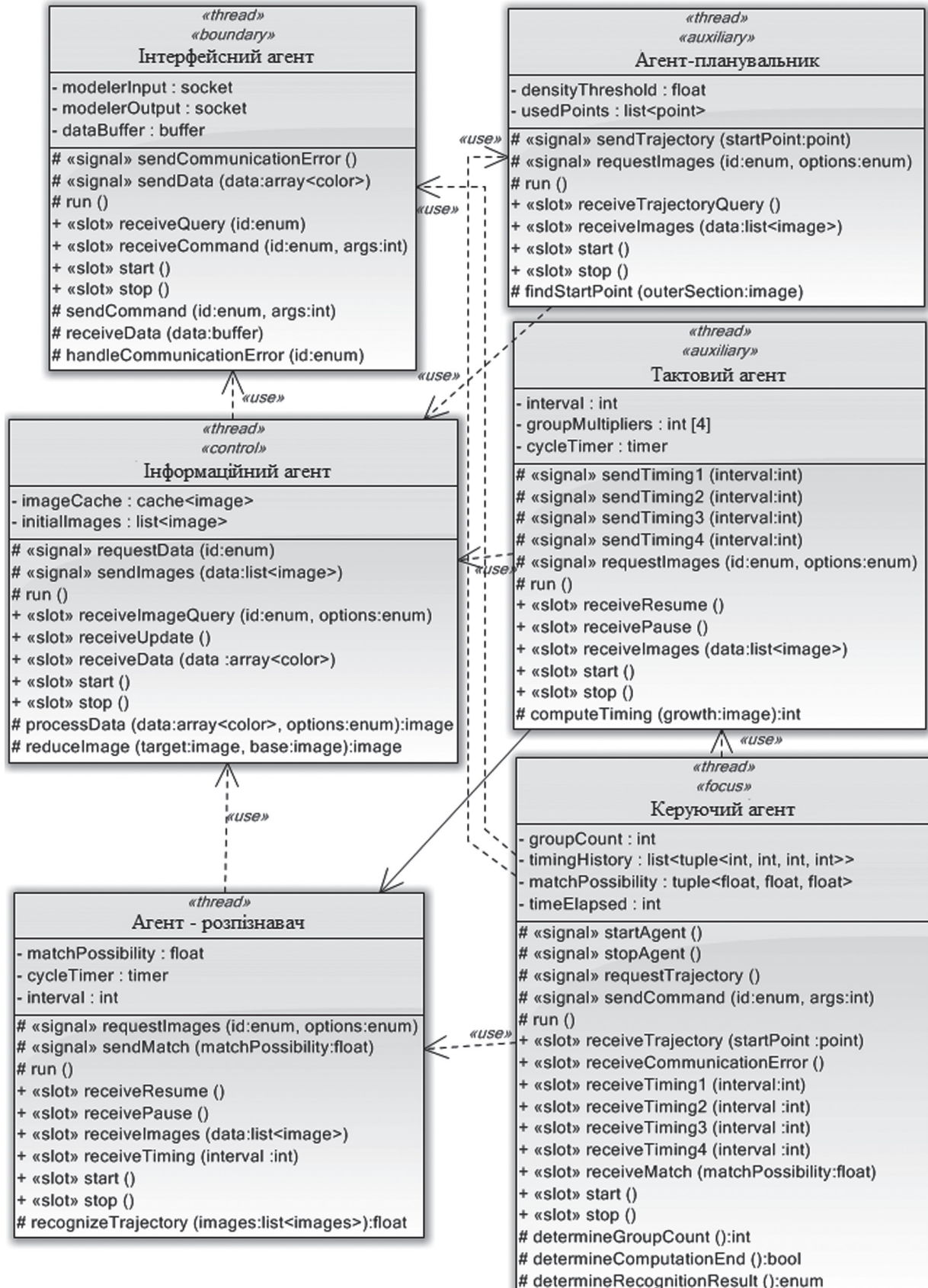


Рис. 6. Діаграма класів БАС

Таким чином, оптимальною низькорівневою основою для реалізації спроектованих агентів є потоки (threads), засновані на транзакціях, тобто працюючі в циклі обробки подій (events). Виділення агентів в окремі процеси в даному випадку найімовірніше буде зайвим, оскільки додаткові витрати і затримки при обміні даними між такими агентами навряд чи будуть компенсовані розподілом обчислювального навантаження між декількома машинами в мережі.

Реалізація системи можлива і за допомогою застосування будь-якого високорівневого набору програмних засобів, що надає готову основу для створення агентів і організації їх спільної роботи (агентного фреймворку).

### Висновки

В даній роботі була розроблена багатоагентна система для вирішення задачі дослідження моделі руху на основі методу розподіленого штучного інтелекту з використанням даних, одержаних з середовища моделювання Modeler, в режимі реального часу. Була складена загальна схема вирішення задачі, спроектовані агенти багатоагентної системи, розроблена структура бази даних, описана загальна архітектура багатоагентної системи.

Результатом розробки є специфікація потрібної багатоагентної системи, складаючих її агентів та бази даних у вигляді діаграм UML, а також опис структури, функцій і порядку роботи системи та примітки щодо її програмної реалізації.

Одержане рішення має наступні переваги:

- повна автоматизація процесу знаходження траєкторій;

- висока ефективність розпізнавання траєкторій завдяки використанню багатьох розпізнавачів статистичних даних, накопичуваних під час дослідження;

- наявність механізмів саморегуляції для мінімізації похибки розрахунку траєкторій та оптимізації споживання системних ресурсів.

Розроблена схема багатоагентної інтелектуальної системи може в подальшому бути реалізована у складі прикладного програмного забезпечення. Це програмне рішення може застосовуватися для підвищення ефективності та автоматизації певних видів діяльності, пов'язаних з дослідженням моделей динамічних систем, спільно з середовищем моделювання Modeler або іншими подібними програмними продуктами.

**Список літератури:** 1. Ручкин К.А. Разработка мультиагентной системы для прогнозирования поведения динамической системы в режиме реального времени / К.А. Ручкин, А.В. Данилов // Искусственный интеллект. – 2011. – №4. – С. 192–199. 2. Ручкин К.А. Разработка компьютерной системы для построения и анализа сечений Пуанкаре / К.А. Ручкин // Искусственный интеллект. – 2009. – №1.

– С. 300–304. 3. Decker K. Extending a Multi-Agent System for Genomic Annotation / K. Decker, S. Khan, C. Schmidt, D. Michaud // AAMAS 2008. – 2008. – pp. 13–25. 4. Rascol V. CASSIOPE: An expert system for conserved region searches / V. Rascol, A. Levasseur, O. Chabrol, S. Grusea, P. Gouret, E. Danchin, P. Pontarotti // BMC Bioinformatics 2009. – 2009. – pp. 128–135. 5. Addis A. From a Generic MultiAgent Architecture to MultiAgent Information Retrieval Systems / A. Addis, G. Armano, E. Vargiu // AAMAS 2008. – 2008. – pp. 48–61. 6. Kanteev M. Multi-Agent Meta-Search Engine Based on Domain Ontology / M. Kanteev, I. Minakov, G. Rzevski, P. Skobelev, S. Volman Skobelev // AAMAS 2007. – 2007. – 13c. 7. Rao V. S. Distributed Agent Mining and Agent Mining Interaction and Integration: A Novel Approach / V. S. Rao, S. Vidiyavathi, G. Ramaswamy // IJRRAS 4. – 2010. – pp. 57–69. 8. Rao V. S. Multi Agent-Based Distributed Data Mining: an Over View / V.S. Rao // International Journal of Reviews in Computing. – 2010. – №5. – pp. 67–82. 9. Sajja P. S. Multi-Agent System for Knowledge-Based Access to Distributed Databases / P. S. Sajja // Interdisciplinary Journal of Information, Knowledge, and Management. – 2008. – №3. – pp. 29–41. 10. Rzevski, G. Magenta Technology: A Family of Multi-Agent Intelligent Schedulers / G. Rzevski, J. Himoff, P. Skobelev. – 2006. – 11c. 11. Camilio C. A Multi-Agent Framework for E-Commerce Automated Negotiation Protocols Evaluation / C. Camilio, R. Nogueira, C. Vinhal // AAMAS 2009. – 2009. – pp. 29–35. 12. DiPippo L. C. A Real-Time Multi-Agent System Architecture for E-Commerce Applications / L. C. DiPippo, V. Fay-Wolfe, L. Nair, E. Hodys, O. Uvarov // AAMAS 2006. – 2006. – pp. 47–58. 13. Pour G. Multi-Agent System Architecture for Quality Control: Moving Toward Autonomic System for E-Commerce / G. Pour // IADIS International Conference e-Commerce 2004. – 2004. – pp. 125–141. 14. Rahal D. Multi-Agent System for Modeling Transport Systems / D. Rahal, F. Rahal, M. Chekroun // European Journal of Scientific Research. – 2010. – №1. – pp. 80–89. 15. Casella J. A Multi-Agent System for Hand-drawn Diagram Recognition / J. Casella, V. Deufemia, V. Mascardi // AAMAS 2007. – 2007. – pp. 54–59. 16. Hempel O. Agent-Based Object Recognition / O. Hempel, U. Boker, G. Hartmann // ICEIS 2000. – 2000. – pp. 33–40. 17. Juchmes R. A Multi-Agent System for the Interpretation of Architectural Sketches / R. Juchmes, P. Leclercq, S. Azar // EUROGRAPHICS 2004. – 2004. – pp. 79–87. 18. Rzevski G. Dynamic Pattern Discovery using Multi-Agent Technology / G. Rzevski, P. Skobelev, I. Minakov, S. Volman Skobelev // TELE\_INFO 07. – 2007. – 9c. 19. Salama G. A Distributed Multi-Agent Model for Face Recognition over Network / G. Salama, A. Fahmy, M. Elbar // ASAT-13. – 2009. – pp. 75–92. 20. Zarandi M. A Multi-Agent Expert System for Steel Grade Classification Using Adaptive Neuro-fuzzy Systems / M. Zarandi, M. Avazbeigi, M. Anssari, B. Ganji // Expert systems. – 2010. – №1. – pp. 161–180. 21. Sas C. Virtual environment trajectory analysis: a basis for navigational assistance and scene adaptivity / C. Sas, G. O'Hare, R. Reilly // Future Generation Computer Systems. – 2005. – №4. – pp. 19–32. 22. Brazier M. T. DESIRE: Modeling Multi-Agent Systems in a Compositional Formal Framework / M. T. Brazier, B. M. Dunin-Keplicz, N. R. Jennings // International Journal of Cooperative Information Systems. – 1997. – №1. – pp. 66–76. 23. Bauer B. UML 2.0 and Agents: How to Build Agent-based Systems with the new UML Standard / B. Bauer, J. Odell // Interdisciplinary Journal of Information, Knowledge, and Management.

– 2010. – №5. – pp. 15-28. **24. Kinny D.** A Methodology and Modelling Technique for Systems of BDI Agents / D. Kinny, M. Georgeff, A. Rao // AAMAS 2001. – 2001. – pp. 101-117. **25. Wooldridge M.** The Gaia Methodology for Agent-Oriented Analysis and Design / M. Wooldridge, N. R. Jennings, D. Kinny // Knowledge Engineering Review. – 1997. – №9. – pp. 27-51. **26. Mylopoulos J.** TROPOS: A Framework for Requirement-Driven Software Development / J. Mylopoulos, J. Castro // Lecture Notes in Computer Science. – 2000. – pp. 55-68. **27. Bernon C.** ADELFE, a Methodology for Adaptive Multi-Agent Systems Engineering / C. Bernon, M. Gleizes, S. Peyruqueou, G. Picard // AAMAS 2006. – 2006. – pp. 95-113. **28. Omicini A.** SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems/ A. Omicini // AAMAS 2005. – 2005. – pp. 37-51.

*Поступила в редколегію 25.02.2013*

УДК 681.3, 004.85, 004.89

**Анализ и проектирование многоагентной системы прогнозирования поведения нелинейной динамической системы в реальном времени / К.А. Ручкин / / Бионика интеллекта: науч.-техн. журнал. – 2013. – № 1 (80). – С. 117-127.**

В работе продолжены исследования [1], связанных с разработкой многоагентной системы прогнозирования поведения нелинейной динамической системы в реальном времени (ППНДСРЧ). Составлена общая схема решения задачи, описана общая архитектура многоагентной системы, разработана структура базы данных, спроектированы агенты многоагентной системы. Предложенная многоагентная система спроектирована на основе метода распределенного искусственного интеллекта с использованием данных, полученных из среды

моделирования Modeler [2] в режиме реального времени. Система состоит из семнадцати агентов: интерфейсного агента, информационного агента, агента-планировщика, управляющего агента, тактового агента и двенадцати специализированных агентов-распознавателей. Приведены спецификацию многоагентной системы, ее агентов и базы данных в виде диаграмм UML, а также описание структуры, функций и порядка работы системы и замечания по ее программной реализации в будущем.

Ил. 6. Библиогр.: 28 назв.

UDC 681.3, 004.85, 004.89

**Analysis and design of multi-agent system predict the behavior of nonlinear dynamical systems in real time / К.А. Ручкин // Bionics of Intelligence: Sci. Mag. – 2013. – № 1 (80). – P. 117-127.**

In the continuation of the study [1], are associated with the drafting of multi-agent system predict the behavior of nonlinear dynamic systems in real time (PPNDSRCH). Ckladena general scheme for solving the problem described overall multi-agent system architecture, developed database structure designed multi-agent system. Over-the proposed multi-agent system is designed based on the method of distributed artificial intelligence using data obtained from the simulation environment Modeler [2] in real time. The system consists of seventeen agents: interface agent, information agent, agent scheduling, managing agent, agent clock and twelve-tsyaty specialized agents recognizers. An specificity certification for multi-system, its agents and database diagrams in UML, and a description of the structure, functions, and on-line system operation and notes about its software implementation in the future.

Fig. 6. Ref.: 28 items.

Н.М. Боргест<sup>1</sup>, М.Д. Коровин<sup>2</sup><sup>1, 2</sup> Самарский государственный аэрокосмический университет  
имени академика С.П. Королева, г. Самара, Россия

## К ВОПРОСУ БАЗОВОЙ ОНТОЛОГИИ МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ

На основе онтологического анализа рассмотрены задачи управления машиностроительным предприятием и, в частности, задачи планирования производства. Проанализированы подходы и принципы создания онтологий машиностроительного предприятия. Особое внимание уделяется мультиагентным технологиям и интересубъективному подходу при описании онтологических свойств акторов на предприятии.

ОНТОЛОГИЯ, ОНТОЛОГИЯ ВЕРХНЕГО УРОВНЯ, АГЕНТ, МУЛЬТИАГЕНТНЫЙ ПОДХОД, ИНТЕРСУБЪЕКТНАЯ ТЕОРИЯ

### Введение

Развитие информационных и коммуникационных технологий за последние десятилетия определило применение междисциплинарных подходов к построению и организации работы информационных систем. В связи с этим возникла проблема обеспечения когнитивной прозрачности при сохранении точности формальной семантики в отображении все более сложного и гетерогенного информационного поля. Решением этой проблемы может быть онтология, общая теория типов объектов и отношений, составляющих конкретные предметные области, иначе концептуальная модель, позволяющая исследователям сфокусироваться на смысле информации, без привязки к конкретным форматам и языкам представления данных.

### 1. Постановка проблемы

Усложнение ИТ инфраструктуры и архитектуры баз данных предприятия диктует необходимость поиска новых подходов к организации хранения информации, перехода к семантически интегрированным базам знаний. Онтологический подход к организации данных предлагает методы решения данной проблемы, однако практическое применение общих онтологий сопряжено с рядом проблем внедрения. Решением части этих проблем может стать реализация специализированной базовой онтологии предприятия, изначально предназначенной для решения практических задач современного машиностроительного предприятия.

### 2. Понятия и принципы

В философии под онтологией понимается наука о сущем, о видах и структуре объектов, свойств, событий, процессов и отношений в различных предметных областях. Исследуя онтологию конкретного предприятия, можно говорить о задаче интеграции знаний на этом предприятии как о высшей форме представления знания и абсолютно недостижимой в силу ограниченности возможностей человека [1]. С учетом имеющихся программно-аппаратных ограничений под онтологией в практическом смысле подразумевают формализацию

и концептуализацию знаний в конкретной предметной области. Формализация знаний предполагает их классификацию. На сегодняшний день не существует общего мнения о том, какие базовые концепты должны составлять онтологию, однако большинство исследователей сходятся во мнении, что онтология должна состоять из классов сущностей предметной области, свойств этих классов, связей между этими классами и утверждений, построенных из этих классов, их свойств и связей между ними.

Онтологические подходы становятся все более популярными, расширяется сфера их применения – современные онтологические модели позволяют описывать нечеткие предметные области [2].

Существует два основных подхода к проектированию онтологии: восходящее и нисходящее проектирование [3]. Сущность восходящего проектирования заключается в последовательном описании предметной области, начиная с самых частных концептов, например, для машиностроительного предприятия это может быть описанием рабочего места токаря с последующим объединением получающихся «минионтологий» в общую систему. Онтологии, получающиеся таким путем, являются узкоспециализированными и трудно применимыми в смежных предметных областях [4]. Нисходящее проектирование онтологий заключается в предварительном построении онтологии на высоком уровне абстракции, где бы описывались наиболее общие, базовые концепты, такие как «класс», «свойство класса», «отношение», общие для многих предметных областей, с последующим доопределением концептов по уже имеющейся классификации. Такие онтологии высокого уровня абстракции называют онтологиями верхнего уровня или «top level ontology» в зарубежной литературе [5].

Онтологии верхнего уровня представляют систему, в рамках которой различные системы могут использовать общую базу знаний, позволяя при этом объединять множество специализированных онтологий более низкого уровня [6]. Концепты, определяемые такой онтологией, являются базовыми

и универсальными для множества предметных областей, что позволяет обеспечивать целостность и непротиворечивость совокупной информационной системы [7]. Стандартные онтологии верхнего уровня иногда называют универсальными онтологиями [8].

На сегодняшний день создано множество онтологических систем верхнего уровня. Среди наиболее известных систем можно назвать SUMO, онтологию Sowa [9], Dolce [10], Clip [11] и ISO 15926-2 [12]. Вышеперечисленные онтологии объединяет возможность определения ключевых концептов через описание их поведения и сценариев использования. Затем эти концепты определяются на основании более общих концептов, заложенных в онтологии верхнего уровня [6]. Такой подход позволяет, с одной стороны, избежать необходимости каждый раз «изобретать велосипед» при определении классов сущностей, а с другой — предоставляет широкие возможности адаптации онтологии под конкретную задачу и в значительной мере упрощает её поддержку.

На рис. 1 представлена многоуровневая структура онтологии машиностроительного предприятия. Онтологии верхнего и нижнего уровней создаются экспертами в предметной области в тесном сотрудничестве со специалистами-онтологами, тогда как задача наполнения онтологии концептами нижнего уровня решается пользователями без привлечения дополнительных специалистов. На практике наиболее хорошо зарекомендовавшим себя методом является наполнение онтологии через специально спроектированные формы-шаблоны, позволяющие человеку без навыков программирования или представлений о том, как устроена онтологическая база знаний, эффективно решать задачу её наполнения [11].

Нисходящее проектирование онтологий имеет ряд преимуществ, например, наличие общей онтологии верхнего уровня позволяет легко обеспечивать интероперабельность между отдельными «дочерними» онтологиями в рамках общей системы понятий.

Для обеспечения и поддержания семантической интеграции информационных ресурсов предприятия используемая онтология должна соответствовать следующим основным принципам [13, 14]:

**Принцип прозрачности:** онтология, предназначенная для общего использования, должна быть понятна для персонала, знакомого с принципами построения и работы онтологий, что обеспечивает поддерживаемость и развиваемость онтологической системы.

**Принцип открытости:** использование открытых стандартов значительно упрощает задачу системной интеграции программных комплексов предприятия.

**Принцип повторного использования доступных ресурсов:** в случае наличия готовых онтологий для конкретных предметных областей (например, онтология склада готовой продукции, онтология инструментального цеха), эффективнее использовать проверенные готовые решения, чем создавать собственный продукт с дублирующим функционалом.

**Принцип консистентности:** онтология предприятия должна обеспечивать целостность, органическую взаимосвязанность и согласованность всех элементов информационной системы предприятия.

**Принцип поддержки версионности продукции:** некоторые виды наукоемкой машиностроительной продукции имеют различный элементный состав от изделия к изделию, что должно находить свое отражение в онтологии.



Рис. 1. Структура онтологии машиностроительного предприятия

Принцип разумной достаточности: если определение не содержит полезной информации по применимости концепта, оно не должно использоваться.

С учетом вышесказанного можно сделать следующий вывод — онтология, претендующая на роль базовой онтологии машиностроения, должна, с одной стороны, быть достаточно абстрактной для охвата всей потребной предметной области, с другой — быть в достаточной степени специализированной для машиностроения с целью снижения затрат и рисков при внедрении.

### 3. Семантическая сеть

Современные системы управления — это программные комплексы высочайшей сложности. Постоянный рост количества элементов таких систем заставил специалистов искать принципиально новые подходы к автоматизированному управлению. Одним из наиболее перспективных стал зародившийся в 1980-х годах метод, получивший название многоагентный или мультиагентный подход. Сущность данного подхода заключается в переходе от централизованного управления «сверху» к самоорганизующейся системе, образованной несколькими взаимодействующими интеллектуальными агентами как программными, так и живыми, для которых описаны их цели и правила взаимодействия. Под интеллектуальностью в этом случае понимается способность агентов обучаться и приспосабливаться.

Задачей онтологии в этом случае является обеспечение взаимодействия агентов, не обязательно оперирующих в рамках одной общей теории. Агент действует в рамках онтологии, если его наблюдаемые действия укладываются в классификацию этой онтологии. Таким образом, онтология определяет «словарь» общения между агентами. Онтологическое согласование может происходить на разных уровнях абстракции, то есть агенты, оперирующие в близких предметных областях, обмениваются более конкретными данными, тогда как слабосвязанные агенты при обмене оперируют сущностями типа «класс», «отношение» и т.п. Использование онтологий позволяет достичь согласованности между агентами, однако не гарантирует полноты описаний. Множественность онтологий реального мира в данном контексте обусловлена различиями внутренних интерпретаций предметных областей разными исследователями [15].

На сегодняшний день одной из наиболее перспективных форм концептуализации знаний признаны семантические сети, представляющие собой модель предметной области в виде ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги задают отношения между ними.

На рис. 2 в качестве примера описания информационной среды предметной области машиностроительного предприятия представлена общая

схема взаимодействия между агентом заказа и агентами рабочих мест в системе внутрицехового планирования завода. Агент заказа содержит полную информацию о заказе, включая технологию, трудоемкость, стоимость и пр. Агент, выполняющий функцию рабочего, представляет собой онтологические свойства исполнителя, такие как разряд, специальность, загруженность и пр. Итогом взаимодействия агентов, обычно интерпретируемым в мультиагентных технологиях как матчинг, является составленное расписание выполнения заказов.

Онтологические модели в силу своей природы, как правило, достаточно объемны. «Масштабный фактор» — возрастание сложности системы из-за увеличения количества связей между её частями — диктует необходимость в новых подходах к управлению сложными системами. Мультиагентный подход, позволяет значительно сократить количество связей в системе, являясь одним из наиболее перспективных способов управления сложными распределенными системами.

Мультиагентный подход при создании интеллектуальных систем основывается на построении комплекса, состоящего из множества агентов. Таким образом, все управление системой осуществляется коллективом агентов, которые адаптируются под решение конкретной задачи.

Мультиагентные системы значительно отличаются от классических систем. В настоящее время основной акцент в разработке таких систем ставится на раздельной обработке и «социальном» поведении агентов. Интересной особенностью мультиагентной системы является возможность представления человека в качестве агента, что открывает широкие возможности при постепенном переходе от операторов к компьютерным агентам.

Стоит отметить, что «агент» — более общее понятие, чем система или ее элементы. Это позволяет при необходимости представлять любую часть сложной системы в качестве агента. Представленная модель пригодна для машинной обработки, что значительно расширяет возможности её использования.

Агенты взаимодействуют между собой в рамках общей внутрицеховой онтологии, которая в свою очередь входит в более общую онтологию предприятия. Результатом работы системы является расписание работ внутри цеха, фрагмент которого приведен в табл. 1. Каждая деталь или сборочная единица (ДСЕ) имеет собственный набор необходимых операций, для которых важна последовательность. Для каждой операции необходимо найти исполнителя и обеспечить правильную технологическую последовательность операций при выполнении ряда условий, например, возможность равномерного распределения нагрузки для рабочих цеха.

Таблица 1

Фрагмент цехового расписания

Название	Номер	Исполнитель	Время начала	ДСЕ
Слесарная	1	Богатов Сергей Викторович	24.01.2013 6:25	74774/113[Приспособление]
Фрезерная	6	Лукшина Надежда Альбертовна	24.01.2013 9:50	74774/113[Приспособление]
Токарная	11	Беребердин Михаил Иванович	24.01.2013 12:20	74774/113[Приспособление]
Слесарная	15	Ерофеев Геннадий Викторович	24.01.2013 15:50	74774/113[Приспособление]
Контроль	15	Сергеев Виктор Борисович	24.01.2013 18:10	74774/113[Приспособление]

Схема взаимодействия агентов приведена на рис. 2. Здесь заказ – более общее понятие, чем ДСЕ, так как один заказ может включать в себя несколько ДСЕ. Визуализация результата может быть выполнена в форме диаграммы Ганта, как показано на рис. 3.

Знания об объекте соотносятся не только с особенностями его взаимодействия со средствами наблюдения, но и с ценностно-целевыми структурами деятельности субъекта. Таким образом, можно вести речь о необходимости создания интерсубъективной теории [14], представляющей собой интеграционную платформу для достижения консенсуса неоднородными агентами относительно способа регулирования проблемной ситуации. Включение человека, его онтологических свойств: восприятия, способностей, возможностей, мотивации и потребностей, в систему, в процесс управления предприятием, существенно расширяет потенциал возможностей самой системы.

Интерсубъективный подход в сочетании с мультиагентным формируют не только иные парадигмы взаимодействия сущностей в управлении предприятием, но соответственно иные критерии такого взаимодействия и сами модели, описывающие бизнес процессы.

**Выводы**

Переход к управлению на основе мультиагентных технологий, базирующийся на онтологическом подходе при описании предметной области, позволяет современному предприятию переходить к экономике реального времени, повышает эффективность использования ресурсов, снижает затраты времени. Реализация мультиагентных технологий поддерживает непрерывное планирование в реальном времени с немедленной реакцией на события, позволяет создать масштабируемую платформу для решения задач высокой сложности.

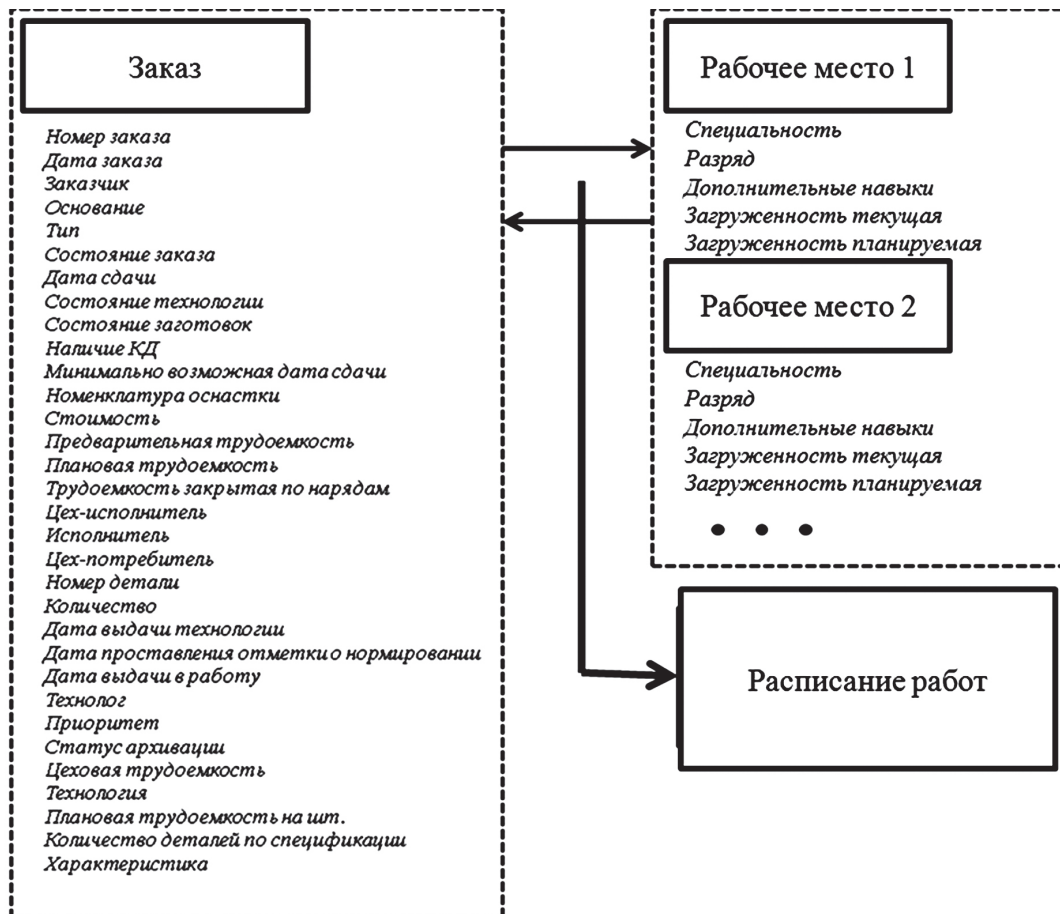


Рис. 2. Схема взаимодействия агентов

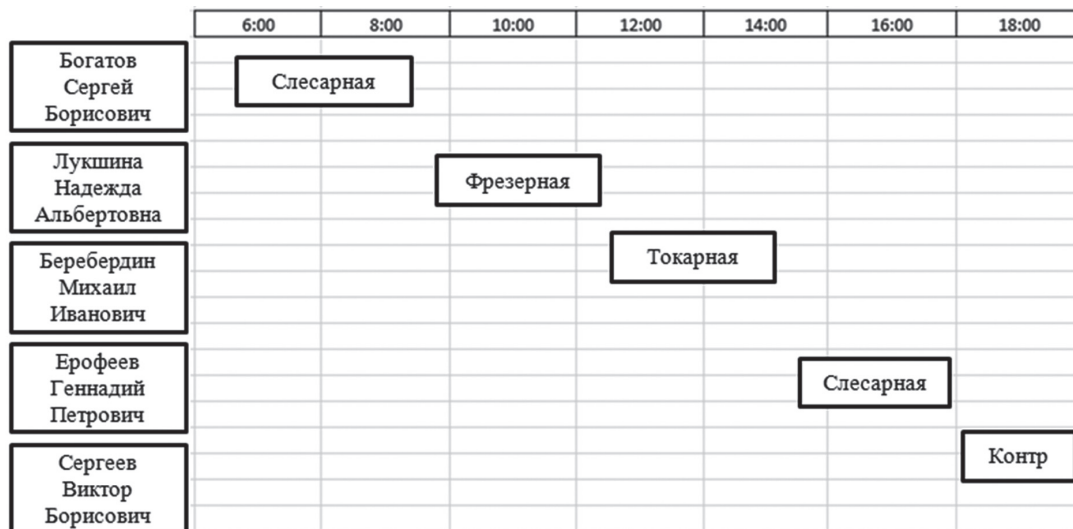


Рис. 3. Фрагмент цехового расписания в форме диаграммы Ганта

Реализация интероперабельности агентов диктует необходимость в обеспечении связанности онтологий их деятельности, что может быть достигнуто в рамках специализированной онтологии верхнего уровня для машиностроительной отрасли. Создание подобных сущностных прикладных моделей позволяет повысить прозрачность принятия решений и ответственность за их исполнение, а также в значительной мере упростить сами задачи ситуационного управления и планирования на предприятии.

Работа выполняется в рамках Государственного контракта № 07.524.12.4022 от 11.07.2012 года по созданию распределенной интеллектуальной системы согласованного управления производственными цехами корпораций машиностроительных предприятий, построенных по сетевому принципу.

**Список литературы:** 1. Mead, G. H. The individual and the social self: Unpublished work of George Herbert Mead (D. L. Miller, Ed.). Chicago: University of Chicago Press, 1982. 2. Потапова Е.В. Модель лингвистической онтологии предметной области с нечеткими семантическими состояниями терминов /Е.В. Потапова// Бионика интеллекта: науч.-техн. Журнал. – 2012. – № 2 (79). – С. 95-102. 3. Building the Semantic Web for the Process Industries using RDF, OWL, and SPARQL for the Integration, Sharing, Exchange, and Hand-over of Plant Lifecycle Information on the basis of ISO 15926.: An all-in explanation of ISO 15926, from data model to implementation. 4. Yang, A. and W. Marquardt, 2004, An Ontology-based Approach to Conceptual Process Modelling. Proceedings of ESCAPE-14, Portugal. 5. N. Casellas, M. Blázquez, OPJK into PROTON: Legal domain ontology integration into an upperlevel ontology. Proceedings of the 3rd International Workshop on Regulatory Ontologies (WORM 2005), volume 3762 of Lecture Notes in Computer Science, pages 846–855. Springer, 2005. 6. G. Nagypál and J. Lemcke. A business data ontology. Data, Information and Process Integration with Semantic Web Services Project, FP6 U 507483, Deliverable D3.3, 2005. 7. A. Abdulaev. Reality, Universal Ontology and Knowledge Systems: Toward the Intelligent World, 2008. 306 p. 8. Jan SCHEFFCZYK, Adam PEASE, Michael ELLSWORTH. Linking FrameNet

to the Suggested Upper Merged Ontology, 2008. 9. Sowa, J., 2000, Knowledge Representation: logical, philosophical, and computational foundations. Brooks/Cole. 10. Gangemi A., N. Guarino, C. Masolo, A. Oltramari, L. Schneider, 2000, Sweetening Ontologies with DOLCE. Proceedings of EKAW 2002. Spain. 11. ISO 15926-2, 2003, ISO-15926:2003 Integration of lifecycle data for process plant including oil and gas production facilities: Part 2 – Data model. 12. Smith B., Against idiosyncrasy in Ontology Development. B. Bennett and C. Fellbaum (Eds.), Formal Ontology and Information Systems, (FOIS 2006). 13. Шведин Б.Я. Онтология предприятия: экспириентологический подход. М.: ЛЕНАНД, 2010. – 240 с. 14. Скобелев П.О. Онтология деятельности для ситуационного управления предприятием [Текст] // Онтология проектирования. – 2012. – № 1. – С. 7–38. 15. Виттих В.А. Ситуационное управление с позиций постнеклассической науки / В.А. Виттих // Онтология проектирования. – 2012. – № 2 (3). – С.7-15.

Поступила в редколлегию 29.11.2013

УДК 65.011.56

Concept of the basic ontology for the engineering domain / N.M. Borgest, M.D. Korovin // Бiоника інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 128-132.

The article describes the problems of ontology in relation to the management of an engineering facility and production planning. The approaches to ontology creation are analyzed. Given representation of the basic ontology of the enterprise allows to visualize the nature of the interactions of basic entities, grounded approach to the formation of adequate criteria and conditions of their interaction.

Fig. 3. Tab. 1. Ref. 15 items.

УДК 65.011.56

До питання базової онтології машинобудування / Н.М. Боргест, М.Д. Коровін // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – Р. 128-132.

У статті розглянуті завдання онтології стосовно до управління машинобудівним підприємством і, зокрема, планування виробництва. Проаналізовано підходи до створення онтологій. Представлений фрагмент базової онтології підприємства дозволяє наочно уявити суть взаємодій базових сутностей, обґрунтовано підійти до формування адекватних критеріїв і умови їх взаємодії.

Лл. 3. Таб. 1. Бiблiогр. 15 найм.

## ОБ АВТОРАХ

<b>Аксак Наталия Георгиевна</b>	99	канд. техн. наук, старший научный сотрудник, доцент кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
<b>Бабкова Надежда Викторовна</b>	25	аспирант кафедры интеллектуальных компьютерных систем Национального технического университета «Харьковский политехнический институт»
<b>Бегунов Андрей Борисович</b>	29	магистрант Национального технического университета Украины «Киевский политехнический институт»
<b>Боргест Николай Михайлович</b>	128	канд. техн. наук, профессор кафедры конструкций и проектирования летательных аппаратов Самарского государственного аэрокосмического университета имени С.П. Королева (Национальный исследовательский университет), директор издательства «Новая техника»
<b>Вавилов Евгений Витальевич</b>	35	аспирант кафедры математического обеспечения компьютерных систем Одесского национального университета им. И. Мечникова
<b>Власенко Наталия Владимировна</b>	93	аспирантка кафедры информатики Харьковского национального университета радиоэлектроники
<b>Вороной Сергей Михайлович</b>	15	канд. техн. наук, доцент кафедры систем искусственного интеллекта Государственного высшего учебного заведения «Донецкий национальный технический университет»
<b>Дрюк Александр Дмитриевич</b>	45	аспирант кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
<b>Дударь Зоя Владимировна</b>	104	канд. техн. наук, профессор кафедры программной инженерии, и.о. зав. кафедрой программной инженерии Харьковского национального университета радиоэлектроники
<b>Егоров Станислав Вячеславович</b>	104	аспирант кафедры программной инженерии Харьковского национального университета радиоэлектроники
<b>Егошина Анна Анатольевна</b>	15	канд. техн. наук, доцент кафедры систем искусственного интеллекта Государственного высшего учебного заведения «Донецкий национальный технический университет»
<b>Ерохин Андрей Леонидович</b>	41	д-р техн. наук, профессор кафедры программной инженерии Харьковского национального университета радиоэлектроники
<b>Заболотняя Татьяна Николаевна</b>	29	канд. техн. наук, старший преподаватель кафедры программного обеспечения компьютерных систем Национального технического университета Украины «Киевский политехнический институт»
<b>Иващенко Георгий Станиславович</b>	108	аспирант кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
<b>Канищева Ольга Валерьевна</b>	25	канд. техн. наук, доцент кафедры интеллектуальных компьютерных систем Национального технического университета «Харьковский политехнический институт»
<b>Керносов Максим Андреевич</b>	54	канд. техн. наук, доцент кафедры интеллектуальных управляющих систем Харьковского национального университета радиоэлектроники
<b>Кораблев Николай Михайлович</b>	108	канд. техн. наук, доцент кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
<b>Коровин Максим Дмитриевич</b>	128	студент Самарского государственного аэрокосмического университета имени академика С. П. Королёва (Национальный исследовательский университет)

<b>Коргут Сергей Анатольевич</b>	99	аспирант кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
<b>Кочуева Зоя Анатольевна</b>	25	старший преподаватель кафедры интеллектуальных компьютерных систем Национального технического университета «Харьковский политехнический институт»
<b>Кучеренко Евгений Иванович</b>	45	д-р техн. наук, профессор кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
<b>Лазаренко Ольга Владимировна</b>	19	канд. техн. наук, доцент кафедры информационных технологий и математики Харьковского гуманитарного университета «Народная украинская академия»
<b>Литвин Олег Олегович</b>	62	канд. физ.-мат. наук, доцент кафедры высшей и прикладной математики Украинской инженерно-педагогической академии
<b>Мантула Елена Вадимовна</b>	112	аспирантка Харьковского национального университета радиоэлектроники
<b>Машталир Сергей Владимирович</b>	98	канд. техн. наук, доцент кафедры информатики Харьковского национального университета радиоэлектроники
<b>Михнова Елена Дмитриевна</b>	98	аспирантка кафедры информатики Харьковского национального университета радиоэлектроники
<b>Панченко Дмитрий Игоревич</b>	19	канд. филол. наук, старший преподаватель кафедры теории и практики перевода Харьковского гуманитарного университета «Народная украинская академия»
<b>Писклакова Ольга Александровна</b>	73	канд. техн. наук, доцент кафедры управления и организации деятельности в сфере гражданской защиты Национального университета гражданской защиты Украины
<b>Попаденко Полина Юрьевна</b>	88	студентка кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
<b>Ручкин Константин Анатольевич</b>	117	канд. физ.-мат. наук, доцент кафедры программного обеспечения интеллектуальных систем Государственного высшего учебного заведения «Донецкий национальный технический университет»
<b>Скопа Александр Александрович</b>	35	д-р техн. наук, профессор кафедры информационных систем в экономике, зав. кафедрой информационных систем в экономике Одесского национального экономического университета
<b>Солодовников Андрей Сергеевич</b>	77	ассистент кафедры биологической и медицинской физики и медицинской информатики Харьковского национального медицинского университета
<b>Стрельцова Надежда Геннадьевна</b>	99	студентка Харьковского национального университета радиоэлектроники
<b>Тянянский Сергей Станиславович</b>	82	д-р техн. наук, профессор кафедры электронных вычислительных машин Харьковского национального университета радиоэлектроники
<b>Чалая Лариса Эрнестовна</b>	88	канд. техн. наук, доцент кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
<b>Чайников Сергей Иванович</b>	77	канд. техн. наук, профессор кафедры системотехники Харьковского национального университета радиоэлектроники
<b>Широков Владимир Анатольевич</b>	3	академик НАНУ, директор Украинского языково-информационного фонда НАНУ, г. Киев

## ПРАВИЛА оформлення рукописів для авторів науково-технічного журналу «БІОНІКА ІНТЕЛЕКТУ»

Науково-технічний журнал «Біоніка інтелекту» приймає до друку написані спеціально для нього оригінальні рукописи, які раніше ніде не друкувались. Структура рукопису повинна бути такою: індекс УДК, заголовок, відомості про авторів, анотація, ключові слова, вступ, основний текст статті, висновки, список використаної літератури.

Відповідно до Постанови ВАК України від 15.01.2003 №7-05/1 (Бюлетень ВАК, №1, 2003, с. 2), стаття повинна мати такі необхідні елементи: постановка проблеми в загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями; аналіз останніх досліджень і публікацій і виділення не вирішених раніше частин загальної проблеми в даній області; формулювання цілей та завдань дослідження; виклад основного матеріалу досліджень з повним обґрунтуванням отриманих наукових результатів; висновки з даного дослідження та перспективи подальших досліджень у даному напрямку.

Статті мають бути виконані в редакторі Microsoft Word. Формат сторінки – А4 (210x297 мм), поля: верхнє – 25 мм, нижнє – 20 мм, ліве, праве – 17 мм. Кількість колонок – 2, з інтервалом між ними 5 мм, основний шрифт Times New Roman, кегль основного тексту – 10 пунктів, міжрядковий інтервал – множник (1,1), абзацний відступ – 6 мм. Обсяг рукопису – від 4 до 12 сторінок (мови: російська, українська, англійська).

УДК друкується з першого рядка, без відступів, вирівнювання по лівому краю.

*Назва статті* друкується прописними літерами; шрифт прямий, напівжирний, кегль 12. *Назви розділів* нумерують арабськими цифрами, виділяють жирним шрифтом. Відступи для назви статті, ініціалів та прізвищ авторів, відомостей про авторів, назв розділів, вступу та висновків, списку літератури: зверху – 6 пт, знизу – 3 пт.

*Анотацію* (мовою статті, абзац 4-10 рядків, кегль 9) розміщують на початку статті, в ній має бути розміщена інформація про результати описаних досліджень.

*Ключові слова* (4-10 слів з тексту статті, які з точки зору інформаційного пошуку несуть змістовне навантаження) наводять мовою рукопису, через кому в називному відмінку, кегль 9.

*Малюнки та таблиці* (чорно-білі, контрастні) розміщуються у тексті після першого посилання у вигляді окремих об'єктів і нумерують арабськими цифрами наскрізно нумерацією за наявності більше ніж одного об'єкта. Невеликі схеми, що складаються з 3-4 елементів виконують, використовуючи вставку об'єкта Рисунок Microsoft Word. Більш складні виконують у графічних редакторах у вигляді чорно-білих графічних файлів форматів .tiff, .jpg, .wmf, .cdr із розділенням 300 dpi. Рисунки мають міститися у текстовому файлі й обов'язково

подаватися окремим файлом з відповідною назвою (наприклад, Рис.1.cdr).

Усі елементи малюнка, включаючи написи, повинні бути згруповані. Усі написи в малюнках і таблицях мають бути виконані шрифтом Times New Roman, кегль у малюнках – 10, у таблицях – 9.

Малюнок повинен мати центрований підпис (поза малюнком), шрифт 9, відступи зверху і знизу по 6 пт. Ширина малюнка має відповідати ширині колонки (або ширині сторінки).

*Формули, символи, змінні*, повинні бути набрані в редакторі формул MathType або Microsoft Equation. Формули розміщують посередині рядка й нумерують за наявності посилань на них у рукописі. Шрифт – Times New Roman. Висота змінної – 10 пунктів, великих і малих індексів – 8 пт, основний математичний символ – 12 (10) пт. Змінні, позначені латинськими літерами, набирають курсивом, грецькі літери, скорочення російських слів і цифри – прямим написанням. Змінні, які є в тексті, також набирають у редакторі формул.

*Список літератури* вміщує опубліковані джерела, на які є посилання в тексті, укладені у квадратні дужки, друкують без абзацного відступу, кегль 9 пт, відступ зверху – 6 пт.

Після списку літератури з відступом зверху 6 пт зазначають дату подання статті до редколегії. Число та місяць задають двозначними числами через крапку. Розмір шрифта – 9 пт, курсив, вирівнювання по правому краю.

*Реферати* (Times New Roman, кегль – 9 пунктів, 3-4 речення) подають російською та англійською мовами. Реферат не повинен дублювати текст анотації.

Разом із рукописом (на аркушах білого паперу формату А4 щільністю 80-90 г/м<sup>2</sup>, надрукований на лазерному принтері, у 2-х примірниках) необхідно подати такі документи:

1. Заяву, яку повинні підписати всі автори.
2. Акт експертизи про можливість опублікування матеріалів у відкритому друці.
3. Рецензію, підписану доктором наук.
4. Відомості про авторів.
5. Електронний варіант рукопису, реферату та відомостей про авторів.
6. Оплату за публікацію.

Необхідно також зазначити один з наступних тематичних розділів, якому відповідає рукопис:

1. Теоретичні основи інформатики та кібернетики. Теорія інтелекту
2. Математичне моделювання. Системний аналіз. Прийняття рішень
3. Інтелектуальна обробка інформації. Розпізнавання образів
4. Інформаційні технології та програмно-технічні комплекси
5. Структурна, прикладна та математична лінгвістика
6. Дискусійні повідомлення

## INSTRUCTIONS for authors of manuscripts of the scientific journal «BIONICS OF INTELLIGENCE»

---

The scientific journal “Bionics of intelligence” accepts for publication original manuscripts which have not been published earlier. The manuscript structure should be as follows: Universal Decimal Classification (UDC) title, authors’ initials and surname (in alphabetical order), abstract, key words, introduction, main text, conclusions, references.

According to the Editorial board resolution, based on the Presidium Convention of Ukraine’s Supreme Attestation Committee of 15.01.2003 №7-05/1 (Bulletin of Supreme Attestation Committee, №1, 2003, p. 2) manuscripts must have the following required elements: introduction (general statement of a problem and its relation to important scientific and practical tasks; analysis of recent research, publications and highlighting of unsolved parts of the general problem in the given field); formulating aims and tasks of research; presentations of the main research material with full substantiation of scientific results obtained; conclusions and perspectives of further research in the given field.

Manuscripts should be submitted in Microsoft Word. Page format - A4 (210x297mm), margins: top – 25mm, bottom – 20mm; left, right – 17mm. Double column format with 5mm spacing, font – Times New Roman, font size – 10 points, line spacing – multiplier (1,1), indentation – 6mm. The manuscript should be from 4 to 12 pages (languages: Russian, Ukrainian, English).

The UDC is published from the first line, without indentation, the alignment is by a left edge. The title is in capital letters; the type is medium bold-faced Roman; type size 12. The names of sections are of extra bold type and numbered in Arabic figures. There are indentions for the names of manuscripts, initials and surnames of authors, information about authors, the names of sections, introduction and conclusions, references: top – 6 pt; bottom – 3 pt.

An abstract (in the language of a manuscript, an indentation is made up of 4-10 lines; type 9) is in the beginning of an article and contains information about the results of described studies.

Key words (4-10 words from the text of an article, which from the point of view of information search bear sense in the language of a manuscript, by way of a comma in nominative case, type 9).

Figures and tables (black-and-white, sharp and of good quality) should be in a text after a first reference in the form of embedded item and numbered separately by Arabic numerals in case of more than one item. All legends of figures and tables, including inscriptions, must be grouped. All inscriptions in figures and tables must be in Times New Roman, font size in figures – 10, in tables – 9. A table title is to the right above the table

(font size – 9). The figure should contain a centered figure legend (outside a figure), font size 9, in the centre, top and bottom indentions – 6pt. The figure width must agree with the column width (or page width).

Equations, symbols, variables should be submitted in Math Type (Equation). Equations are centered and numbered in case of references in the text. The font – Times New Roman. The size of variable – 10 points, superscript and subscript characters – 8 pt, a main math. symbol – 12 (10) pt. Variables, designated by Latin letters, should be italicized; Greek letters, abbreviations of Russian words and figures should be set in Roman type. Variables which are in the text are also submitted in Math Type (Equation).

References, submitted to the state standards, include published sources that are referred to in the main text in square brackets, without an indentions, 9pt., top indentation – 6 pt.

The date of receiving an article by the Editorial board is designated after the references with top indentions – 6 pt. Date and month should be given in numbers by way of a full stop. The font size – 9 pt, italic type, alignment should be done on the right edge.

Abstracts should be submitted in two languages: Ukrainian and Russian (Times New Roman, 9 pt, 3-4 sentences). The text of a resume must not duplicate an abstract.

The following documents must be submitted together with a manuscript:

1. An application of the following form signed by all the authors:

”You are kindly requested to accept the paper (authors’ full names and the name of an paper should be indicated) in ..... pages (the number of pages should be indicated) for publication in the scientific journal “Bionics of intelligence”. We guarantee the payment.

Information about the authors (surname, first name and patronimic of each authors, place of work, degree, academic status, contact telephone, mailing and electronic addresses should be indicated).

Signatures of authors”.

2. The text of a manuscript on A4 format white color sheets of 80-90gr/m2 density typed on a laser printer.

3. A certificate of expertise about a possibility of having the materials published in the press.

4. A review signed by a doctor of sciences.

5. Information about the authors.

6. An electronic variant of a manuscript, an abstract and information about the authors (on a 3.5” diskette or by electronic mail).

7. A receipt of payment for publication.

**АЛГЕБРО-ЛОГІЧНІ ЗАСОБИ МОДЕЛЮВАННЯ ПРИРОДНОЇ МОВИ**

Проведено аналіз алгебро-логічної структури природної мови. Розглянуто концептуально-методолгічний підхід до мови людини, що дозволяє сприймати її як деяку алгебру, а її тексти – як формули цієї алгебри.

МОВА ПРИРОДНА, АЛГЕБРА ПРЕДИКАТИВ, ВІДНОШЕННЯ, АЛГЕБРА ПРЕДИКАТНИХ ОПЕРАЦІЙ

**Вступ**

Формальним моделям семантико-синтаксичних структур мови відводиться вирішальна роль у сучасній проблематиці комп'ютерної лінгвістики та системах штучного інтелекту (ШтІ). Це пов'язано з необхідністю створення програмно-апаратного комплексу генерації та аналізу речень природної мови (ПМ).

**1. Дослідження алгебро-логічної структури природної мови**

У роботі використовується апарат алгебри предикатів [1]. Множина  $U$  може бути як скінченною, так нескінченною. У першому випадку простір  $U^m$  називатимемо скінченним, а в іншому – нескінченним.

$$P(x_1, x_2, \dots, x_n) = \begin{cases} 0, & \text{якщо } (x_1, x_2, \dots, x_n) \notin T \\ 1, & \text{якщо } (x_1, x_2, \dots, x_n) \in T. \end{cases} \quad (1)$$

Згідно з (1) можливий перехід від будь-якого відношення  $T$  до відповідного йому предикату  $P$ . Предикат  $P$ , що знаходимо по (1), називатимемо характеристичною функцією відношення  $T$ .

**2. Шляхи автоматизації обробки мовної інформації**

У даний час в системах штучного інтелекту машинний словник та комплекс програм (тезауруси) використовуються, як правило, для виконання будь-якої однієї функції.

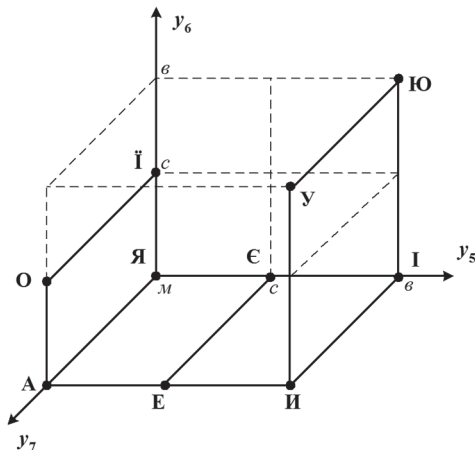


Рис.1. Тривимірний простір ознак для голосних фонем

**Висновки**

У статті запропонована узагальнену структуру інтелектуальної системи, яка відповідає новій інформаційній технології рішення задач на ЕОМ, що орієнтовані на досягнення високорівневої технології обробки мовної інформації (отримання нової якості). Істотно новим в роботі є розширення алгебри скінченних предикатів (АСП). Тепер вона охоплює не тільки скінченні предикати, а також – нескінченні. Тепер область її рекомендованого застосування розширена та охоплює довільні відношення, які далі будемо описувати за допомогою ДКАП.

**Список літератури:**

*Поступила до редколегії 15.02.20012*

УДК 519.62

**Алгебро-логические средства моделирования естественного языка / Г.Г. Четвериков, И.Д. Вечирская // Бионика интеллекта: науч.-техн. журнал. 2007. №1 (66). С. 00-00.**

В статье рассматриваются перспективные направления развития современных цифровых устройств, сетей и систем. Утверждается, что развитие средств вычислительной техники является основой автоматизации умственной деятельности человека.

Ил. 5. Библиогр.: 7 назв.

UDK 519.7

**Algebra-logical tools of modeling natural language / G.G. Chetverikov, I.D. Vechirskaya // Bionics of Intelligense: Sci. Mag. – 2007. №1 (66). С. 00-00.**

In article the perspective directions of modern digital devices, networks and systems development are considered. The carried out analysis shows means of computer facilities development is a baseline of automation of the man intellectual activity.

Fig. 5. Ref.: 7 items.

Видавництво здійснює остаточне форматування тексту відповідно вимогам друку.

**Адреса редакції:**

Україна, 61166, м.Харків, пр. Леніна 14, ХНУРЕ к.127, тел. 702-14-77, факс 702-10-13, e-mail: ira\_se@list.ru, bionics@kture.kharkov.ua

*Наукове видання*

**БІОНІКА ІНТЕЛЕКТУ**  
**інформація, мова, інтелект**

Науково-технічний журнал

№ 1 (80)

2013

Головний редактор — *М. Ф. БОНДАРЕНКО*

Відповідальний редактор — *Ю. П. Шабанов-Кушнарєнко*

Заступник відповідального редактора — *Г. Г. Четвериков*

Відповідальний секретар — *І. Д. Вечірська*

Коректор — *Л. М. Денісова*

Комп'ютерна верстка — *О. Б. Ісаєва*

Рекомендовано Вченою Радою  
Харківського національного університету радіоелектроніки  
(протокол № 20 від 29.03.2013)

**Адреса редакції:**

Україна, 61166, Харків-166, просп. Леніна, 14,  
Харківський національний університет радіоелектроніки, к. 127, 285  
тел. 702-14-77, факс 702-10-13,  
e-mail: [bionics@kture.kharkov.ua](mailto:bionics@kture.kharkov.ua)

---

Підписано до друку 22.03.2013. Формат 60 x 84 <sup>1</sup>/<sub>8</sub>. Друк ризографічний.  
Папір офсетний. Гарнітура Newton. Умов. друк. арк. 17,3. Обл.-вид. арк. 16.  
Тираж 100 прим. Зам. № .

Надруковано в навчально-науковому видавничо-поліграфічному центрі ХНУРЕ  
61166, Харків-166, просп. Леніна, 14