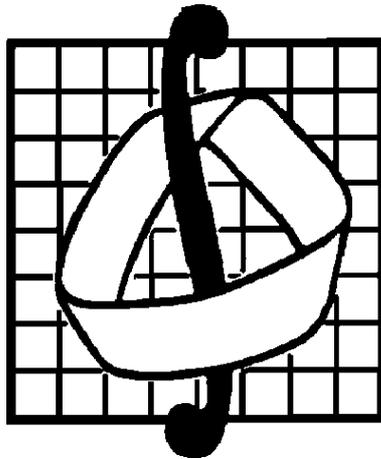


МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА



Механико-математический факультет

МАТЕРИАЛЫ
IX Международной конференции
"Интеллектуальные системы и
компьютерные науки"

(23-27 октября 2006 г.)

ТОМ 2
часть 1

Издательство механико-математического факультета МГУ
2006

УДК 519.95, 519.14, 519.1, 519.6

Издание осуществлено при поддержке
Российского фонда фундаментальных исследований
по проекту № 06-01-10-114

Материалы IX Международной конференции "Интеллектуальные системы и компьютерные науки" (23-27 октября 2006 г.), том 2, часть 1. - М.: Изд-во механико-математического факультета МГУ, 2006.

Сборник содержит работы участников IX Международной конференции "Интеллектуальные системы и компьютерные науки", проходившей на механико-математическом факультете МГУ им. М. В. Ломоносова с 23 по 27 октября 2006 г. при поддержке Российского фонда фундаментальных исследований (проект № 06-01-10-114). Сборник адресован научным сотрудникам, преподавателям, аспирантам и студентам, работающим и интересующимся тематикой математических проблем теории интеллектуальных систем и их приложений.

Научное издание

МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И КОМПЬЮТЕРНЫЕ НАУКИ" (23-27 октября 2006 г.) Под общей редакцией академика Садовниченко В. А., проф. Кудрявцева В. Б., проф. Михалева А. В.

В составлении и редактировании сборника принимали участие: Строгалов А. С., Носов В. А., Уварова Т. Д., Холоденко А. В., Галатенко А. А.

Ответственный за выпуск Строгалов А. С.

© Механико – математический факультет МГУ, 2006

ПРЕДИСЛОВИЕ

IX Международная конференция «Интеллектуальные системы и компьютерные науки», материалы которой публикуются в настоящих сборниках (т.1 и т.2) проходила на механико-математическом факультете МГУ им. М.В.Ломоносова с 23 по 27 октября 2006 г. при поддержке Российского фонда фундаментальных исследований (проект № 06-01-10-114).

Оргкомитетом конференции до начала ее работы были разосланы информационные письма в ведущие научные центры и университеты России и стран СНГ, в ряд зарубежных научных центров и университетов, отобраны наиболее интересные доклады и сообщения для заслушивания их на пленарных и секционных заседаниях.

В работе конференции принимало участие около 300 участников (ведущие ученые, преподаватели, научные сотрудники, аспиранты и студенты) из Азербайджана, Алжира, Армении, Белоруссии, Великобритании, Германии, Грузии, Иордании, Казахстана, Канады, Киргизии, КНР, Пакистана, России, Сербии, Сирии, Словении, США, Узбекистана, Украины, Франции, Черногории, Японии.

Работа конференции проходила по следующим направлениям:

- распознавание образов;
- информационные процессы и технологии;
- интеллектуальное программное обеспечение;
- базы данных и знаний;
- принципы принятия решений;
- экспертные системы и решатели задач;
- моделирование систем и процессов в технике, естественных и гуманитарных науках;
- дискретная оптимизация;
- теория дискретных функций;
- автоматы и роботы с элементами искусственного интеллекта;
- компьютеры и нейрокомпьютеры;
- нечёткая математика и её приложения;
- компьютерная алгебра;
- теория кодирования и криптография;
- параллельные вычисления

Содержание пленарных и секционных докладов отражены в настоящих материалах конференции. Тексты публикуются в авторской редакции (исправлены замеченные опечатки).

Fuzzy sets in the plural model of uncertainties: statistical interpretation

Donchenko V. S.,

*professor, Kyiv National Taras Shevchenko University.
2, Academician Glushkov prospectus, building 6, 03680, Kyiv, Ukraine
voldon@unicyb.kiev.ua*

Classical conception of Fuzzy Subsets (ClasFsS) proposed Lotfi Zadeh (Fuzzy Sets.// Information and Control, 8(3). June 1965. pp. 338-353), see also methodological view in Кофман А. (Введение в теорию нечетких множеств.- Г.: Радио и связь. 1982.- 322 с.) seemed to propose the practice a new method to manipulate with uncertainty. This method from the very beginning considered to be alternative to the ones already had been in use that time: statistical, maxmin, interval. The proposition to take into account and formalize the idea of the intermediate, transitional domains between “crisp” alternatives was the essence of the ClasFsS.

As it was mentioned earlier from the very beginning the ClasFsS considered by it founders on one hand as being alternative to those had been being in use by the moment of birth of the theory (“isolationism”), and on the other hand as the theory alternative to the classical, “crisp”, set theory. Particularly, ClasFsS considered to have nothing mutual with the statistical methods. Both of these pretensions seem to be symptoms of the theory coming into being.

Indeed, as relating set theory ClasFsS is in the “naive” faze and nothing like “crisp” set theory axioms there exit. As to the last, in the (Donchenko V.S., [1]) (see also Donchenko V.S.[2],[3]) the attempt is represented to propose some form of the abstraction axiom. Also, the full absence of somewhat that one may be nominated to be logical calculus characterizes the situation in ClasFsS now, though there are attempts to say about “Fuzzy logic”. But the one called “Fuzzy logic” is simply analogue of “crisp” propositions calculus (А.Кофман, 1982). It is reasonably to note opportunely, that functions of this “fuzzy calculus” are not full in the space of all function on “fuzzy propositions” as it is in Boolean algebra.

Also, pretensions of ClasFsS to be alternative to statistical method which namely is the way to investigate uncertainty through the frequencies of results are not fruitful for the ClasFsS because all the power of statistical methodology including interpretation turned out to be cut off. Statistical interpretation is natural for a membership function [1,2]. Besides, the considerations of that work demonstrating the gap in the object of uncertainty in the definition of ClasFsS pointed out the way to make the definition of ClasFsS correct in this aspect. This modification of the ClasFsS embodied in conception of Modified Fuzzy Sets (MoFS) in the paper has been already referred to [1] and earlier publications [2]. Namely, in MoFS classical membership function $\mu(e), e \in E$ becomes to be a function of two arguments: from $e \in E$ and from “crisp” predicate $P \in \wp$ – or correspondent “crisp” set. In such variant of determination “property $P \in \wp$ is described fuzzy way by the modified membership function $\mu^{(P)}(e), e \in E, P \in \wp$ ” with preserving general restriction $\mu^{(P)}(e) \in [0,1], e \in E, P \in \wp, P$ being a parameter..

Definition. By the Modified Fuzzy Set (MoFS) we will mean the pair $(E, \mu^{(P)}(e), P \in \wp)$, where E – abstract set (supporter) and $P \in \wp$ with the \wp - the set of “crisp” predicate on abstract “crisp” universal set U_P (or correspondent “crisp” subset of U_P).

MoFS Model Example: generalized variants of logit- and probit-regression

Logit- and probit- regressions and its Generalized variants are the best example of MoFS. As it is known, in these variants of the regressions are the dependence of the Bernoulli distribution parameter on the vector $e \in E = R^m$ is considered. This dependence has the next $\beta \in R^m$ parameterization:

$$P_e \{Y = 1\} = G(e^T \beta),$$

Where Y - Bernoulli - random variable, $G(z)$ appropriate known distribution function or the tail of the distribution.

Parameter $\beta \in R^m$ is to be estimated via observations (sample) $(e_1, y_1), \dots, (e_n, y_n)$:

$$e_i \in R^m, i = \overline{1, n} \text{ - non-random,}$$

$$y_i \in \{0,1\}, i = \overline{1, n} \text{ -}$$

the values (realizations) of independent Bernoulli - random variable $Y_i, i = \overline{1, n}$ with the parameters, correspondingly:

$$P_{e_i} \{Y_i = 1\} = G(e_i^T \beta), i = \overline{1, n}.$$

In this model case:

- $E = R^m$;
- P as a predicate defined by the relation: $P = \{Y = 1\}$;
- $\mu^{(P)}(e) = G(e^T \beta)$.

Evidently, $\mu^{(P)}(e)$ is parameterized by $\beta \in R^m$.

General character of the MoFS model example: statistical interpretation of the MoFS

In the paper [1] (see [2]) two theorems have been proved made possible statistical interpretation of the ClasFsS and MoFS. These theorems, formulated for discrete and non discrete supporters E are the next.

Theorem 1. For any finite collection of the ClasFsS $(E, \mu_{\underline{A}_i}(e)), i = \overline{1, n}$ with the one and the same supporter E one may find discrete probability space (Ω, B_Ω, P) , collection of the evens $A_i \in B_\Omega, i = \overline{1, n}$ and complete collection of the events $H_e : H_e \in B_\Omega, e \in E$, – within this probability space, that all of the membership functions $\mu_{\underline{A}_i}, i = \overline{1, n}$ may be represented as the systems of conditional probabilities in the next way:

$$\mu_{\underline{A}_i}(e) = P\{A_i | H_e\}, \text{ for any } e \in E, i = \overline{1, n}.$$

Theorem 2. Given the:

- (E, \mathfrak{F}, m) - the space with a measure;
- $(E, \mu_{\underline{A}_i}(e)), i = \overline{1, n} \mu_i(e), i > 0$ a collection on Fuzzy subsets with the equal supporters E ;
- all of the membership functions $\mu_{\underline{A}_i}(e), i = \overline{1, n}$ are $\mathfrak{F}, \mathfrak{L}$, measurable (\mathfrak{L} - Borel σ -algebra on R^l),

then:

- exist probability space (Ω, B_Ω, P) ,
- exist ξ discrete random S_p – valued random variable on (Ω, B_Ω, P) with S_p is n -element set with elements say $S_i, i = \overline{1, n}$;
- exist η random E – valued random variable on (Ω, B_Ω, P)

such, that for any $i = \overline{1, n}$

$$\mu_{\underline{A}_i}(e) = P\{\xi = S_i | \eta = e\},$$

where $P\{\xi = S_i | \eta\}$ – conditional distribution of r.v. ξ respectively r.v.. η .

Remark 1. Both of the theorems demonstrate, that for ClasFsS exist latent “crisp” predicates-events (or correspondent “crisp» sets): $A_i \in B_\Omega, i = \overline{1, n}$ for the first case and $\{\xi_i = S_i\} \in B_\Omega, i = \overline{1, n}$, - which are characterized in a Fuzzy way. For the MoFS variants, as it has been demonstrated by the logit-, probit- model example, these predicates are presented in the modified definition patently.

MoFS and MoFS model example in the context of plural models of uncertainties

When saying about ClasFsS or MoFS role in uncertainty description it is interesting to create “general platform” in which basic theory for uncertainty manipulating can take their own places. It is likely the so call “plural” model of uncertainties to be such platform.

“Plural” uncertainties model start from the conception of “observation” and “observation situation”.

When saying “observation situation” we mean “conditions” (denoted by κ) plus “observation” by itself. In it’s turn “conditions” in the “observation situation” consist of “varying part”(denoted by x) and on default part (denoted by f). As to “observation” then compulsory part of them is the response y on the conditions κ . But there is no precise meaning of the “observation”. To put it more precisely “observation” may be interpreted in the next three variants:

$$\text{"observation"} = \begin{cases} y \\ (x, y) \\ (\kappa, y) \end{cases}.$$

It is necessary to say, that standard meaning of a sequence of "observations"(real or virtual) is $(x_1, y_1), \dots, (x_N, y_N)$ while it is necessarily to be $(\kappa_1, y_1), \dots, (\kappa_N, y_N)$.

Definition. "Plural" model of uncertainties is the model, based on response y plurality in the sequence of "observations": real or virtual.

So, when saying about uncertainties one have to answer himself what content of "observation" is in use.

Indeed, plurality in y take place in deterministic case when "conditions" are of the form $\kappa_1 = (x_1, f), \dots, \kappa_N = (x_N, f)$, but "observations" treat to be y_1, \dots, y_N . This case may be interpreted as the "latent" parameter case, as it qualified in physics.

Plurality in y under just and the same $\kappa \equiv \kappa_i, i = \overline{1, N}$ is the object of application of statistical method. As it is well known all result observing or may be observed in observations are described and the frequencies (may be it's limits: probabilities) of the results or collections of the results (events) are considered.

Classical regressions illustrates the application of the statistical methods when there is plurality in y in observations with common κ , when response y is real and variability in x take place. Application of Least Square Method (LS) or it's modifications are common in this case.

Generalized variants of logit- and probit-regressions illustrates the application of the statistical methods when there is plurality in y in observations with common κ , when response y_i is binary: $y \in \{0, 1\}$ - and variability in x take place. Application of MLM (Maximum Likelihood Method) characterize this case of plurality in observations of binary response y .

And, at last, in minmax approach varying part of the conditions is considered to consist of two parts:

$$x_i = (x_i^{(1)}, x_i^{(2)}), i = \overline{1, N},$$

while each of observations treats to be

$$(x_i^{(1)}, y_i), i = \overline{1, N}, \text{ or more precisely } ((x_i^{(1)}, f), y_i), i = \overline{1, N},$$

with $x_i^{(2)} \in X_2$, where X_2 is known.

MoFS approach or modification of the ClasFsS within the plural model of uncertainties is interpreted just as the generalized logit- and probit- regression. Indeed, accordingly to basic statistical interpretation theorem from [1,3] membership function $\mu^{(P)}(e), e \in E$ has the uncertainty object P . So, MoFS observations may be treated as

$$(e_i, y_i), i = \overline{1, N} : e_i \in E, y_i = \begin{cases} 1, \text{ when } P \text{ is observed} \\ 0, \text{ when } P \text{ is not observed} \end{cases}.$$

Collections of the MoFS: situation

MoFS definition of the a fuzzy set as a pair $(E, \mu^{(P)}(\cdot)), \mu : E \rightarrow [0, 1]$ with the membership function $\mu^{(P)}(\cdot)$ being the function of two arguments with one of them fixed ($P \in \wp$) let the problem of classification or clusterization to be considered: ascription each of elements $e \in E$ to the one of the K classes, described by predicates $P_k \in \wp, k = \overline{1, K}$ from a MoFS collection $(E, \mu_k^{(P_k)}(\cdot)), P_k \in \wp, k = \overline{1, K}$ with

$$\mu_k^{(P_k)}(\cdot) : \sum_{k=1}^K \mu_k^{(P_k)}(e) \begin{matrix} \leq \\ \geq \end{matrix} 1,$$

Predicates collection $P_k \in \wp, k = \overline{1, K}$ from MoFS collection $(E, \mu_k^{(P_k)}(\cdot)), P_k \in \wp, k = \overline{1, K}$ may be interpreted as a collection of the alternatives which may take place for each of the elements of $e \in E$ with some probabilities, described by $\mu_k^{(P_k)}(e), e \in E, k = \overline{1, K}$.

We will say, that MoFS collection $(E, \mu_k^{(P_k)}(\cdot)), P_k \in \wp, k = \overline{1, K}$, describe the situation for the elements $e \in E$ or the situation made concrete by $e \in E$. Ascription the $e \in E$ under consideration to the one of K classes, described by predicates $P_k \in \wp, k = \overline{1, K}$, we will call the classifying the situation.

Classifying the situation conception

When interpreting MoFS collection as a situation, each of the $e \in E$ make the situation concrete while membership functions $\mu_k^{(P_k)}(e), k = \overline{1, K}$ for each of the fix $e \in E$ evince the “degree of appearance” for the $P_k \in \wp, k = \overline{1, K}$. Such conception of the situation make it possible to estimate the situation for that or this $e \in E$ by the maximum (or minimum) of the confidence in each of $P_k \in \wp, k = \overline{1, K}$ for fixed $e \in E$ and correspondingly to make ascription the element $e \in E$ to one of the classes, determined by $P_k \in \wp, k = \overline{1, K}$. Such approach to classifying the situation namely realizes the idea embodied in MLM (Maximum Likelihood Method). It is reasonable to remark that in MLM this idea is realized in a posteriori form: when having the observations.

Definition. Function $\hat{P}(e), e \in E, \hat{P} : E \rightarrow \{P_1, \dots, P_K\}$, determined on the common E of the MoFS collection $(E, \mu_k^{(P_k)}(\cdot)), P_k \in \wp, k = \overline{1, K}$ (situation) by the relation:

$$\hat{P}(e) = P_{k^*}, k^* = \arg \max_{k=\overline{1, K}} \mu_k^{(P_k)}(e), e \in E \quad (2)$$

said to be situation estimation for $e \in E$.

Situation Model example: clustering the distributions, probe sets (a priori data)

Simple example may demonstrate the conception of classifying the situation. This example relates to the observations of the distributions with known densities. MoFS membership functions with transparent statistical content are constructed being namely probabilities of so called “probe” crisp set-events. Be the limiting be applied the classification by the minimum or maximum of the density functions (“a priori” maximum likelihood classification) appears.

References

1. Donchenko V. Fuzzy sets: Abstraction axiom, Statistical Interpretation. Observations of Fuzzy Sets. //International Journal “Information Theories and Applications”.– V.13, №3.– 2005.–p.233-239.
2. Donchenko V. S. Conditional distributions and Fuzzy sets. // Bulletin of Kiev University. Series: Physics and Mathematics, №3,4 1998. (In Ukrainian)
3. Donchenko V. S. Statistical models of observations and Fuzzy sets. // Bulletin of Kiev University. Series: Physics and Mathematics, №1, 2004. (In Ukrainian)

Ontologies for computer science research¹

Tatiana Gavrilova, Seppo Puuronen

e-mail: gavr_csa@rambler.ru, sepi@cs.jyu.fi

*Saint-Petersburg State Polytechnical University, Russia,
University of Jyväskylä, Finland,*

The paper presents a structured approach to facilitate students' research activity in computer and information sciences based on methodology of knowledge engineering. It features ontology or conceptual domain structure design and

¹ The work is partly supported by Russian Foundation for Basic Research (grant 04-01-00466) and grant from Academy of Finland

refinement as a kernel and guide for a fruitful and meaningful research activity. Such approach may help graduate and post-graduate students to work as knowledge analysts and to shape the proper framework for their study.

Key words: ontology design and development, knowledge engineering, research activity.

1 Introduction

Any research is aimed to enhance the humanity knowledge pool. The need of knowledge exchange and reuse became a global problem for scientific and research community with the exponential growth of Internet.

Doing research is a multi-faceted creative and challenging activity. Young researchers use a lot of informal rules-of-thumb advice that may help but not a systematic guidelines. Major recommendations deal with library work, citation and other important but not essential components of research activity. Web resources now help to streamline the process of finding, selecting, and entering information from the Web, corporate databases, and reference materials into a paper. We are speaking about not syntax but semantic of study. Not HOW but WHAT? Of course the skilful and experienced supervisor controls the process.

This paper may be helpful are for the self-guided students and young supervisors.

In addition, it details an approach to promoting integrity during the education of researchers, including how to develop an effective thesis. Providing a framework for research this paper may be essential for anyone concerned about methodology of scientific study.

Good hint for better understanding of research activity may be found in [Järvinen, 2001] where main research approaches are presented. Fig.1 describes visually the taxonomy of main methods. This paper deals with conceptual analytical-approach which studies reality and identifies the information models.

All researchers are knowledge (or information) workers. They speak different languages – hardware specialist’s language is different to language used by those who do business modelling. They use these languages to specify and to elucidate the domain.

Ontology may serve as a common language to all the researchers as an ontology is an explicit specification of a conceptualization [Gruber, 1993]. Describing ontology is describing the skeleton of the problem. It is not only description, it is an explanation and clarification of new knowledge that was worked out.

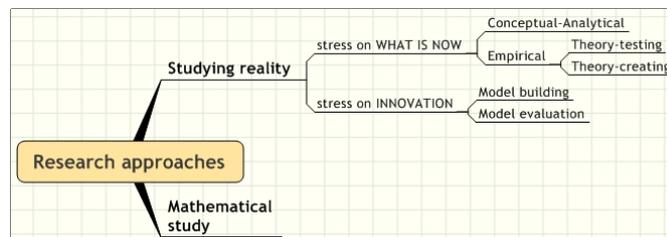


Fig.1. Taxonomy of research approaches.

2. Ontological engineering – first steps

Ontological engineering (Fig.2) comprises different fields related to ontology design, development and application.

From a philosophical viewpoint, “ontology” (without the indeterminate article and with the uppercase initial) is the branch of philosophy which deals with the nature and the organization of reality [Guarino, 95].

Now ontologies aim at capturing domain knowledge in a generic way and providing a commonly agreed understanding of a domain, which may be reused and shared across applications and groups [Chandrasekaran, 99]. Neches and colleagues [Neches, et al, 91] gave classical definition as follows “An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary”. Later, in 1993, Gruber’s definition (see before) becomes the most referenced on the literature.

Ontology is a set of distinctions we make in understanding and viewing the world. There are a lot of other definitions of this milestone term [Neches et al ,1991; Gruber, 1993; Guarino et al, 1995; Gomez-Peres et al, 2004].

1. Ontology as an informal conceptual system.
2. Ontology as a formal semantic account.
3. Ontology as the structured vocabulary.

All these definitions together clarify the ontological approach to knowledge structuring on one hand, on the other hand give enough freedom to the open thinking. So ontological engineering gives the intuitively clear representation of domain structure, research techniques, main concepts and relationship among them.

Other definitions emerge as a consequence of how their authors build and use ontologies. We can distinguish here between top-down or bottom-up approaches depending on what was first the ontology or the KB. Swartout and colleagues [Swartout, 97] define the word ontology as follow: "an ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base". This definition is based on the fact that they have built SENSUS (a broad ontology with more than 50,000 terms) and they use it as a basis for building domain specific ontologies by identifying those terms in SENSUS that are relevant to a particular domain and then prune the skeletal ontology using heuristic. The result of the prune mechanism is the skeleton upon which the KB is built. So, we can say that this is a top down approach since the ontology was the starting point.

The opposite approach was taken at the KACTUS project [Bernaras, 1996], where the ontology is built after a process of abstraction of the content already represented in a knowledge base. Ontologies are often equated with taxonomic hierarchies of classes, but class definitions, and the subsumption relation, but ontologies need not be limited to these forms [Gruber, 1995]. Visual approach to present ontology is not only compact but also very comprehensive. It makes ontology a powerful mind tool (Johanssen, 1998; GavriloVA, Voinov, 1996).

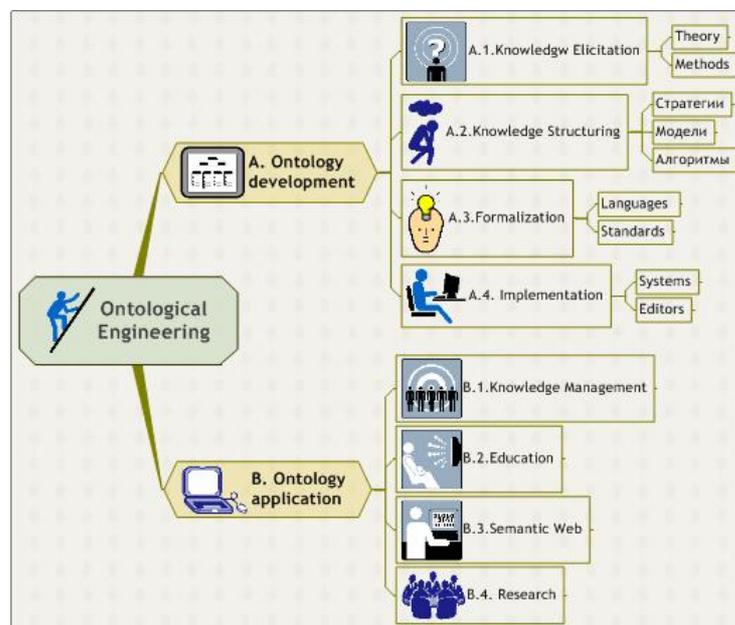


Fig. 2. Ontological Engineering Architecture

Ontologies are useful structuring tools, in that they provide an organising axis along which every researcher can mentally mark his vision in the information hyper-space of domain knowledge. Rather often we can't express all the information in one ontology, so subject knowledge storage includes a set of ontologies. Some problem may occur when jumping from one ontological space to another. But constructing of meta-ontologies may help.

Ontology development also faces the knowledge acquisition bottleneck problem. The ontology developer comes up against the additional problem of not having any sufficiently tested and generalised methodologies recommending what activities to perform and at what stage of the ontology development process these activities should be performed. That is, each development team usually follows their own set of principles, design criteria and steps in the ontology development process. The absence of structured guidelines and methods hinders the development of shared and consensual ontologies within and between teams, the extension of a given ontology by others and its reuse in other ontologies and final applications [Guarino, Giaretta, 1998].

Until now, few domain-independent methodological approaches have been reported for building practical ontologies [Mizogushi, Bourdeau 2000; Swartout, Patil, Knight, Russ, 1997; Fensel, 2001]. These methodologies have in common that they start from the identification of the purpose of the ontology and the need for domain knowledge acquisition. However, having acquired a significant amount of knowledge, major researchers propose a formal language expressing the idea as a set of intermediate representations and then generating the ontology using translators. These representations bridge the gap between how people see a domain (implicitly) and the languages in which ontologies are formalised (explicitly).

This paper proposes the most explicit way for ontology design - to use the visual representation in a form of a tree or set of trees. Ontology developers (who are unfamiliar with or simply inexperienced in the languages in which ontologies are coded, e.g. DAML, OIL, RDF) may find it difficult to understand implemented ontologies or even to build a new ontology. It is easier for any researcher simply to draw the ontology using well-known to everybody “pen and pencil” technique then to study these languages. That is why visual approach is so natural that can be used without any pre-requisite experience.

Figures 3 and 4 illustrate the idea how ontology bridge the gap between chaos of unstructured data (names of different models and techniques for knowledge mapping and representation) and clear understanding of modern classification. Our approach shows that ontology development process needs some creative efforts of meta-concepts definition that helps to name the groups and to structure the chaos. Figure 2 propose to define two meta-concepts or classes – one aimed at informal knowledge mapping (as it is in mind) and other involving formal and semi-formal machine representation models.

Although there are some methods that are rather powerful and versatile [Kremer, 1998], the knowledge analyst (every ontology developer should become an analyst) is still weakly supported while working for a set of ontologies to describe his/her main research results. This process is the most important and critical stage in the research thesis development life cycle - transition from first drafts and vague ideas to conceptualisation by understanding and realisation of the subject information structure and main problem solving ways. The student has to do this sophisticated procedure alone or but better to ask for help of supervisor who should have some experience as professional analyst.

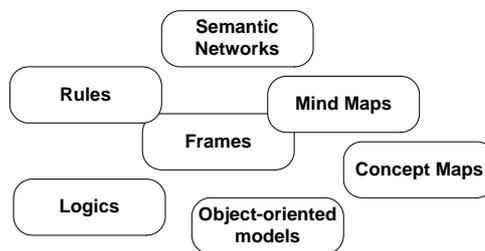


Fig. 3. Unstructured set of Knowledge Mapping Techniques

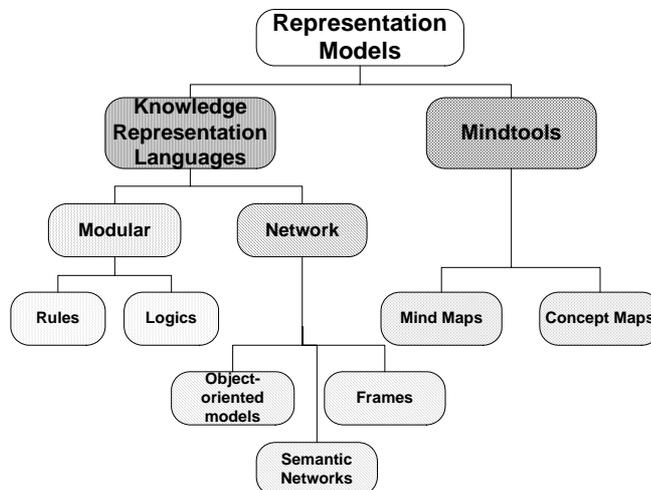


Fig. 4. Ontology “Knowledge Representation Models”

But there is a need of some preliminary analysis of data and documents before ontology may be developed.

3. Object-structured Approach.

For that purpose a special methodology Object-Structured Analysis (OSA) has been developed [Gavrilova and Voinov, 1992-2000], which is intended to help knowledge analyst to perform the most informal step of knowledge acquisition, concluding in prior conceptual structuring of the subject domain. The approach presents the enhancement of classical structured analysis methodology [Sowa, 1994; Yourdon, 1990] and object-oriented Analysis OOA [Booch, 1993] to knowledge engineering.

OSA is based on decomposition of subject domain into several (3-8) strata (Tab.1). The number of strata is considered by the analyst. This multi-step and time-consuming procedure is methodological base for effective constructing of subject ontologies.

Table 1. Matrix for OSA

1	WHAT FOR Knowledge	Strategic Analysis: Targets, Aims, Requirements, Constraints.
2	WHO Knowledge	Organisational or Historical Analysis: Main Researchers, Human Resources, Actors.
3	WHAT Knowledge	Conceptual Analysis: Main Concepts, Processes, Entities and Relationships between them.
4	HOW TO Knowledge	Functional Analysis: Main Algorithms, Decision Procedures, Business Processes Modelling, Decision Making Models.
5	WHERE Knowledge	Spatial Analysis: Geography, Environment, Communications, etc.
6	WHEN Knowledge	Temporal Analysis: Historical Dates, Schedules, Time Constraints, etc.
7	WHY Knowledge	Causal Analysis: Explanations to Decision Making Models.
8	HOW MUCH Knowledge	Economical Analysis: Resources, Losses, Incomes, Revenues, SWAT, etc.

Level →	Domain Level in general (u_1)	Problem Level (u_2)	Sub-Problem Level (u_3)	(u_n)
Stratum ↓				..	
<i>Strategic Analysis</i> s_1	E_{11}	E_{21}	E_{31}	E_{i1}	E_{n1}
<i>Organisational Analysis</i> s_2	E_{21}				
<i>Conceptual Analysis</i> s_3	E_{31}				
<i>Functional Analysis</i> s_4	E_{41}				
<i>Spatial Analysis</i> s_5	E_{51}				
<i>Temporal Analysis</i> s_6	E_{61}				
<i>Causal Analysis</i> s_7	E_{71}				
<i>Economical Analysis</i> s_8	E_{81}				
s_m	E_{m1}				E_{mn}

Filling that matrix is performed into two steps:

Step1. Global (vertical) analysis, i.e. decomposition of the heterogeneous domain information into the groups related to mentioned above methodological strata.

Step 2. Local analysis of each individual stratum (horizontal), concluding in maintenance of gradually detailed structures. The number of levels depends on peculiarities of the subject domain and could vary dramatically for different strata. From the point of view of methodology the number of levels $n < 3$ indicates ill-structured domain knowledge.

The first level (or column 2 in the table) corresponds to the discipline information as a whole. The second one corresponds to the problem that is studied now. The others may correspond to particular sub-problems, depending on the required reasonable deepness of detailing. The procedure of the described analysis may be performed both in top-down and bottom-up strategies, including their possible mixture.

The formation of strata with more or less definite meaning as described in Tab.1 allows avoiding many traditional didactic mistakes in teaching and learning. The minimal obligatory set of strata for the course structuring development is:

s3: Conceptual Structure or subject ontology.

s4: Functional Structure or main problem solving procedures.

Other strata are designed and developed if needed by subject peculiarities, e.g. spatial and temporal analysis strata (s5 and s6) may be formed in those disciplines which study construction or management where the issues of scheduling, real-time operations, real object manipulation are substantial.

Step 1 algorithms may be sketched in such form:

1.1: Gather all the data and knowledge of discipline identification

1.2: Select a set of N strata to be formed ($N \geq 3$).

1.3: For each i-th stratum select a subset of all available information, relevant to that stratum and represent it in way appropriate to that stratum (see below).

1.4: If there remains unused bulk of information, increase number of strata and repeat step

1.3. Otherwise, begin the horizontal analysis of each declared stratum.

Step 2 is horizontal analysis of strata that depends on the number of columns in OSA matrix and may be performed in two ways: deductive (top-down) and/or inductive (bottom-up). As the most essential stratum is s3 (WHAT-analysis), the horizontal analysis for it is concluded by resulting conceptual structure or a set of the domain ontologies.

Analogous algorithms were developed and practically tested and evaluated by the authors during developing of distance learning courses for different branches of computational science and for artificial intelligence (AI).

4. Main ontologies to facilitate research

However there are a lot of serious works about ontology design [Fensel, 2001; Gomez, 2004; Misogoshi, 2000] still practical ontology development is not a routine procedure. However it is very close to frame based representation formulated by Minsky [Minsky, 1974]. It's an irony but a truth that most of the modern innovative silver bullets for conceptual structuring where old solutions from practice of artificial intelligence studies.

Figure 5. illustrates our the structured view over ontology pre-theory. This approach describes ontology's classifications based on different criteria. Any researcher has to understand what type of ontology may help to clarify his/her research results.

The MindManager tool [Mindmanager, 2003] used for this picture layout is one of best visual tools for some types of ontologies presentation.

Knowledge in ontologies is mainly formalized using five kinds of components: concepts, relations, functions, axioms and instances [Gruber,1993]. In practice the main are concepts and relations.

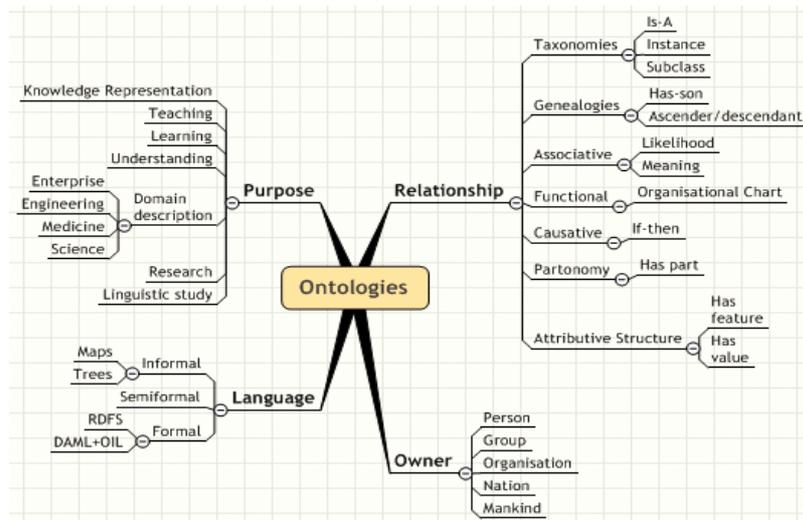


Fig. 5. Ontology Structure

That is why the major methodological classification is based on types of relationships between the concepts.

a. Classification based on type of relations

- *Taxonomies* that organise concepts into sub- super-concept tree structures. The most common forms of these are Specialisation relationships commonly known as the 'is a kind of' relationship. Partitive relationships describe concepts that are part of other concepts.
- *Genealogical* feature functions are Father-of or Predecessor-of.
- *Associative* relationships that relate concepts across tree structures. Associative relationships that represent, for example, the functions, processes a concept has or is involved in and other properties of the concept.
- *Functions* are a special case of relations in which the n-th element of the relationship is unique for the n-1 preceding elements. Formally, functions are defined as: $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$.
- *Causative'* relationships (IF -> THEN relationship)
- *Partonomies (HAS-PART relationship)*
- *Attributive trees (HAS-ATTRIBUTE and HAS-VALUR relations)*

b. Classifications based on language used

Uschold and Gruninger (Uschold, 1996) have distinguished four kind of ontologies depending on the kind of language used to implement them. They are:

- *Informal* ontologies if they are written in natural language. These languages are called *Vocabularies*, they support the creation of purely hand-crafted ontologies with simple tree-like inheritance structures. For example "The Gene Ontology".
- *Semi-formal* ontologies are those which are written in a frame-based system. This provides greater structure. Frame-based systems are based around the notion of frames or classes which represent collections of instances (the concepts of the ontology). Each frame has an associated collection of slots or attributes which can be filled by values or other frames.
- *Formal ontologies* are defined in a language with formal semantics, theorems and proofs of such properties as soundness and completeness. A language for which is alternative to frames is logic, notably Description Logics (DLs).

c. Classification based on Purpose

Depending on purpose ontologies are usually classified in the following categories:

- *Knowledge Representation ontologies*

They capture the representation primitives used to formalize knowledge in knowledge representation paradigms and don't commit themselves to any particular domain.

Examples: Frame-Ontology (Gruber, 93), which captures the representation primitives (classes, instances, slots, facets, etc.) used in frame-based or object oriented languages.

- *Top level ontologies*

In a top level or upper level ontology, the following characteristics are desirable:

- Universal*, that is, every concept one can imagine can be correctly linked into the upper ontology in appropriate places.
- Articulate*, that is, the distinctions that are made in the ontology are both necessary and sufficient for most purposes.

Examples: Sowa's Top Level Ontology, Top level of particulars of Guarino's group, IEEE Standard Upper Ontology, Upper Cyc® Ontology.

- *Linguistic ontologies*

The main characteristic of this kind of ontologies is that they are bound to the semantics of grammatical units (words, nominal groups, etc.).

Regarding other features, they form quite a heterogeneous group of resources, used mostly in natural language processing.

Examples: CoreLex, EDR Electronic Dictionary, EuroWordNet, Generalised upper Model, Goi-Taikai's ontology, Mikrokosmos' ontology, SENSUS, ThoughtTreasure Ontology, Wordnet.

- *General/Common ontologies*

They include vocabulary related to things, events, time, space, causality, behaviour, function, etc. the well known general and domain independent ontologies are listed in the examples: Cyc, E-cl@ss, NAICS (North American Industry Classification System), SCTG (Standard Classification of Transported Goods), UNSPSC.

- *Domain ontologies*

These are reusable in a given domain. They provide vocabularies about the concepts within a domain and their relationships, about the activities that take place in that domain, and about the theories and elementary principles governing that domain.

Examples:

Chemistry ontologies: CHEMICALS, Environmental pollutants, Engineering ontologies: EngMarth, PhysSys, Ontology for Engineering Mathematics, PLANET.

Enterprise Ontologies: ProPer, Process Specification Language Ontology, TOVE,

Medical Ontologies: GALEN, Gene Ontology, MeSH, ON9, UMLS, SNOMED

Research Ontologies: AIFB, (KA)2, CERIF ontology (Common European Research Informat), SWRC Semantic Web Researcher Community

- *Meta-ontologies, also called Generic Ontologies or Core Ontologies*

They are reusable across domains. The most representative example could be a mereology ontology (Borst, 1997) which would include the term part-of. Dublin Core (Weibel, 95) provides a vocabulary for describing the contents of on-line information sources.

- *Task ontologies* provide a systematized vocabulary of the terms used to solve problems associated with tasks that may or may not be from the same domain. These ontologies provide a set of terms by means of which to generically describe how to solve one type of problems. They include generic names, generic verbs, generic adjectives and others in the scheduling tasks.

- Application ontologies contain the necessary knowledge for modelling a particular domain.

d. Classifications based on owner

It is one of the most obvious and simple classifications. See Figure 5.

The given classification may be used as a roadmap to understand what set of ontologies is really a must and which may be used as complimentary ones.

5. Research framework based on ontologies

Traditional approach to research may be formulated in general as a set of 8 sequential steps proposed by Jenkins [Jenkins, 1985] to describe any research process. It is a simplification as the process is really iterative:

1- Idea

2- Library research

3- Research topic

4- Research strategy

5- Experimental design

6 -D data capture

7- Data analysis

8- Publish results

Ontological approach can facilitate practically all the named steps with the aid of visual mapping schemes which serve as a visionary mindtool and structured framework. Any researcher is in a role of knowledge analyst and he has to describe *concepts* representing entities or 'things' within a domain or approaches, tasks, functions, actions, strategies, reasoning processes, etc. Then is the turn of *relations* describing the interactions between concepts or a concept's properties.

Figure 6 illustrates some substantial features of ontology development.

First, one should elucidate the level of granularity of abstraction, it is extremely important to find the proper balance between generalization and detailing.

Second, there are different types of cognition [Norman, 2001] from behavior (experiential) and from decision making (reflective). We develop this idea by continuing reflective branch by deductive: logical and case-based.

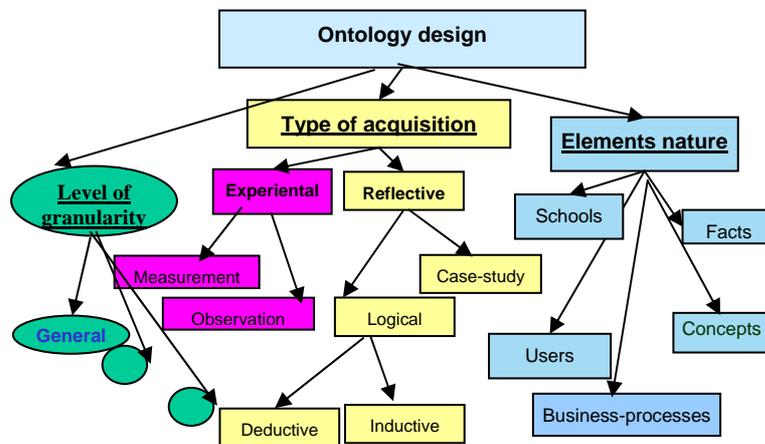
Logical way may be based on deductive (top-down) and inductive (bottom-up) reasoning:

- Experiential and
- Reflective
 - Logical
 - Deductive
 - Inductive
 - Case-based study.

Third, one who is building ontology is forced to give up with fuzzy “Victorian style” writing and turn to minimal and precise semi-formal definition of elements or

- Facts, goals,
- Business-processes, reasoning processes, technological processes,
- Schools, research approaches, strategies,
- Concepts, tasks, functions, actions
- Users, customers, companies, groups, etc.

Fig. 6.



Ontology design features

As developing ontology is tough, time-consuming intelligent process one may have a seduction to borrow ontology. There are a number of public domain libraries that are accessible via Web but they can't truly help your own research ontology development process. But they can give some general ideas:

- DAML repository (<http://www.daml.org/ontologies/>). \
- Ontolingua Server repository (www-ksl-svc.stanford.edu:5915/).
- Universal repository (<http://www.ist-universal.org/>).
- SHOE repository (<http://www.cs.umd.edu/projects/plus/SHOE/onts/>).
- WebODE (<http://babage.dia.fi.upm.es/webode/>)
- WebONTO (<http://kmi.open.ac.uk/projects/webonto/>)
- Ontosaurus (<http://www.isi.edu/isd/ontosaurus.html>)

Some design criteria set of principles that have been proved useful in the development of ontologies are [Arpirez et al., 1998; Bernaras et al., 1996; Borgo, 1996; Gruber, 1995]:

- *Clarity and Objectivity,*
- *Completeness ,*
- *Coherence ,*
- *Maximum monotonic extendibility ,*
- *Minimal ontological commitments*
- *Ontological Distinction ,*
- *Diversification of hierarchies,*
- *Modularity,*
- *Minimization of the semantic distance between sibling concept,*

- *Standardization of names* whenever is possible.

Taking into consideration all these we can propose one simple ontology development algorithm in a nutshell comprising 7 steps:

1. Collect all the relevant information, select and verbalise all essential concepts (glossary).
2. Reveal types or relations between concepts and try to make hierarchies and present them visually (trees).
3. Detail the concepts via top-down strategy and form meta-concepts via bottom-up strategy.
4. Exclude repetitions, synonyms, excessiveness and contradictions.
5. Split big trees into sub-trees and edit it.

This algorithm was used for developing several practical research ontologies [Gavrilova et al, 2005].

Then new approach to conduct research may be proposed:

- Problem-ontology definition
- Ontology of reviewed approaches
- Experiment framework design
- Data structure design and ontology development
- Mathematical modeling and main results ontology design
- Programme implementation (for CS)
- Structured writing up the findings

Conclusion

Ontological framework scaffolds the research activity. But ontological engineering is rather easy for «old» sciences with good structure. Researchers in new, multi-disciplinary and ill-structured disciplines of computer science as HCI, cognitive science, information management, etc. will face a bunch of difficulties in design and development phases. Ontologies also are rather subjective.

Our paper presents one of the first attempts to show the visionary role of ontologies. They are good for better self-understanding of research and then for knowledge sharing.

Acknowledgements.

The authors are grateful to Mathiew Koshy and Feng Guayan for some materials from their study.

Bibliography

1. Arpírez, J. C., Gómez-Pérez, A., Lozano, A. Pinto, H. S. (1998) (ONTO)2Agent: An ontology-based WWW broker to select ontologies. Workshop on Applications of Ontologies and Problem-Solving Methods. European Conference on Artificial Intelligence (ECAI'98). Brighton (United Kingdom). -pp.16-24.
2. Bernaras A., Laresgoiti I., Corera J. (1996) Building and Reusing Ontologies for Electrical Network Applications. Proceedings of the European Conference on Artificial Intelligence (ECAI96). ECAI 96.
3. Booch G. (1993) Object Oriented Analysis And Design With Applications. Pearson Education.
4. Borgo, S., Guarino, N., Masolo, C. (1996) Stratified Ontologies: the case of physical objects. In proceedings of the Workshop on Ontological Engineering. Held in conjunction with ECAI96. – pp. 5-15. Budapest.
5. Borst W.N. (1997) Construction of Engineering Ontologies. University of Twente. Enschede, NL-Centre for Telematica and Information Technology.
6. Chandrasekaran, B., Johnson, T. R., Benjamins, V. R. (1999) “Ontologies: what are they? why do we need them?”. IEEE Intelligent Systems and Their Applications. 14(1). Special Issue on ontologies. – pp. 20-26.
7. Fensel, D. (2001) "Ontologies: A silver bullet for Knowledge Management and electronic commerce", Springer.
8. Gavrilova T.A., Voinov A. (1996) Visualized Conceptual Structuring for Heterogeneous Knowledge Acquisition // Proceedings of Int. Conf. on Educational Multimedia and Hypermedia EDMEDIA'96, MIT, Boston, USA. - pp. 258-264.

9. Gavrilova, T., Feng, G. (2005) Ontology-based Framework to User Modelling in Decision Support // In Proceeding of Fifth Conference of International Society for Knowledge Organization ISKO, France. - pp 139-149
10. Gavrilova, T., Sosnovsky S. Development of Educational Ontology for C-Programming // Proceedings of XI-th Conference "Knowledge-Dialogue-Solution, KDS 2005", Varna, Bulgaria. - pp.127-131.
11. Gavrilova, T., Dicheva D., Sosnovsky S., Brusilovsky P. (2005) Ontological Web Portal for Educational Ontologies // In Proc. Of "Applications of Semantic Web Technologies for E-Learning Workshop (SW-EL'05)" in conjunction with 12th Int.Conf. on Artificial Intelligence in Education (AI-ED'05), Amsterdam. -pp 19-29.
12. Brusilovsky P., Yudelso M., Gavrilova T. (2005) Towards User Modeling Meta Ontology// in Lecture Notes in Artificial Intelligence (LNAI 3538) Ed. Ardissono L., Brna P. & Mitrovic A. Proceedings of 10th International conference User Modeling UM 2005, Springer. – pp.448-452.
13. Gavrilova, T., Laird, D. (2005) Practical Design Of Business Enterprise Ontologies // In Industrial Applications of Semantic Web (Eds. Bramer M. and Terzyan V., Springer,. - pp.61-81.
14. Gómez-Pérez, A., Fernández-López, M., Corcho, O. (2004) Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web Springer.
15. Guarino, N., Giaretta, P. (1995) Ontologies and knowledge bases. towards a terminological clarification". Toward Very Large Knowledge Bases. Ed. IOS Press. – pp.. 25-32.
16. Gruber, T. R. (1993) A translation approach to portable ontology specifications. Knowledge Acquisition.Vol. 5.
17. Gruber, T. R. (1995) Toward Principles of the Design of Ontologies Used for Knowledge Sharing. International Journal of Human Computer Studies. 43.
18. Fensel D., Musen M. (2001). The Semantic Web:A Brain for Humankind. IEEE intelligent, March.
19. Jonassen, D.H. (1998). Designing constructivist learning environments. In C.M. Reigeluth (Ed.), Instructional design models and strategies, 2nd ed. Mahwah, NJ: Lawrence Erlbaum.
20. Järvinen P. (2001) *On Research Methods*. Tampereen Yliopistopaino Oy, Juvenen-Print, Tampere.
21. Jenkins A.M. (1985) *Research Methodologies and MIS Research*. In Mumford et al, - pp.103-117.
22. Lenat, D.B., Guha, R.V. (1990) Building large knowledge-based systems. Addison-Wesley Publishing Company, Inc.
23. Mindmanager (2003) <http://www.mindjet.com>
24. Minsky M. (1974) A framework for representing knowledge", MIT-AI Laboratory memo 306,
25. Mizogushi, R. and Bourdeau J. (2000) Using Ontological Engineering to Overcome Common AI-ED Problems. International Journal of Artificial Intelligence in Education, volume 11, pp.1--12.
26. Norman D. (1993) Things that make us smart. Perseus Books., Cambridge, Massachusetts.
27. Noy, N. F. (2002) Ontology Development 101: A Guide to creating your First Ontology, Stanford University
28. Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T., Swartout, W.R. (1991) "Enabling technology for knowledge sharing". AI Magazine. 12(3). pp. 36-56.
29. Sowa, J. F. (1984) Conceptual Structures: Information Processing in Mind and Machine. Reading, Massachusetts, Addison-Wesley.
30. Studer R., Benjamins R., Fensel D. (1998) Knowledge Engineering: Principles and Methods". DKE 25 (1-2). - pp:161-197.
31. Swartout B.; R. Patil; K. Knight; T. Russ. (1997) Toward distributed use of large-scale ontologies. In AAAI-97 Spring Symposium Series on Ontological Engineering.
32. Uschold M., King M. (1995) Towards a Methodology for Building Ontologies. Workshop on Basic Ontological Issues in Knowledge Sharing..

33. Uschold M., Gruninger M. (1996) *Ontologies: Principles Methods and Applications*. Knowledge Engineering Review.
34. Wertheimer M. *Productive Thinking*. Enl.ed. L.: Ass. Book Publ., 1966. 302 p.
35. Yourdon, E. (1989) *Modern Structured Analysis*, Prentice-Hall Bristol, Computer Science Department.

Development of an ITS for self-instructing of Chinese hieroglyph

Liu Yong,
Moscow State University

1 Characteristics of the problem domain

Foreign language instruction delivered on computers is not new, but the enhancement of such system for instruction of the oldest language in the world — Chinese hieroglyphs, is still an open topic. Computer-assisted language learning (CALL) has been used in Chinese language instruction for some time with varying degrees of success [1]. This CALL approach to computer-delivered instruction generally uses predefined branching routines and static error feedback messages to guide learners. More adaptive, individualized instruction is feasible with AI techniques that model learner performance and deliver goal-based and remedial instruction to move the learner through the materials [2]. This kind of computer-based instruction is the basis for Intelligent Tutoring System (ITS), but it is widely accepted that there are still primary deficiencies of ITS in domains of Chinese character instruction [3][4], concretely, the weakness of pedagogical application according to Chinese hieroglyph system.

The domains of Chinese learning is fairly characteristic comparing with those that have been most used in recent ITS development (mathematics, physics, electronics). Those instructing subjects such as mathematics are more tractable and formally defined. But the instructing of Chinese language must include a certain amount of hieroglyphs, within which does not exist a clear hierarchy of relation as in nature science. The learners who study Chinese characters differ greatly in their acquisition order and way [5]. Those domain-independent tutoring systems can not satisfy the requirement of Chinese instructing. A set of domain-specific mechanisms are needed to organize teaching materials, make training strategies and manage individual learner's "learning information base".

The mastery of Chinese characters is one of the most difficult tasks in learning Chinese. The dictionary, *Hanyu Dazidian*, published by Hubei Cishu Publishing House (Hubei, China) in 1986, contains over fifty-six thousand characters. Of course, in general it is not necessary for us to know so many characters. The "Modern Chinese Commonly-Used Word List"², compiled in 1987, includes the most frequently-used 2,500 characters, as well as the second most frequently-used 1,000 characters. Thus it comprises 3,500 characters altogether. China ministry of education requires the student in senior middle school can recognized 2900 Chinese characters. So, a hard task lies in front of the learners who wants to mater even very simple Chinese.

Except the large number of characters, another reason of difficulties in learning Chinese is the lack of correlation between the forms and the pronunciations. Although there are character components indicating the pronunciation or meaning, the learner has to first grasp a certain number of basic components and characters before he can comprehend the rules and apply them to his study of new characters.

Aiming at smoothing these difficulties, we tried to provide beginners with a fast and efficient way to master 500 basic characters³. We are building a system for the Russian learners' self-instruction of Chinese characters on a personal computer.

2 Self-instructing system of Chinese hieroglyph

Our normal target user is person who is not assumed to have the available time for long-term academic course. Rather, he wishes to study the characters in a self-paced, extracurricular fashion, or to develop a

² Xiandai Hanyu Changyongzi Biao (Modern Chinese Commonly-Used Word List), compiled by the national language committee and national education committee of China in 1987.

³ The size of the character set can be configured by the teacher interface according to learner's background.

multifaceted view of them according to his interests. The user wants to grasp and to memorize characters' important properties, to initiate his own "learning information base" on them. Later he might gradually enlarge and interrelate his personal knowledge items.

The system's goal is twofold: the first part is to model an interactive information base on characteristic properties of Chinese characters, which would lend itself to personalized self-instruction in a tree exploration mode; and the second is to build the base on which we may subsequently explore an Intelligent Tutoring System architecture, which would take into account the expertise already gained by the learner over prior sessions.

An overall view of the system is as Figure 1. In Central objects in the information base (sketched by the ellipse) of the system are: the current standard information base on characters (which merges basic and special material on their properties, textual, ideographic, and sound data), and the learner's current profile (data he already accessed, record from session results, or a global report). The main functional handlers in the system are as follows: a learner and a teacher interface provide interactive environment, two schedulers act as organizer of exercises and analyzer of learning profile. The authoring module can be eliminated after the first configuring, so that the system could work as a single issued version of self-instruction system.

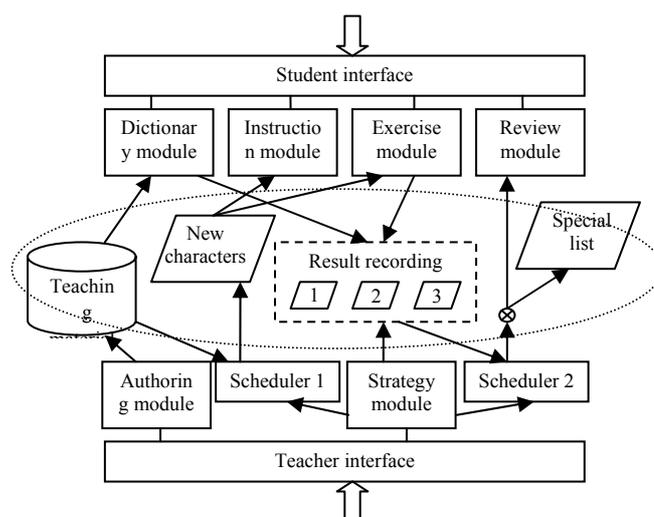


Figure 1 system architecture

The interactive information base provides the learner with basic universal properties of the characters (morphology, intrinsic meaning in Russian), extended with a quite comprehensive set of language aspects (phonetics, radical structure, contextual or pragmatic attributes). The dictionary module enables search or explore the characters with detailed properties, while instructing module controls the learning of characters grouped by scheduler-1. The results of exercises and dictionary searching (not all of them, only some ranked search results) are sent into learning-information records. For observing and then guiding the user's learning session, these records are divided into different levels according to the training strategy. In our system prototype, there are three level: level 1 – mastery, level 2 – with occasional errors, level 3 – not mastered or difficult to remember. The scheduler 2 works as an analyzer of session results and a controller of review content. For those characters which the learner can not remember after certain number of sessions (depend on the strategy), they are collected in a special list, in order to give the learner another way to study it (print it out, for example), and to avoid those characters block the learning recycle in system.

3 Conclusion and future work

The system mainly focuses on Chinese characters instruction and training, we see it as a prototype of a tutoring system of primary Chinese language, while the authoring function and the expert model are weak points.

A rewarding aspect lies in the scopes of future system developments: intelligent approach to the idea in the context of the discovery and learning of lexical properties of the Chinese characters in the learner's individual profile. We expect progress from patient observation and modeling of user activity, and from the availability of multiplatform software development tools, merging different classes of functions, heading towards polymorphic or multiple-view knowledge bases.

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Mr. Strogalov A. S., for all the guidance and constructive discussions in the work.

References

- [1] Li Jianguo, Zhang Xiaozhen, The Design and Implementation of Multimedia Intelligent Tutoring System for Chinese Characters // IEEE First International Conference on Multi-Media Engineering Education Proceedings, Volume 6, Issue 8, Jul 1994, Page(s):459 - 463
- [2] Yang Shanlin, GuTiejun, Ma Junxi, The Application of Chinese E-Learning Technology Standardization in Web-based Intelligent Tutoring System // Contemporary education technology, Issue No.04 2004
- [3] Yue Dongjian, Chai Peiqi. Features of Chinese Computer Aided Language Learning System. //Mini-Micro System. Vol.22 No.17 Beijing, July 2001.
- [4] Лю Юн, Развитие компьютерной вспомогательной системы по языковому обучению и концепт языкового обучения.// Естественные и технические науки, №5, М. 2005
- [5] 刘珣, 《对外汉语教育学引论》, 北京语言大学出版社, 2002 (Liu Xun, Principles of teaching Chinese language as a foreign language, Publish company of Beijing language and culture university, Beijing, 2002)

Application recognition of the bases of images

Mahmudova Shafaqat Jabrayil qizi

Managing sector

Institute of Information Technologies

of the National Academy of sciences of Azerbaijan

A31141, Baku, street. F.Agajeva 9

depart9@iit.ab.az shafag2@iit.ab.az

Informing and computerizing in the modern society is gradually getting greater range. By influencing to all processes in the society, computers change the activity form of man in general and enter into the new areas of the practice. The issue that people learn and apply new information technologies (computers, program provision, new types of mass information media) to different fields leads to creation of new modern systems.

Database, bank of database and knowledge, inquire-information and expert systems, as well as books and school supplies are the sources of information.

Bank of database in the modern society means information system consisting of the following elements.

1. Calculation System;
2. One or some databases;
3. Collection of applied programs (annex of database).

Knowledge base – is a collection that belongs to fields of some subjects and is formally submitted for implementation of the view on base of them.

Expert system – is a complex of computer program provision that assists in making decision that is based on the life of human. Information that is previously received from the best specialists of any fields is used in the expert systems.

Database – is the objective form of submission and organization of systematized collection of database. There are cross notifications among writings of different types of database. Database provides keeping of information and its quicker and more comfortable appeal to changing information.

Management system of database – consists of united language and program provision and is used for organization of base of given information, inserting and editing of writings and etc.

Management system of database includes the following language components:

1. Descriptive language of changing database, which defines the logical structure of changing database;
2. Operation on changing database – language that provides inserting, editing and choice;
3. The language that provides the organization of structured queries;
4. The language that provides query by example.

Systems of the knowing in the modern time are spreading mostly. It is difficult to show such a field of the science and the production activities that doesn't use them or will not apply them within the next years. Systems of the complicated big recognition in the modern time are technical means intended to find out the signs of the objects and to measure the parameters commenting them; Objects are the collection of algorithms of the recognition that turns the introduction information into certain exits.

Relation database is the total of attitudes keeping all information kept in the database, VB. But users will be able to accept such database as total of the table. It is used for information often undergone to changes from the database.

One of the basic superiorities of relation approach to the formation of the database (VB) is that users relation VB get opportunity to effectively work in lines and columns without need for knowledge in real formation of database in simple and clear notion terms of schedules and in foreign memory.

Some of stages of projecting of relation database have been described below:

- To define the objects included into database (source of database);
- To define the relations among objects;
- To define the main peculiarities of objects;
- To find out the relations existing among properties of objects;
- To define relations existing among schedules defined on base of relations kept within objects of database;
- By providing the integrity of database, to define the operations implemented during formation and changing of information existing in the schedule.
- To find out indexes that are necessary for accelerating the implementation of queries;
- To take security issues into consideration – to define authorities and users;
- To prepare procedures to create reserve copies and restore files.

Although the presented list is not full, it allows creating opinion about necessary procedures. Each step of the projecting depends on the results of previous projects, and as long as it is exact, there will need less iteration in future.

The following procedures are of more importance:

- To keep all necessary datum in database;
- To minimize the number of information about schedules kept in database;
- To normalize the schedules in order to facilitate the problems connected with upgrading and liquidation of datum.

Some of mentioned purposes are clear, but others are not clear fully. It doesn't take less importance. Let us look through each of them in a very short form. Combination operations happen in result of the issue that database are taken from different schedules in management system of relation database, new schedules are created on base of them and previously defined dependence are defined. Thus, they appear during implementation queries, but not during creation of schedules. Technology of the database multimedia, geological place of the free shaped text files of the information, HTML and XML and so on., keeping of documents and surrounding methods helping to search relation on considerably pulls off from the outside of frame of the VB. In this mean, management system of database organized for recognition of copies is of great importance.

It is possible to understand the recognition of the copies as information process. Thus, recognition is realized with the help of the recognition systems and intellectual information channels having entry and exit. Information about signs of the object represented in the exit is given in the system. Organization and exploitation of the automating system of recognition of copies is a very complicated issue. In organization of such systems, we may look thorough issues of different types. One of such issues is creation of descriptions base and recognition of data description that are popular with past period.

There have been done different works in the field of the recognition of the copies and creation of the descriptions base in the world. Some of famous management systems of descriptions base have been indicated in figure 2. For instance, thinness or thickness of contours of description, bending of the fulcrum lines and etc.

There are many algorithms for recognition and comparison of the description with descriptions existing in database. In this article, the creation of management system of descriptions base and recognition of the description have been considered from another aspect. The creation of the management system of the descriptions base consists of the following stages:

1. To create the descriptions base;
2. To read the descriptions base to the temporary field;
3. To define the classifiers for recognition of the descriptions;
4. To compare and recognize the descriptions;
5. To apply the fractal pressing methods to the descriptions.

Fragment method algorithm of recognition has been used in the management system of the descriptions base organized by us.

Descriptions base is usually organized on base of scanner descriptions. In order to effectively use the descriptions existing in the description base, they are read temporary field named ListBox in the form of list. In the temporary field, the description is kept in the graphical form and in the form of file, and diminished form of the description is kept in form of icon. Descriptions are usually organized on base of scanner descriptions. Logical structure of most descriptions is as the following:

- Name of the description;
- Index of the description;
- Colors degree of the description (black-white, gray);

- Maximum sizes;
- Common attributes;
- Size of the description.

It is possible to increase the number of the fields existing in the logical structure depending on the descriptions.

Files of the graphic type are included in the base in accordance with each writing. In this case, there is made appeal to graphic files when each new writing is inserted, and the necessary description is inserted in base. If there are many writings in database, to apply to graphic files every time leads to time loss. One of the rules to insert the descriptions in the database is to read the descriptions to ListBox of the computers. In this case, the little form of the description and the name of file is red to the same field. Descriptions to the database, is included in the list form from the framed area. In this case, the time is saved.

It is one of main questions to organize the classifiers pursuant to signs of the descriptions. Two forms of classification method are offered in the organization of classifiers:

Intentional. Characteristic feature of the Intentional methods consists of it that different properties of the recognition signs and their relations are used. Such elements, separate marks may be marks of interval of signs and relation matrixes of signs.

Extensional. Characteristic feature of the Extensional methods consists of it that free diagnostic mark of the each learnt object is given in the big or little measure. In essence, this method is near clinical approach. Thus, it looks at people as whole system, which each of people is individual and has a special diagnostic mark.

Different fragments of the descriptions are kept in the base pursuant to classifiers. When arbitrary description is given, this description is divided into certain fragments and is compared with fragments that conform to the classifier. The parts corresponding to the description are selected and a temporary description arises on base of this choice. After this, temporary descriptions are compared with basic descriptions and it is defined if the description is recognized or not. If the description is recognized, this recognition is defined with certain percentage. It is clear that the recognition cannot be 100%.

As a final stage, it is the possible to look at the issue that the descriptions are kept in base in the pressed form. In this case, it is possible to use different methods of pressing. And one of them is the fractal pressing method. Pressing of database is one of the most urgent matters, which appear when information is kept and transmitted on communication cannels. Let us define what information and database are. These two words are not synonyms. If information is everything that we can accept, then, databases are means for submitting, keeping and transmitting the information. It is the clear that it is possible to present any information in form of database. Besides, it is possible to submit the same information by means of database given in different numbers. The main problem in which the science engages is that information should be submitted in such a form that its volume comforting to database is minimal.

A model and implementation of universal engine for neural systems

Okulovskiy Yuri, *Ural State University*
Russia, Yekaterinburg, Lenin sreet, 51
e-mail: okulovsky@forgex.ru

Introduction

Neural networks are mathematical model of biological neural systems. Neuronetworks can be used in various areas – fuzzy logic, approximate solving of NP-hard tasks, dynamical optimization etc.

When developing application, which uses neural computations, developers have difficulties, because:

- Neuronetworks' development requires a specific theoretical knowledge;
- Neural system can hardly be debugged;
- Only one network is needed to solve particular task. However, to find which one is best, many of them should be tested, and hence, developed.

Therefore, demand to creation of a neuronetworks' library, which could be built in external application, exists. Many such libraries were developed. The most known is a Matlab's neural network package. Aside of this, many smaller projects exist and are available. The problem is they all have disadvantages, like:

- Impossibility to integration. There is no natural way to integrate them in some application, because they are not supposed to work with external program;

- Closed architecture. Most of solutions can only operate with a very particular neuronet's class (such as multi-layered perceptrons) and a strict set of algorithms. They cannot be extended, because their developers just have not considered this need;
- Language dependence. If even there is possibility to add components, the language for development is strictly limited;
- Documentation problems. If even there is many supported components, there is almost no way to find the required one;
- Errors. Small libraries without support team can work improperly or not work at all.

Our task was to develop and implement a neuronetworks' library without such disadvantages. Since there a lot of networks, learning algorithms etc, this library is supposed to be opened. This means that anyone can widen collection of components. There are, therefore, two groups of system's users. Firsts uses a system's core and develop new components. Seconds uses both core and component's library. To make the system popular, requirements of both these groups should be fitted.

Developers' requirements are supposed to be:

- Language independence. Anyone should have a possibility to write on customary programming language;
- Absence of expandability's limitations. There should be a way to add anything into a system;
- Developments' locality. Programmer should do only his job to write useful code. Components interaction, loading, saving and other routine must be done by system itself.

Users' requirements are as follows:

- Minimized amount of programming, where it is possible;
- Block architecture, which would allow to assembly system from components;
- Documentation system with possibility to search;
- A possibility to built a neural calculation into application without difficulties;
- Automatic tester, which would allow fast training and testing of systems.

Let us now introduce some definitions. Neuron is a basic calculation unit. It consists of several input channels and one output channel. With each input channel, its weight is associated. Neuron transforms input signals by applying non-linear function to their weighted total. Neuronet is a set of interconnected neurons. Its weights vector is a concatenation of neurons' weights vectors. Neuronetwork input vector is a set of signals to specified channels. Some neurons in networks are marked as output neurons, and their outputs forms network's output vector.

Neuronetworks can learn to give specified answers in response to input vector. A pair of input and answer is called learning sample. Learning is performed by changing networks' weights. We will call the rule that changes weights as "learning algorithm". It is not generally recognized definition, but it is more convenient for our purposes.

Learning algorithm requires a measure of neuronetwork's error. This measure is called a goal function. For more information about neural systems, see [1]

The set of neuronetwork, learning algorithm and goal function is called a neurosystem. Parts of neurosystem are components. If needed, other component's types may be developed and integrated into neurosystem. An algorithm, which handles neurosystems, is called an engine.

Engine, its functions and model

The first function of engine is an analyzing input vectors. This means a simple redirection of input vector to neuronetwork. Such functions as saving and loading neurosystem are trivial.

Learning can roughly be described as followed sequence of actions:

- (A) Take input vector, analyze it and the calculate goal function's value.
- (B) Run learning algorithm, and then apply its direction to network
- (C) Calculate goal function again

There must be said that goal function is typically calculated based on output vector and answer. However, sometimes answer vector depends from output. In that case, engine obtains some oracle, which calculate answer vector in stage (A).

Sequence above is an online learning algorithm. An offline version means that a set of samples is given, and after learning network must give a right answer to all the vectors from set. In that case, analogical sequence applies to samples' set.

Neurosystem should also story components (algorithm, network, etc) and provide its interaction. The main problem about interaction is compatibility problem. Assume that MLP is a multi-layered perceptron component; QG is a quickest descend algorithm's component. QG requires a gradient vector. It can invoke corresponded MLP method, which would run BPE algorithm, and then use its result. Now, someone has developed new network's component, for example, Elman's net (EN). EM theoretically can learn with QG. Therefore, the problem of connecting new component with existed ones appears.

Solution is a requirement for BPE method to have a standardized signature. If EM, MLP, or some other components have method with this signature, QD may invoke it and so they are compatible with QD. In most of modern programming languages, a natural way to provide such standardization is using interfaces. Interface can be considered as a set of signatures, which must be implemented in component if component implements interface.

Now we will study learning algorithms and goal functions more closely. Let us consider a followed learning algorithm:

$$\bar{q}(t) = \eta \cdot \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right) \quad (1)$$

$$\bar{m}(t) = \bar{q}(t) + \gamma \cdot \bar{q}(t-1) \quad (2)$$

$$\bar{r}(t) = -v \cdot (w_1, \dots, w_n) \quad (3)$$

$$d(t) = \bar{m}(t) + \bar{r}(t) \quad (4)$$

E is a goal function, w_1, \dots, w_n are weights, $d(t)$ is algorithm's direction. (1) is called a quickest descend rule, (2) is a moment rule, (3) is linear resurrection. It is clear, that both moment rule and resurrection can be included or excluded from algorithm. Therefore, any combination is possible. Development each combination as individual component would mean exponential growth of components count. It is avoided by using composite algorithms.

For example, sequential composition of two algorithms means that second algorithm should run after the first one and corrects its output. Composition of (1) and (2) is an example of such composition. Parallel composition means that both algorithms are to run independently, and then their directions are to add. Example is a combination of (1-2) and (3). Composition is formally an algorithm component, but all its functions are to run underlying algorithms and manipulate their outputs.

By using compositions, complex algorithms can be built from basic ones. Of course, performance will decrease. However, if some composite algorithm appears to be very good for particular task, it can always be rewritten as a monolith.

That example clearly shows that components should have a possibility to have other components inside them. Another example is goal function. In trivial case, goal function is a simple mathematical form of output and answer. Most often, it is a norm of their difference or its degree.

In offline learning, however, goal function must take into account not one pair of output and answer, but a several of them. More often, it is done by summarizing trivial function's values over learning set. Therefore, every mathematical form should have its multi-sample analog. This also is done by development a formal component, which would wrap underlying trivial function. When invoked, wrapper should call underlying function over samples and then sum results.

Other versions of wrapper are possible. For example, in online mode wrap can store samples and then use them as sample set. This behavior is called "pseudo-offline" mode. Old samples may be excluded from set, or their influence may decrease. Sometimes, there is a need to combine two different neurosystem. For example, BAM's output can be given to perceptron for further analysis. There is no general approach for such assembled systems to learn. However, it is useful to let system learn separately, and then use their combination only for analysis. That is easily done in assumption that neurosystem is formally a component, and therefore can be placed into another neurosystem.

Of course, assembled neurosystem has different behavior. However, from outside it does not differ from basic one. So, the signature of external method (such as analysis) is described in interfaces. That also allows developing new versions of engines, compatible with existed external applications.

As a platform for implementation, Microsoft .NET Framework was chosen. It is a platform with full and natural support of all object-oriented capability. Additionally, .NET Framework provides language integration and language independence. This means that development of components may be done in any language without restrictions. Moreover, .NET applications are independent from operational system and hardware. Finally, .NET supports a reflection mechanism, which allows solving some additional problems. For more information about .NET Framework and its advantages, see [2]

Extensibility

In this section, we will discuss possibilities to extension. As the main tool to provide extensibility, events were used. Events can roughly be described as sets of functions. When event is raised, all associated functions are invoked. By modifying event, external functionality can be added to component.

There are several points in learning process, where engine has alternatives. For example, if goal function's value is low enough after first evaluation, learning algorithm may be cancelled. On the contrary, if after learning value remains high, learning may be repeated.

A flexible way to provide these alternatives is using managing events. When engine is on the choice point, it raises event and pass Boolean parameter to it. Associated functions may set this parameter to constant or perform some more complex activity. Then engine continue execution with respect to value of parameter.

Another area to use events is a development of new component's types. Functionality of algorithms, goal functions and neuronetworks is implemented in described model. However, there may be need to add new component type inside neurosystem. For example, it can be component type, which erases connections in neuronetwork. Example is Optimal Brain Damage rule (OBD).

At first, neuronetwork must implement some interface, which provides methods to erase connections. Secondly, OBD must be launched during learning process. More often, it is launched after algorithm is done and its direction is applied. For this purpose, learning process is marked up with event as minutely as possible. So new components may catch these events and act.

Events also carry all the data about conditions of their raising. It will allow acting differently depending of these conditions.

The last thing to count is a problem with altering network's state. Networks' state is a data, which determines output in response to input. Usually network's state is unambiguously determined by networks' weights.

Components can change network's state during their work. Example is learning algorithm that tries a few directions' multipliers to choose the best. If these changes will not be rolled back, applying of direction is not correct. Moreover, some networks (like Elman's network) change their state even when analyzing vectors. Therefore, the system of control over changes and their rolling back was developed. When component wants to change networks' direction, a backup point is created. After necessary work is done, changes may be rolled back or committed. Committing changes is meaningless when learning algorithms are discussed, but it may be useful to new components types.

Since many components may change the state, there may be need to store a stack of backup points. Therefore, this possibility was also designed.

Associated problems

In addition to described model, some more features are being developed. The first is a documentation system. It would allow:

- 1) Mark a components' fields and methods with comments in a natural language inside program code;
- 2) Search among installed modules for a proper component;
- 3) Create a document (for example, XML-document) with a description of neurosystem, then create instances of all objects and assembly them properly
- 4) Create a description of existed neurosystem in natural language

This documentation system will allow creating neurosystems without writing program code. Reflection mechanism makes it easier to components' developers.

Second feature is automatic tester. Tester is an application, which safely handles learning process. It allows a user-friendly way to tune parameters, restart work after system's shutdown and other routine tasks. To make this tester work with particular problem, some methods (generating inputs and answers) must be developed.

The Bibliography

- [1] Simon Haykin. Neural Networks – a comprehensive foundation, 2nd edition // Prentice Hall, 1999
- [2] Hoang Lam, Thuan L. Thai. .NET Framework Essentials, 3rd edition // O'Reily, 2003.

Two-waves interaction in cylinder

Alexander Shermenev

Wave Research Center,

Russian Academy of Sciences,

Moscow, 117942, Russia,

E-mail: sher@orc.ru,

tel. +7(495)3918640

Computer algebra system is applied for studying the interaction of two acousical waves in a cylinder within quadratic approximation. When the cylinder coordinates are used, the usual perturbation techniques in separation of variables method inevitably leads to a series of overdetermined systems of linear algebraic equations for the unknown coefficients (in contrast with the Cartesian coordinates).

However, if we introduce formally a solution of the »rst system of this series, all these overdetermined systems become compatible for the special case of the nonlinear acoustical wave equation.

Using the added solution and quadratic polynomials of the Bessel functions of radius, we express explicitly the coefficients of the »rst two harmonics. It gives solutions describing two waves interaction which are found with the same accuracy as the nonlinear acoustical equation is derived.

This is a generalization of results given in [1], where it was proved that nonlinear monochromatic waves are described by quadratic polynomials of Bessel functions. The similar approach was used in [2] for describing the long periodic water waves on a slope in the high-order shallow water approximation.

The Bibliography

[1] Shermenev, A. Separation of variables for the nonlinear wave equation in cylindrical coordinates. *Physica D: Nonlinear Phenomena*, 212:3-4(2005) pp 205-215.

[2] Shermenev, A. 2003 *Nonlinear Periodic Waves in Shallow Water*. LNCS 2630, Springer-Verlag, pp 375-386.

Bitemporal Multi-level Secured Database for Intelligence Applications

Worawit Meanrach and Suphamit Chittayasothorn

Department of Computer Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology, Ladkrabang, Bangkok 10520, Thailand.

Multi-level secured (MLS) database is a security management which provides security clearance levels for both users and data. A traditional MLS user can read the data which is written by someone with a lower security level and can write so that someone else higher can read it as well. In this paper, an extension to the MLS database concept is applied so that users with lower clearance level can see the information which is written by higher-level users. This kind of information is considered as commands given to lower-level users by their higher ups. Bitemporal timestamp is also included at the tuple level. Since lower-levels user can see the commands and can write reports back to the higher-level ones, the database system is suitable for intelligence applications with several security clearance levels.

1 Introduction

In an intelligence application where users are classified into a hierarchy of security clearance levels and database objects are also classified accordingly, conventional databases need to be extended in order to satisfy this multi-level secured requirement. This kind of database is called multi-level secured (MLS) databases. Conventional classification levels include TS (Top Secret), S (Secret), C (classified) and U (Unclassified) [1,2]. The TS level is the top level such that $TS > S > C > U$. Each level may have several members. These members of the same level cannot see other same level users' data but can see all lower-level data. A typical user hierarchy is as shown in Fig. 1.

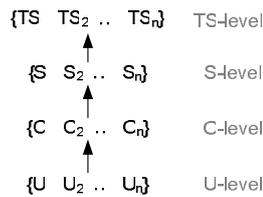


Fig.1 A user hierarchy for the MLS model

Most MLS databases are represented by the relational database model with some MLS-specific integrity rules. The most popular ones are those proposed by Bell and La Padula [3]. The first rule is called the simple security property. It states that a subject (user) can read the objects (data) if its clearance (classification) level is equal to or higher than those of the objects. The second rule is called the *-Property. It states that a subject can write an object only when its clearance level is the same or lower that the classification level of the object. In summary, a subject can read an object which is written by someone with a lower security level and can write/overwrite an object so that someone else higher can read it as well.

According to the above rules, it is clear that they are applicable to “report” type of applications. In some applications, the “command” type of information written by users with higher clearance levels for their supervisees that have lower clearance levels is required. We therefore introduce the new concept of commands. Fig.2 shows our required MLS database activities that include both report (pull) and command (cmd) activities.

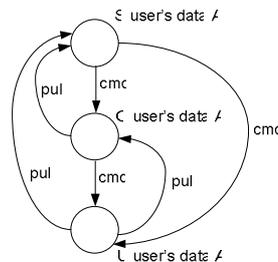


Fig 2. Report and command activities among adjacent clearance levels

In the case that lower-level data seem trustworthy, higher-level users may pull or up level data entered by lower-level users to be their own data. When the original data is later updated, the up-level one is also updated accordingly. This reflects the idea that all users can look at lower-level data and try to bring the most trustworthy one to their own level.

2 Temporal Database

A temporal database is a database that supports some aspects of time, not counting user-defined time (an uninterpreted attribute domain of date and time). There are three major categories based on the relational model namely the valid time state table, transaction time state table and bitemporal tables. Conventional database applications nowadays assume that all the facts which are kept in the database are true. This assumption follows the 2-value logic: "If found then true, if not found then false." If a fact is updated, the new fact replaces the old one. For example, John Doe decided to change his first name by the end of 2002 and became Peter Doe. After the update took place, there is no John Doe in the database. Any new reports and queries that refer to the person, even the ones that refer back to the data before the end of 2002, refer to him as Peter Doe.

A kind of temporal database table called "valid time state table" has been introduced. Valid time start and valid time end are attached to the end of each tuple of a relation. The insertion of each tuple to such a relation requires a timestamp of the valid time of the facts on the tuple. This insertion activity is trivial. The hard parts are delete and update operations.

A temporal delete operation is not the same as a conventional delete one. It may involve marking the old tuple to be valid up to the deletion date, insert a new tuple and delete the existing tuple. The actual operations required depend on the relationship between period of validity of the fact to be deleted and the period of applicability of the deletion. Temporal updates face similar challenges.

A transaction time state table is intended to be used for facts whose state is not supposed to be changed according to time. For example, the height of a mountain is assumed not to be changed with respect to human life. However, with better measurement technology, different values of the height are recorded for each measurement time. The transaction time state table keeps facts and the time those facts are recorded (or believed to be true).

Bitemporal database tables keep facts whose states can be changed in time (valid time) and the time that they are recorded. [4]

3 Multilevel Secure Bitemporal Data Model

Since the intelligence application requires reports from lower-level users to their supervisors and the facts that they report may change over time, temporal capability needs to be introduced to the database. Multilevel secure database with valid time was introduced by []. However, we found that not only the time that the facts are valid need to be recorded, but also the time that the lower-level agents report to the supervisor need to be recorded as well. The later is actually the time that the users believe that the facts are valid. For example, agent A may believe that John Smith was in London during the first half of 2005 and was in Paris during the second half of the same year. He believes so and reports to his supervisor on January 1, 2006. Later, he obtained new information and believes that John Smith was in Rome during the entire 2006 instead. He believes so and reports to his supervisors on March 20, 2006. A Multilevel Secure Bitemporal Data Model (MSBD) has been proposed in [5]. The general format of an MSBD relational scheme is as follows:

$$R(A_1, C_1, VT_1, TT_1, A_2, C_2, VT_2, TT_2, \dots, A_n, C_n, VT_n, TT_n, TC, CC)$$

Each attribute A_i has a classification level C_i . It shows the level of the user who originally enters the data. VT_i and TT_i are valid time and transaction time of each attribute A_i . TC is the tuple class which reflects the classification level of the tuple and CC is the classification level of the tuple creator. However, it is found that to preserve time consistency among valid time and transaction time of the attributes, it is more desirable to have only the timestamps of the tuple as follows:

$$R(A_1, C_1, A_2, C_2, \dots, A_n, C_n, VT, TT, TC, CC)$$

4.0 Illustrated Examples

CODENAME	CN_C	FULL NAME	FN_C	DEPARTMENT	D_C	VT_S	VT_E	TT_S	TT_E	TC	CC
conan	S1	JAME HENRY	S1	WATCH	S1	2005-02-01	9999-12-31	2005-01-05	2006-10-05	S1	S1
conan	S1	JAME HENRY	S1	ANTI NARCOTIC	S1	2005-02-01	2006-11-01	2006-10-05	9999-12-31	S1	S1
conan	S1	JAME HENRY	S1	ANTI NARCOTIC	S1	2006-11-01	9999-12-31	2006-10-05	9999-12-31	S1	S1
gin	C1	ADAM JONE	C1	WATCH	C1	2006-11-01	9999-12-31	2006-10-05	9999-12-31	C1	C1
cola	U1	JOHN SMITH	U1	WATCH	U1	2006-11-01	9999-12-31	2006-11-01	9999-12-31	U1	U1

Fig 3. A DETECTIVES relation instance

JOB_ID	JLC	CODENAME	CN_C	VT_S	VT_E	TT_S	TT_E	TC	CC
337	C1	gin	C1	2006-09-10	9999-12-31	2006-09-05	9999-12-31	C1	S1
337	C1	gin	C1	2006-09-10	9999-12-31	2006-09-06	9999-12-31	C1	C1
337	U1	cola	U1	2006-09-10	9999-12-31	2006-09-10	9999-12-31	U1	C1
337	U1	cola	U1	2006-09-10	9999-12-31	2006-09-10	9999-12-31	U1	U1

Fig 4. A JOB_DET relation instance

An intelligence application is developed on MSDB relations. Fig. 3 shows the DETECTIVE relation instance. Conan is a high-level agent with the classification level S. He was believed to work for the WATCH department since February 1, 2005 but is now believed to work for that department only until the end of October 2005 and works for the ANTI NARCOTIC department since November 1, 2005. Fig. 4 shows a job assignment relation. The S level agent Conan assigns the job 337 to gin by writing the first tuple of the relation shown. Gin who detects a command tuple (TC < TS) accepts the assignment and insert his own tuple (the second one). Gin then in turn, assigns the job to his lower-level agent Cola who also accepts the job. Operations on the database are complicated and require further elaboration.

References

- [1] Jukic N, Vrbsky SV. "Asserting Beliefs in MLS Relational Models". SIGMOD Record, Vol. 26, No. 3, pp. 30-35, 1997.
- [2] Jukic N., Vrbsky S., Parrish A., Dixon B, Jukic B. "A Belief-Consistent Multilevel Secure Relational Data Model". Information Systems, Vol. 24, No. 5, pp. 377-402, 1999.
- [3] Bell, D. E. and LaPadula, L. J. "Secure Computer Systems: Unified Exposition and Multics Interpretation." MTR-2997, MITRE (1975).
- [4] R. Snodgrass. "Managing Temporal Data A Five-Part Series". Timecenter Technical Report TR28, Department of Computer Science, University of Arizona, 1998

Применение генетических алгоритмов для анализа криптографических протоколов

Айдаров Юрий Рафаэлевич,

ст. пр. кафедры процессов управления и информационной безопасности

Пермского государственного университета

Адрес: 614051, Пермь, ул. Звонарева, д. 3, кв. 69

E-mail: aidar@psu.ru

В наши дни все большую распространенность получают различные формальные модели описания криптографических протоколов, предполагающие дальнейший автоматизированный анализ с помощью некоторых алгоритмов. В первую очередь это связано с тем, что самые современные методы шифрования не гарантируют соответствующего уровня защиты, если протокол обмена информацией, в первую очередь ключами шифрования, содержит возможности для несанкционированного доступа.

Одним из наиболее эффективных методов моделирования протоколов являются графовые структуры. [1, 2] Вершины графа при этом соответствуют некоторым состояниям информационного обмена. Следовательно, анализ сводится к поиску некоторого пути в графе, который приводит в незащищенное состояние. На основе найденного пути строится модель атаки на исследуемый протокол.

В системе Murphi[1], разработанной в Стэнфордском университете, реализованы две стратегии поиска: поиск в ширину и поиск в глубину. Инструмент вероятностной проверки моделей Prism[2], созданный в Бирмингемском университете, выбирает очередную вершину на основе случайной величины. При этом веса вершин, соответствующие вероятности выбора вершины, определяются с помощью применения правил специальной логики (продукций).

Оценку эффективности каждого из методов можно производить по общему количеству рассмотренных вершин (состояний) и максимальному суммарному объему памяти, который был использован для хранения текущей информации о переборе.

Система Murphi соответствует использованию стека (поиск в глубину) и очереди (поиск в ширину) для хранения вершин графа в процессе обхода. В системе Prism реализован своеобразный поиск в глубину, в ходе которого хранится определенная информация о вероятности перехода к некоторым другим вершинам графа.

Автор предлагает использовать для обхода графа состояний поиск пути в графе на основе очереди с приоритетами. При этом с одной стороны количество памяти, которое требуется для хранения состояний больше чем для поиска в глубину, с другой стороны меньше чем для поиска в ширину. Эксперименты показывают, что скорость подобного поиска не меньше, чем у ставших уже традиционными методов поиска в ширину и в глубину. Существенным преимуществом использования очередей с приоритетами является дополнительная возможность применения методов эвристического поиска, например, генетических алгоритмов (ГА). Основной проблемой, которая усложняет применение ГА для решения задач анализа криптографических протоколов, является необходимость скрещивания исходных решений таким образом, чтобы в результате получалось корректное решение задачи. Предлагается решить этот вопрос за счет изменения в ходе работы ГА приоритетов вершин графа в очереди, что открывает возможности для использования параллельных вычислений в процессе обхода.

В ходе проведенных экспериментов автору удалось показать эффективность использования очередей с приоритетами совместно с ГА. При этом в качестве функции оценивания ГА использовалось количество новых для злоумышленника фактов, которые могут быть получены с помощью взаимодействия с моделью, хотя окончательно вопрос выбора функции оценивания не закрыт.

Ссылки

1. <http://verify.stanford.edu/dill/murphi.html>
2. <http://www.cs.bham.ac.uk/~dxp/prism/index.html>

Программное обеспечение для инвестиционного бизнес-проектирования

Альханагтах О. Ж.,

Белорусский государственный университет,

22002 Минск, ул. Радужная, д.8/2, кв. 215

e-mail: omarhanaqtah@yahoo.com

Автор данной статьи рассматривает вопросы эффективности внедрения организациями программного обеспечения для управления инвестиционными проектами с учетом мирового опыта и статистики крупнейших мировых консалтинговых компаний. В статье освещаются вопросы критериев принятия решений о выборе и применении программного обеспечения, а также приводится обзор отечественных и зарубежных программных продуктов для расчета и управления инвестиционными проектами.

Ключевые слова: программное обеспечение, инвестиционный проект, анализ проекта, система управления проектами, эффективность использования программного обеспечения, бизнес-процессы, корпоративная система управления проектами, глобальные иерархические структуры, интеграция, системы высшего/низшего класса, функции систем планирования и управления, работы проекта, ресурсы и затраты, назначение ресурсов, команда проекта, контроль за ходом выполнения проекта, отчеты по проекту, рынок программного обеспечения, рейтинговые оценки, Gantt, PERT, GAAP, IAS, UNIDO, COMFAR, PROPSPIN, Microsoft Project, Project Expert, Spider Project, Primavera Systems, Open Plan, Альт-Инвест, Инвестор, Аналитик.

Значительный выигрыш во времени реализации инвестиционного проекта образовался от применения точных математических методов в управлении сложными комплексами работ, что стало возможным благодаря развитию вычислительной техники. Применение в компании формализованных методов управления проектами позволяет более обоснованно определять цели инвестиций и планировать инвестиционную деятельность, более полно учитывать проектные риски, оптимизировать использование имеющихся ресурсов и избегать конфликтных ситуаций, контролировать исполнение составленного плана, анализировать фактические показатели и вносить своевременную коррекцию в ход работ, накапливать, анализировать и использовать в дальнейшем опыт реализованных проектов.

В настоящее время в США уже сложились глубокие традиции использования систем управления проектами во многих областях жизнедеятельности. Причем, основную долю среди планируемых проектов составляют небольшие по размерам проекты. Например, исследования, проведенные еженедельником «Infoworld» (США), показали, что 50 % - ти пользователей в США требуются системы, позволяющие поддерживать планы, состоящие из 500-1000 работ, и только 28 % пользователей разрабатывают расписания, содержащие более 1000 работ. Что касается ресурсов, то 38 % пользователей приходится управлять 50-100 видами ресурсов в рамках проекта, и только 28 % пользователей требуется контролировать более чем 100 видов ресурсов. В результате исследования были определены также средние размеры расписаний проектов: для малых проектов – 81 работа и 14 видов ресурсов, для средних – 417 работ и 47 видов ресурсов, для крупных проектов – 1198 работ и 165 видов ресурсов. Данные критерии могут служить отправной точкой для менеджера, обдумывающего полезность перехода на проектную форму управления деятельностью собственной организации. Как видно, применение системы управления проектами может быть эффективным и для очень небольших проектов.

Необходимость использования программного обеспечения для управления инвестиционными проектами обоснована тем, что, согласно исследованиям, проведенным Standish Group, в странах США и западной Европы около 90 % проектов не были успешно выполнены из-за перерасхода средств или отклонения от сроков. Более того, 33 % проектов были прекращены до стадии завершения. Примерно 1/3 из проектов, выполняемых мелкими, средними и крупными компаниями, превысила бюджет в среднем на 150-200 %. Более чем 1/3 из обследуемых компаний сообщали о превышении сроков выполнения проектов на 200-300 %. На неудачные, в том числе нереализованные, проекты в мире тратится около 145 миллионов долларов в год.

По данным консалтинговой фирмы Interthink (США) в настоящее время подавляющее большинство компаний США и Канады используют формализованные подходы к управлению проектами. Статистика использования компаниями методов управления проектами выглядит так:

- 1) в компании не используются формализованные подходы управления проектами (2,4 %);
- 2) методы управления проектами используются в проектах разработки новых продуктов (60,1 %);
- 3) методы управления проектами используются в управлении внутренними проектами (55,7 %);
- 4) методы управления проектами используются в капитальных проектах (42,7 %);
- 5) методы управления проектами используются при работе с заказчиками (47 %);
- 6) в компании внедрен полный проектно-ориентированный подход для проектов во всех областях (22,5 %).

В США и европейских странах проводятся крупномасштабные исследования эффективности использования программного обеспечения для управления проектами. Обзор, проведенный Институтом управления проектами США (PMI), включает данные, полученные более чем от 100 североамериканских компаний и профессионалов в области управления проектами. Результаты эффективности использования компаниями США программного обеспечения для управления проектами выглядят так:

- 1) не эффективно (1 %);
- 2) низкий уровень эффективности (5 %);
- 3) приемлемо (15 %);
- 4) эффективно (41 %);
- 5) очень эффективно (38 %).

Также по результатам обзора были получены следующие результаты: специалисты различных отраслей США сошлись во мнении, что прирост эффективности при использовании информационных систем управления проектами составляет 21 % по отношению к показателям компаний, не использующих в своей деятельности информационные системы управления проектами. При оценке прироста эффективности в данном исследовании использовалась модель оценки PMI (PM Value Model), которая предполагает рассмотрение прироста эффективности процессов управления проектами по следующим разделам: финансовые параметры, процедурные и организационные параметры, рост эффективности и обучение персонала, работа с клиентами. Как показывает опыт США, Канады, европейских стран внедрение программного обеспечения позволяет существенно повысить эффективность реализации проектов компании.

Естественно, что с расширением круга пользователей систем проектного менеджмента происходит расширение методов и приемов их использования. Зарубежные компьютерные журналы регулярно публикуют статьи, посвященные системам для управления проектами, включающие советы пользователям таких систем и анализ использования методики сетевого планирования для решения задач в различных сферах управления.

Постоянное совершенствование методов управления организацией, подкрепляемое использованием современного программного обеспечения является условием успешного функционирования компании на рынке. Одной из задач руководителя стало использование прогресса в информационных технологиях, чтобы не отстать от конкурентов. Однако на передний план выходят задачи выбора, эффективного внедрения и использования подобных систем. Программное обеспечение для управления проектами характеризуется несколькими признаками, которые, по сути, являются требованиями, определяющими его эффективное применение в организации.

1. *Распределение функций системы по ролям в проекте.* Поскольку в реализацию проектов вовлечено множество сотрудников организации, поэтому корпоративная система управления проектами (Enterprise Project Management – EPM) не может состоять из одного программного продукта «на все случаи жизни». Пользователям, находящимся в разных подразделениях на разных уровнях управления, для выполнения своих обязанностей в любой момент времени должна быть доступна актуализированная информация о проекте. Так, например, высшему руководству требуются удобные формы представления информации о портфеле проектов, отчеты по отклонениям, дающие возможность проследить за ходом выполнения отдельных проектов, соблюдением бюджетных и временных ограничений. Аналитикам требуются средства для моделирования возможных рисков и вероятных сценариев развития многих проектов в комплексе, с учетом их взаимного влияния друг на друга.

2. *Поддержка глобальных иерархических структур.* EPM-система должна обеспечивать планирование, анализ и контроль всех проектов компании в рамках единой структуры или иерархии проектов. Иерархический подход к управлению проектами позволяет формировать бюджеты на любом уровне детализации.

3. *Контроль одновременного использования ограниченных ресурсов компании во многих проектах.* Рабочее время сотрудников компании – важнейший экономический ресурс. EPM-система призвана помогать руководству принимать решения о том, кто и в каком проекте будет участвовать. Она дает возможность планировать ресурсы на основе ролей (специальностей) задолго до того, как будут известны конкретные исполнители работ.

4. *Обеспечение обмена информацией внутри команды проектов в режиме реального времени.* Лица, принимающие решения должны обладать всей необходимой им информацией для принятия адекватных решений в кратчайшие сроки. Так, руководители проектов, имея свободный доступ к актуальной информации, могут оценивать различные варианты развития событий и выбирать наиболее оптимальный с точки зрения сроков, трудозатрат и стоимости.

5. *Наглядное представление информации о состоянии проектов.* Применение программного обеспечения способствует повышению прозрачности проектов в масштабах всей организации, что непременно ведет к повышению вероятности успешной реализации проектов. Кроме того, программные продукты содержат инструменты для анализа взаимного влияния проектов. Информация представляется в удобной и понятной всей форме, таким образом, максимальное влияние уделяется решению конкретных задач для успешного достижения целей проекта.

6. *Готовность к интеграции.* ЕРМ-система должна иметь возможность интеграции с финансовыми и другими информационными системами предприятия. Интеграция может осуществляться на основе как стандартных, так и специально создаваемых модулей. Вопросы интеграции играют важную роль в обеспечении качества передачи информации в компании. Как только происходят какие-либо изменения в проектах, специалисты всех подразделений компании должны иметь возможность сразу же увидеть последствия этих изменений.

К решению о выборе программного обеспечения для управления проектами в различных организациях приходят разными путями. Самый короткий путь, как правило, связан с решением, основанным лишь на рекламной информации продавца системы. Определенную помощь в выборе системы могут оказать сравнительные обзоры программного обеспечения, публикуемые в компьютерных изданиях. Однако необходимо иметь в виду, что в обзорах, публикуемых в компьютерных журналах, преимущественно оценивается мощность и изящность реализации функций пакетов с программистской точки зрения. Вопросы же управленческой эффективности применения пакетов в той или иной ситуации отходят на второй план.

Западные обзоры программного обеспечения для управления проектами традиционно разделяют программы, доступные на рынке, на две широкие группы: системы высшего класса (high-end) и системы низшего класса (midrange). При попытке ответить на вопрос о критериях классификации систем выясняется, что основным признаком для отнесения программы к тому или иному классу является ее цена. Если система продается по цене ниже \$ 1000, то она автоматически попадает в разряд систем низшего класса. Если же установленная на систему цена превышает \$ 1000 (а лучше, чтобы она была выше \$ 3000), то система имеет все шансы быть отнесенной к высшему классу. Развитие информационных технологий последних лет практически свело на нет различия между системами по объемным показателям мощности систем (размеры планируемого проекта по работам и ресурсам, скорость пересчета проекта). Даже дешевые пакеты сегодня способны поддерживать планирование проектов, состоящих из десятков тысяч задач и использующих тысячи видов ресурсов. Изучая матрицы сравнения основных функций систем, также достаточно трудно найти существенные пробелы в той или иной системе. Выявить отличия в реализации отдельных функций часто удается лишь при детальном изучении и тестировании системы.

Выбор программного обеспечения совершается на основе составления перечня задач, для решения которых требуется система управления проектами. Необходим анализ характера деятельности организации с точки зрения возможности и целесообразности применения проектной формы планирования и управления: какая деятельность может планироваться в виде проектов, насколько детально необходимо планировать и контролировать проекты. Для укрупненного описания и анализа проекта на прединвестиционной стадии в большей степени подходит специализированное программное обеспечение анализа проектов, которое позволяет выполнить оценки основных показателей рентабельности проекта в целом и обосновать эффективность капиталовложений.

Примером системы для анализа проектов является хорошо известная на российском рынке программа Project Expert. Необходимо отметить, что для описания плана инвестиций в Project Expert используются традиционные подходы сетевого планирования, предполагающие разбиение проекта на комплекс взаимозависимых задач и описание требуемых для их выполнения ресурсов. В Project Expert реализованы Gantt и PERT диаграммы. Однако если управление проектами в организации не завершается обоснованием инвестиций и существует потребность в контроле за ходом реализации проекта, то необходимо переходить к программному обеспечению управления проектами.

Успешная разработка технико-экономического обоснования еще не означает успеха всего проекта. Даже весьма перспективный проект может потерпеть неудачу из-за ошибок планирования и реализации. Для выбора пакета необходимо определение перечня реализованных функций планирования и управления:

- только планирование или планирование и контроль хода проекта;
- планирование и контроль лишь сроков выполнения работ;

- планирование и контроль финансовых вложений без детального планирования использования ресурсов;
- детальное планирование использования ресурсов;
- многопроектное управление.

Необходимо определить примерные требования к размерности проектов и детальности планирования, организационной структуре управления и отчетности: сколько проектов будет вестись одновременно и будут ли они взаимозависимыми, каково примерное количество задач в одном проекте, сколько видов ресурсов будет задействовано в одном проекте, и как будут разделяться ресурсы между проектами. Кроме того, на выбор пакета влияют специфические требования управления в конкретной предметной области. Например, специальные требования к отчетности или необходимость расчета дополнительных показателей, необходимость интеграции системы с другими приложениями или нормативными базами данных и т.п.

Рассмотрим основные функции систем календарного планирования.

Средства описания комплекса работ проекта, связей между работами и их временных характеристик.

Средства описания и типы планирования задач (выполнить как можно раньше, как можно позже, работы с фиксированной датой начала/окончания, возможность привязки длительностей задач к объему назначенных ресурсов, вычисляемые резервы времени (полный, свободный) и т.д.). Средства установки логических связей между задачами. Многоуровневое представление проекта. Поддержка календаря проекта, поддержка календарных ресурсов.

Средства поддержки информации о ресурсах и затратах по проекту и назначения ресурсов и затрат отдельным работам проекта. Ведение списка наличных ресурсов, возможность задания нормального и максимального объемов работ. Поддержка ресурсов с фиксированной стоимостью и ресурсов, стоимость которых зависит от длительности их использования. Расчет требуемых объемов ресурсов. Ресурсное планирование (выделение перегруженных ресурсов и использующих их задач, автоматическое/командное выравнивание профилей загрузки ресурсов (с учетом ограничений по времени или с учетом ограничения на ресурс, с учетом приоритетов задач).

Средства контроля за ходом выполнения проекта. Средства отслеживания состояния задач проекта (фиксация плана расписания проекта, средства ввода фактических показателей состояния задач (процент завершения)). Средства контроля за фактическим использованием ресурсов (бюджетное количество и стоимость ресурса, фактическое количество и стоимость ресурса, количество и стоимость ресурсов, требуемых для завершения работы).

Графические средства представления структуры проекта, средства создания различных отчетов по проекту. Диаграмма Гантта (часто совмещенная с электронной таблицей и позволяющая отображать различную дополнительную информацию), PERT диаграмма (сетевая диаграмма). Средства создания необходимых для планирования отчетов (отчет по состоянию выполнения расписания, отчеты по ресурсам и по назначению ресурсов, профиль ресурса, отчет по стоимости).

Классические системы календарного планирования в последнее время дополняются программными продуктами, которые позволяют:

- добавить или улучшить отдельные функции управления проектами, например, анализ рисков, учет рабочего времени исполнителей, расчет расписания при ограниченных ресурсах;
- интегрировать системы управления проектами в корпоративные управленческие системы;
- настроить универсальное программное обеспечение на специфику управления проектами в конкретной предметной области (например, интеграция со сметными системами для строительных проектов).

Хотелось бы отметить, что хотя выбор системы должен быть оптимальным, не стоит забывать, что программная система управления проектами – это лишь мощный инструмент, позволяющий если не исключить, то сильно уменьшить количество рутинных операций. Ниже рассмотрим признанные компьютерные программы по управлению инвестиционными проектами.

Для того, чтобы самостоятельно рассчитать и проанализировать инвестиционный проект, провести анализ финансово-хозяйственной деятельности для оценки инвестиционной привлекательности своего предприятия с помощью собственных специалистов, необходимо их обучение современным методам инвестиционного проектирования и использование специальных компьютерных программ. Инвестиционное проектирование, и тем более анализ, - процесс весьма трудоемкий, а также требующий специальных знаний не только в сфере экономики, а также тот факт, что инвестиционное проектирование с учетом новых реалий и требований времени – вопрос малоизученный, и породило множество программных пакетов для инвестиционного проектирования и анализа. Принцип работы данных программ примерно одинаков. Сначала вводится набор параметров. В результате расчета получается полный финансовый отчет, данные

которого можно исследовать с помощью присутствующих здесь аналитических инструментов. В качестве исходных данных используется информация о производственной программе, маркетинговый план, схема финансирования проекта. Методики анализа и общие подходы к расчетам примерно одинаковы, однако есть и особенности, отличающие различные программы друг от друга.

Хотелось бы отметить, что интерес к методам планирования и управления проектами год от года все возрастает. Это обусловлено, во-первых, тем, что методология управления проектами позволяет реализовать подход к проекту как к системе, предусматривающей не только выработку решений по развитию потенциала системы, но и обеспечению их осуществления. Во-вторых, позволяет при разработке сложных систем различного назначения эффективно распоряжаться выделенными на реализацию проекта ресурсами, учитывая при этом такие факторы, как затраты, стоимость, прибыль, риск.

Сегодня на российском и белорусском рынках существует около десятка компьютерных программ для расчета и сравнительного анализа инвестиционных проектов, как отечественных, так и зарубежных. К таким системам относятся пакеты COMFAR и PROPSPIN, созданные в UNIDO – ООН по промышленному развитию, а также российские пакеты, такие как Project Expert, Аль-Инвест, Инвестор, и др. Рассмотрим некоторые аспекты указанных программ.

Системы COMFAR и PROPSPIN разработаны UNIDO и прошли международную сертификацию. В основу данных систем положена методика, которая подходит для обоснования не только новых инвестиций, но и для разработки проектов по финансово-экономическому и техническому оздоровлению компаний, их расширению и модернизации. Благодаря объективным возможностям, имеющимся у UNIDO в области распространения своих изданий, эта методика стала принятым во многих странах стандартом по оценке эффективности инвестиций. В ней рассматриваются проблемы маркетинга, выбора месторасположения проекта, сырья, технологии и оборудования, поднимаются вопросы анализа трудовых ресурсов и воздействия капиталовложений на окружающую среду. В то же время, учитывая достоинства данной методики (в частности, детально разработанная информационная база технико-экономического обоснования), следует отметить, что отдельным аспектам анализа инвестиций (например, комплексной оценки разнообразных критериев эффективности проекта, анализа проектного риска и дифференцированного воздействия инфляции) не уделено достаточно внимания, а проблемы оценки систематического риска, распределения ограниченных средств финансирования среди альтернативных возможностей, технологии оперативного анализа и аудита вовсе не исследованы.

Система COMFAR (Computer Model for Feasibility and Reporting) применяется в настоящее время в ряде российских финансовых институтов для финансового анализа и оценки экономической эффективности инвестиционных проектов, ориентированных, в первую очередь, на привлечение иностранных инвестиций. Последнее объясняется тем, что результаты расчетов эффективности и представление выходной информации в системе COMFAR соответствуют принятым международным стандартам, что, в свою очередь, необходимо для успешного проведения переговоров с иностранными инвесторами. Недостатки системы:

- несоответствие налогового блока отечественным условиям налогообложения;
- принятый в системе годичный шаг расчета (в период строительства возможно полгода). Если же необходим меньший шаг расчета (а это случается часто), необходимо применять специальные меры;
- жесткая заданность перечня исходных данных при ограниченности их количества. Если проект предусматривает большое число видов оборудования (например, два и более видов основного оборудования), то данные о нем приходится агрегировать, что приводит к неточностям в учете амортизации (т.к. по агрегированному оборудованию ее приходится вычислять вне системы). Также существуют трудности и по вводу некапитализируемых затрат;
- трудность учета инфляции. Даже для постоянного уровня инфляции в системе не предусмотрено текущее изменение коэффициентов пересчета иностранной и местной валют. Если же необходимо учитывать переменную во времени инфляцию, текущие цены и затраты приходится задавать извне;
- невозможность модификации пользователем формул («закрытый» характер пакета);
- исходная информация для проведения расчетов не соответствует структуре калькуляции затрат и данных отчетности на отечественных предприятиях. Самостоятельный пересчет данных (вне данной программы) – процедура довольно трудоемкая и требующая специальной подготовки аналитиков. Вместе с тем вполне возможна разработка специальной программы, обеспечивающей необходимый пересчет данных.

Таким образом, хотя программа COMFAR является неким общепризнанным международным эталоном, ее цена и функциональные характеристики вряд ли привлекут к ней внимание наших пользователей.

PROPSPIN ((PRO)ject Profile Screening and Pre-appraisal Information System) предназначен для: формулирования инвестиционного проекта; исследования последствий изменения выбранных параметров; подготовки двух или более сценариев, основанных на различных предположениях относительно перспектив проекта. Отличительной чертой PROPSPIN является интегрированность. Это означает, что пользователь

одновременно видит на экране и входные данные (возмущающие воздействия), и их финансовые последствия. Отчет PROSPIN представляет собой законченный вариант финансового профиля проекта с учетом заданных ограничений. Однако пакет не является средством проведения полного финансового анализа, а служит для быстрого просмотра различных вариантов (для выявления их), которые будут пригодны для дальнейшего рассмотрения. Большим недостатком пакета является отсутствие учета фактора инфляции, а также отсутствие влияния рыночных факторов на цену и объем выпускаемой продукции, ресурсы и т.д.

В отличие от пакета COMFAR пакет *Аль-Инвест* имеет: налоговый блок, полностью соответствующий российским условиям; возможность настраивать блоки входных данных на условия, соответствующие реальной ситуации (налоги, показатели инфляции, объем экспорта и условия реализации продукции на внутреннем и внешнем рынках (в кредит, с авансовыми платежами), задержки платежей). Свободное владение пакетом требует от пользователя изучения принципов работы с электронными таблицами, т.к. случайные непрофессиональные корректировки пользователя могут привести к серьезным нарушениям работы системы. Недостатком данного продукта также является выдача одного и того же набора выходных показателей для инвесторов и других участников проекта.

Программа *Инвестор* отличается проработанностью методики, привязкой к российскому законодательству и российскими стандартами анализа. Конечно, финансовый анализ, будь то российский или международный, имеет общие принципы, но терминология формы отчетности и многие другие детали взяты из российской, а не международной практики. Однако в функции блока финансового анализа входит отображение российских стандартных форм бухгалтерской отчетности в международных стандартах GAAP и IAS. Недостатком программы является непригодность работы с иностранными инвесторами. Отдельно следует отметить, что в последнее время функционал данной программы не развивается компанией-разработчиком, а он включен в последние версии другой системы компании-разработчика – Аналитик.

Project Expert – наиболее распространенная в России программа («закрытого» типа), фактически сочетающая в себе системы инвестиционного анализа и управления проектами. Пакет позволяет системно производить качественный анализ проекта по 40 позициям (реальность концепции проекта, качественные показатели, общественная значимость, рыночный потенциал и т.д.). Расчет показателей проекта производится в соответствии с методикой UNIDO, но при сравнительно больших возможностях задания условий реализации. Ответы экспертов оцениваются по пятибалльной шкале, после чего оценки суммируются. Как правило, если экспертов несколько, то вычисляется средняя оценка. Анализ рисков выполняется по 71 позиции, охватывающей 11 стадий проектов. Здесь эксперт выбирает лишь уровень риска – высокий, средний, низкий по каждой позиции и формулирует соответствующие комментарии по выбранному уровню. Использование балльных или вероятностных оценок не предусмотрено. Непредусмотрены также поправки на риск (увеличение требуемой нормы дисконта). Как модуль качественного анализа, так и модуль анализа рисков являются открытыми для пользователя. Пользователь может сформировать свой опросный лист (в соответствии со стоящими перед ним задачами) для качественного анализа проекта. Он может также самостоятельно сформировать факторы риска и комментарии к ним в соответствии со специфическими особенностями проекта. Определенным недостатком программы можно считать отсутствие возможности создавать варианты проекта на любой стадии его разработки и проводить сравнительный анализ по всему набору показателей, имеющихся в программе. Также графическое представление информации в этом пакете является более слабым, чем в пакетах COMFAR или Альт-Инвест.

Open Plan – полностью русифицированная система планирования и контроля крупных проектов и программ. Основные отличия системы: мощные средства ресурсного и стоимостного планирования, эффективная организация многопользовательской работы и возможность создания открытого, масштабируемого решения для всего предприятия. *Open Plan* поставляется в двух вариантах – Professional и Desktop – каждый из которых отвечает различным потребностям исполнителей, менеджеров и других участников проекта.

Primavera Systems является признанным лидером на рынке программного обеспечения для управления проектами. *Primavera Enterprise*, программное обеспечение, построенное по модульному принципу, позволяет создавать целостную корпоративную систему управления проектами. Это инструмент, предназначенный для многопроектного планирования и контроля всех проектов в организации, включая бюджетирование проектов, календарно-сетевое планирование проектов, управление ресурсами, анализ проектов и портфелей проектов по показателям, управление рисками в проекте и т.д. *Primavera Enterprise* – это масштабируемое решение, настраиваемое в соответствии с индивидуальными потребностями и задачами каждого участника проекта. Форма представления информации в *Primavera Enterprise* настраивается в соответствии с единым корпоративным стандартом, что позволяет каждому участнику – от руководителя

проекта до исполнителей на местах – сосредоточиться именно на его направлении деятельности и существенно облегчает процесс принятия решений. Primavera Project Planner (P3) используется для управления проектами в области строительства, машиностроения, архитектуры, IT и телекоммуникаций. P3 – признанный стандарт среди высокопроизводительного программного обеспечения, применяемого для календарно-сетевого планирования, ресурсного и стоимостного анализа. P3 разработана для крупномасштабных и многоплановых проектов. Работая с проектами, содержащими до 100000 работ, P3 не имеет ограничений по ресурсам и количеству целевых планов. Сетевая версия P3 может использоваться для управления как сложными многоуровневыми иерархическими проектами, так и комплексами распределения проектов. Однопользовательская версия предназначена для работы с небольшими проектами на уровне высшего звена управления. Кроме P3, компанией Primavera Systems поставляется облегченная система для управления проектами – SureTrak. Этот полностью русифицированный продукт ориентирован на контроль выполнения небольших проектов и/или фрагментов крупных проектов. Может работать как самостоятельно, так и совместно с P3 в корпоративной системе управления проектами.

Российская разработка *Spider project* отличается мощными алгоритмами планирования использования ограниченных ресурсов и большим количеством дополнительных функций. Система спроектирована с учетом большого практического опыта, потребностей, особенностей и приоритетов российского рынка. Spider Project поставляется в двух версиях – Professional и Desktop.

На сегодняшний день *Microsoft Project* (MS Project) является самой распространенной в мире системой управления проектами. Во многих западных компаниях MS Project стал привычным дополнением к Microsoft Office даже для рядовых сотрудников, которые используют его для планирования графиков несложных комплексов работ. Последней версией системы является MS Project 2003, которая имеет три модификации: Standard - для индивидуального использования, Professional – обеспечивает дополнительные возможности по анализу проекта и планированию ресурсов, Project Server 2003 – платформа для организации группового (вместе с Project Standard 2003) и корпоративного (вместе с Project Professional) решения для управления проектами.

В настоящее время MS Project является лидером среди программного обеспечения для управления проектами предприятия как на российском, белорусском, так и на западных рынках. На данный момент большинство пользователей этой системы применяют последнюю версию – связку MS Project Professional 2003 с Project Server 2003. Наиболее востребованными функциями данной системы считаются календарное планирование, отслеживание хода выполнения, составление отчетов. Что касается управления материальными ресурсами, оно, по мнению пользователей, оставляет желать лучшего. При внедрении MS Project не возникает практически никаких технических трудностей. Однако особого внимания на этом требует подготовка персонала, что подразумевает учет всех тонкостей работы с данной системой, изучение его реальных возможностей и функций.

В таблице представлен обзор западного рынка программного обеспечения по управлению проектами с рейтинговыми оценками (по данным 2006 г.).

Второе место на отечественных рынках программных продуктов по управлению проектами принадлежит Spider Project, которая обошла таких грандов, как Primavera и Open Plan. Это вполне заслуженный результат, учитывая самые богатые на сегодняшний день возможности планирования и развитые средства коммуникации команды проектов в процессе исполнения. Интерес к данному программному продукту объясняется постоянным непрекращающимся развитием функций данной системы с учетом специфики управления проектами в России, а также сопутствующих процессов. Фирма-разработчик оказывает постоянную техническую поддержку пользователям, которая охватывает не только аспекты работы с данной системой, но и теоретические нюансы управления проектами, и накопленный практический опыт в данной области.

Primavera for Enterprise является менее распространенным программным продуктом по управлению проектами, тем не менее, по мнению многих экспертов, его возможности намного превосходят функционал, реализованный в MS Project, что делает его ориентированным на использование в крупных организациях, осуществляющих большие проекты. В целом, пользователи данного продукта оценивают его как надежный и гибкий инструмент для использования при управлении проектами. При этом наиболее востребованными функциями пакета являются расчет критического пути и расчет загрузки ресурсов. В числе выгод, которые приносит использование данной системы управления проектами, выделяется прозрачность и большая предсказуемость. Также очень важным является то, что результаты расчетов данной системы позволяют обосновывать перспективные решения.

Однако, с методологической точки зрения, для успешного функционирования системы управления проектами разработка *бизнес-процессов* должна предшествовать настройке программного обеспечения. Отработанные и регламентированные бизнес-процессы во многом сами по себе гарантируют слаженную

работу предприятия, образуя некоторую внутреннюю систему управления. Каждое подразделение выполняет свои функциональные обязанности, предоставляя информацию руководству для принятия управленческих решений и координации работы предприятия. Дальнейшая автоматизация ряда рутинных бизнес-процессов за счет использования программного обеспечения лишь позволяет дополнительно ускорить их выполнение. В связи с этим, большим заблуждением является мнение, что программное обеспечение может решить все внутренние проблемы на предприятии. А, как правило, причины этих проблем заключаются в отсутствии бизнес-процессов, отвечающих существующим требованиям рыночных отношений. Находясь под влиянием такого подхода, руководители не уделяют должного внимания анализу и оптимизации существующих бизнес-процессов на предприятии. В результате программное обеспечение выходит на первый план и автоматизируется существующий «беспорядок». Процессы управления, несмотря на то, что уже автоматизированные, по сути, являются прежними. Как следствие, руководство предприятия не получает ожидаемой отдачи от системы управления проектами (СУП).

Особенно важную роль описание бизнес-процессов играет в двух случаях:

1) В случае создания СУП на базе разрабатываемого программного обеспечения. В данном случае крайне необходимо разработать бизнес-процессы, максимально отвечающие требованиям к СУП, так как от этого напрямую будет зависеть эффективность использования программного обеспечения.

2) В случае, если в рамках создания СУП предполагается осуществление интеграции между различными информационными системами. Здесь без описания бизнес-процессов, заложенных в две смежные интегрируемые информационные системы, и процесса обмена информацией между ними, практически невозможно приступить к разработке интеграционного модуля. В обоих случаях качество описания процессов закладывает фундамент под дальнейшие работы по разработке программного обеспечения или интеграционных решений, внося свой вклад в эффективное функционирование системы управления проектами в целом.

В заключение хотелось бы отметить постепенное, хотя и довольно осторожное внедрение практики использования специализированного программного обеспечения для управления проектами на отечественных предприятиях и фирмах. При этом успешное использование системы управления проектами определяется не столько набором его функций, а готовностью самой организации принять к использованию данный инструмент.

Обзор западного рынка программного обеспечения по управлению проектами с рейтинговыми оценками (2006 г.)

	Microsoft Project	eProject	@task	Celoxis	eStudio	Copper	Milestones	Smooth Projects	MinuteMan	Project KickStart
Рейтинг	золото	серебро	бронза	4	5	6	7	8	9	10
Цены										
Цена	\$599.00	\$0	\$0	\$0	\$0	\$999.00	\$219.00	\$0	\$49.95	\$129.95
Плата за обслуживание (в мес.)	\$0	\$39.50	\$30.00	\$7.95	\$50.00	\$0	\$0	\$39.00	\$0	\$0
Количество пользователей	1	1	1	1	Не огранич.	50	1	1	1	1
Общий рейтинг										
Рейтинги										
Простота использования										
Функциональные возможности										
Простота инсталляции										
Поддержка/помощь										

Функции проектного управления	■ ■ ■ ■	■ ■ ■ ■ □	■ ■ ■ ■ □	■ ■ ■ ■ □	■ ■ ■ ■ □	■ ■ ■ ■ □	■ ■ ■ ■ □	■ ■ ■ ■ □	■ ■ ■ ■ □	■ ■ ■ ■ □
Управление ресурсами	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
Совместная работа	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■
Функциональные возможности										
Клиент-модуль	■	■		■	■	■	■	■	■	■
ASP-модуль	■	■	■	■	■	■	■	■	■	■
Проектное управление										
Управление работами	■	■	■	■	■	■	■	■	■	■
• Назначение / Переназначение работ	■	■	■	■	■	■	■	■	■	■
• Отслеживание хода исполнения работ	■	■	■	■	■	■	■	■	■	■
• Обратная связь работ	■	■	■	■	■	■	■	■	■	■
• Назначение / Срок завершения работ	■	■		■	■	■	■	■	■	■
• Подчиненность работ	■	■	■	■	■	■	■	■	■	■
• Периодичность работ	■	■	■	■	■	■	■	■	■	■
Календарное планирование	■	■	■	■	■	■	■	■	■	■
• Календари	■	■	■	■	■	■	■	■	■	■
• Временная диаграмма	■	■	■	■	■	■	■	■	■	■
• События	■	■	■	■	■	■	■	■	■	■
Диаграмма Гантта	■	■	■	■	■	■	■	■	■	■
Формирование отчетов	■	■	■	■	■	■	■	■	■	■
• Статистика	■	■	■	■	■	■	■	■	■	■
• Объем работ										
• Денежные ресурсы										
• Клиенты (покупатели, заказчики)	■	■	■	■	■	■	■	■	■	■
Документирование	■	■	■	■	■	■	■	■	■	■
Бюджетирование	■	■	■	■	■	■	■	■	■	■
Метод СРМ	■	■	■	■	■	■	■	■	■	■
Схема проекта (шаблон)	■	■	■	■	■	■	■	■	■	■
Предметная область	■	■	■	■	■	■	■	■	■	■
Вехи	■	■	■	■	■	■	■	■	■	■
Исходный план проекта	■	■	■	■	■	■	■	■	■	■
Управление ресурсами										
Детализация ресурсов	■	■	■	■	■	■	■	■	■	■
Способности к оптимизации	■	■	■	■	■	■	■	■	■	■
Детализир. схема исполнения работы	■	■	■	■	■	■	■	■	■	■
Материалы / Поставщики	■	■	■	■	■	■	■	■	■	■

Отметка начала / завершения	■	■	■	■	■	■	■	■	■	■
Импортирование ресурсов	■	■	■	■	■	■	■	■	■	■
e-mail адреса	■	■	■	■	■	■	■	■	■	■
Затраты	■	■	■	■	■	■	■	■	■	■
Комментарии к ресурсам	■	■	■	■	■	■	■	■	■	■
Группирование многочисленных ресурсов	■	■	■	■	■	■	■	■	■	■
Совместная работа										
Инструментальная панель	■	■	■	■	■	■	■	■	■	■
Объединяющий центр совместной работы	■	■	■	■	■	■	■	■	■	■
Календарь / Временная диаграмма для группы исполнителей	■	■	■	■	■	■	■	■	■	■
Отслеживание исполнения и назначений	■	■	■	■	■	■	■	■	■	■
Выведение отчетов / Документирование	■	■	■	■	■	■	■	■	■	■
Форумы	■	■	■	■	■	■	■	■	■	■
e-mail интеграция	■	■	■	■	■	■	■	■	■	■
Поддержка/Помощь										
Телефонная помощь	■	■	■	■	■	■	■	■	■	■
Справочник	■	■	■	■	■	■	■	■	■	■
Часто задаваемые вопросы	■	■	■	■	■	■	■	■	■	■
База данных справочной информации	■	■	■	■	■	■	■	■	■	■
Средства обучения	■	■	■	■	■	■	■	■	■	■
Пользовательский форум	■	■	■	■	■	■	■	■	■	■
Online формы	■	■	■	■	■	■	■	■	■	■
e-mail адреса	■	■	■	■	■	■	■	■	■	■

Методы обработки информации о пространственно распределенных объектах

Андрянов Д. Е., E-mail: AndrianovDE@inbox.ru
 Муромский институт Владимирского государственного университета
 602264, г.Муром, Владимирской обл., ул.Орловская, 23

В докладе рассматривается метод обработки информации о геоинформационных объектах на основе топологических отношений. Данный метод позволяет формализовать

топологическую составляющую пространственных объектов и на ее основе реализовать алгоритмы вычисления взаимоотношений. Полученные данные позволяют значительно расширить спектр решаемых задач в современных геоинформационных системах.

Современные геоинформационные технологии находят применение в самых различных сферах деятельности человека, как в связи с его профессиональными потребностями в пространственной информации, так и в связи с познавательными потребностями. Особо интенсивно геоинформационные системы (ГИС) внедряются в системы управления городскими службами, начиная от создания земельного кадастра и заканчивая телекоммуникационными системами управления энергетическим обеспечением. Применение систем, построенных с использованием новых информационных технологий, обусловлено необходимостью инвестиций, которые можно привлечь при условии полной прозрачности управления территорией и коммуникациями.

В таких системах эффективность хранения большого объема пространственной информации определяется методами обработки. Основной составляющей любой ГИС является пространственный анализ, реализовать который возможно при использовании информации о пространственном расположении объекта и его взаимосвязи с другими объектами инфраструктуры. Геометрическое описание объекта и его местоположение выполняется на основе топологии и топологических отношений. Теоретически топологические характеристики объектов, используемые в современных ГИС, мало формализованы [1]. Поэтому, в настоящее время, остро стоит вопрос формализации взаимных отношений объектов в ГИС.

Вычисление топологических отношений осуществляется алгоритмически либо эвристически. Механизм, основанный на обычной схеме выбора решения достаточно подробно описан в [4]. Содержательно решение об определении того или иного отношения представляет собой альтернативный подход, выбранный лицом принимающим решение (ЛПР) при рассмотрении текущих условий из множества возможных согласно поставленной им цели.

Можно выделить три класса топологических отношений (внутриобъектные, межобъектные и концептуальные). Каждый класс определяет большую группу топологических отношений, которые отражают элементы внутренней или внешней топологии. Указанные классы могут перекрываться, взаимодополнять друг друга. Внутренние – определяют целостность объекта, как совокупности элементарных геометрических частей и семантического контекста. Концептуальные – топологические отношения, устанавливающие наиболее общие правила расположения объектов относящихся к разным классам [3]. Межобъектные – топологические отношения, устанавливаемые между парой объектов [2].

Мы остановимся на топологии между объектами, т.к. эти отношения в первую очередь используются при построении алгоритмов обработки данных в муниципальных геоинформационных системах (МГИС) и геоинформационных системах предприятий (ГИСП), где плотность информации на единицу площади значительно высока. Каждый объект необходимо рассматривать как некоторую сущность состоящую из геометрических примитивов, а ГИС помогает человеку правильно трактовать набор этих примитивов.

Поэтому, прежде чем разрабатывать алгоритмы вычисления топологических отношений определим общий метод. Для вычисления топологического отношения между объектами будем использовать бинарное отношение R упорядоченных пар элементов на множестве объектов X , т.е. $R \subseteq X \times X$, и это означает, что для $\forall x, y \in X$ топологическое отношение xRy имеет место тогда, когда $(x, y) \in R$, тем самым определяется топологическое отношение.

При этом совокупность всех пар (x, y) может быть задана:

- $n \times n$ матрицей, элемент a_{ij} которой на пересечении i -й строки и j -го столбца равен единице, если $x_i R x_j$, $i, j = \overline{1, n}$, и нулю – в противном случае;

- графом, когда каждому элементу $x_i \in X, i = \overline{1, n}$, однозначно соответствует вершина графа, которая соединяется дугой с вершиной $x_j \in X$ и направлена от x_i к x_j , если $x_i R x_j$, а когда $i=j$, то дуга представляется петлей при вершине x_i ;

- верхним и нижним сечениями $R^+(x)$ и $R^-(x) \forall x \in X$,

$R^+(x) = \{y \in X / (y, x) \in R\}$ - верхнее сечение (срез) для фиксированного $x \in X$,

$R^-(x) = \{y \in X / (x, y) \in R\}$ - нижнее сечение (срез) для фиксированного $x \in X$,

для пустого отношения $R = \emptyset$, т.е. когда R не выполняется ни для одной из пар объектов

$(x, y) \in X, R^+(x) = R^-(x) = \emptyset, \overline{R^+}(x) = X \setminus R^+(x), \overline{R^-}(x) = X \setminus R^-(x), \forall x \in X$,

где \bar{R} – отрицание (дополнение) отношения R , т.е. $xRy = y\bar{R}x$, $\bar{\bar{R}} = R$.

На основе одного из вышеперечисленных представлений возможно сгенерировать алгоритмы вычисления топологических отношений, количество которых в первую очередь ограничивается поставленной задачей.

Установление топологических отношений базируется на геометрическом расположении объектов на карте. Существуют следующие варианты геометрического расположения: касание, пересечение, включение, изолированность.

Указанные варианты геометрического расположения могут быть дополнены двумя числовыми характеристиками, которые являются наиболее важными при определении взаимного расположения объектов – расстояние (величина характеризующая степень удаленности двух объектов, выраженную в единицах измерения принятых в рассматриваемой системе) и относительное расположение или угол между характеризующими объект элементами, например, минимальным и максимальным диаметрами.

Примем следующие допущения: в ГИС могут реализовываться простые и сложные топологические отношения; простые топологические отношения подразделяются на виды и типы.

Разделение топологических отношений на виды будем производить по комбинации объектов участвующие в топологическом отношении.

Типизацию топологических отношений будем производить по вариантам геометрического расположения и числовым характеристикам.

На основании сказанного выше выделим следующие типы топологических отношений: изолированность, пересечение, вложенность, соседство, близость, удаленность, относительное расположение [3]. Для вычисления этих топологических отношений в дальнейшем на основе предложенного метода будет разработана группа алгоритмов для задач МГИС.

Список литературы

1. Журкин И. Г., Наумов С. В. Итерационная модель представления линейных пространственных данных // Информационные технологии. 2003, №6. с. 11-16.
2. Горбачев В. Г. Что такое «топологические» отношения в цифровой картографии и для чего топологические отношения нужны в геоинформатике? // www.integro.ru – сайт ГИС Ингео ЗАО ЦСИ «Интегро», 1997 г.
3. Садыков С. С., Андрианов Д. Е., Еремеев С. В. Формальное определение топологических отношений между картографическими объектами в ГИС // Обработка информации: методы и системы: Сборник научных статей – М.: Горячая линия – Телеком, 2003, с. 73-78.
4. Катулаев А. Н., Северцев Н. А. Математические методы в системах поддержки принятия решений. М.: Высшая школа, 2005. – 311 с.

Модель интеллектуальной системы управления деятельностью коллективов

*к.т.н., доцент Антонова Елена Ивановна,
Рябцев Тимофей Владимирович, Бенгер Роман Владимирович,*

*Дальневосточный государственный университет,
г. Владивосток, ул. Октябрьская 25, 69000*

E-mail: ant@imcs.dvgu.ru

Управление в широком понимании этого термина – непрерывный процесс воздействия на объект управления (личность, коллектив, технологический процесс, предприятие, государство) для достижения оптимальных результатов при наименьших затратах времени и ресурсов.

Управление существует более 7 тысяч лет, оно социально, так как началось с развития человеческих отношений. Тенденции развития управления – методы, отношение к объекту управления – в течение всего времени были направлены на улучшение социальной среды работников и более тонкое взаимодействие с каждым человеком с целью учета его индивидуальных особенностей и потребностей. Это вполне естественно, поскольку без людей нет организации. Без нужных людей ни одна организация не сможет

достичь своих целей и выжить. А это крайне тяжело без эффективного управления трудовыми ресурсами – одного из важнейших аспектов теории и практики управления. Для того чтобы предприятие работало эффективно, необходимо правильно организовать труд работников, при этом постоянно контролируя деятельность работников, используя различные методы управления.

Специфический вид управленческой деятельности, объектом которой является коллектив работников - персонал, получил название управления персоналом (кадрами). Специфический, потому что в управлении людьми необходимо учитывать такую неопределенную, ненадежную вещь, как «человеческий фактор».

Ненадежные элементы в контексте управления – это элементы, о состоянии которых нельзя получить достоверную информацию и поведение которых невозможно спрогнозировать с высокой степенью вероятности. В теории управления ненадежными элементами, прежде всего, являются люди. Системы, элементами которых являются люди, обладают рядом особенностей, которые затрудняют процесс управления. Это напрямую связано с биологической, эмоциональной особенностью людей, что и делает их ненадежными элементами.

Достижения новой информационной технологии сегодня уже позволяют рассматривать возможность автоматизации процессов управления людьми при решении ими профессиональных задач, являясь одним из основных резервов повышения эффективности управления. Десятилетия развития локальных сетей и сети Интернет предложили множество технологий для организации общения, как между людьми, так и между целыми предприятиями и программными комплексами, которые также могут быть использованы в контексте управления. Анализ существующих программных систем позволяет сделать вывод о том, что автоматизацию процесса управления понимают, прежде всего, как информационное обеспечение лиц, принимающих решения.

На концептуальном уровне проектирования системы управления ненадежными элементами необходимо четкое определение общей аксиоматики описания **управленческих процессов**, т.е. фиксация исходных положений дальнейшего исследования.

Модель системы управления – совокупность представлений о системе. Необходимо определиться с основными структурными элементами модели управления. В совокупности эти элементы должны образовывать полный и непротиворечивый **набор моделей**, характеризующий все базовые компоненты системы управления ненадежными элементами.

Применительно к задаче управления ненадежными элементами при решении профессиональных задач набор элементов модели системы управления трансформируется в систему, включающую:

- 1) модель профессиональной деятельности, определяющую цели и задачи деятельности, план решения, ограничения и взаимосвязь ресурсных потоков;
- 2) модель «ненадежного элемента» (человека) как объекта и как субъекта управления, определяющая ролевые взаимоотношения в коллективе;
- 3) модель управленческого процесса.

Модель профессиональной деятельности (МПД)

Цель разработки модели: формализация способа описания и планирования деятельности, описания множества состояний выполняемой деятельности и взаимосвязей ее элементов, формализация алгоритма оценки эффективности деятельности на всех этапах по заданным критериям. На основе МПД формируется система агрегированного представления планируемых бизнес-процессов, описание взаимосвязи параметров ресурсных потоков, а также представление отслеживаемых параметров выполнения деятельности для возможного вмешательства системы с целью обеспечения выполнения плана решения «ненадежными элементами».

Модель человека - «ненадежного элемента» (МОЧ)

В модели человека выделим:

1) Субъект управления, который осуществляет управленческую и организационную работу, принимает решения и обеспечивает достижение поставленных целей. Субъект управления называют управляющей системой.

2) Объект управления, на который направлено управленческое воздействие в целях обеспечения функционирования и развития системы. Объект управления иногда называют управляемой системой.

Управление осуществляется некоторым субъектом по отношению к некоторому объекту (или объектам). Оправдано рассмотрение субъектов и объектов управления в иерархии управления, поскольку на разных ступенях иерархии один и тот же объект управления может быть как субъектом, так и объектом. Процессу управления всегда должно предшествовать определение субъекта и объекта управления в заданной иерархии

отношений между ними, а также определение свойств и характеристик, отрицательно влияющих на выполнение профессиональной деятельности людьми.

Цель разработки модели: формализация особенностей управляющей и управляемой систем (субъекта и объекта управления), определение возможных «блокирующих» воздействий (тормозящих осуществление деятельности по заданному плану решения), определения возможных и оправданных способов мотивации для предотвращения «блокирующих» воздействий данного «ненадежного элемента».

На основе МОЧ формируется система представления «ненадежного элемента» (человека) как субъекта и объекта управления.

Модель управленческого процесса (МУП)

Цель разработки модели: формализация необходимых мероприятий по достижению показателей эффективности, а также формализация применения приемов защиты от блокирующих воздействий «ненадежных элементов» и применения способов мотивации. Необходимо определение алгоритма, позволяющего вмешаться в ход выполняемой деятельности и применить необходимые для данной ситуации и для данного «ненадежного элемента» управляющие процедуры, с целью обеспечения выполнения плана решения задач.

Управленческое мероприятие – действия системы (процедуры управления), направленные на «ненадежный элемент» с целью обеспечения выполнения им профессиональной деятельности по заданному плану решения.

Таким образом, при построении модели системы управления имеет место цепочка итераций:

1) в МПД: описание планируемой деятельности и критериев эффективности функционирования системы управления для осуществления деятельности в соответствии с планом;

2) в МЧ: описание субъекта и объекта управления (выделение параметров наблюдения и управления «ненадежного элемента», влияющих на профессиональную деятельность);

3) в МУП: описание управленческих мероприятий для обеспечения эффективного выполнения «ненадежными элементами» запланированной деятельности.

Здесь необходимо отметить, что в процессе функционирования системы, МУП характеризуется воздействием (использование конкретного управленческого мероприятия из МУП) на функционирование объекта и субъекта управления из МЧ и, в связи с этим, необходимостью пересмотра процессов, реализуемых в МПД, с целью развития и поддержания адекватности МПД изменяющимся условиям.

В рамках решения задачи «управления ненадежными элементами» нас интересует проблема разработки процедур управления людьми при решении профессиональных задач. К числу наиболее эффективных процедур управления, которые можно применить в рамках решения нашей задачи можно отнести следующие:

1. Способ управления на основе требований *должностных инструкций*. В соответствии с должностными инструкциями определяются роли участников системы управления, обязанности, возможности, правила решения профессиональных задач. Требования инструкций, регламентирующие действия сотрудников согласно установленным обязанностям в коллективе, являются основой выделения людских ресурсов для решения подзадач.
2. Способ управления на основе *поощрительных и штрафных санкций*. Для стимулирования выполнения правильности задач в короткие сроки к участникам системы управления применяются поощрительные санкции. В случае невыполнения задачи на участников налагаются штрафные санкции.
3. Способ управления, в котором учитываются *интересы и психологические особенности личности*. Для выполнения задачи подбирается такой участник в систему управления, у которого интересы совпадают с темой задачи, и его психологические особенности максимально удовлетворяют требованиям задачи.
4. Способ управления, при котором учитывается *личное участие работника в принятии решений*. На определенном шаге выполнения работы система управления дает возможность участнику выбрать один из дальнейших путей решения задачи, предусмотренным планом решения. Таким образом, участнику в системе управления предоставляется «свобода» выбора дальнейшего решения задачи, но в рамках установленного плана решения.
5. Способ управления, при котором ведется *учет и контроль работника*. На каждом этапе выполнения задачи в системе расставляются временные контрольные точки, в которых идет проверка выполнения задачи участником - укладывается ли он в отведенные сроки. Строятся графики успеваемости всех участников решения.

б. Способ управления, при котором ведется *информирование* участников системы. Субъекту сообщается о текущем положении дел объекта, где отслеживается ход выполнения задачи, сроки выполнения и успеваемость.

Эти методы должны быть положены в основу реализации интеллектуальной системы управления людьми, учитывающей особенности человека и направлены на повышение «надежности» человека как объекта субъекта управления.

Расширяемая интеллектуальная интернет-система по ХИМИИ¹

с.н.с. **Артемьева И. Л.**, м.н.с. **Рештаненко Н. В.**, н.с. **Цветников В. А.**,
e-mail: artemeva@iacp.dvo.ru, reshtanenko@iacp.dvo.ru

Химия является сложной предметной областью, которая состоит из нескольких крупных разделов (например, физической, органической и аналитической химии), которые в свою очередь делятся на более мелкие (например, химическая кинетика, химия фосфорорганических соединений, рентгено-флуоресцентный анализ). При решении прикладных задач специалисты данной области используют знания многих ее разделов. Кроме того, данная область развивается, что ведет к появлению новых знаний и новых задач, а также к изменению онтологии предметной области. Поэтому программная система, автоматизирующая профессиональную деятельность в данной области, должна учитывать указанные свойства этой области.

Для решения научных, практических и образовательных проблем предметной области «Химия» в работе предлагается использовать интеллектуальную Интернет систему для коллективного развития информационной базы данной области и ее использования при решении задач на компьютере. Такая система должна позволять изменять онтологию и знания предметной области (а также, как следствие, изменять множество решаемых задач) и иметь средства адаптации методов решения задач к этим изменениям.

Интернет система для химии включает информационное наполнение (рис.1), оболочку информационного наполнения, программное наполнение и блок администрирования.

Разработка интеллектуальной системы основывается на многоуровневой модели онтологии данной



Рис. 1. Информационное наполнение интеллектуальной системы по химии

области. Структура многоуровневой модели химии приведена на рис. 2. Третий уровень онтологии

¹ Работа выполнена при финансовой поддержке ДВО РАН, проект 06-III-A-01-005 «Разработка теоретических основ создания и экспериментальное исследование интеллектуальных систем, основанных на многоуровневых моделях некоторых разделов химии», а также Российского фонда фундаментальных исследований, проект 06-07-89071-а «Исследование возможностей коллективного управления в семантическом вебе информационными ресурсами различных уровней общности».

определяет термины, общие для всех разделов данной области; второй уровень – термины, используемые при определении онтологии каждого раздела, а первый уровень определяет онтологии разделов. Знания предметной области описываются в терминах онтологии первого уровня.

Примерами терминов онтологии третьего уровня являются следующие термины: типы сущностей, сущности, собственные свойства сущностей, совместные свойства сущностей. Каждый тип сущностей представляет собой некоторое множество сущностей; каждая сущность может иметь имя, быть представлена числом, быть логическим значением либо кортежем значений. Множества сущностей различных типов не пересекаются. Характеристикой каждой сущности является ее тип. Каждая сущность может иметь свой набор собственных свойств.

Модель онтологии каждого раздела (компонентов онтологии второго уровня) получается из модели онтологии третьего уровня заданием значений параметров онтологии третьего уровня, т.е. заданием набора типов сущностей, изучаемых соответствующим разделом. Например, для органической химии сущности бывают химическими элементами, веществами, органическими соединениями, реакциями и т.д.

Онтология первого уровня определяет свойства сущностей каждого типа. Схема определения свойства фиксируется в онтологии третьего уровня. Конкретный набор сущностей каждого типа, а также значения их свойств задаются в знаниях раздела.

Онтология второго уровня состоит из множества модулей (каждый модуль соответствует разделу предметной области). Онтология первого уровня для каждого раздела также является модульной. Определение каждого модуля онтологии первого уровня базируется на онтологии второго уровня соответствующего раздела. База знаний, в свою очередь, является модульной. Ее модули соответствуют модулям онтологии первого уровня.

Онтология первого уровня и ее модель для физической химии описаны в работе [1]. Структура онтологии первого уровня и ее модель для органической химии, а также метод ее построения описаны в работах [2, 3]. Онтология первого уровня и ее модель для рентгено-флуоресцентного анализа описаны в работе [4]. Модель онтологии представлена на языке прикладной логики [5, 6].

Информационное наполнение интеллектуальной системы по химии также должно включать актуальную теоретическую информацию различных разделов предметной области, представленную в книгах, журналах и на Интернет сайтах.

Оболочка информационного наполнения предоставляет набор программных средств, обеспечивающих возможность использования содержимого информационного наполнения.

Блок администрирования администрирует пользователей и контролирует жизненный цикл всей Интернет системы. Всех возможных пользователей интеллектуальной системы по химии можно условно разделить на носителей информации, обслуживающих пользователей, и прикладных пользователей. Обслуживающие пользователи – это администратор, следящий за функционированием всего банка знаний. Сопровождающие пользователи – программисты, которые создают и встраивают новые программные средства для решения прикладных задач. Носители информации – специалисты-эксперты, которые являются не только специалистами предметной области химии, но и инженерами знаний, способными расширять имеющуюся онтологию добавлением новых разделов, а также добавлять новые знания в базу знаний в соответствии с онтологией первого уровня. Эксперты также имеют возможность просмотра и оценки всех вновь созданных модулей.

Прикладные пользователи в свою очередь разделяются на классы: исследователь, лаборант и гость. Исследователь – специалист в области химии, занимающийся исследовательской и/или прикладной деятельностью. Лаборант – пользователь, решающий конкретные прикладные задачи. Гость – пользователь, одноразово использующий программу. Исследователь занимается исследовательской и/или прикладной деятельностью. Он может просматривать всю информацию информационного наполнения. Для индивидуальной работы он может создавать свою копию всего или части информационного наполнения, где может производить любые изменения, необходимые ему для работы. Если в процессе своей исследовательской или экспериментальной деятельности он получил существенные результаты, пополняющие базу данных, то он посылает запрос на пополнение информационного наполнения администратору, который принимает решение о возможности ее добавления в информационное наполнение. Лаборант решает конкретные прикладные задачи. Он может просматривать всю информацию из информационного наполнения системы. Лаборант может использовать программные средства из программного наполнения для решения прикладных задач. Помимо этого лаборант имеет возможности гостя. Гость может просматривать информационное наполнение.

Программное наполнение интеллектуальной системы для химии включает (рис. 3): средства редактирования информационного наполнения и набор программных средств для решения прикладных химических задач. Средства редактирования информационного наполнения включают редактор онтологии

(всех трёх уровней), редактор знаний и редактор теоретического материала. Редактирование онтологии и знаний осуществляется посредством редакторов и графического компонента (интерфейсной оболочки для пользователей), использующих понятную для химиков терминологию и графические представления. Знания, введенные с помощью интерфейсной оболочки, переводятся во внутреннюю структуру представления базы знаний.

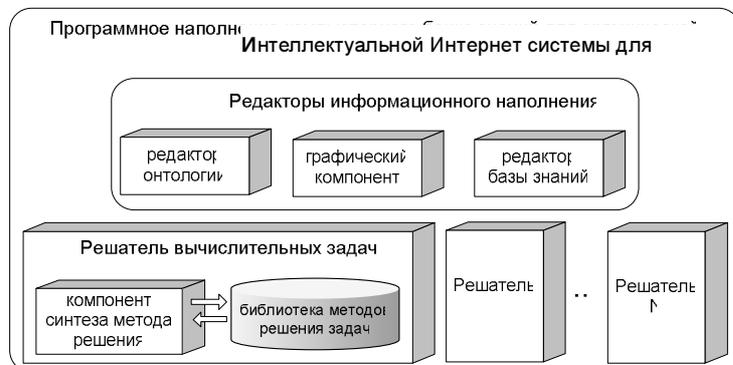


Рис. 3. Программное наполнение интеллектуальной системы по химии

Другими компонентами программного наполнения являются решатели задач (например, классификации, вычислительных задач, задач проверки непротиворечивости знаний и онтологии). Поскольку важным свойством программной системы является ее адаптация к расширению множества решаемых задач, то решатель вычислительных задач должен иметь средства настройки на изменение множества задач. Такая настройка может выполняться посредством автоматического синтеза методов решения задач. При синтезе методов решения задач по спецификации задачи проблемой является автоматическое разрешение ограничений предметной области, задаваемых онтологией и знаниями этой предметной области. Если ограничения имеют вид линейных равенств, то их разрешение сводится к решению системы линейных уравнений. Но в настоящее время не для всех видов ограничений существуют методы их решения. Поэтому компонентами решателя вычислительных задач химии являются подсистемы для решения систем уравнений, а также библиотеки методов (подпрограмм) решения элементарных задач. Синтез метода решения задачи состоит в построении последовательности вызова этих подпрограмм (методы такого построения разработаны для пакетов прикладных программ и известны из литературы).

Программное наполнение может расширяться средствами автоматического решения новых видов ограничений, если методы решения ограничений других классов станут известны.

Решатель должен позволять сохранять сгенерированные методы решения задач, после чего может быть выбран оптимальный (вручную и автоматически). Решатель должен предоставлять возможность сохранения оптимального метода в библиотеку методов и самостоятельного внесения пользователем метода решения задачи в терминах, определяемых решателем.

Программное наполнение также должно включать подсистему настройки автоматизированного рабочего места (АРМ) для различных пользователей. У каждого класса пользователей есть ряд задач, которые пользователь может решать с помощью интеллектуальной системы, поэтому для каждого класса пользователей есть свой АРМ. Подробное описание потребностей различных классов пользователей и функций интеллектуальной системы, реализующих эти потребности, дано в работе [7].

Список литературы

1. Артемьева И. Л., Цветников В. А. Фрагмент онтологии физической химии и его модель // Электронный журнал "Исследовано в России", 2002, № 3, с. 454-474. (<http://zhurnal.ape.relarn.ru/articles/2002/042.pdf>)

2. Артемьева И. Л., Высоцкий В. И., Рештаненко Н. В. Модель онтологии предметной области (на примере органической химии) // Научно-техническая информация, сер.2, 2005, № 8, с. 19-27
3. Артемьева И. Л., Высоцкий В. И., Рештаненко Н. В. Описание структурного строения органических соединений в модели онтологии органической химии // Научно-техническая информация, сер.2, 2006, № 2, с. 11-19.
4. Артемьева И. Л., Мирошниченко Н. Л. Модель онтологии рентгено-флуоресцентного анализа // Информатика и системы управления, № 2, 2005, с. 78-88.
5. Клещев А. С., Артемьева И. Л. Необогатенные системы логических соотношений. Часть 1. // Научно-техническая информация, сер.2, 2000, №7. С. 18-28.
6. Клещев А. С., Артемьева И. Л. Необогатенные системы логических соотношений. Часть 2. // Научно-техническая информация, сер. 2, 2000, № 8, с. 8-18.
7. Артемьева И. Л., Рештаненко Н. В. Специализированный компьютерный банк знаний предметной области «Химия» // Искусственный интеллект, №4, 2004, с. 235-245.

Экспертная система диагностики вирусного гепатита²

Артюхин В. В.,

кафедра «Вычислительные системы и моделирование»

Пензенского государственного педагогического университета им. В. Г. Белинского.

Адрес: 440052, г. Пенза, ул. Средне-Кутузовская, 18.

e-mail: death-stalker@yandex.ru

Важной проблемой службы переливания крови является ранняя диагностика вирусного гепатита. Решение этой задачи мотивируется тенденцией значительного распространения гемотрансмиссивных заболеваний (заболеваний, передающихся при переливании донорской крови). В связи с постоянной трансформацией вирусов гепатита не представляется возможным выработать чёткие критерии диагностики этого заболевания [1]. По этой причине совершенствование методов лабораторной диагностики нередко задерживается. Известны эффективные методы лабораторной диагностики вирусных гепатитов: иммуноферментный анализ (ИФА) и полимеразно-цепная реакция (ПЦР). Однако они являются дорогостоящими методами. Затраты на проведение ИФА и ПЦР превышают современные затраты на оказание лечебных мероприятий. Таким образом, с целью массового обследования населения (скрининга) применение иммуноферментного анализа и полимеразно-цепной реакции не является целесообразным. Поэтому необходимо использовать менее дорогостоящие и более доступные методы диагностики, в том числе, компьютерные методы диагностики. В медицине достаточно популярны статистические методы. Однако применение методов статистического анализа в рассматриваемой области имеет объективные ограничения из-за нечёткой трактовки моделей нормы, адаптации и патологии. Поэтому последние годы характеризуются широким применением нейросетевых технологий в медицинской практике [2, 3]. Следует указать, что нередко диагностическая эффективность прогнозных моделей варьирует в пределах 70 – 85% [4]. Таким образом, важной задачей применения нейронных сетей в медицинских исследованиях является необходимость повышения достоверности прогноза.

Целью данной работы является исследование и создание нейронных сетей, позволяющих эффективно распознавать лабораторную характеристику вирусного гепатита для его скрининга, без привлечения дорогостоящих методов ИФА и ПЦР, а также создание экспертной системы для проведения скрининга.

Подготовка данных, исследование и обучение сетей с помощью метода обучения Левенберга-Марквардта были нами подробно описаны в предыдущих работах [5]. После того как была выбрана оптимальная архитектура нейронной сети и количество используемых показателей, обучение сети проводилось с использованием алгоритма градиентного спуска. Выбор данного алгоритма связан с тем, что остальные алгоритмы обучения требуют больших ресурсов компьютера, что не всегда выполнимо в

² Работа выполнена в рамках гранта А04-3.16-625 "Нейросетевая диагностика вирусного гепатита у хирургических больных" для поддержки научно-исследовательской работы аспирантов вузов Федерального агентства по образованию

условиях лечебных учреждений различного уровня. Поэтому целесообразно использовать более медленный, но менее ресурсоемкий алгоритм градиентного спуска. В связи с тем, что обучение производится очень редко, скорость обучения не имеет решающего значения. В ходе экспериментов подбирались оптимальное значение шага обучения, обеспечивающее достижение приемлемой ошибки на тестовом множестве и достаточно гладкий процесс обучения. При использовании 6 показателей оптимальным оказался шаг обучения, равный 0,1. При этом ошибка сети на тестовом множестве составила 12%. Для обучения потребовалось 6849 циклов обучения. При использовании 25 показателей оптимальный шаг обучения оказался равным 0,001. Сеть обучилась за 1332 цикла, ошибка на тестовом множестве составила 10%.

Экспериментально подобранные архитектура и параметры нейронных сетей использованы при реализации нейросетевой экспертной системы. Система разработана в среде разработки Borland C++ Builder. Для хранения данных использована СУБД Microsoft Access.

Система позволяет проводить работу в режимах пользователя и администратора. В режиме пользователя система позволяет зарегистрировать пациента, ввести все необходимые данные, провести диагностику конкретного пациента, используя ранее обученную сеть и занести результаты диагностики в базу данных. В режиме администратора система позволяет обучить нейронную сеть, просмотреть все информационные карты обследования пациентов. Также администратор может управлять политикой безопасности в системе. Он может создать нового пользователя с определенными правами или создать новый пароль для уже существующего пользователя.

Рассмотрим возможности, предоставляемые системой.

При нажатии на кнопку "Первоначальная регистрация пациента" появляется форма "История болезни", содержащая три закладки: "Критерии госпитализации", "Информационная карта обследования" и "Данные стационарного обследования". Данная форма позволяет зарегистрировать впервые поступивших пациентов и добавить данные об уже занесенных в базу пациентах.

Закладка "Критерии госпитализации" содержит поля для ввода первичной информации о пациенте: номер амбулаторной карты, фамилия, имя, отчество, дата рождения, дата поступления, домашний адрес, место работы, профессия, пол, краткая история заболевания и жалобы больного, предварительный (рабочий) диагноз и сопутствующий диагноз.

Закладка "Информационная карта обследования" содержит поля для ввода результатов общего, биохимического и иммунологического анализов крови. При занесении анализов не обязательно заполнять все поля, для диагностики вирусного гепатита главное иметь данные о билирубине, аспаргатамино-трансферазе, аланинамино-трансферазе, щелочной фосфатазе, гамма-глутамил-транспептидазе и тимоловой пробе.

Закладка "Данные стационарного обследования" содержит поля для ввода информации о пациенте на момент окончания лечения: название отделения, номер истории болезни, номер обследования, дата выписки, койко-день, компьютерный диагноз, основной клинический диагноз, осложнения и рекомендации.

Разработанная система имеет механизм обучения. При нажатии на кнопку "Обучение многослойной сети" открывается форма, позволяющая пользователю управлять обучением нейронной сети. В системе предусмотрена настройка администратором параметров обучения сети. При нажатии на кнопку "Параметры обучения" перед пользователем открывается форма, позволяющая настроить параметры обучения сети. Пользователь может выбрать вариант обучения сети по 6 или 25 показателям. Также можно задать количество нейронов скрытого слоя, цель и шаг обучения. Можно установить максимальное количество циклов обучения и границы для интерпретации ответов сети.

После задания всех параметров следует подготовить данные для обучения сети. Для этого следует нажать на кнопку "Подготовка к работе обучающих данных" на форме "Обучение нейронной сети". После нажатия этой кнопки будет сформировано обучающее и тестовое множество. Затем все данные будут масштабированы. Далее необходимо создать нейронную сеть, удовлетворяющую заданным ранее параметрам. Для этого необходимо нажать кнопку "Создание сети". При нажатии на эту кнопку происходит создание сети с заданными параметрами. Также происходит инициализация сети, то есть случайным образом задаются первоначальные значения для всех весов и смещений в сети. Значения весов и смещений лежат в интервале $[-1;1]$. После инициализации сети пользователь может увидеть значения всех весов и смещений.

При нажатии на кнопку "Обучение" запускается процесс обучения сети. По завершении обучения пользователь увидит количество циклов, потребовавшихся для обучения и ошибку сети на обучающем множестве. Также можно увидеть выход сети при подаче конкретного вектора.

После обучения необходимо протестировать обученную сеть. При нажатии на кнопку "Тестирование" на обученную сеть подается тестовое множество и вычисляется значение ошибки на этом множестве. Если ошибка сети на тестовом множестве оказывается больше установленной пользователем величины, процесс

обучения следует повторить. Обучение происходит до тех пор, пока ошибка сети на тестовом множестве не станет меньше установленной.

Разработанная нейросетевая экспертная система позволяет диагностировать вирусный гепатит по 6 или 25 показателям. Ввод данных для диагностики может осуществляться двумя способами, не зависимо от количества используемых показателей:

- 1) загрузка информации из базы данных;
- 2) ввод данных с клавиатуры.

Для проведения диагностики необходимо загрузить обученную сеть. После нажатия кнопки "Поставить диагноз", данные пациента будут поданы на вход сети и обработаны. В окне "Компьютерный диагноз" пользователь увидит диагноз, поставленный сетью и степень уверенности сети в правильной диагностике. На основании диагноза будут выданы рекомендации. Если данные для диагностики были загружены из базы данных, то после проведения диагностики необходимо занести компьютерный диагноз в базу.

Тестирование разработанной экспертной системы на реальных данных показало результаты, практически совпадающие с результатами, полученными при использовании MATLAB.

Список литературы

1. Калнберз В. К. Гемотрансфузия в современной хирургии / В. К. Калнберз, М. В. Калнберза, Ю. В. Жукова // Проблемы гематологии и переливания крови, 1999. – №2. – С. 9 – 10.
2. Горбань А. Н. Нейронные сети на персональном компьютере / А. Н. Горбань, Д. А. Россиев. – Новосибирск: Наука. Сибирская издательская фирма РАН, 1996. – 276 с.
3. Статистический и нейросетевой методы идентификации и прогнозирования в медицине / В. Б. Лифшиц, Т. И. Булдакова, С. И. Суятинов, С. В. Колентьев // Информационные технологии. – 2004. – №3. – С. 60 – 63.
4. Щетинин В. Г. Применение компьютерных «нейронных сетей» в клинической лабораторной диагностике / В. Г. Щетинин, А. А. Соломаха // Клиническая лабораторная диагностика. 1998. №10. С. 21 – 33.
5. Соломаха А. А., Марьин В. К., Горбаченко В. И., Артюхин В. В. Нейросетевая диагностика вирусных заболеваний печени для объективизации эндоэкологической ситуации в регионах России // Экология и промышленность России, 2005, февраль, – С. 40 – 41.

Построение модели подсистемы комплексной защиты информации как системы случайных величин

Арьков П. А.,

*кафедра Информационной безопасности Волгоградского государственного университета
г. Волгоград ул. 50-лет Октября 30-14
lpr11107@rol.ru*

В настоящее время системы защиты информации для информационной системы создаются практически для каждой организации. При этом используются подходы и решения которые должны отвечать общим требованиям Федеральной службы по техническому и экспортному контролю. Однако основным недостатком данного подхода является то, что в нем требуется только наличие соответствующих средств защиты в системе и не учитывают вероятности их преодоления.

Предлагается модель подсистемы комплексной защиты информации основанная на теории графов и теории вероятностей с помощью которой рассчитываются вероятность преодоления средств защиты за заданное время ΔT .

Рассматриваемая модель атаки на информационную систему представляет собой граф $G(V, E)$, где $V = A \cup T$. Вершинами графа являются два подмножества $A = \{A_1, A_2, A_3, \dots, A_m\}$ ресурсов информационной системы и $T = \{T_1, T_2, T_3, \dots, T_n\}$ различных угроз для этих ресурсов. Наличие угрозы T_i для ресурса системы A_j будет выражаться в присутствии ребра E_{ij} инцидентного вершинам T_i и A_j и направленного от T_i к A_j .

Система безопасности должна быть системой с полным перекрытием, т.е. системой где для каждой угрозы ресурсу должно иметься, по крайней мере, одно средство противодействия. На графе G это выразится в наличие как минимум одного средства защиты для каждого ребра E_{ij}.

Различные средства противодействия угрозам представляются в виде множества S = {S₁, S₂ ... S_M}.

Таким образом, можно перейти к модели подсистемы защиты информации, которая представляет собой множество графов {G_{ij}}, i=1...n, j=1...m. Каждый из этих графов состоит из истока (угрозы T_i) и стока (ресурса A_j). Между истоком и стоком расположено множество вершин отображающих различные средства защиты {S₁, S₂ ... S_M} связанных между собой в соответствии с тем в какой последовательности необходимо их преодолевать.

Рассматриваются все возможные пути достижения вершины A_j из вершины T_i в графах G_{ij}. Каждый путь представляет собой последовательность средств защиты {S₁, S₂ ... S_N}, которые необходимо преодолеть для получения доступа к защищаемому ресурсу.

Т.е. путь преодоления подсистемы защиты может быть представлен как система случайных величин (T₁, T₂, ... T_N), где T_k, (k=1..N) – это величина представляющая собой время затрачиваемое на преодоление k-го средства защиты, распределенная по определенному закону. Каждому средству защиты S_k ставится в соответствие функция f_k(T_k) k=1..N, представляющая собой плотность вероятности для времени затрачиваемого на преодоление k-го средства защиты. Принимая величины (T₁, T₂, ... T_N) независимыми можно легко перейти от совокупности функций плотностей вероятностей отдельных величин к плотности вероятности системы случайных величин в целом (1).

$$f(T_1, T_2, \dots, T_N) = f_1(T_1)f_2(T_2)\dots f_N(T_N) \quad (1)$$

Рассмотрим данную систему случайных величин в виде N-мерного пространства. Тогда вероятность преодоления подсистемы защиты за время ΔT фактически представляет собой вероятность попадания случайной точки в этом N-мерном пространстве в некоторую область D которая определяется в соответствии с (2).

$$D = \begin{cases} T_1 + T_2 + \dots + T_N \leq \Delta T \\ T_1 \geq 0 \\ T_2 \geq 0 \\ \dots \\ T_N \geq 0 \end{cases} \quad (2)$$

Т.е. величины принимаемые T₁, T₂, ... T_N неотрицательны и их сумма не должна превышать ΔT. Для расчета вероятности P_{пр} преодоления системы защиты за заданное время ΔT используется формула (3):

$$P_{пр} = \int_D \dots \int f(T_1, T_2, \dots, T_N) dT_1 dT_2 \dots dT_N \quad (3)$$

Таким образом, используя данную модель осуществляется расчет вероятности преодоления подсистемы защиты информации по определенному пути за заданное время. Получены результаты расчета вероятности преодоления системы защиты для случая, когда величины (T₁, T₂, ... T_N), где T_k, (k=1..N), распределены по нормальному и Пуассоновскому закону.

Построение таксономии «элементы» для интеллектуальных систем автоматизированного проектирования

Ахремчик О. Л., доцент
Россия, 170026, Тверь, Никитина, 22,
Тверской государственной технической университет,
кафедра автоматизации технологических процессов,
Email: axremchic@mail.ru

Обобщается опыт построения таксономий для баз данных и знаний интеллектуальных систем. Выделяются особенности, отражаемые таксономией. Вводится определение элемента. Рассматриваются способы определения тождественности элементов. Операции построения таксономии проводятся при

ограничении на набор свойств элементов. Представляется правило принадлежности элементов таксономии. Рассматривается описание класса при введении отношения "тип-подтип". Переход от типа к подтипу предлагается обеспечивать на основе декомпозиции свойств типа и агрегации свойств подтипа. В качестве закона совпадения типов предлагается проверка свойства гомоморфизма на упорядоченных универсумах, представляющих ветви таксономии.

Построение таксономии рассматривается на примере разработки интеллектуальной САПР систем автоматизации и управления технологическими процессами. Первым шагом построения машинно-обрабатываемых описаний объектов и процессов проектирования, используемых в интеллектуальных САПР, является введение классов. Используя онтологический подход можно говорить о необходимости построения таксономии "элементы", являющейся детализацией общей системы понятий, применяемой для построения моделей процесса проектирования. Таксономия задает способ расположения объектов по классам, характеризующий отличия между ними. Таксономия отражает: существование определенного свойства у элемента; принадлежность элемента к определенному классу; существование отношений между элементами, входящими в состав проектируемой системы автоматизации.

Предметной областью построения таксономии "элементы" является совокупность технических средств и программных модулей, структурированных логически и организованных физически для достижения целей, определяемых задачами автоматизированного проектирования систем управления. Система управления представляется: $S = E \cup P \cup R$, где S – модельное описание системы; E – множество элементов системы; P – множество свойств элементов; R – множество отношений, заданных на P и E .

Понимая под целью проектирования построение набора описаний элементов и связей между ними, отражающих существенные свойства разрабатываемой системы элементом будем называть упорядоченную пару $\Theta = (t, e)$: $e \in T$, $e \in E$, $t \in T$, $t \subset E$, где E – множество значений (технических средств и программных модулей), используемых при создании новой системы; T – множество типов элементов, такое, что каждый тип $t \in T$ есть множество значений $e \in E$.

Составляющие таксономии $\Theta_1 = (t_1, e_1)$ и $\Theta_2 = (t_2, e_2)$ тождественны, если $t_1 = t_2$ и $e_1 = e_2$.

Под связью элементов Θ_1 и Θ_2 в системе автоматизации понимается неупорядоченная пара (Θ_1, Θ_2) . Связи (Θ_1, Θ_2) и (Θ_3, Θ_4) тождественны, если

$$\Theta_1 = \Theta_3 \text{ и } \Theta_2 = \Theta_4 \text{ или } \Theta_1 = \Theta_4 \text{ и } \Theta_2 = \Theta_3.$$

Учет введенных понятий при построении таксономии позволяет снизить число ошибок, связанных с появлением одного и того же значения с разными типами.

Объекты (t, e) организуются в подгруппы, каждая из которых имеет обобщенное имя. Использование подгрупп основывается на выполнении условий, позволяющих представить подгруппы путем установления связи элемента с именем класса: в подгруппу могут добавляться новые элементы; существующие элементы могут удаляться; в подгруппу может входить любое число элементов одного и того же типа.

При построении таксономии над элементами выполняются основные операции: добавление элемента в подгруппу; удаление элемента из подгруппы; просмотр свойств элемента: изменение свойств элемента. Операции проводятся в рамках ограничения, определяющего фиксированный набор свойств элемента. Введение ограничения связано с зависимостью системообразующего оператора проектируемой системы со свойствами элемента.

Введем обозначения E_{tax} – множество элементов, составляющих таксономию, T_{tax} – множество типов элементов, C_{tax} – множество связей между элементами, обеспечивающих появление системообразующих признаков в процессе проектирования, R_{tax} – множество неупорядоченных пар (t_1, t_2) типов, для которых в C_{tax} существует связь между элементами типов t_1 и t_2 . Принадлежность элементов, используемых при проектировании систем автоматизации, классам таксономии определяется: $T_{\text{tax}} = \{ t : \exists \Theta = (t, e), \Theta \in E_{\text{tax}} \}$;

$$R_{\text{tax}} = \{ (t_1, t_2) : \exists \Theta_1, \Theta_2; \Theta_1 = (t_1, e_1) \in E_{\text{tax}}, \Theta_2 = (t_2, e_2) \in E_{\text{tax}}, (\Theta_1, \Theta_2) \in C_{\text{tax}} \}.$$

Потенциально на элементах могут задаваться отношения порядка: некоторое отношение порядка над T_{tax} ; некоторое отношение порядка τ над элементами типа E_{tax} для каждого $t \in T_{\text{tax}}$; некоторое отношение порядка над подмножеством множества T_{tax} , содержащим элементы, связанные с t в R_{tax} ; отношения порядка над проекциями связей C_{tax} . Разработка таксономии на основе эвристических приемов и экспертных методов сводится к выделению отношений порядка вида τ .

Увеличение семантической мощности описания типов элементов состоит в определении отношений "тип-подтип" в пределах множества T_{tax} . Элемент типа t' является элементом типа t , если

$$t \succ t',$$

где \succ – обозначение иерархического частичного порядка, представляемое деревом над множеством T_{tax} . Дерево формально описывается связным графом без контуров.

Операторы перехода от типа к подтипу $\psi: T_T \rightarrow T_{\text{SUBT}}$ являются сложными функциями многих переменных. При их реализации используются функции ψ' такие, что аргументы из списка ψ' входят в

список аргументов ψ и любой результат ψ имеет базу, все компоненты которой есть элементы списка результатов ψ' . Применение ψ к конкретному кортежу свойств типа в целях получения свойств подтипа сводится к вычислению: кортежей, описывающих свойства подтипа путем применения функций декомпозиции свойств типа элементов, являющихся аргументами ψ к составляющим кортежа, описывающего свойства типа; кортежа, описывающего свойства типа путем агрегации свойств подтипа элемента.

Для данного типа $t \in T_{\text{tax}}$ определим $[t]$ – максимальный тип, по отношению к которому t является подтипом. Максимальный тип является единственным и определяет группы элементов, входящие в заданный список технического задания на проектирование и задающих структурные свойства разрабатываемой системы. Введение максимального типа модифицирует определение тождественности элементов таксономии до проверки выполнения условий $[t_1] = [t_2]$ и $e_1 = e_2$.

Пусть $t_{\min}(t, e)$ – наименьший тип t' для которого $(t, e) = (t', e)$ в смысле тождественности, определенной выше. Предположим, что $([t], e) \in E_{\text{tax}}$ и $t_{\min}([t], e)$ существует. Элемент (t, e) не входит в таксономию, если t и $t_{\min}([t], e)$ не сравнимы по отношению \succ . В противном случае некоторый новый элемент таксономии $(t, e) \in E_{\text{tax}}$.

Необходимость учета отличий элементов приводит к использованию подтипов для подтипов. Если в пределах подтипа выделяется промежуточный подтип, то: данный подтип действует в течение всего жизненного цикла таксономии; данный подтип, возможно, является более общим подтипом для других подтипов.

Класс элементов как максимальный тип характеризуется кортежем свойств:

$$[t] = \langle P_1, P_2, \dots, P_k \rangle.$$

Новый элемент определяется как подтип максимального типа, наследующий все его свойства. Кроме этого он имеет дополнительные свойства, отражающие детализацию описания. Для регистрации существенных свойств исходят из результата идентификации элемента в составе проектируемой системы и окружающей среды. В ходе построения таксономии рассматривается логическая и классификационная идентификация.

Построение таксономии "элементы" по свойствам отражается следующими рассуждениями. Пусть $E^* = \{E_1^*, E_2^*, \dots, E_n^*\}$ – множество элементов данного типа, выпускаемых промышленностью. Для множества E^* существует набор свойств P . Тогда

$$\forall e^* \in E^* : \exists k \in P_0 \Rightarrow \exists E_0^* = \{e_{01}^*, e_{02}^*, \dots, e_{0m}^*\} : \forall e_j^* \in E^*, e_j^* \in e_{0i}^*,$$

где E_0 – множество типов элементов E^* ; e_{0i}^* – i -й тип (класс); P_0 – декартово произведение свойств, входящих в набор свойств типа элементов.

С каждым свойством связано некоторое ссылочное отношение, задаваемое экземплярами элементов. Для определения отношений используется таксономический универсум U . Если вершина $u \in U$ и дуги, исходящие из u , входят в u_1, u_2, \dots, u_n в указанном порядке, то универсум $u(u_1, u_2, \dots, u_n)$ задается записями вида $u_i(u_j)$ при направлении связи от i -ой вершины к j -ой. Таксономия "элементы" представляет упорядоченный универсум, вершины которого являются подтипами элементов, экземплярами элементов, свойствами. Примером раскрытия универсума является:

$$\text{SUB} = \text{Id } U \{ \text{микроконтроллеры, промышленные компьютеры} \},$$

где SUB – обозначение универсума с частичным порядком, Id – отношение тождества.

Закон совпадения типов при отнесении к подтипу отражает свойство простого гомоморфизма: если U и V – универсумы и ψ – отображение из множества вершин U в множество вершин V , т. е. $\psi: U \rightarrow V$. U и V обладают простым гомоморфизмом, если:

$$|\psi(u)| = |u|, \psi(u_i) = \psi(u)_i, |u| \geq i, \forall u \in U,$$

где $|u|$ – число дуг, исходящих из вершины u ; u_i – вершина, на которую указывает i -я дуга, исходящая из вершины u .

Если V – упорядоченный универсум некоторым отношением SUB ("тип-подтип") и $\psi: U \rightarrow V$ такое, что

$$|\psi(u)| = |u|, \psi(u_i) = \psi(u)_i \in \text{SUB}, |u| \geq i, \forall u,$$

то данное выражение определяет не простой гомоморфизм, а выпуклое отображение, задающий обобщенный закон совпадения типов в таксономии.

Проведенный комплекс исследовательских работ позволяет представить набор составляющих информационного обеспечения интеллектуальных САПР: множество типов элементов; множество экземпляров элементов данного типа; множество подтипов данного типа; отношения порядка, позволяющие

формировать составные части систем управления из элементов; граф перехода от типа элементов к подтипу и экземплярам. Система понятий, отражающая названия классов, индивидуальна и развивается как по декларативной, так и по процедурной составляющей, порождая базу терминов, по которым идентифицируются элементы в таксономии.

Комбинаторная модель и методы оптимизации сетей с технологией АТМ

Баклан А. Н.,

*Институт кибернетики им. В.М.Глушкова НАНУ,
Украина, 02091, г. Киев, ул. Ревуцкого, 11-Г, кв. 13
E-mail: a_baklan@yahoo.com*

Широкое распространение телекоммуникационной технологии АТМ делает актуальной разработку моделей и методов оптимизации сетей с технологией АТМ. Технология АТМ (от Asynchronous Transfer Mode – асинхронный режим передачи) – это технология высокоскоростной одновременной передачи трафика всех видов (данные, голос и видео) в сетях с коммутлируемыми каналами [1]. Любой передаче данных в сети АТМ предшествует организация виртуального соединения (коммутлируемого или постоянного) между отправителем и получателем данных, что упрощает процедуры маршрутизации. Передача данных осуществляется путем обмена фиксированными 53-байтовыми пакетами данных (т.н. ячейками). Термин «асинхронный» в названии технологии означает, что ячейки, принадлежащие одному соединению, могут поступать в канал связи в непредсказуемые моменты времени (т.е. нерегулярно). Существует несколько протоколов верхнего уровня, обеспечивающих передачу поверх АТМ трафиков других типов.

При создании соединения АТМ может требоваться определенная пропускная способность на всем пути между двумя конечными узлами этого соединения, а также устанавливаться требования к показателям качества услуг (Quality of Service). К основным показателям качества относятся:

- средняя задержка в передаче ячеек – CTD (Cell Transfer Delay);
- вариация средней задержки;
- доля (или вероятность) потерянных ячеек – CLR (Cell Loss Ratio).

Для обеспечения качества услуг технология АТМ имеет несколько категорий услуг (видов трафика) (Categories of Service):

- а) CBR (Constant Bit Rate) – передача с постоянной скоростью;
- б) VBR (Variable Bit Rate) – передача с переменной скоростью;
- в) ABR (Available Bit Rate) – передача с доступной скоростью;
- г) UBR (Unspecified Bit Rate) – передача с неустановленной скоростью.

Введение категорий услуг увеличило преимущества АТМ, сделав эту технологию пригодной практически для неограниченного спектра применений.

Рассмотренная выше специфика технологии АТМ не позволяет использовать традиционные методы оптимизации телекоммуникационных сетей для решения практических задач.

Например, для сетей с технологией АТМ, содержательную постановку одной из наиболее важных задач оптимизации телекоммуникационных сетей – задачи оптимального выбора пропускных способностей каналов связи – можно сформулировать следующим образом.

Задана структура сети АТМ, состоящая из коммутаторов, соединенных каналами связи. Для каждого канала связи определена его длина. Заданы также требования в передаче потоков трафиков CBR, VBR и ABR для каждой упорядоченной пары коммутаторов. Кроме того, для каждого канала связи заданы величины потоков трафиков CBR, VBR и ABR. Пропускная способность каждого канала связи пропорциональна пропускной способности базового канала. Известны также удельные стоимости каналов связи разной пропускной способности на единицу длины. Необходимо для всех каналов связи выбрать такие количества базовых каналов для трафика CBR, общего трафика VBR и ABR, а также часть общего трафика, занятую трафиком VBR, при которых стоимость сети будет минимальной. При этом должны выполняться следующие ограничения на показатели качества обслуживания: CLR и CTD для трафика CBR, CLR и CTD для трафика VBR, CTD для трафика ABR.

Постановка задачи оптимального выбора пропускных способностей каналов связи для разных видов трафика приведена в работе [2]. Автором предлагается обобщенная постановка рассматриваемой задачи, которая является задачей дискретной оптимизации и относится к NP-сложным задачам.

Для решения задач дискретной оптимизации наибольшее распространение получили приближенные алгоритмы. Это объясняется несколькими причинами. Во-первых, подавляющее большинство таких задач относится к NP-сложным задачам, точное решение которых является проблематичным даже с использованием мощной вычислительной техники [3]. Во-вторых, целевые функции имеют, как правило, большое количество локальных экстремумов. В-третьих, во многих прикладных задачах входные данные задаются с погрешностями, что делает нецелесообразными те существенные вычислительные затраты, которые необходимы для нахождения их точного решения. В-четвертых, положенные в основу приближенных схем идеи метаэвристики позволяют создавать алгоритмы, которые могут решать не одну, а целый класс близких по постановке оптимизационных задач [4].

Предлагаемая автором комбинаторная модель задачи позволяет применить следующие приближенные методы для ее решения:

- локальный поиск;
- локальный поиск с повторами;
- имитационный отжиг;
- ускоренное имитационное моделирование;
- генетические алгоритмы.

С этой целью были разработаны несколько приближенных алгоритмов, для исследования эффективности которых было написано программное обеспечение и проведены вычислительные эксперименты.

Список литературы

- [1] Назаров А. Н., Симонов М. В. АТМ: технология высокоскоростных сетей. М.: ЭКО-ТРЕНДЗ, 1997.
- [2] Зайченко О. Ю. Оптимізація характеристик мереж з технологією АТМ // Системні дослідження та інформаційні технології. - 2002. - №3. - С.57- 73.
- [3] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. - М.: Мир, 1982.
- [4] Blum C., Roli A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison // ACM Computing Surveys. – 2003. – Vol. 35, №3. – P. 268–308.

Искусственный интеллект в Ghostgrid

Баклановский Максим, Корнев Дмитрий

Екатеринбург, мат-мех УрГУ

Ghostgrid – это система распределенных вычислений, удовлетворяющая следующим требованиям. Во-первых, она динамична, то есть число машин, подключенных к Ghostgrid может варьироваться в большом диапазоне. Любая машина в любой момент времени может получить доступ к ресурсам грида, расплачиваясь своими ресурсами. Учитывая простоту подключения к гриду, а так же постоянный рост числа машин в сети Интернет, Ghostgrid может быстро вырасти, поэтому мы изначально закладываем в него механизмы самоорганизации. Во-вторых, грид должен удовлетворять требованию максимальной отказоустойчивости не только в целом, как Интернет, но и по отношению к каждому подключенному к ней узлу. В-третьих, система прогнозирования времени выполнения задач должна иметь в среднем высокую точность прогнозов, с возможностью повышения этой точности при необходимости. В-четвертых, система должна быть управляема. В этой статье описываются некоторые базовые понятия, которые мы закладываем в основу Ghostgrid.

Искусственный интеллект является неотъемлемой частью системы распределенных вычислений Ghostgrid. Под искусственным интеллектом мы понимаем И-модули принятия решений в условиях нечеткой информации о постоянно меняющихся внешних условиях, на основе распространяемых в системе новостных пакетов, таких как прогноз, прогноз-результат, рейтинг. Каждый узел Ghostgrid целеустремлен. При этом критерии оценки своего текущего состояния узел вырабатывает сам в соответствии с заложенной в нем нормативной базой. Для достижения своей цели узлы используют самообучающиеся И-модули выбора стратегий.

В системе Ghostgrid все машины делятся на «клиентов» и «серверов». Основной объем вычислений происходит на клиентах, в то время как сервера ответственны за поддержание целостности грида и формирование его динамичной структуры. Каждый клиент приписан к той или иной динамически

формируемой группе узлов, называемой облаком. За взаимодействие клиента и грида отвечает сервер клиента. При этом каждый клиент самостоятельно выбирает себе сервер на основании рассылаемых другими серверами и клиентами новостных рассылок, в которых в том числе распространяются характеристики серверов, такие как количество клиентов, подключенных к ним, вычислительная активность, пропускная способность каналов и другие. Новостные рассылки являются основной структурообразующей единицей Ghostgrid. По этим новостным рассылкам каждый клиент может составить для себя рейтинг того или иного сервера. По данному рейтингу возможно построить несколько стратегий выбора оптимального сервера. Первая, самый очевидная, но не самая эффективная – это выбор наилучшего сервера. Другим вариантом является случайный выбор из группы лучших серверов. Но в таком случае будет происходить отталкивание от более слабых серверов, что наверняка приведет к недоиспользованию ресурсов грида. Применения каких-либо алгоритмов из антагонистических игр вызывает сомнение по тем причинам, что, во-первых, необходима постоянная самообучаемость, во-вторых, набор параметров, от которых зависит принятие решения, может меняться в процессе жизни грида, и в-третьих, каждый узел находится в условиях нечеткой информации, поскольку получает новостные рассылки в основном лишь от группы узлов (облако новостной рассылки), с которыми он взаимодействует. На наш взгляд, здесь будет необходимо использовать И-модуль, который, самообучаясь, и обеспечит быстрое решение данной задачи выбора клиентом сервера.

В новостных рассылках будет необходимо передавать большие объемы информации большому количеству адресатов. Для обеспечения достаточной гарантии доставки этих рассылок существует два пути, которые демонстрируются мультикастовыми протоколами NORM и ALC. Отказ от обратной связи неприемлем. Неприемлемо и жертвовать скоростью в гриде. Поэтому, мы будем использовать нечеткие обратные связи. В стеке протоколов И-модули будут использоваться не только на прикладных уровнях, но возможно и для решения задач по контролю уровня заполнения каналов, по маршрутизации. Наше видение грида таково, что искусственный интеллект его пронизывает, причем на всех уровнях. В Ghostgrid для формирования облаков новостных рассылок тоже используется И-модуль, поскольку он позволяет определять, с кем в данный момент данному узлу общаться наиболее выгодно для достижения целей. Обратной связью в данном случае может быть информация о том, насколько другие узлы информированы о данном узле. Таким образом, узлы будут объединяться в некие нечеткие социумы, в которых они будут взаимодействовать по средствам рассылки. При этом формализовать механизм выбора узлом того или иного социума не представляется возможным, поскольку факторы, влияющие на принятие решения, для каждого узла могут быть различными, а информация нечеткая. Важно отметить, что решение каждый узел принимает самостоятельно, основываясь на его собственном понимании вопроса выгоды. Также не исключено, что какой-то узел может оказаться в нескольких социумах одновременно.

Из предыдущей задачи сразу вытекает задача идентификации: каким образом определить, к какому социуму относится узел. Такую задачу можно решить с помощью алгоритма, учитывающего активность общения каждого узла с другими узлами. Но отследить, каким образом социумы эволюционировали во времени, какой из них в какой переходил, с кем объединялся и так далее без использования И-модуля не представляется возможным. Но данная задача схожа с распознаванием образов, для которого традиционно используется искусственный интеллект. В Ghostgrid предусмотрена система прогнозирования длительности выполнения задач. Это осуществляется с помощью И-модуля, который при загрузке задачи на Ghostgrid выдает прогноз. После выполнения задачи, результат сравнивается с прогнозом, рассылаются новостные пакеты и сервера, распределяющие задачи по вычислительным узлам и входящие в облако этой новостной рассылки досамообучаются. Напомним, что задача может поступить на выполнение с любого узла Ghostgrid, после чего сервера распределяют эту задачу на основании информации о других узлах. Эти данные они получают из новостных рассылок. Важно, что рассылки не обязательно ограничиваются одним социумом. Узлы способны определять значимость рассылаемой ими информации, согласно чему расширяя или сужая круг адресатов.

Мы понимаем, что вскоре настанет такой момент, когда администрировать грид вручную станет невозможным. Для этих целей мы предлагаем ввести «странников», которые бы перемещались по гриду, собирая информацию и осуществляя ту или иную политику, заданную администратором, с помощью взаимодействия с некоторыми механизмами управления серверами, такими как коэффициенты в нейронных сетях их И-модулей. Для достижения своей цели странник также постоянно самообучается воздействуя на грид и оценивая результаты своего воздействия.

Таким образом, мы стремимся построить принципиально новую систему, в которой бы отсутствовали четкие критерии, а искусственный интеллект, распределенный по всем узлам, пронизывал бы ее на всех уровнях архитектуры.

Комплекс программ для решения систем нелинейных интегро-дифференциальных уравнений (ИДУ), содержащих производные и интегралы дробного порядка

Бандурин Н. Г.

доктор технических наук, профессор кафедры строительной механики Волгоградского государственного

архитектурно-строительного университета;

E-Mail: bandurin_n@bayard.ru

Игнатьев В. А.

доктор технических наук, профессор, заведующий кафедрой строительной механики Волгоградского государственного

архитектурно-строительного университета;

400074, Волгоград-74, Волгоград, Академическая ул., 1, тел.: (8442) 94-23-08

Программный комплекс предназначен для численного решения систем нелинейных интегро-дифференциальных уравнений, содержащих производные и/или интегралы нецелого порядка (небольшого размера одно-, двух и трехмерные краевые и начально-краевые задачи). Задаваемые в качестве исходных данных функции могут иметь кусочно-непрерывный характер.

Системы ИДУ, решаемые с помощью программ рассматриваемого комплекса, должны удовлетворять следующим условиям: система уравнений должна быть корректной и не жесткой; ее решение должно быть достаточно хорошо представимо алгебраическими полиномами; число уравнений должно быть равно числу неизвестных функций; для каждой неизвестной функции в системе уравнений должно содержаться по крайней мере одно уравнение с производными этой функции наивысшего порядка по всем координатам на всей области интегрирования; эти производные не должны быть смешанными; должен быть обеспечен полный набор начальных и краевых условий. Для получения решения, представляемого в текстовой или графической форме, достаточно ввести в память компьютера текст уравнений и необходимые числовые параметры задачи.

Решение начально-краевой задачи для ИДУ получается в результате последовательного выполнения больших шагов длиной H_t в положительном направлении временной оси t . Каждый такой шаг в свою очередь делится на n_t малых шагов, длина которых $h_t = H_t / n_t$. При решении многомерных задач отрезки интегрирования на других осях могут содержать другое число шагов (n_x или n_y). Разрешающая система линейных алгебраических уравнений на большом шаге формируется относительно узловых приращений неизвестных функций [1], поэтому интегралы и производные как целого, так и дробного порядка должны быть выражены через узловые значения неизвестных функций и/или их шаговых приращений. В случае целого порядка производной вектор ее узловых значений выражается через узловые значения собственно функции по формуле дифференцирования [1,2]. Производные и интегралы дробного порядка α ($0 < \alpha < 2$, $\alpha \neq 1$) вычисляются по формулам трапеций, полученным в [3] и [4] соответственно.

При дискретизации производных и интегралов целого порядка многочленами степени n ($2 \leq n \leq 9$) с шагом h_t [5] оценка погрешности будет $O(h_t^{n+1})$, в то время как для интегралов и производных дробного порядка с использованием формулы трапеций - $O(\delta^2)$. Следовательно, для того чтобы интегралы и производные дробного порядка вычислялись с оценкой $O(h_t^{n+1})$ необходимо выполнение равенства $\delta = h_t^{(n+1)/2}$, то есть шаг следует уменьшить на несколько порядков. Для вычисления значений функций в пределах малого шага h_t с оценкой погрешности $O(h_t^{n+1})$ в программах выполняется интерполяция этой функции на большом шаге H_t с помощью формулы Лагранжа степени n . Таким образом, за счет увеличения объема вычислительной работы сохраняется точность, достигаемая при решении уравнений, не содержащих производных и интегралов дробного порядка. В то же время порядок метода значительно понижается, так как в дополнительных узлах вычисление невязки не выполняется.

Для обеспечения возможности задавать на разных участках оси t систему уравнений различными выражениями вводится функция $g(a, b)$ такая, что

$$g(a, b) = \begin{cases} 1, & \text{если } t \in [a, b]; \\ 0, & \text{если } t \notin [a, b]. \end{cases}$$

Эффективность применения пакета программ иллюстрируется ниже результатами решенных тестовых примеров.

Пример 1. Уравнение 2-го порядка, описывающее колебания массы m в промежутке времени $0 < t < 16$ при условии, что сила сопротивления пропорциональна производной дробного порядка 1.5, а внешняя сила, равная 8, действует только в период времени от 0 до 1 [5]

$$[mD^2w(t) + \beta D^{1.5}w(t) + kw(t)]g(0, 20) - 8g(0, 1) = 0.$$

Начальные условия: $w(0) = \dot{w}(0) = 0$.

График решения этого уравнения, полученного с помощью рассматриваемых универсальных программ для $m = 1.0$, $\beta = 0.5$ и $k = 0.5$, имеет 4 экстремума и практически совпадает с графиком, содержащимся в работе [5], который был получен авторами в результате решения этого уравнения путем сведения одного интегро-дифференциального уравнения к системе дифференциальных уравнений и последующего ее решения.

Пример 2. Для нелинейного уравнения

$$w_{xx} + w_{yy} - \alpha w_t(1 + w_t^2) + 2\sin(x + y - t) - \alpha \cos(x + y - t)(1 + \cos^2(x + y - t)) = 0,$$

имеющего точное решение $w(x, y, t) = \sin(x + y - t)$, необходимо найти решение на равномерной сетке с шагом $h \approx 0.2$ и $n_t = 10$ в области $\{0 \leq x, y \leq \pi/2, 0 \leq t \leq 600\}$. Краевые и начальные условия:

$$w(0, y, t) - \sin(y - t) = 0, \quad w(\pi/2, y, t) - \cos(y - t) = 0, \quad w(x, 0, t) - \sin(x - t) = 0,$$

$$w(x, \pi/2, t) - \cos(x - t) = 0, \quad w(x, y, 0) - \sin(x + y) = 0.$$

Максимальное отклонение приближенного решения от точного при $\alpha = 1$ оказалось равным: на первом большом шаге - $\delta = 4.221 \cdot 10^{-7}$, на последнем 268-м шаге - $\delta = 4.259 \cdot 10^{-7}$. Видно, что δ остается практически неизменным даже на большом промежутке оси t , что объясняется абсолютной устойчивостью решения уравнения, которое моделирует устойчивый физический процесс распространения тепла в прямоугольной пластинке. При $\alpha = -1$ точное решение остается прежним, но численное решение разрушается немедленно, откуда, по-видимому, можно сделать вывод, что не существует устойчивый физический процесс, описываемый таким уравнением.

Пример 3. Краевая задача для существенно нелинейного интегро-дифференциального уравнения

$$y'' + yy' + \exp(y') + \int_0^{36} sy(s)ds + \int_0^{36} y(s)y''(s)ds + \sin x(1 - \cos x) - \exp(\cos x) - \sin 36 + 36\cos 36 - 0.25\sin 72 + 18 = 0.$$

Решение следует найти на отрезке $R = \{0 \leq x \leq 36\}$.

$$\text{Краевые условия: } y(0) = 0, \quad y'(36) + \int_0^{36} y(s)ds - 1 = 0.$$

Точное решение: $y(x) = \sin x$. Начальное приближение: $y_0 = 0$.

Задача была решена при делении отрезка интегрирования на 150 частей, причем максимальная ошибка оказалась равной $7.5 \cdot 10^{-9}$.

Пример 4. Система нелинейных интегро-дифференциальных уравнений, содержащих интеграл дробного порядка 0.5

$$U_{xx} - U_t - \sin(U) + UV + J^{0.5}(U) + \int_0^x \int_0^t U_t(\alpha, \beta)V(\alpha, \beta)d\alpha d\beta - \sin(x+t) - xt + 1 = 0, \quad V + U + \int_0^x \int_0^t U(\alpha, \beta)d\alpha d\beta = 0.$$

Краевые и начальные условия: $U(0, t) - t = 0$, $U(1, t) - t - 1 = 0$, $U(x, 0) = 0$.

Решение следует найти в области $\{0 \leq x \leq 1, 0 \leq t \leq 4\}$.

Ниже приводятся значения функций $U(0.4, 4.0)$ и $V(0.4, 4.0)$, вычисленные при разных значениях шага сетки h .

h	$U(0.4,4.0)$	$V(0.4,4.0)$
0.2	2.5834	-5.4792
0.1	2.5779	-5.4693
0.05	2.5780	-5.4690
0.025	2.5780	-5.4689

Получено достаточно устойчивое решение.

Список литературы

1. Бандурин Н. Г. Новый численный метод порядка n для решения интегро-дифференциальных уравнений общего вида // Вычислительные технологии. 2002. Т. 7, № 2. С. 3-10.
2. Игнатьев В. А., Игнатьев О. В. О численном решении дифференциальных и интегродифференциальных уравнений на основе матриц интегрирования и дифференцирования. / Вестник ВолгГАСУ. Серия: Технические науки. Вып. № 4 (12). Волгоград, 2004. с. 140-154.
3. Edwards J. T., Ford N. J., Simpson A. C. The numerical solution of linear multi-term fractional differential equations: systems of equations // Journal of Comp. and Appl. Math. 2002. V. 148. P. 401-418.
4. Diethelm K., Ford N. J., Freed A. D., Luchko Yu. Algorithms for the fractional calculus: A selection of numerical methods // Comput. methods in appl. Mech. and engrg. 2004. № 8. P. 3-34.
5. Trinks C., Ruge P. Treatment of dynamic systems with fractional derivatives without evaluating memory-integrals // Computational Mechanics. 2002. № 29. P. 471-476.

Нейросетевое обучение с подкреплением для моделирования поведения конкурентов в экономических играх

Бендерская Е. Н., доцент, Ахин М. Х.,
*Санкт-Петербургский государственный политехнический университет,
195251, Санкт-Петербург, ул. Политехническая, д. 29
Ice_Phoenix@rambler.ru, bender@sp.ru*

Современный мир, с его постоянно увеличивающейся скоростью жизни, ставит все новые и новые задачи перед специалистами всех отраслей. В области экономики одной из таких задач является задача прогнозирования поведения компаний на мировых и прочих экономических рынках. Сложность такого прогнозирования очень велика, и для решения этой проблемы используются различные методы и модели, реализация большей части которых стала возможна только благодаря появлению мощных вычислительных средств, способных справиться с огромным объемом данных.

Но практически все модели, используемые в настоящее время, обладают одним существенным недостатком – в их основу, как правило, закладывается какой-либо определенный алгоритм, с той или иной степенью точности отражающий процессы, протекающие на анализируемых рынках. В случае возникновения на рынке ситуации, не учтенной при разработке алгоритма, прогноз окажется некорректным, что может привести к значительным финансовым и другим потерям [1].

В работе рассматривается проблема принятия экономических решений в условиях частичной или полной априорной неопределенности в изменяющемся во времени окружении.

Целью данной работы является создание методики проектирования моделей, свободных от недостатков алгоритмических, путем использования аппарата нейронных сетей и мультиагентного подхода, с последующим выполнением прогнозов на их основе.

Предлагается использовать возможности современных нейросетевых технологий для разработки ядра информационно-обучающей системы игрового характера. Система предназначена для построения моделей, ситуационного моделирования игровых ситуаций и обучения на их основе принятию оптимальных с экономической точки зрения решений. Исследуется возможность создания методики обучения на основе интерактивного мониторинга результатов обучения с учетом различного характера моделируемых игр, отличающихся степенью априорной неопределенности и динамики окружения.

Особенность разрабатываемого обучающего комплекса нового поколения состоит в способности не только имитировать окружающую среду и поведение условного «противника» по определенным алгоритмам, но и вырабатывать собственное поведение обучающего агента в зависимости от поведения обучаемого.

Основные поведенческие закономерности окружающей среды предлагается рассматривать через формальные игровые модели. Существует несколько разновидностей общих моделей игр, которые допускают нейросетевую реализацию и которые могут использоваться для моделирования экономических процессов [2, 3]:

- Операционные игры (игры с заранее жестко заданным порядком действий-операций каждого агента);
- Антагонистические игры, в том числе:

1. Матричные игры (конечное множество стратегий);
2. Бесконечные антагонистические игры (бесконечное множество стратегий);

- Неантагонистические игры (бескоалиционные и кооперативные игры);

- Позиционные игры (учитывают динамику конфликтов) с полной или неполной информацией, антагонистические и кооперативные, в том числе:

1. Одношаговые игры (игры, разыгрываемые за одну итерацию);
2. Многошаговые игры (игры, представляющие собой некий повторяющийся процесс);
3. Дифференциальные игры (игры с бесконечным числом шагов);

- Рефлексивные игры (игры, учитывающие сложные взаимозависимости между агентами, т.н. рефлексия, за счет анализа и использования общих и частных знаний, имеющихся в системе).

В данной работе для моделирования выбраны многошаговые игры, т.к. они по своей природе являются итерационными, легко поддаются реализации в нейросетевом базисе, занимают промежуточное положение по сложности среди существующих игр и в тоже время являются достаточно приближенными к реальным условиям принятия решений.

На примере формальной экономической задачи, известной как «как «задача о деятельности холдинга консалтинговых компаний» [4], проводится исследование возможности применения нейронных сетей прямого распространения для моделирования подобных ситуаций. Сети этого класса наиболее просты по своей структуре и легко поддаются как программной, так и аппаратной реализации. Известно, что в задачах аппроксимации произвольных функций, нейронные сети прямого распространения позволяют получить максимально возможную точность оценок (в заданных условиях наличия обучающих примеров) и уже используются во многих системах прогнозирования значений экономических индикаторов наряду со стандартными подходами [5]. Если будет показано, что данные сети способны давать корректные прогнозы для рассматриваемого класса задач, то при необходимости их внедрение в существующие системы прогнозирования сможет быть выполнено быстро и с минимальными затратами ресурсов.

Рассматриваемая задача представляет собой многошаговую игру, в которой обучающие последовательности генерируются во время самого процесса игры, т.е. никакого начального набора знаний, на котором мы могли бы инициализировать нейронную сеть нет, и ответов на отдельные обучающие примеры (промежуточные ситуации) тоже нет.

Поэтому обучение нейронной сети строится на основе метода обучения «с подкреплением» (или обучения со штрафами и поощрениями, называемого иногда обучением с критиком), который является вариантом основного принципа динамического программирования Беллмана в приложении к нейронным сетям. Данный метод позволяет значительно сократить число анализируемых вариантов во время определения оптимальной стратегии благодаря более полному учету получаемой в результате моделирования информации. Кроме того, применение именно такого обучения является адекватным по отношению к задаче, для которой используется нейронная сеть.

Для анализа различных вариантов поведения применяются т.н. «жадные» стратегии, которые позволяют как определять оптимальное на данный момент с точки зрения выигрыша поведение, так и исследовать другие варианты для поиска стратегий, максимизирующих возможный выигрыш в будущем.

После проведения моделирования был проведен его подробный анализ, показавший, что получаемые стратегии являются непроигрышными. По результатам анализа были выдвинуты предложения по развитию структуры нейронных сетей для получения прогнозов, более близких к оптимальным.

По результатам исследовательской работы можно сказать, что применение нейросетевого подхода для решения обозначенной проблемы выбора оптимального поведения в условиях неполной информации подтверждает целесообразность развития данного подхода для решения рассматриваемого класса задач. Результаты, полученные в данной работе, свидетельствуют о том, что нейронные сети прямого распространения способны успешно решать задачи экономического прогнозирования, связанного с многошаговой реализацией различных ситуаций с противоречивыми интересами участников экономического процесса. Можно ожидать, что при наличии дальнейших работ в этом направлении будут предложены конфигурация модели и стратегия обучения, которые позволят получать за разумное время оптимальные стратегии и оставят возможность для дальнейшей адаптации к изменениям окружения.

Предлагаемый в работе имитационно-игровой подход на базе нейросетевых технологий, благодаря мультиагентной структуре, способности к самоорганизации и опоре на экономические принципы при построении моделей, позволит легко адаптироваться к потребностям различных задач социально-экономического прогнозирования.

Список литературы

1. Кан М. Технический анализ. - СПб.: Питер, 2003. - 288 с.
2. Петросян Л. А., Зенкевич Н. А., Семина Е. А. Теория игр. - М.: Высшая школа, 1998. - 304 с.
3. Губко М. В., Новиков Д. А. Теория игр в управлении организационными системами. - М.: Синтег, 2002. - 148 с.
4. Терехов С. А. Адаптивные нейросетевые методы в многошаговых играх с неполной информацией. // Лекции по нейроинформатике. Научная сессия МИФИ-2005. Нейроинформатика-2005. – с. 92-135.
5. Головкин В. А. Нейронные сети: обучение, организация и применение. - М.: Радиотехника, 2001. - 256 с.

Сравнительный анализ методов обработки выходных сигналов хаотической нейронной сети

Бендерская Е. Н., доцент;

Жукова С. В.,

Санкт-Петербургский государственный политехнический университет,

195251, Санкт-Петербург, ул. Политехническая, д. 29

bender@sp.ru , flyingbug2@mail.ru

Модель хаотической нейронной сети (ХНС) относится к классу рекуррентных нейронных сетей, одному из самых перспективных, и предложена для решения задач кластеризации сравнительно недавно [1, 2]. Данная модель является наглядным примером того, как могут быть использованы результаты исследований из смежной с нейросетевой тематикой области нелинейных динамических систем для получения нового эффективного метода решения задач кластеризации. Актуальность задач кластеризации, решение которых позволяет выявлять скрытые закономерности, знания в заданном наборе данных, остается большой, особенно в условиях все возрастающего потока информации во всех областях деятельности человека. Большинство разработанных за последнее время новых алгоритмов кластеризации базируются на давно известных классических подходах и в основном направлены на учет априорной информации о кластерах, что лишает их универсальности и делает сильно зависимыми от точности задания априорной информации, поэтому разработка и исследование других, альтернативных подходов является важными.

Задача кластеризации может быть охарактеризована как сложная, так как относится к задачам обучения без учителя (классы, на которые должно быть разбито исходное множество объектов (данных), не заданы,

может быть указано только их количество), и в общем случае она должна быть решена в условиях отсутствия какой-либо априорной информации о кластеризуемых объектах. Разработка систем, способных решать трудно формализуемые задачи, проводится в современных исследованиях на основе использования нейробиологического прототипа – мозга человека. Мозг представляет собой высокопроизводительную систему с неустойчивой нелинейной динамикой, которую создают миллионы нейронов, объединенные в кластеры по принципу функциональной декомпозиции. Поэтому естественным является объединение теории нейронных сетей и теории нелинейных систем и хаоса для извлечения информации из заданного набора данных.

Хаотическая нейронная сеть основана на нелинейном логистическом отображении [3], реализуемом каждым отдельным элементом (нейроном) системы (нейронной сети) и на глобальной связи элементов в системе [4], структура которой задается как логическая решетка. При этом главный используемый для кластеризации эффект, характеризующий работу полученной нелинейной динамической системы - эффект синхронизации, возникающей при определенном сочетании параметров ХНС.

Использование свойств хаоса для решения задач кластеризации основывается на обнаружении аналогии между дендограммой, являющейся результатом работы иерархического метода кластеризации, и бифуркационной диаграммой, описывающей динамику нелинейной системы. В функционировании ХНС [2, 5] удается объединить достижения теории нейронных сетей и теории хаоса.

Необходимость определения подходов для настройки нейронной сети указанного типа вызвана тем, что отсутствие методики выбора параметров ХНС не позволяет воспользоваться в полной мере ее возможностями. Выполненных работ в этом направлении очень мало, как и литературы по хаотической нейронной сети, несмотря на относительно большое число публикаций по нелинейным системам. Это связано с новизной рассматриваемой модели с точки зрения нейросетевого ее воплощения и решения задачи кластеризации. Рассмотренные в работах [4, 6 и др.] аспекты функционирования систем с глобальной связью и эффектов кластеризации (связанных с образованием с течением времени работы системы групп элементов, обладающих схожей динамикой, т.е. проявление эффекта синхронизации) не позволяют связать между собой параметры системы и реальную задачу кластеризации некоторого набора исходных данных. Для использования динамики нелинейной системы, обладающей свойством синхронизации, в распознавании образов (кластеризации), исходные данные должны предопределять состав и параметры системы. Частично такой подход представлен в [1, 2] и базируется на вычислении значений весовых коэффициентов ХНС w_{ij} с учетом взаимного расстояния между точками x_i условного изображения:

$$w_{ij} = \exp\left(-\frac{|x_i - x_j|^2}{2a^2}\right),$$

где a - масштабирующая константа, вычисляемая как среднее расстояние между k -соседями. При этом используется тот факт, что при формализации в задаче кластеризации каждый объект может быть графически интерпретирован как точка в D -мерном пространстве признаков.

Особый интерес представляет рассмотрение различных вариантов обработки результатов работы ХНС. Так как динамика ХНС отражает распознанные кластеры, то для того, чтобы выявить синхронные нейроны, которые после переходного процесса образовали кластеры, (каждый нейрон представляет свою точку кластеризуемого изображения), необходимо выполнить постобработку последовательностей выходов ХНС. Размер последовательностей соответствует числу отсчетов времени наблюдения за функционированием ХНС.

Возможно несколько подходов в обработке – грубая и точная обработка. В работах [1, 2] предлагается использовать грубую обработку с привлечением понятия взаимной информации, вычисляемой классическим способом по Шеннону через энтропию Больцмана. При этом значения выходов закругляются до нулевых и единичных с последующим вычислением вероятностей нахождения нейронов в каждом из этих двух состояний (на периоде наблюдения) и условных вероятностей нахождения в каждом из двух состояний. Считается, что нейроны синхронизированы, если их взаимная информация достаточно велика (превышает выбранный порог). Такой подход требует достаточно большого периода наблюдения и рассмотрения нескольких вариантов кластеризации с разными значениями порогов. Загрубление при обработке значений выходов ХНС в большей степени способствует образованию меньшего числа кластеров – в пределе все точки окажутся в одном кластере, т.к. индивидуальная информация о хаотических аттракторах, по которым двигаются нейроны (последовательность изменений значений их выходов) теряется.

Точная обработка предполагает задание в качестве требования к синхронным нейронам точного равенства значений их выходов [7]. Но для этого должен полностью завершиться переходный процесс, кроме того, такое жесткое требование резко ограничивает возможности ХНС с точки зрения

обрабатываемых данных. В [3, 6] показано, что синхронный режим для системы с глобальной связью возможен только при определенном сочетании ее параметров (инерционной и диссипативной составляющих). При переходе к ХНС это выражается в значениях весовых коэффициентов, а они зависят от исходного изображения. При таком подходе будет часто наблюдаться другое крайнее решение задачи кластеризации – каждая точка образует соевой кластер.

От эффективности (с точки зрения времени, сложности, а главное степени загрубления) постобработки выходных сигналов ХНС, зависит качество (с точки зрения точности разбиения данных на кластеры, что можно проверить для тестовых задач) решения задача кластеризации. Поэтому предлагается проводить расчет взаимной информации нейронов, учитывая не только значения выходов нейронов, относящиеся к одному такту, но и на интервале переменной длины. При этом предполагается, что нейроны могут колебаться синхронно, но в противофазе. Этот способ призван уменьшить степень загрубления при использовании вероятностной обработки.

Второй предлагаемый способ обработки, связан со снижением жесткости требований по точному совпадению значений выходов путем введения допустимой невязки (рассогласования выходов). Еще один «невероятностный» вариант обработки занимает промежуточное положение между двумя представленными выше - учитывается направление изменений значений выходов (увеличение/уменьшение), а также привлекается информация об абсолютных значениях изменений выходов в текущем отсчете или на интервале.

Для проверки применимости и областей предпочтительного использования предложенных вариантов обработки значений выходов ХНС, была разработана имитационная модель ХНС и создан набор тестовых изображений различной степени сложности и с различными характеристиками (количество точек, их плотность, число кластеров, зашумленность). Результаты экспериментов показали, что альтернативные способы обработки выходных сигналов ХНС позволяют сократить вычислительные затраты как за счет уменьшения периода ожидания завершения переходного процесса, так и за счет уменьшения периода наблюдения, и при этом расширить возможности ХНС по обработке разреженных изображений с небольшим числом точек.

Список литературы

1. Angelini L., F. De Carlo, Marangi C., Pellicoro M., Stramaglia S. Clustering Data by Inhomogeneous Chaotic Map Lattices //Physical review letters, v. 85, №3, 2000, pp.554-557.
2. Ormerod C., Bordes N. S., Pailthorpe B. A. Characterising coupled map lattices //Proceeding of HPC-Asia 2001, Sept 24-28, 2001, pp.10-16.
3. Иванова А. С., Кузнецов С. П. О динамике на пороге возникновения хаоса в модельных системах, построенных на базе логистических отображений с инерционной и диссипативной глобальной связью. // Известия вузов. - Прикладная нелинейная динамика, т.10, 2002, №6, с. 42-53.
4. Иванова А. С., Кузнецов С. П. Волна кластеризации в цепочке систем, каждая из которых содержит набор элементов с внутренней глобальной связью. // Известия вузов. - Прикладная нелинейная динамика, т.11, 2003, №4-5, с. 80-88.
5. Бендерская Е. Н., Жукова С. В. Решение задач кластеризации с использованием хаотической нейронной сети //Сборник научных трудов 7-ой всероссийской научно-технической конференции «Нейроинформатика-2005», ч.1, М., 2005. – с. 54-60.
6. Kaneko K. Clustering, Coding, Switching, Hierarchical Ordering, and Control in Network of Chaotic Elements // Physica D., v.41, №2, 1990, pp.137-172.

О структурной идентификации автоматов

Богомолов Сергей Анатольевич

Доцент кафедры прикладной математики

Саратовского государственного социально – экономического университета

Большинство различного рода задач теории систем изначально или путем последовательных уточнений постановки задачи доопределяются таким образом, что каждая из них имеет единственное решение. Наличие нескольких альтернативных решений почти всегда воспринимается как неокончательный результат.

Формально это означает, что необходимо сформулировать совокупность условий, представляющих собой условие единственности решения соответствующей задачи.

В [1] принцип единственности связывается с тем, что каждая модель системы должна доопределяться внешним образом с целью достижения ее единственности. В [2] приведено определение принципа единственности на основе понятия изоморфизма: в задаче реализации для достаточно полных данных существует единственная модель в том смысле, что все модели, объясняющие данные изоморфны друг другу. Приводятся примеры единственности моделей в смысле изоморфности всех минимальных моделей с минимальной размерностью пространства состояний.

Под моделью системы будем понимать конечный детерминированный автомат с фиксированными входным и выходным алфавитами, а в качестве данных – конечное множество вход – выходных слов, реализуемых автоматом.

Идентификация систем в постановке задачи по Заде [3] применительно к автоматным моделям состоит в определении по множеству вход – выходных слов неизвестного автомата (диаграммы переходов) из некоторого класса.

В работе [4] в качестве доопределения предложена структуризация множества вход – выходных слов, представляющих совокупность данных (переход от множества простых экспериментов к множеству кратных экспериментов – следам автомата) и необходимые свойства автоматов (неприводимость относительно реализации заданного следа) позволяющие осуществить однозначную структурную идентификацию автомата по структурированным данным.

В настоящем сообщении рассматривается вопрос об определении свойств и структуры следов автомата, однозначно его идентифицирующих (с точностью до неприводимости, относительно реализации) и содержащих лишь необходимую информацию для однозначной идентификации автомата (неприводимых следов), а также определении свойств автоматов, имеющих единственный неприводимый след, однозначно их идентифицирующих.

Понятие неприводимости следа относительно свойства «идентифицируемости» автомата определяется конструктивно с использованием некоторого преобразования экспериментов, образующих след, названного операцией редукции (сокращения) следа. Эксперимент есть результат применения операции редукции, если для некоторого вход – выходного слова $a\beta$ подэксперимент $R(a\beta)$ удаляется из эксперимента R , а слово $a\beta$ направляется в некоторый подэксперимент удаляемого эксперимента. Эксперимент, полученный в результате применения операции редукции назовем редуцированным подэкспериментом исходного эксперимента. След автомата, однозначно его идентифицирующий (с точностью до неприводимости) назовем неприводимым, если любой след, полученный применением операции редукции к экспериментам, входящим в след, не является следом, однозначно идентифицирующим автомат. Показана конечность множества неприводимых следов, однозначно идентифицирующих автомат, и определена структура и свойства автоматов, имеющих единственные неприводимые следы.

Список литературы

1. Бир С. Кибернетика и управление производством. М.: ИЛ, 1965.
2. Калман Р. Идентификация систем с шумами // УМН. 1987. Т.40. Вып. 4 (244). С.27-41.
3. Современные методы идентификации систем / Под ред. П. Эйкхоффа. М.: Мир, 1983.
4. Богомолов С. А. О восстановлении поведения автоматов по экспериментам. Дискретная математика, т.1, вып. 1, 1989, с. 135-146

Пространство интероперабельности в одноранговых средах

доц. Брейман А. Д.,
Аникши М. А.,

ГОУ ВПО «Московский государственный университет

приборостроения и информатики» (МГУПИ)
г. Москва, 107996, ул. Стромынка, д.20, каф. ИТ-4
maxdog28@mail.ru, it4_adb@mgpi.edu

Под интероперабельностью системы обычно понимают способность ее компонентов к взаимодействию между собой для достижения целей, стоящих перед системой. Для программного обеспечения интероперабельность можно определить как возможность обмена данными между программными комплексами, использования одинаковых протоколов и форматов представления информации. В распределенных базах данных (БД) с централизованным управлением интероперабельность обеспечивается за счет наличия общей глобальной схемы или нескольких интегрируемых в одну схему, отображающих расположение ресурсов в сети [1].

Глобальная схема позволяет не только представлять в БД информацию о местонахождении ресурса, но и определять взаимоотношения между представлениями распределенных данных. В подобных БД могут возникать сложные взаимосвязи между онтологическим описанием данных и их количественными значениями, но централизованная схема ресурсов гарантирует максимальную масштабируемость по отношению к этим факторам, характеризующим гетерогенность хранимой информации. Предел масштабируемости при этом определяется максимальными размерами глобальной схемы, количеством хранимых данных и вычислительными возможностями серверов. Таким образом, семантические взаимосвязи хранимых данных и их свойств является зависимыми от их отображений в глобальной схеме БД. Поскольку интероперабельность системы в этом случае определяется семантикой данных, она может быть охарактеризована как свойство глобальной схемы, приводящее к упорядоченному хранению информации.

Распределенные БД, основанные на одноранговой модели (одноранговые БД, ОБД), для которой характерно равноправие входящих узлов с точки зрения предоставления и потребления ресурсов, существенно отличаются от БД с глобальной схемой. Поскольку узлы ОБД могут в произвольное время подключаться и отключаться, возникает необходимость обеспечения поиска не только собственно хранимых данных (файлов, отношений, XML-документов и ресурсов других типов), но и узлов ОБД. Можно выделить два основных подхода к организации поиска объектов (ресурсов и узлов) в ОБД: распределенные хеш-таблицы (Distributed Hash Tables, DHT) [2] и семантические индексы [3]. Оба подхода обеспечивают простой и эффективный доступ к данным, реализованы во многих системах и продолжают совершенствоваться.

Распределенные хеш-таблицы задают единое пространство идентификаторов, на которое отображаются ресурсы и узлы ОБД с помощью функции хеширования. Основная идея состоит в том, что узел, обладающий некоторым идентификатором N , принадлежащим к этому пространству, отвечает за все ресурсы, которым функция хеширования назначила идентификаторы, близкие к N . Для обеспечения доступа к данным каждый узел хранит локальную таблицу маршрутизации — набор ссылок на другие узлы. Каждая ссылка содержит идентификатор узла и сетевой адрес, используемый для доставки сообщений по транспортной сети (например, IP-адрес). Очевидно, что чем больше таблица маршрутизации, тем быстрее будут получены ответы на запросы. С другой стороны, чем больше таблица маршрутизации, тем больше ресурсов тратится на поддержание ее актуальности, в особенности — для систем с высокой частотой подключений/отключений узлов.

Поскольку функция хеширования гарантирует только получение корректного идентификатора, все точки пространства идентификаторов имеют равную вероятность доступа к потенциально требуемому ресурсу. Это означает, что все узлы обладают одинаковой возможностью и необходимостью к взаимодействию, то есть интероперабельность всех узлов системы одинакова. Подобной интероперабельностью также обладает централизованный вариант хеш-таблиц. Такое пространство отличается от неструктурированного

возможностью доступа к любой его части. Они имеют одинаковое распределение идентификаторов в пространстве, но разную семантическую характеристику взаимодействия.

Поскольку узлы не обладают семантической связанностью, неструктурированные одноранговые системы вынужденно используют широковебчательные рассылки, доставляя все сообщения каждому узлу. В централизованных системах аналогичную роль играет метод полного перебора при поиске в массивах неупорядоченных данных. Эта аналогия позволила увидеть применимость некоторых методов сортировки данных для определения графов взаимодействия между узлами в структурированных сетях.

Структурированная сеть, очевидно, несет в своей структуре указания на возможности взаимодействия между узлами, т.е. обладает семантикой, но не может выявить различий между ними из за несемантических идентификаторов.

Применение семантических индексов получило широкое распространение в одноранговых файлообменных сетях. В отличие от распределенных хеш-таблиц, для создания индексов используются сами данные — либо в виде названий хранимых файлов, либо в виде списка часто встречающихся слов, что позволяет учитывать семантику взаимосвязей между узлами. В результате однородное пространство, констатирующее только наличие взаимодействия, преобразуется в пространство, описывающее семантическую связанность. С геометрической точки зрения равномерная сетка, на пересечении линий которой находились идентификаторы узлов в хеш-таблицах, была искривлена под действием неравномерных семантических связей.

При этом, однако, пропадает возможность применения алгоритмов сортировки, используемых в хеш-таблицах, так как теперь интероперабельность в пространстве идентификаторов теряет однородность. В централизованной БД, имея упорядоченный набор данных, можно выбрать алгоритм обработки, наиболее эффективный именно для данного типа упорядоченности. Пространство же семантических индексов формируется на основе свойств самих узлов. Взаимодействующие узлы, при семантической схожести некоторого признака, получают общий префикс в индексе. Можно сказать, что пространство индексов формируется на основе семантики отношений между узлами, потенциальной целесообразности их взаимодействия или интероперабельности. Естественно, основной целью определения пространства индексов является возможность использования семантических метрик, а возможность выявления интероперабельности узлов вторична. Это пространство индексов условно может быть представлено в виде сети, состоящей из объектов, содержащих семантические метрики и индексы узлов, получаемых уже после ее создания. Если бы семантика отношений между любой парой узлов в такой сети была одинаковой, как в распределенных хеш-таблицах, эти две сети не являлись бы однотипными.

Системы распределенных индексов не предоставляют каких-либо гарантий для времени передачи сообщений до узла, отвечающего за некоторую точку в пространстве идентификаторов, поскольку количество переходов при маршрутизации заранее неизвестно. Другими словами, пространство идентификаторов, не учитывающее семантические характеристики отношений между узлами, не обладает структурой, которая бы упрощала доступ к ресурсам по сравнению с широковебчанием или полным перебором.

По аналогии с глобальной централизованной схемой можно сделать вывод, что первичная составляющая системы может быть использована для построения пространства также и в ОБД. Графы взаимодействия между узлами, где расстоянием является семантическая метрика, не всегда просто изображаются. Например, в двумерном пространстве нельзя изобразить граф, состоящий из четырех узлов, все взаимосвязи между которыми одинаковы. Для этого пришлось бы вводить дополнительные измерения. В то время как при использовании первичной метрики системы мы получаем простое представление ее базовых свойств, пусть и не в евклидовом пространстве. Семантическая метрика достаточна для управления системой, так же как точное определение глобальной схемы данных достаточно для управления централизованной или федеративной БД. Можно сказать, что метрика семантической взаимосвязи между ресурсами в глобальной централизованной схеме данных является аналогом метрики интероперабельности в одноранговых средах.

Чем выше суммарная интероперабельность системы, тем больше ее возможности по выполнению поставленных задач. Если она очень высока, но система равномерно распределяет идентификаторы в пространстве интероперабельности, это не говорит о том, что структура не эффективна. Это лишь показывает, что лучшим вариантом взаимодействия между узлами такой системы является либо переход к централизованному подходу к обработке информации, либо равномерное распределение вычислений между узлами системы.

Так как семантические связи непостоянны и могут изменяться в зависимости от целей, например, индексации различного контента в файлообменных сетях, или получения различных показаний и точности характеристик в запросах к сенсорной сети, изменяется и само пространство интероперабельности. Очевидно, что в централизованных базах данных запросы к ним не приводят к непосредственному

изменению ее организации, так как связь всегда фиксирована относительно глобального отображения. В динамических одноранговых системах частое изменение связей может приводить к полному обновлению структуры пространства интероперабельности, которое, как и централизованная схема отображения, должна быть непротиворечива, что сказывается на затратах ресурсов на ее организацию и поддержание работоспособности.

Частота изменения свойств узла является его неотъемлемой характеристикой и очевидно, что при ее возрастании уменьшается интероперабельность между узлами. Тем меньше искривление пространства интероперабельности, с одной стороны приводящее к целесообразности централизованного решения, а с другой показывающее низкую эффективность выполняемого процесса, все больше сводимого в таком случае к широковебательным рассылкам. К очевидным плюсам можно отнести возможность параллельного выполнения задач, что демонстрируют алгоритмы агрегации и кластеризации данных в сенсорных сетях, основанные на свойствах семантической схожести, но они являются производными по отношению к организуемому пространству и сами по себе не влияют на него. Так же и алгоритмы глубинного анализа данных в БД не влияют на первичную организацию обрабатываемых ими информационных ресурсов.

Все подходы, использующие семантические индексы, основаны на учете отдельных характеристик, например имени хранимого файла, и их создание может сравниваться с алгоритмом кластеризации, аналогичной группировке данных. Развитие одноранговых систем идет по пути создания полностью децентрализованных аналогов СУБД, для которых очевидным является не только хранение данных и определение их связи между собой, но и их онтологическое представление, представляющее смысл информации. В семантических индексах подразумевается, что смысловое содержание каждого документа определяется только содержанием в нем определенных слов. Свойства, на основании которых определяется пространство интероперабельности, являются фиксированными относительно некоторого гарантированного глобального домена знания, что и обеспечивается этим утверждением. Такой подход в централизованной БД равнозначен тому, что все числа имеют одинаковый смысл и различаются только количеством, не обладая качественными характеристиками.

Наиболее совершенными одноранговыми решениями на сегодняшний день являются одноранговые системы управления данными, применяющие средства описания онтологий и данных, такие как OWL и XML. Такие описания позволяют получить полный децентрализованный аналог СУБД, повысить качество получаемых результатов и масштабируемость системы. Вместе с тем они серьезно влияют на принципы организации пространства интероперабельности, поскольку обычно подразумевается, что люди могут хранить тексты с одинаковым смысловым содержанием в разных манерах изложения, но они независимо друг от друга имеют одинаковое представление о структуре этого документа и его онтологического содержания [3].

Внедрение стандарта на метаданные вполне достижимо, например, в случаях перехода федеральной БД в ОБД, или в крупных файлообменных сетях. Но на практике в сравнительно небольшой динамичной одноранговой структуре, обладающей либо очень сложной системой представления данных, либо большим количеством динамичных онтологий и описаний информации, что возможно, например, в случае организации работы коллективов, решающих постоянно сменяющиеся специфические задачи в области производства или исследовательской деятельности, это оказывается невозможным. В таких сетях фиксированное определение признака теряется и то же становится относительным по отношению к другому признаку узла. Это означает больше, чем снижение интероперабельности и потенциальную централизацию. Логически необходимой тогда оказывается организация еще одной одноранговой сети, создающей пространство интероперабельности структур онтологий и описаний данных. Данная проблема обоснованно игнорируется в задаче управления данными и относится больше к области извлечения информации. Но даже решения с применением алгоритмов этой области не всегда могут гарантировать правильность полученных глобальных схем и требуемое качество системы. Одним из решений этой проблемы, возможных на уровне организации ресурсов узлов, служит создание так называемых чистых одноранговых систем управления данными, в которых узел хранит только свою часть XML схемы и децентрализованный элемент знания о системе.

Список литературы

1. Aberer K. et al. Emergent Semantics Systems. // In ICSNW 2004, Lecture Notes in Computer Science 3226. — IFIP, 2004. — pp.14-43.

2. Tanner T. Distributed Hash Tables in P2P Systems — A literary survey. – Helsinki University of Technology, 2005.
3. Crespo A., Garcia-Molina H. Semantic Overlay Networks for P2P Systems. // In Proc. of the 29th VLDB conference, Berlin, Germany, 2003.
4. Sartiani C., Manghi P., Ghelli G., Conforti G.: XPeer: A Self-organizing XML P2P Database System. // Proceedings of the First International Workshop on Peer-to-Peer Computing and Databases (P2P&DB), Heraklion, Crete, Greece, 2004.

Применение многоагентных систем для моделирования процесса обучения студентов

**Варфоломеев А. Г., к.ф-м.н., доцент,
Марахтанов А. Г.,
Петрозаводский Государственный Университет
e-mail: marahthanov@petsu.ru
185001, г. Петрозаводск, ул. Северная, 4-52**

Моделирование различных социальных групп (социальных систем) является, в настоящее время, очень популярной и актуальной задачей. Особенно интересным и перспективным выглядит использование в подобных моделях элементов систем искусственного интеллекта.

В данной работе предполагается построить модель студенческой группы учебного заведения, с целью решения следующей проблемы. Замечено, что при произвольном формировании студенческих групп, успеваемость групп различается. Согласно предположению, успеваемость группы в целом зависит от многих параметров. В том числе, от межличностных отношений между студентами группы, от качества преподавания, расписания занятий, социальных условий проживания студентов.

Построив модель студенческой группы, мы сможем исследовать, насколько эти факторы в действительности оказывают влияние, какие из них более приоритетны в сравнении с другими, а также каким образом, меняя их, можно добиться большей успеваемости. Кроме того, создание подобной модели может способствовать использованию её для апробации нововведений в области образования, что особенно актуально в настоящее время.

Для решения описанной выше проблемы можно использовать различные, в том числе ставшие уже классическими, подходы. Подобная модель может базироваться на методах системного анализа (если рассматривать социальную группу, как систему), имитационного моделирования, исследования операций, теории игр и ряда других. Однако во всех этих методах существует ряд ограничений, не позволяющих использовать их в нашем случае. Так, в системном подходе, любое сложное взаимодействие представляется через работу более простых и независимых подсистем, которые в случае рассмотрения социальных систем часто оказываются взаимозависимыми друг от друга или, по крайней мере, существенно влияют друг на друга. В исследовании операций процесс принятия решений заключается в нахождении оптимальной стратегии поведения в заданных условиях, которые, однако, на самом деле постоянно меняются и пересматриваются. При имитационном моделировании формализованная модель объекта или процесса никак не может пополняться или уточняться «на ходу», любые изменения требуют остановки процесса

моделирования, ручного внесения изменений в модель и перезапуска процесса. В теории игр правила поведения игроков также заранее определены и известны для всех игроков, им известны все участники и т. д. [3]

Большинства перечисленных выше недостатков можно избежать, используя многоагентные системы (МАС). Ключевым элементом этих систем становится интеллектуальный агент, способный воспринимать ситуацию, принимать решения и коммуницировать (общаться) с другими объектами. При этом агенты взаимодействуют в определенной среде. Главное отличие МАС от существующих жестко организованных программных систем состоит в наличие нового свойства – способности к самоорганизации. Данной возможности были лишены перечисленные выше модели, в них структура взаимодействий между объектами задавалась до начала функционирования системы. Однако новые подходы в области синергетики, изучающей самоорганизацию, подсказывают, что процесс образования устойчивых пространственно-временных структур и состояний равновесия нельзя не учитывать в процессе работы модели.

Термин «агент» достаточно популярен, и используется для обозначения автономного программного модуля, искусственного организма, виртуального деятеля. В нашем случае, под «интеллектуальным агентом» будем понимать любые физические или виртуальные единицы, у которых присутствуют как минимум четыре из обозначенных выше функций, а именно [1, стр. 106]:

- когнитивная (наличие пополняемой символической модели внешнего мира),
- рассуждающая (способность на основании построенной модели внешнего мира производить рассуждения и на их основе принимать самостоятельные решения или выполнять действия, изменяющие среду),
- коммуникативная (наличие модели общения, состоящей из моделей участников, процесса и желаемого результата общения),
- ресурсная (наличие ресурсов, определяющих поведение).

В зависимости от типа агента, среды, механизма отношений между агентами выделяют различные типы МАС. В условиях нашей задачи наиболее удобно определять МАС, как децентрализованную систему, в которой управление происходит за счет локальных взаимодействий между интеллектуальными агентами, существующими в общей среде.

Построение МАС для решения поставленной задачи целесообразно начинать с проектирования самого агента. Выделив возможные типы агентов системы, необходимо определить их структуру и возможные действия, на основании социальной теории, а также характеристики внутренних компонентов структуры индивида реального общества, которые имеют главное значение для решения поставленных задач моделирования. Появляясь на свет, агенты обладают различными характеристиками (свойствами). Часть свойств может быть у одних агентов, но отсутствовать у других.

В простейшем случае, решая нашу задачу, можно выделить «агента-студента», как основного субъекта системы. Каждый «агент-студент» обладает набором ресурсов, таких, как деньги, знания, свободное время, а также набором потребностей, задаваемых минимальным и желаемым уровнем знаний, денег и времени. При этом, в зависимости от менталитета, агент может тратить свое время либо на приобретение знаний, либо на приобретение денег (работа), либо транжирить время. Менталитет агента может меняться в зависимости от его взаимодействий с другими агентами, воздействия внешней среды или изменения ресурсов самого агента. Объем доступных для потребления ресурсов определяется состоянием внешней среды системы.

Взаимодействия между агентами должны строиться исходя из 4-х возможных причин общения [1, стр. 185]:

- совместимость целей (общая успеваемость группы увеличивает количество ресурсов - знаний);
- отношение агентов к ресурсам (агенты могут делиться знаниями, без потери своих знаний, друг с другом, совместное использование знаний повышает их объем);
- опыт агентов (дружеские связи между агентами способствуют установлению совместных целей и дополнению ресурсов);

- обязательства агентов друг перед другом (получение определенного ресурса от другого агента предполагает его возврат в том или ином виде).

При этом на основании опыта прошлых взаимодействий и собственного менталитета, у агентов могут возникать различные типы формирования общих целей с другими агентами, например, агент a_1 может быть по отношению к агенту a_2 : благонамеренным (стремиться к благу, как для себя, так и для a_2), эгоистичным (стремится к своему благу, отношение к a_2 - нейтральное), злонамеренным (стремится к своему благу, даже за счет неудачи a_2), альтруистичным (стремится к благу a_2 , даже за счет неудачи себя) и т. д. При математическом описании отношений между агентами удобно использовать аппарат нечетных отношений, когда взаимодействие характеризуется числом от -1 до 1.

Кроме того, каждый агент имеет различную способность к убеждению или подчинению. В общем виде, ее можно представить как поле, притягивающее (отталкивающее) агентов друг от друга. Это поле, во многом, и будет определять отношения между агентами.

Воздействия внешней среды можно рассматривать как некое поле, оказывающее влияние на агентов (к примеру, на минимальный и желаемый уровень денежных ресурсов) и на доступные к использованию ресурсы. Сами агенты также способны оказывать влияние на внешнее поле.

Для математической интерпретации в обозначенных выше случаях поле понимается как функция $S: F \rightarrow R^k$, которая задана на фазовом пространстве F и удовлетворяет уравнению поля $\hat{A}(S) = f$, где \hat{A} – некоторый оператор. Под фазовым пространством понимается система координат, каждая из которых соответствует разным величинам интенсивности одной характеристики. [2, стр. 29]

Для упрощения дальнейшей работы с агентом, его удобно представить в виде совокупности взаимосвязанных элементов, связи между которыми определяются архитектурой агента. Описанный выше «агент-студент» может быть описан в виде вертикальной InteRRap-архитектуры, представляющей агента как множество уровней, связанных через управляющую структуру и использующих общую базу знаний. [4]

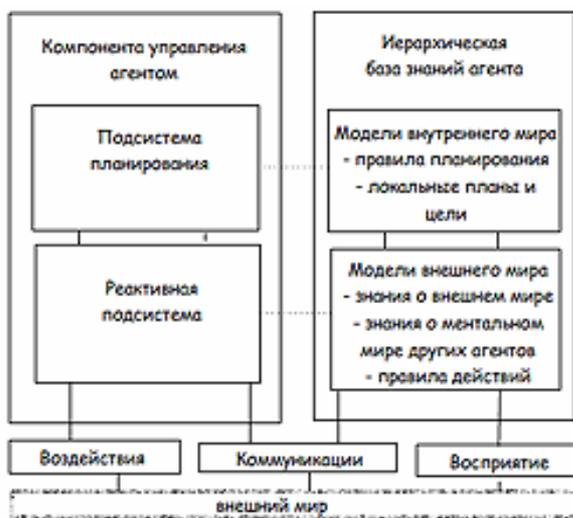


Рис. 1. InteRRap-архитектура агента

На рисунке 1 представлен пример подобной архитектуры. Если в базе знаний агента хранится возможная реакция на полученное из внешнего мира событие, она осуществляется сразу реактивной подсистемой. В нестандартных ситуациях идет обращение к системе планирования.

С развитием подобной модели имеет смысл ввести новые типы агентов – преподавателей, работодателей, а также дифференцировать ресурсы знаний, ввести новые типы ресурсов (к примеру, с учетом гендерных отношений между студентами).

Список литературы

- [1] Тарасов В. Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика – М.: УРСС, 2002
- [2] Гуц А. К., Коробицын В. В., Лаптев А. А. и др. Математические модели социальных систем – Омск: ОмГУ, 2000
- [3] Скобелев П. О. Открытые мультиагентные системы для поддержки процессов принятия решений при управлении предприятиями // Известия Самарского научного центра Российской академии наук – 2001 – Т. 3, № 1
- [4] Muller, J. – P., Pischel M., Thiel M. Modeling Reactive Behaviour in Vertically Layered Agent Architectures // Intelligent Agents / Ed. By M. Wooldridge and N. R. Jennings. – Berlin: Springer-Verlag. – P.261-276

Методика оценки безопасности интеллектуального здания

Васильев В. И., Лохматов М. В.

*Уфимский Государственный Авиационный Технический Университет
Кафедра Вычислительной Техники и Защиты Информации
Республика Башкортостан, г.Уфа,
ул. Генерала Горбатова, д. 3/2 кв. 99, тел. 8 903 355 0 355
E-mail: marat-ufa@yandex.ru*

Первым делом при проектировании любого сооружения рассчитывают экономическую привлекательность проекта. Использование современных технологий, таких как интеллектуальное здание, существенно увеличивает объем будущих инвестиций. Перед проектировщиками встает вопрос выбора оборудования и состава подсистем в целом (стоит ли включать систему контроля и управления доступом, должна ли эта система быть отечественной или зарубежной). Ответом зачастую является опыт предыдущих проектов, интуиция либо простейшие методы принятия решений.

Целью данной работы была разработка методики, позволяющей свести проблему к задаче многокритериальной оптимизации, способы решения которой известны.

Введение

В настоящее время все большую популярность приобретает идея интеллектуального здания, отличительными особенностями которого являются:

1. способность оптимально реагировать на изменения в происходящих в здании процессах;
2. сочетание преимуществ децентрализованных систем и единого центра управления всеми подсистемами;
3. структурированный подход к построению инженерных систем здания;
4. возможность внесения изменений с минимальными затратами.

Основной проблемой при проектировании подобных систем является доказательство эффективности выбранного решения. В связи с этим актуальной становится разработка методологии оценки эффективности интеллектуального здания.

Основная часть

Предлагаемая методика состоит из следующих шагов:

1. Составить дерево целей построения интеллектуального здания. Глобальная цель разбивается на цели, цели на подцели и т.д. как показано на рисунке 1.



Рисунок 1 - Дерево целей

2. Составить дерево критериев или показателей достижения поставленных целей. Критерии можно условно разделить на 4 группы:

- показатели качества назначения (например, вероятность обнаружения возгорания);
- показатели надежности (например, наработка на ложное срабатывание);
- показатели помехоустойчивости (например, наработка между отказами);
- экономические показатели (например, суммарные затраты на систему).

3. Определить состав необходимых подсистем, которые можно разделить на три группы:

1) интегрированный комплекс технических средств безопасности:

- системы видеонаблюдения (СВН);
- охранно-пожарной сигнализации;
- автоматического пожаротушения (САПТ);
- противодымовой защиты;
- оповещения о пожаре и управления эвакуацией людей;
- контроля и управления доступом (СКУД).

2) комплекс систем жизнеобеспечения:

- системы отопления;
- вентиляции и кондиционирования воздуха (HVAC);
- удаленного мониторинга и управления электроснабжением;
- освещения и управления освещением;
- учета энергоносителей;
- контроля и управления лифтами;
- аварийного контроля инженерных систем).

3) комплекс систем информатизации:

- локальная вычислительная сеть;
- системы приема эфирного и спутникового телевидения;
- радиотелефонии;
- телефонной сети;
- проведения конференций с синхрорепродукцией;
- электроснабжения;
- средства оперативной радиосвязи персонала).

4. Определить функции, выполняемые каждой из подсистем и их взаимосвязь.

Например, функция системы пожарной сигнализации (СПС): обнаружение очага возгорания. Реакции: СПС фиксирует факт и время события, отображает место сработавшего датчика на поэтажных планах; система управления лифтами опускает их на первый этаж; HVAC отключает приточную вентиляцию в данном помещении и активизирует систему дымоудаления в коридорах; СВН направляет камеры на сработавший датчик и присваивает этим камерам высший приоритет; САПР запускает модули пожаротушения; СКУД автоматически разблокирует помещения на путях эвакуации; система управления освещением включает аварийное освещение в зоне тревоги; система оповещения автоматически информирует персонал о ЧП.

Также на данном этапе необходимо построить модель функционирования интеллектуального здания с помощью методологии IDEF0. Пример IDEF0 диаграммы, иллюстрирующей реакцию оператора системы безопасности на возникновение пожара, приведен на рисунке 2.

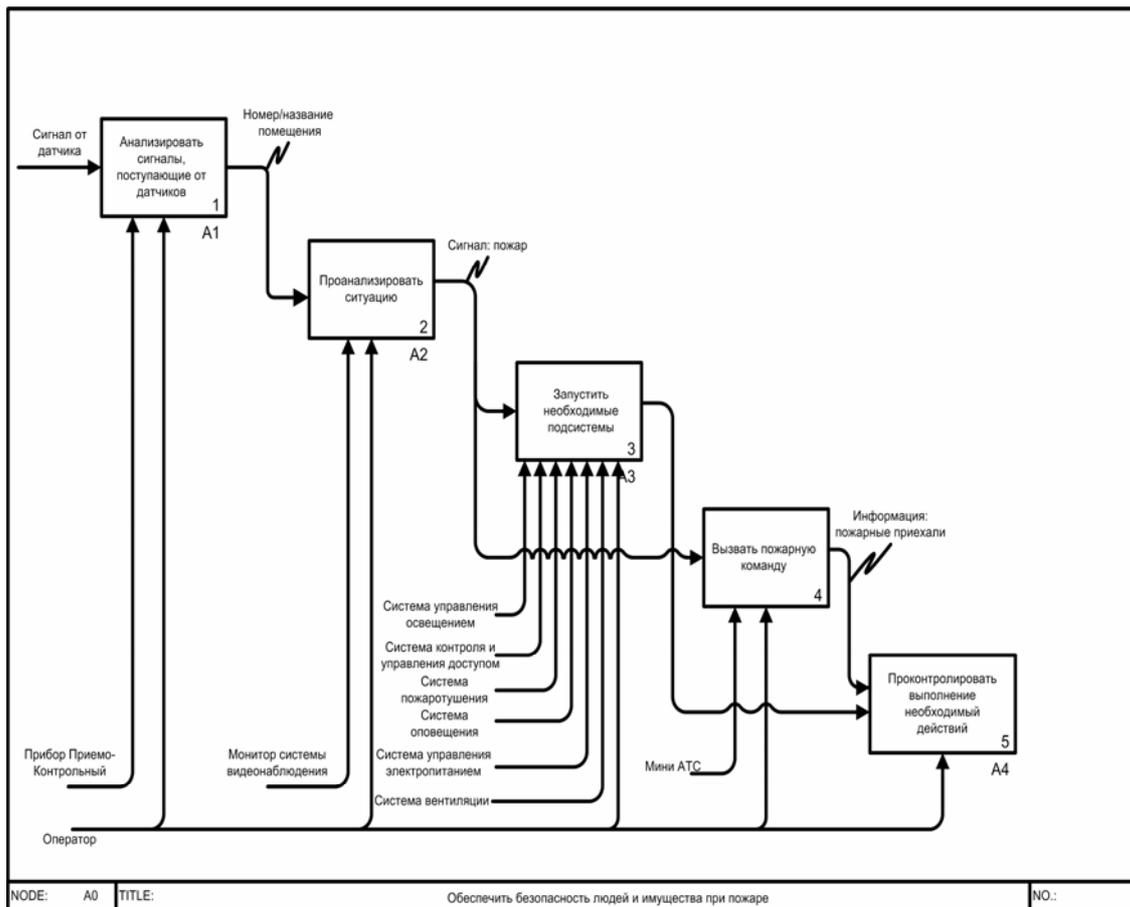


Рисунок 2 - IDEF0 диаграмма.

5. Решить многокритериальную задачу оптимизации.

Существуют следующие группы методов решения многокритериальных задач:

- линейная свертка показателей;
- лексикографические методы, основанные на доминировании критериев;
- интерактивные методы предпочтительного выбора с помощью итерационных процедур сужения исходного множества объектов;
- аксиоматический подход с использованием теории полезности.

В предлагаемой модели используется метод смещенного идеала - представитель группы интерактивных методов. Его идея в том, чтобы сравнить показатели рассматриваемой системы с эталонными требованиями и дать соответствующие рекомендации как это показано на рисунке 3.

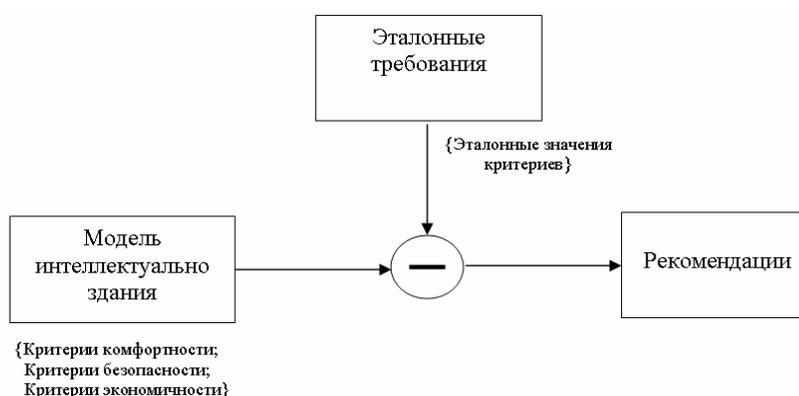


Рисунок 3 - Выработка рекомендаций

Выводы

Предложенная методика позволяет произвести выбор наиболее предпочтительного варианта реализации системы в случае, когда существует несколько альтернатив. Если вариант - единственный, методика позволяет оценить насколько он близок к идеальным требованиям.

Новизна подхода заключается также в применении IDEFO диаграмм для построения модели функционирования интеллектуального здания, которые ранее применялись лишь при моделировании бизнес-процессов предприятия.

Список литературы

1. С. В. Черемных, И. О. Семенов, В. С. Ручкин, Моделирование и анализ систем. IDEF-технологии: практикум. - Москва, Радио и связь, 2003.- 249с.
2. Н. Г. Топольский, Концепция создания интегрированных систем безопасности и жизнеобеспечения. - Москва, ВИПТШ МВД РФ, 1994.- 87 с.

Оптимизация процесса обработки данных в системах распознавания буквенно-цифровых меток движущихся объектов

Веснин Е. Н.,
научный сотрудник,
vesnin@imit.ru

Царев В. А.,
зав. кафедры ПО ВТ и АС, к.т.н., доцент,
vats@imit.ru

Институт менеджмента и информационных технологий (филиал)
Санкт-Петербургского государственного
политехнического университета в г. Череповце,
162600, Вологодская обл., г. Череповец, ул. Гоголя, д.33а

Введение

Во многих практических приложениях существует необходимость осуществлять контроль и учет объектов по нанесенным на них идентификационным меткам (ИМ). Это могут быть изделия, имеющие какие-либо регистрационные номера: контейнеры, автомобили, объекты подвижного состава железнодорожного транспорта и др. Организация такого контроля возможна при помощи опτικο-электронных систем, позволяющих повысить его эффективность.

Системы распознавания (СР) буквенно-цифровых меток движущихся объектов, как правило, функционируют в условиях значительной априорной неопределенности. Осреднение состояния среды позволяет спроектировать систему, оптимально работающую только при определенных внешних условиях. Всякое же отклонение состояния среды от среднего приводит к неоптимальности функционирования такой системы.

Именно поэтому, на этапе проектирования математического и программного обеспечения рассматриваемых систем необходимы эффективные модели организации вычислительного процесса, обладающие возможностью адаптации к изменяющимся внешним условиям.

Основные этапы решения задачи распознавания

При решении задачи распознавания ИМ выделяют следующие основные этапы:

1). Ввод исходной информации. В рассматриваемых системах изображения формируются при помощи неподвижной видеокамеры, в поле зрения которой перемещаются объекты, подлежащие идентификации. В ЭВМ изображение представляется в виде матрицы, каждый элемент которой описывает яркость соответствующей точки изображения, и может принимать одно из целочисленных значений на отрезке $[0; 2^L - 1]$, где L – глубина цвета.

2). Обнаружение движения. На данном этапе на основе анализа последовательности кадров принимается решение о наличии объекта контроля (ОК) в кадре. Задача осложняется тем, что помимо объекта контроля в контролируемой зоне возможно наличие «посторонних» движущихся объектов и колебаний самой регистрирующей камеры.

3). Локализация ИМ. Целью выполнения данного этапа является указание областей на кадре, которые возможно содержат изображение метки. Данная задача также не является тривиальной, поскольку изображения реальных сцен могут содержать множество элементов, схожих с ИМ (надписи, рисунки, конструктивные элементы и др.).

4). Сегментация ИМ выполняется для выделения на изображении ИМ отдельных символов. При этом часть символов может быть пропущена, а часть выделена ложно. Сложности на данном этапе в основном обусловлены тем, что некоторые символы иногда оказываются разорванными, а другие наоборот слившимися. Также доставляют проблемы неплоский характер информационного носителя, смещения символов или их частей относительно местоположения в строке, недостаточное разрешение изображений (слишком малый размер символов).

5). Распознавание. На этом этапе изображения выделенных символов (описанных на языке априорного словаря признаков) относятся к одному из классов априорного алфавита. Из полученного набора символов формируются возможные варианты распознавания ИМ. Типовыми проблемами на данном этапе являются:

разнообразии форм начертания символов; различные символы могут обладать сходными очертаниями (U и V, S и 5, Z и 2, G и 6); искажения формы: вращения с изменением наклона символов. Кроме того, необходимо выделить радиометрические искажения: дефекты освещения, тени, блики, неравномерный фон. При формировании вариантов распознавания ИМ используются ограничения на допустимый синтаксис и геометрию ИМ.

6). Принятие решений. На данном этапе по результатам обработки текущего и некоторого количества предшествующих кадров принимается решение об объекте контроля. Поскольку на каждом кадре может быть локализовано несколько зон, каждой из которых соответствует свое множество вариантов распознавания, здесь следует сформировать из всего многообразия решений подходящее.

На каждом этапе также может выполняться необходимая предобработка изображения.

Следует отметить, что для каждого этапа в настоящее время разработано множество различных алгоритмов [1-2], характеризующихся своей эффективностью (выраженной в некоторой характеристике качества) и быстродействием.

Организация процесса обработки данных в системах распознавания буквенно-цифровых меток движущихся объектов

Процесс обработки каждого кадра видеопоследовательности состоит из последовательного выполнения алгоритмов обработки кадра. Число этапов обработки в общем случае может быть различным. Каждый этап также может быть представлен множеством альтернативных реализаций. Последовательность выполнения алгоритмов и их параметров будем называть *стратегией распознавания*.

Обычно в системах распознавания для обработки каждого кадра используется единственная стратегия распознавания, выбранная исходя из оптимальности функционирования при среднем состоянии среды. В некоторых системах для повышения эффективности распознавания используют входной видеобuffer.

Недостатками существующих подходов к построению систем такого рода являются низкая эффективность при значительном отклонении параметров внешней среды от средних значений и неэффективное использование вычислительных ресурсов. Так при росте интенсивности движения объектов контроля или уменьшении доступности вычислительных ресурсов может произойти переполнение буфера видеоданных и часть кадров (или даже объектов контроля) будет пропущена. Низкая интенсивность движения ОК, напротив, приведет к недоиспользованию буфера видеоданных, и как следствие, неэффективному использованию вычислительных ресурсов.

К факторам внешней среды, оказывающим влияние на эффективность систем распознавания, можно отнести интенсивность и скорость движения объектов контроля, а также доступность вычислительных ресурсов. В качестве характеристик, описывающих эффективность функционирования рассматриваемых систем, используют: вероятность правильного распознавания ИМ ОК, вероятность неверного распознавания ИМ ОК, вероятность пропуска ОК, вероятность ложной тревоги, время идентификации ОК.

Предлагаемая модель системы идентификации

Авторами предлагается новая модель организации вычислений (рис.), обладающая возможностью изменять стратегию распознавания при изменении внешних условий и позволяющая тем самым улучшить характеристики функционирования системы в целом. Системы, построенные на базе данной модели, обладают рядом особенностей. Вся информация, используемая в процессе распознавания, хранится в таблице «рабочая информация». Изменять данные в этой таблице может только управляющий модуль. При изменении доступности вычислительных ресурсов, выбираются такие стратегии распознавания, которые позволяют наиболее полно использовать имеющиеся ресурсы.

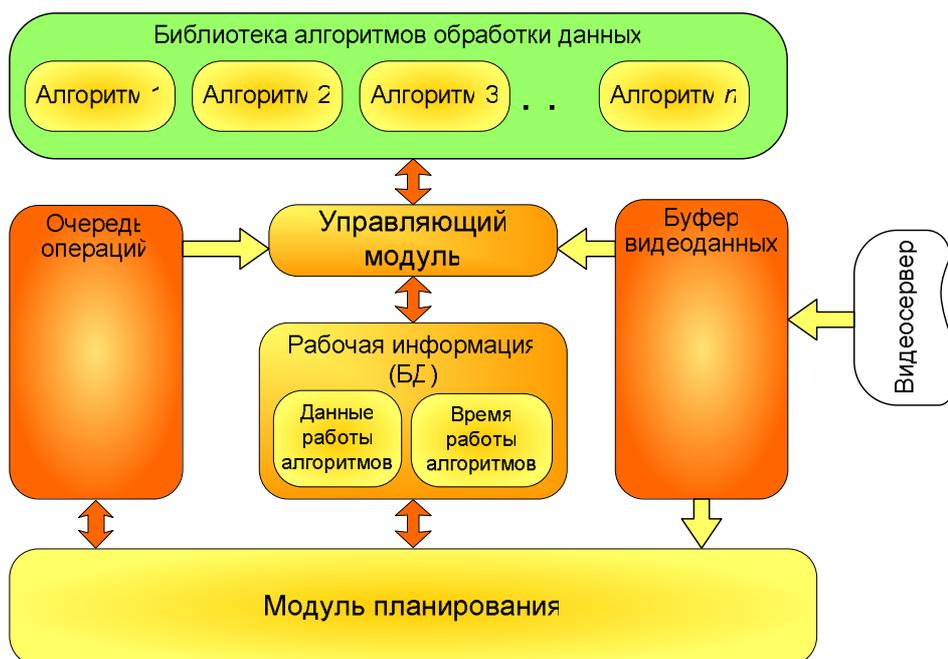


Рис. Предлагаемая модель системы распознавания

Кратко опишем функции основных модулей предложенной модели.

Видеосервер с некоторой частотой ν помещает кадры в видеобуфер.

Модуль планирования на основе анализа изменений количества записей в видеобуфере и результатов обработки ряда предыдущих кадров выбирает стратегию обработки кадров и записывает в очередь операции последовательность вызова алгоритмов и их параметров, соответствующих данной стратегии.

Управляющий модуль поочередно извлекает записи из очереди операции и выполняет алгоритмы, соответствующие данным записям. Также задачами управляющего модуля является формирование входных данных для алгоритмов из таблицы «рабочая информация» и запись в данную таблицу результатов работы алгоритмов. В таблицу «рабочая информация» записывается также время выполнения каждого алгоритма. Это необходимо для уточнения моделей алгоритмов и адекватного планирования.

Как видно, предлагаемая модель систем идентификации является достаточно универсальной - позволяет в процессе работы изменять стратегию функционирования системы. Благодаря хранению всей рабочей информации в единой таблице обеспечивается единообразный доступ к необходимым данным и их хранение, что позволяет учитывать историческую информацию. За счет разделения данных и операций над ними упрощается при необходимости организация параллельных вычислений.

Планирование процесса распознавания

Исходными данными при проектировании алгоритма планирования являются множество стратегий распознавания $\{R\}$ с описывающими их характеристиками и ограничение на количество записей в видеобуфере $N_{\text{доп}}$ (задержка на время идентификации ОК).

На вход СР могут поступать кадры следующих типов: H_1 – без ОК (движения), H_2 – содержащие ОК, но не содержащие ИМ, H_3 – содержащие ОК и его ИМ. Множеству стратегий соответствуют матрицы E и T , описывающие их эффективность и среднее время работы соответственно. Элемент E_{ij} матрицы E содержит оценку эффективности использования стратегии R_i для обработки кадра типа H_j . Элемент T_{ij} матрицы T является оценкой среднего времени анализа кадра типа H_j с помощью стратегии R_i . Вместо оценок среднего времени анализа кадров можно использовать доверительные интервалы для их среднего значения, нижние и верхние границы которых можно свести в матрицы T_L и T_U .

Введем в рассмотрение следующие события, связанные с результатами анализа кадра: A_1 – отказ на этапе обнаружения ОК; A_2 – отказ на этапе локализации ИМ; A_3 – отказ на этапе распознавания ИМ; A_4 – предложен вариант распознавания ИМ.

Каждая стратегия характеризуется распределениями вероятностей возможных исходов при анализе кадров различных типов. Данные распределения можно собрать в матрицу B , элемент B_{ij} которой является вероятностью $P(A_j | H_i)$ исхода A_j при анализе кадра типа H_i .

Краткосрочное планирование

Краткосрочное планирование заключается в выборе стратегии анализа следующего кадра на основании результатов анализа текущего. При этом необходимо обеспечить максимальную эффективность его обработки за минимальное время. Для этого необходимо предсказать тип следующего кадра. В работе исследована эффективность определения типа следующего кадра с использованием аппарата скрытых Марковских моделей и формулы Байеса.

Рассмотрим методику определения типа обработанного кадра на основе результатов его анализа.

Введем в рассмотрение матрицу переходов M . Элемент M_{ij} матрицы M означает вероятность появления на входе системы в момент $n + 1$ кадра типа H_j при условии, что в предыдущий момент времени n поступил кадр типа H_i . Эти вероятности не зависят от n и определяются лишь состоянием внешней среды. Будем считать, что для всех i выполняется соотношение $\sum_j M_{ij} = 1$.

Обозначим $H_i(n)$ – событие, характеризующее появление кадра типа H_i в момент времени n , а $A(n)$ – исход обработки n -го кадра. Тогда с учетом введенной ранее переходной матрицы и формулы полной вероятности вероятность того, что следующий $(n+1)$ кадр будет типа H_i , определится следующим образом:

$$P(H_i(n+1) | A(n)) = \sum_{k=1}^3 P(H_k(n) | A(n)) M_{ik}, \quad (1)$$

при этом по формуле Байеса имеем:

$$P(H_i(n) | A(n)) = \frac{\pi(H_i(n)) P(A(n) | H_i(n))}{\sum_{k=1}^3 \pi(H_k(n)) P(A(n) | H_k(n))}, \quad (2)$$

где $\pi(H_i(n))$ – априорная вероятность кадра типа H_i .

Согласно принятому в теории скрытых Марковских моделей допущению о зависимости текущего состояния только от предыдущего в данной формуле вместо априорных вероятностей предлагается использовать их оценки, основанные на результате анализа предыдущего кадра:

$$P'(H_i(n) | A(n)) = \frac{P(H_i(n) | A(n-1)) P(A(n) | H_i(n))}{\sum_{k=1}^3 P(H_k(n) | A(n-1)) P(A(n) | H_k(n))}.$$

В результате получаем рекуррентную формулу для оценки $P(H_i(n+1) | A(n))$; при этом полагаем $P(H_i(1) | A(0)) = \pi(H_i)$.

В качестве альтернативного варианта оценки априорных вероятностей кадров различных типов $\pi(H_i(n))$ могут быть применены методы математической статистики для анализа результатов работы системы за предшествующий период τ . При этом следует учитывать, что если τ выбрать слишком большим, то эти значения могут оказаться малоприменимыми для принятия решения о выборе стратегии распознавания. Если же интервал времени, на котором производится оценка, будет мал, то возможны случаи, когда априорные вероятности некоторых типов кадров окажутся нулевыми. Например, в случае распознавания номеров автомобилей, интенсивность движения объектов контроля может быть весьма неоднородной, что может быть обусловлено наличием регулируемого перекрестка или сезонными и суточными колебаниями интенсивности движения. В промышленном производстве неоднородность интенсивности движения объектов контроля определяется особенностями технологического процесса. Исходя из вышесказанного наиболее целесообразно использовать несколько временных интервалов различной длительности τ_s для оценки

$\pi(H_i(n))$. На каждом интервале τ_s производится статистическая оценка априорных вероятностей кадров различных типов и на основании (1), (2) определяются вероятности различных типов для следующего кадра $P_{\tau_s}(H_i(n+1)|A(n))$. Затем формируется итоговое значение $P^*(H_i(n+1)|A(n))$ в соответствие с определенной функцией, аргументами которой являются

$$P_{\tau_s}(H_i(n+1)|A(n)).$$

Однако, как показала практика, использование формулы Байеса для оценки типа кадра является устойчивым по отношению к ошибкам оценки распределения априорных вероятностей π . Можно показать, что ошибки обусловленные «плохой» оценкой априорных вероятностей могут возникнуть лишь, когда выполняется условие $b_{\max1}/b_{\max2} \geq \max(x)/\min(x)$, где $b_{\max1}$ – максимальное значение в некотором столбце матрицы B , $b_{\max2}$ – второе максимальное значение в том же столбце матрицы B .

При использовании скрытых Марковских моделей в процессе работы можно с помощью алгоритмов Баума-Уэлша или Витерби [3] переоценить параметры, описывающие скрытую Марковскую модель $\{M, B, \pi\}$.

После того как определен тип кадра, выбирается такая стратегия его обработки, с помощью которой удастся обработать его наилучшим образом при сохранении количества записей в видеобufferе в допустимых пределах. Если в текущий момент времени буфер видеоданных содержит $N(n)$ записей, то за время анализа текущего кадра ожидаемое количество записей в нем составит $N(n+1) = N(n) - 1 + T_{ij} \cdot v$, здесь T_{ij} – среднее время анализа кадра типа H_j при использовании стратегии R_i , v – частота формирования кадров (Гц). Т.о. необходимо выбрать стратегию, обеспечивающую $N(n+1) < N_{\text{доп}}$ и обладающую наибольшей эффективностью обработки кадров соответствующего типа.

Долгосрочное планирование

Для планирования процесса обработки и анализа изображений необходимо также иметь и долгосрочные стратегии, способные обеспечить более эффективное использование вычислительных ресурсов и видеобufferа. Они также позволяют в некоторой степени нивелировать неточности обусловленные использованием скрытых Марковских моделей на этапе краткосрочного планирования, в основе которых лежит предположение о независимости текущего состояния системы от предыстории ее функционирования (считается, что текущее состояние зависит только от предыдущего).

Одним из источников информации о вероятностях появления кадров различных типов может служить интенсивность движения объектов контроля. Если известна плотность распределения интервалов времени между появлениями объектов контроля, то можно оценить вероятность появления объекта в зоне контроля в некотором интервале времени (например, соответствующем времени формирования кадров, находящихся в видеобufferе). Информация о доле кадров с ИМ среди кадров с ОК позволяет оценить вероятности появления кадров различных типов.

При помощи скрытой Марковской модели предлагается моделировать поведение системы на протяжении планируемого периода и осуществить выбор множества оптимальных стратегии распознавания.

На основании результатов распознавания последовательности кадров имеется возможность прогнозировать продолжительность пребывания ОК и ИМ в зоне контроля и тем самым оценить количество кадров различных типов.

На основании полученных оценок распределения типов кадров в различные моменты времени $p(H_i(n))$ можно определить ожидаемое время обработки последовательности кадров $[n_1, n_2]$ с использованием различных стратегий:

$$T_{ожс}(R_k) = \sum_{n=n_1}^{n_2} \sum_{i=1}^3 T_{ki} p(H_i(n)),$$

В качестве допустимых выбираются стратегии для которых выполняется неравенство $T_{ожс} \leq (n_2 - n_1 + 1) \cdot v$. Выбор стратегий при краткосрочном планировании можно осуществлять среди определенных на данном этапе.

На основе информации о доверительных интервалах для ожидаемого времени анализа кадров различных типов, выполняются оценки минимального и максимального значений ожидаемого времени анализа последовательности кадров и с их учетом осуществляется выбор подходящих стратегий анализа.

Экспериментальные результаты

В докладе представлены результаты вычислительных экспериментов на примере системы распознавания регистрационных номеров автотранспортных средств. Исследованы различные варианты построения системы:

1). Без входного буфера видеоданных ($N_b=1$);

Каждый кадр обрабатывается при помощи единственной стратегии. Буфер видеоданных содержит последний полученный на текущий момент времени кадр. После завершения обработки кадра начинается обработка кадра, находящегося в буфере и т. д.

2). С буфером видеоданных ($N_b>1$)

Система, как и прежде, функционирует по детерминированному алгоритму, но имеется возможность буферизации видеоданных. После завершения обработки кадра начинается обработка первого находящегося в буфере кадра и т. д.

3). С буфером видеоданных ($N_b>1$) и адаптивным выбором стратегии распознавания.

Результаты моделирования подтвердили эффективность предложенной модели СР в широком диапазоне изменения внешних условий (скорость движения ТС в зоне контроля, изменение доступности вычислительных ресурсов).

Дальнейшие исследования

В дальнейшем планируется разработать алгоритм планирования процесса распознавания с учетом лишь частичной обработки видеопоследовательности, что как ожидается должно дополнительно улучшить характеристики проектируемых систем распознавания буквенно-цифровых меток движущихся объектов.

Список литературы

- [1] Журавлев Ю. И., Рудаков К. В., Гуров С. И., Дюкова Е. В., Кутуков Г. П., Матюнин С. Н., Местецкий Л. М. Состояние и перспективы развития исследований в области обработки и распознавания видеоинформации (аналитический обзор). Журнал «Инженерное образование» №1 Январь 2005.
- [2] Бондаренко А. В., Галактионов В. А., Горемычкин В. И., Ермаков А. В., Желтов С.Ю. Исследование подходов к построению систем автоматического считывания символьной информации. Препринт ИПМ им. М.В.Келдыша РАН
- [3] Biological Sequence Analysis, Durbin, Eddy, Krogh, and Mitchison, Cambridge University Press, 1998.

Оболочка экспертных систем ХС# 1.0: Подход к автоматизированному извлечению онтологий и их применению в компоненте объяснения

Гадиятулин Р. А.

*Пермский государственный университет
E-mail: gadiatulinn@perm.ru, Chuprina@psu.ru*

В настоящее время экспертные системы (ЭС) остаются одним из наиболее широко применяемых на практике приложений искусственного интеллекта (ИИ). Накоплен большой объем баз знаний (БЗ) созданных не только в различных инструментальных средах, но и в различных парадигмах: продукционной,

логической, на семантических сетях, фреймовой. Совместное использование знаний из таких разнородных систем затруднено в виду того, что обычно традиционные базы знаний функционируют только в своем инструментальном окружении. Проблема тиражирования знаний существует не сама по себе, а в связи с необходимостью решения конкретной задачи с использованием существующих баз знаний

В рамках проекта XG# предлагается решение перечисленных выше проблем: разработано специальное инструментальное окружение для декларативного описания грамматики языка, учитывающей специфику ПО; все остальные компоненты ЭС генерируются автоматически. Это позволяет не только создавать адекватные специфике ПО языки представления знаний, но и решить проблему тиражирования знаний, а также целый круг задач, связанных с отладкой, модификацией, интеграцией и тиражированием баз знаний. В основе предлагаемого подхода – активное использование метазнаний не только для описания семантики системы, но и активное использование онтологий связанных как с описанием ПО и пользователя. Однако, проектирование и разработка онтологий, то есть онтологический инжиниринг, не является тривиальной задачей. Он требует от разработчиков профессионального владения технологиями инженерии знаний — от методов извлечения знаний до структурирования и формализации.

Рассматривается возможность автоматизации извлечения знаний более высокого уровня из предметных баз знаний продукционных ЭС, такой подход можно условно назвать rule-mining и применение приобретенных таким образом знаний для решения перечисленных ниже задач:

- помощь в отладке, модификации и интеграции баз знаний;
- познавательная цель: создавая онтологию, эксперт анализирует связи понятий проблемной области
- тиражирование онтологических знаний;
- тиражирование баз знаний (трансформация из одной оболочки экспертных систем в другую);
- облегчение ввода в систему новых знаний, обнаружение и исправление ошибок при вводе (модификации) знаний;
- повышение интеллектуальности объяснения хода логического вывода (обоснование правил проблемного уровня и выдача некоторой дополнительной информации).

В рамках работы предлагается методика извлечения метазнаний из "сырых" предметных баз знаний.

Идея этого метода такова: считать дополнительным знанием сами структуры хранения знания о проблемной области, если шире – то все, что содержится в экспертной системе и среде ее выполнения и из чего можно извлечь онтологические метазнания на основе некоторых эвристик и помощи пользователя.

Итогом исследований стал вывод, что процесс извлечения онтологических метазнаний из существующих предметных баз знаний можно до некоторой степени автоматизировать, тем самым, упростив труд эксперта. Реализована компонента системы XG# для извлечения онтологий и редактор онтологий в формате OWL. Создание онтологии понятий предлагаемым методом выглядит как последовательность следующих этапов:

1. Автоматически по базе знаний строится дерево целей (данная возможность играет роль дополнительного средства отладки баз знаний);
2. Система выполняет анализ каузальных зависимостей и формирует список возможных связей;
3. Пользователю-эксперту предлагается ответить на ряд вопросов, после чего формируется некоторый вариант онтологии;
4. Если вариант, сгенерированный системой не устраивает эксперта, он имеет визуальный инструментарий (редактор онтологий) для модификации онтологии.

Результаты

Использование, полученных таким образом, онтологических знаний обеспечит развитие возможности выдачи подсказок при редактировании базы знаний, и в частности, в клаузе объяснения правила.

Объяснения с использованием онтологии могут генерироваться "на лету" с учетом нужного уровня детализации.

Включение компоненты для работы с онтологиями в XG# откроет широкие возможности применения в образовательном процессе: индивидуальное задание на составление онтологии ПО поможет студентам познакомиться на практике с семантическими сетями и с технологическими аспектами вывода на СС. При этом существует возможность учета не только связей между понятиями ПО, но и онтологического профиля пользователя. Уровень пользователя тоже может выводиться из онтологических знаний – профили пользователя могут быть представлены явно в виде метазнаний (как часть онтологии).

Обработка многомерных сигналов на основе стохастических моделей вейвлет коэффициентов

Гай В. Е.,

602264 Владимирская область, г.Муром,
ул. Радиозаводское шоссе д.48, кв. 93,
e-mail: iamuser@inbox.ru

При решении задач обработки многомерных сигналов и, в частности, изображений, как правило, используется некоторая вероятностная модель, описывающая статистические связи между отсчетами.

Разработка алгоритмов обработки сигналов на основе многомерного вейвлет преобразования требует построения математических моделей, описывающих связь как между элементами изображения, находящимися на одном уровне разложения, так и между элементами, расположенными на соседних уровнях разложения исходного изображения.

Анализ гистограмм вейвлет – коэффициентов различных изображений показывает, что они имеют резкий пик на нуле и продолжительный спад в обе стороны от начала координат, значительно отличаясь от кривой гауссова распределения [1].

Если считать, что коэффициенты вейвлет разложения в каждом частотном диапазоне независимы и имеют одинаковое распределение, то статистическая модель будет полностью определяться безусловной статистикой вейвлет – коэффициентов. Более точно, по сравнению с гауссовым, аппроксимировать гистограмму вейвлет коэффициентов позволяет обобщенное гауссово распределение

$$P_{\alpha,\beta}(x) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} \exp\left(-\left(\frac{|x|}{\alpha}\right)^\beta\right),$$

где показатель β определяет форму распределения, при $\beta = 2$ $P_{\alpha,\beta}(x)$ соответствует гауссовой плотности, $\beta = 1$ $P_{\alpha,\beta}(x)$ соответствует плотности Лапласа, параметр α определяет среднеквадратическое отклонение распределения.

Основным недостатком такой модели является то, что коэффициенты вейвлет – преобразования рассматриваются как независимые, что не соответствует действительности. Таким образом, для более качественного описания статистических свойств изображения необходимо использовать совместные законы распределения спектральных коэффициентов.

При анализе коэффициентов двумерного вейвлет преобразования реальных изображений, можно видеть, что большие по величине коэффициенты появляются в одних и тех же относительных местоположениях в частотных диапазонах на соседних масштабах и направлениях.

Зависимости между амплитудами коэффициентов могут быть описаны с использованием случайного поля с пространственно изменяющейся дисперсией. Моделирование данных зависимостей можно осуществить с использованием гауссовой масштабируемой смеси [2, 3].

При этом с каждым вейвлет - коэффициентом ω_i связывается некоторая переменная, описывающая скрытое состояние $S_i \in \{S, L\}$. Значение S_i определяет, какая из двух компонент в модели смеси определяет ω_i .

Состояние S соответствует гауссовому распределению с нулевым средним и с малой дисперсией σ_S^2 , состояние L соответствует гауссовому распределению с большой дисперсией σ_L^2 .

При этом безусловная плотность распределения вероятностей $f(\omega_i)$ описывается как комбинация условных плотностей

$$f(\omega_i) = p_i^S * g(\omega_i; 0, \sigma_{S,i}^2) + p_i^L * g(\omega_i; 0, \sigma_{L,i}^2), \text{ где } p_i^S = 1 - p_i^L.$$

Величины p_i^S и p_i^L могут быть интерпретированы, соответственно, как вероятности того, что значение ω_i будет маленьким или большим.

Описание зависимостей между уровнями разложения

Для описания зависимостей между уровнями разложения используются вероятностные деревья, которые связывают переменную характеризующую скрытое состояние каждого вейвлет коэффициента с переменной характеризующей скрытое состояние каждого его потомка. Это приводит к созданию графа зависимостей имеющего такую же топологию квадродерева как и вейвлет коэффициенты.

Каждая полоса частот представлена квадродеревом (при этом подразумевается, что полосы частот являются независимыми). Каждая связь предок \rightarrow потомок имеет соответствующую матрицу переходов между состояниями

$$A_i = \begin{bmatrix} p^{S \rightarrow S}_i & p^{S \rightarrow L}_i \\ p^{L \rightarrow S}_i & p^{L \rightarrow L}_i \end{bmatrix}, p^{S \rightarrow L}_i = 1 - p^{S \rightarrow S}_i, p^{L \rightarrow S}_i = 1 - p^{L \rightarrow L}_i.$$

Величины $p^{S \rightarrow L}_i, p^{L \rightarrow S}_i$ определяют вероятность того, что значение состояния вейвлет коэффициента будет изменяться при переходе от текущего масштаба к следующему. Для того, чтобы величины вейвлет коэффициентов сохранялись при переходе от масштабу к масштабу (т.е. чтобы у небольшого по величине вейвлет коэффициента все потомки были тоже были небольшими, а у большого по величине вейвлет коэффициента все потомки по величине также были большими) необходимо чтобы

$$p^{S \rightarrow S}_i > p^{S \rightarrow L}_i; p^{L \rightarrow L}_i > p^{L \rightarrow S}_i.$$

Модель скрытого Марковского дерева (СМД) определяется дисперсией смесей $\sigma_{S,i}^2, \sigma_{L,i}^2$, матрицей переходов между состояниями A_i и вероятностью p^L_i того, что корневой узел будет иметь большое значение для каждого i .

Объединяя данные параметры в вектор Θ , получаем модель СМД, которая представляет собой параметрическую модель для совместной функции плотности распределения вероятности $f(w|\Theta)$ вейвлет-коэффициентов в каждом из трёх частотных диапазонах (диапазоны рассматриваются как статистически независимые).

Универсальная модель СМД

Модель СМД описывает основные статистические свойства вейвлет – преобразования реальных изображений. Для описания модели в предложенной выше форме требуется $4J$ параметров, где J -число уровней разложения.

Для описания модели СМД с уменьшенным количеством параметров учитываются следующие свойства вейвлет коэффициентов:

1. Экспоненциальное уменьшение значений величины вейвлет – коэффициентов при переходе от одного уровня разложения к другому.

2. Высокую степень постоянства значений вейвлет – коэффициентов на конечных уровнях разложения.

На основе данных свойств вейвлет – преобразования можно определить функциональную форму параметров модели. Уменьшение величин вейвлет коэффициентов и изменение постоянства вейвлет – коэффициентов может быть описано с использованием шаблонов, которые показывают как дисперсии смеси и вероятности переходов изменяются в соответствии с изменением уровня разложения.

Так как дисперсии смесей СМД характеризуют величину вейвлет – коэффициента, можно потребовать, чтобы они уменьшались экспоненциально при переходе к более грубому уровню разложения:

$$\sigma_{L,j}^2 = C_{\sigma S} 2^{-j\alpha L}, \sigma_{S,j}^2 = C_{\sigma S} 2^{-j\alpha S}.$$

Так как величина вейвлет – коэффициентов, которые соответствуют контурам уменьшается медленнее, чем тех, которые представляют гладкие области, требуется, чтобы $\sigma_{S,j}^2 < \sigma_{L,j}^2$ для всех масштабов, а также

$\alpha_s \geq \alpha_L$. Таким образом, данные параметры описывают модель изображения, поведение спектра которого характеризуется как $1/f$.

Четыре параметра $C_{\sigma S}, C_{\sigma L}, \alpha_S, \alpha_L$ описывают безусловные плотности вейвлет коэффициентов.

Из анализа ансамблей реальных изображений следует, что изменение значений матрицы переходов можно аппроксимировать экспонентой (при рассмотрении значений матрицы одновременно на нескольких уровнях разложения). Введём следующую экспоненциальную форму матрицу состояний переходов:

$$A_j = \begin{bmatrix} 1 - C_{ss} 2^{-\gamma_{sj}} & C_{ss} 2^{-\gamma_{sj}} \\ \frac{1}{2} - C_{LL} 2^{-\gamma_{Lj}} & \frac{1}{2} + C_{LL} 2^{-\gamma_{Lj}} \end{bmatrix}.$$

Таким образом, универсальная модель СМД записывается в виде следующих девяти параметров:

$$\Theta_m = \{\alpha_s, C_{\sigma S}, \alpha_L, C_{\sigma L}, \gamma_S, C_{SS}, \gamma_L, C_{LL}, p_{j_0}^L\}.$$

Для определения параметров модели УСМД можно использовать два подхода:

- использовать алгоритм ожидания – максимизации для определения параметров на основе исходного изображения; данный подход не является предпочтительным, так как время, затраченное на обучение, может оказаться большим, чем в случае модели СМД;

- непосредственное определение параметров модели УСМД по ансамблю изображений.

При использовании второго подхода коэффициенты снижения значения дисперсии и постоянства оценивались аппроксимацией линии (с использованием логарифмического масштаба) для каждого уровня разложения. Универсальные параметры, полученные при обучении модели следующие:

$$\Theta_m = \left\{ \alpha_s = 3.1; C_{\sigma_s} = 2^{1.1}; \alpha_L = 2.25; C_{\sigma_L} = 2^{1.1}; \gamma_S = 1; C_{SS} = 2^{2.3}; \gamma_L = 0.4; C_{LL} = 2^{0.5}; p_{j_0}^L = \frac{1}{2} \right\}$$

Алгоритм фильтрации изображений на базе модели СМД

Пусть исходное изображение x искажено аддитивным гауссовым шумом n с нулевым средним и дисперсией σ_n^2 .

После вейвлет-преобразования получим:

$$y = w + n',$$

где n' - гауссовский шум с дисперсией σ_n^2 , w - коэффициенты вейвлет преобразования сигнала без шума [4,5].

Подход к фильтрации изображения состоит в построении модели СМД применительно к y и затем использовании данной модели для определения оценки w на основе y .

Для построения модели СМД, которая определяется набором параметров Θ , на основе y используется алгоритм ожидания – максимизации. Аддитивный белый гауссовский шум n' с нулевым средним увеличивает дисперсию каждой компоненты смеси $\sigma_{i,m}^2$ на σ_n^2 . Другие параметры модели не изменяются под действием аддитивной шумовой компоненты.

Так как w - реализация случайного поля с совместной плотностью распределения вероятности $f(w | \Theta)$, модель которого строится на основе СМД, то для осуществления оценки исходного сигнала из зашумлённого, применим байесовский подход.

Для получения минимальной среднеквадратической оценки вейвлет коэффициентов из параметров модели СМД Θ и искажённого шумом изображения y используется следующее выражение:

$$\hat{w}_i = E[w_i | y, \Theta] = \sum_{q \in \{S, L\}} p(S_i = q | y, \Theta) \frac{\sigma_{q,i}^2}{\sigma_n^2 + \sigma_{q,i}^2} y_i$$

Анализ результатов фильтрации тестовых изображений показал, что данный подход обеспечивает лучшее подавление шума по сравнению с традиционными вейвлет алгоритмами, основанными на автоматическом выборе порога [5].

Возможные перспективы развития данного подхода к описанию изображений заключается в разработке различных алгоритмов (или адаптация классических алгоритмов с определённой модификацией) обработки изображений, решающих поставленные задачи более качественно, чем классические методы.

Список литературы

1. Жизняков А. Л., Вакунов Н. В. Вейвлет-преобразование в обработке и анализе изображений. – М.: Государственный научный центр Российской Федерации – ВНИИ Геосистем, 2004.-102 с.
2. Andrews D., Mallows C., Scale mixtures of normal distributions. Royal Statistic Society, 36, pp. 99-102, 1974
3. T. M. Cover and J. A. Thomas, Elements of Information Theory, Wiley-Interscience, New York, 1991.
4. Donoho D., Johnstone I. Adapting to unknown smoothness via wavelet shrinkage, J. Amer. Stat. Assoc., vol. 90, pp. 1200-1224, 1995
5. Chapman H., Kolaczyk E., McCulloch M. Signal denoising using adaptive Bayesian wavelet shrinkage, in IEEE Int. Symp. Time-Frequency and Time-Scale Analysis, Paris, pp. 225-228, 1996

Общие подходы к классификации инфокоммуникационных услуг

к.т.н. доц. **Гайваронская Г. С.,**
Павлов С. В.,
Одесская государственная академия холода
Украина, г. Одесса (65082), ул. Дворянская, 1/3.
E-mail: semyen@rambler.ru.

В данной статье рассмотрен подход к классификации инфокоммуникационных услуг. Приведены методы исследования и его результаты.

In this article an approach to infocommunication services classification is researched. The methods of research and its results are presented.

Ключевые слова: методика исследования инфокоммуникационных услуг – классы инфокоммуникационных услуг – модель представления классов инфокоммуникационных услуг

Keywords: method of research of infocommunication services – classes of infocommunication services – model of presentation of classes of infocommunication services.

Введение

Стремительное развитие научно-технического прогресса в области телекоммуникаций и информатики обеспечивает в настоящее время возможность предоставления обширного спектра инфокоммуникационных услуг широкому кругу пользователей.

Инфокоммуникационная услуга (услуга информационного общества) предполагает автоматизированную обработку, хранение или предоставление по запросу информации с использованием средств вычислительной техники, как на входящем, так и на исходящем конце соединения.

Число этих услуг постоянно увеличивается с появлением новых технологий, способных предоставлять более качественные и ресурсоемкие услуги, увеличением потребностей в предоставлении узкоспециализированных услуг, а также с изменениями в экономической, социальной и политической жизни страны.

В результате такое огромное количество инфокоммуникационных услуг представляет трудности, которые может ощутить любой оператор связи или консультант, пытаясь обосновать необходимость вложения средств пользователя в предоставление определенного набора услуг, как, впрочем, и определить последний, а тем более определить требования к показателям сети, предъявляемых данными услугами. Поэтому появляется необходимость в классификации этих услуг. Классификация услуг, прежде всего, необходима для быстрого и качественного определения требований к оборудованию, так как для предоставления каждой инфокоммуникационной услуги необходимо выполнить определенный набор требований (например, к скорости информационного доступа или к коэффициенту ошибок). При появлении новых услуг данная классификация позволит определить возможность предоставления таких услуг в сети.

На сегодняшний день научные исследования, посвященные анализу инфокоммуникационных услуг, находятся на начальном этапе. Это обуславливается тем, что необходимость проведения подобного рода исследований появилась недавно, а также потребностью не только охватить все услуги, существующие на рынке, но и предусмотреть появление новых. Нормативные и законодательные документы Украины, а так же рекомендации и стандарты международных организаций не определяют классификацию услуг.

Методика исследования.

Для проведения классификации инфокоммуникационных услуг, выделения отдельных классов услуг по совокупности ряда параметров и определении требований этих классов услуг к сети и оборудованию необходимо решить следующие задачи:

1. Необходимо сформировать список тех услуг, которые фигурируют на сегодняшнем рынке.
2. После определения наиболее полного каталога услуг, необходимо определить показатели, необходимые для их предоставления.
3. Проанализировать качественно все параметры, по которым возможна классификация услуг. К таким параметрам, например, можно отнести: скорость установления соединений, скорость и объемы передаваемой информации для каждого типа услуг, допустимое абсолютное значение и допустимые вариации задержки информации в сети, допустимый коэффициент ошибок и др. Определить существенные показатели.
4. После определения необходимых параметров, сформировать классы инфокоммуникационных услуг.
5. Сформулировать требования, выдвигаемые классами инфокоммуникационных услуг на процесс развития сети.

Данная методика классификации инфокоммуникационных услуг реализована на практике.

Был сформирован список услуг, которые фигурируют на сегодняшнем рынке. Рассмотрен широкий спектр услуг, включая передачу текстовых и электронных сообщений, передачу данных по специализированным и неспециализированным сетям с разными методами коммутации в различных режимах переноса информации. Рассмотрены также услуги голосовой связи, телеконференций и услуги информационных служб (такие как информационно-справочные, видео по запросу, доступа к информационным ресурсам и другие), а так же услуги мультимедиа.

Были сформированы показатели, необходимые для предоставления данных инфокоммуникационных услуг на основе рекомендаций, разработанных сектором телекоммуникаций Международного союза электросвязи. При выборе параметров учтены требования как пользователей так и операторов связи, предоставляющих услуги.

Далее выбраны наиболее существенные параметры (те параметры, которые остаются неизменными вне зависимости от того, какие методы и средства используются для предоставления услуг), определяющие требования к сетям и оборудованию. Это следующие параметры:

- вид услуги – определяет, является услуга основной, дополнительной либо же интеллектуальной;
- вид передаваемой информации – характеризует, каким видом информации оперирует данная услуга;
- информационная скорость доступа – показывает, какая скорость необходима для качественного предоставления ИКУ;
- допустимое время задержки – приемлемое время задержки данных, при которой не теряется их информационный смысл;
- готовность сети – коэффициент готовности при связи между терминалами пользователей;
- допустимый коэффициент ошибок – число, которое определяет порог ошибок, при котором полученные данные будут восприняты правильно;
- время доступа к службе – определяет время, которое необходимо для получения какой-либо услуги;
- служба – организация или обеспечение связи, требуемой для предоставления конкретного вида услуг.

Другие параметры не были учтены в силу того, что они либо не принципиальны к предоставлению услуг, либо изменяются в зависимости от того, как реализована связь для предоставления конкретной услуги. Например, не будем учитывать такой показатель как «освобождение», потому как этот параметр будет варьироваться в зависимости от того, как реализована служба связи.

Затем была проведена классификация по следующим показателям:

- служба;
- вид услуги;
- вид передаваемой информации;
- информационная скорость доступа;
- допустимое время задержки;
- готовность сети;
- допустимый коэффициент ошибок;
- время доступа к службе.

Данная классификация необходима для того, чтобы качественно оценить все указанные показатели.

В результате классификации выяснилось, что некоторые показатели не подходят для определения классов инфокоммуникационных услуг.

При классификации услуг по службе, посредством которой они предоставляются, выяснилось, что данная классификация будет являться неоднозначной. Информационная скорость предоставления услуг, а также допустимое время задержки, готовность сети, допустимый коэффициент ошибок службы передачи данных зависит от технологии сети, на которой реализована эта служба.

Проанализировав данные, которые были получены после формирования показателей, необходимых для предоставления данных услуг, можно сделать вывод, что для предоставления как основных, так дополнительных и интеллектуальных услуг используются одни и те же параметры (если только услуга не ориентирована на расширение параметров предоставления услуг, например, увеличение полосы пропускания для передачи данных). Расширение полосы пропускания возможно лишь в ТМ службах, которые позволяют это сделать (например, в узкополосной цифровой сети интегрального обслуживания увеличить общую пропускную способность возможно лишь до 2048 кбит/с). Поэтому с точки зрения вариантов классификации показатель вида услуги не меняет структуру классификации, поэтому его можно не учитывать.

Вид передаваемой информации может быть следующим: текстовые и электронные сообщения, неподвижные и подвижные изображения, речь, вызов, аудио, видео, данные, мультимедиа.

Показатели предоставления инфокоммуникационных услуг варьируются в широких диапазонах. Для удобства работы и для большей конкретизации определим в этих диапазонах условные промежутки.

Данные, полученные при анализе инфокоммуникационных услуг свидетельствуют о том, что значительная группа услуг требует полосы пропускания не более 64 кбит/с. Узкополосная цифровая сеть интегрального обслуживания ограничена скоростью 2048 кбит/с, можно выделить второй промежуток – 64...2048 кбит/с.

Третий и четвертый промежуток определяем, опираясь на определения широкополосной цифровой сети интегрального обслуживания – 2048...143360 кбит/с и более 143360 кбит/с.

Допустимое время задержки для не интерактивных услуг (которые предполагают промежуточное накопление и автоматическую обработку) составляет в зависимости от класса обслуживания сообщения: не более 4 часов; 4...12 часов; более 12 часов.

Для интерактивных услуг и услуг реального времени определены следующие границы: менее 500 мс (речь, видео); 500...1800 мс (передача данных); более 1800 мс (резервируется на случай выявления интерактивной услуги, допускающей задержки более 1800 мс).

В рекомендации Международного института электросвязи (Н.221) приведена взаимосвязь, характеризующая соотношение между показателем коэффициента ошибок и процента ошибочных блоков при кодировании с помощью циклического избыточного кода.

Эта взаимосвязь дает представление о количестве ошибочных блоков при определенном коэффициенте ошибок. Можно сделать вывод, что промежуток 10^{-8} ... 10^{-5} будет центральным с точки зрения возможности восстановления ошибочных пакетов.

Большинство ИКУ требуют времени доступа к службе в диапазоне 10...20 секунд. Левая и правая последовательность вне данного промежутка определяют еще два промежутка.

Анализируя все вышеуказанные промежуточные классификации и оценив параметры, которые являются решающими в определении классов услуг, можно приступать к формированию классов инфокоммуникационных услуг по решающим показателям.

Совмещение всех промежуточных классификаций показывает, что привязка к виду передаваемой информации влечет за собой двойные диапазоны показателей. Например, для передачи видео возможна информационная скорость доступа в промежутках 4...2048 кбит/с и 2048...143360 кбит/с. Не возможно будет определить пороговый класс услуг и выдвинуть конкретные требования к сети и оборудованию.

Чтобы определить непротиворечивые классы, необходимо избавиться от показателя вида передаваемой информации. Этот параметр был необходим для определения вида информации, являющегося наиболее ресурсоемким с точки зрения полосы пропускания.

Именно полоса пропускания, на мой взгляд, является наиболее важным параметром предоставления инфокоммуникационных услуг. От нее в первую очередь зависит возможность предоставления конкретного типа услуг. Например, мы не сможем предоставлять видео повышенного качества, используя полосу пропускания телефонной сети общего пользования.

Итак, оценив наиболее важный показатель предоставления услуг, и определив диапазоны других, ранее приведенных параметров, можно приступить к выполнению классификации.

Сначала определим количество классов. Их будет четыре, так как определяющий показатель (информационная скорость доступа), определен в четыре промежутка. Показатель допустимого времени задержки необходимо будет указывать с помощью двух полей (для услуг, предоставляемых в режиме реального времени, и для услуг, предполагающих промежуточное накопление и обработку).

Можно было бы организовать пять классов (один класс – отдельно для не интерактивных услуг, полоса пропускания которых на сегодняшний день, как правило, не превышает 64 кбит/с). Однако стремительное развитие технологий в скором времени может дать возможность любому пользователю просматривать и отправлять видео сообщения на свой, например, «мультимедийный почтовый ящик» (в службах голосовой связи такую функцию для голосовых сообщений выполняет «голосовой почтовый ящик»). Тогда уже необходима большая пропускная способность. Поэтому показатель допустимого времени задержки будем указывать с помощью двух полей.

Теперь определим количество подклассов для каждого класса. Их будет три, так как все диапазоны показателей услуг (кроме информационной скорости доступа) условно разделены на три промежутка. Данное разделение является наиболее не громоздким, удобным и не противоречивым.

После определения количества классов и их подклассов, можно представить таблицу классов инфокоммуникационных услуг.

Приведенная таблица полностью отображает сформированные классы ИКУ, однако не является наглядной, этому способствуют повторяющиеся части, подобные во всех подклассах. Из тех частей, которые уникальны для каждого класса, сформируем куб. Повторяющиеся же данные определим как таблицу, справедливую для всех подклассов.

Полученная в результате преобразований модель представлена на рисунке 1.

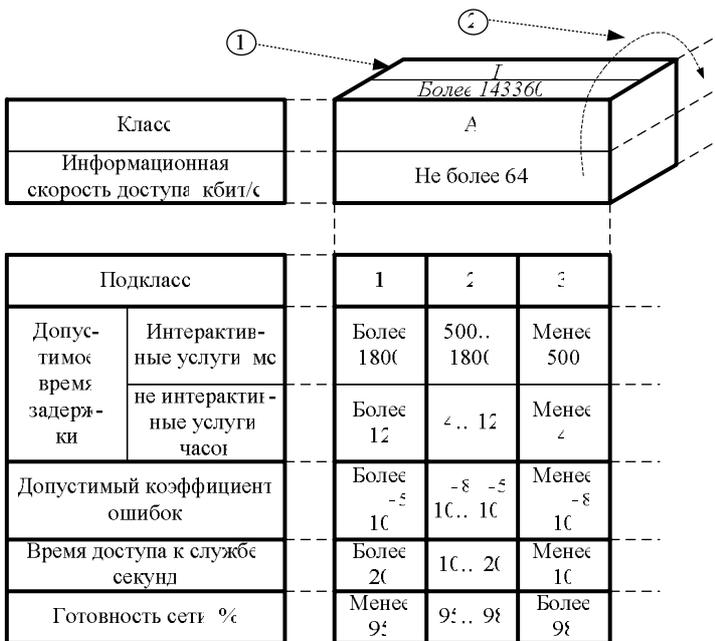
Подвижный куб в развернутом виде выглядит так, как показано на рисунке 2.

Приведенная структура хороша тем, что она наглядна и очень гибка к изменениям.

Класс		А			Б			В			Г		
Информационная скорость доступа, (кбит/с)		Не более 64			64...2048			2048...143360			Более 143360		
Подкласс		А1	А2	А3	Б1	Б2	Б3	В1	В2	В3	Г1	Г2	Г3
Допустимое время задержки	Интерак-тивные, (мс)	Более 1800	500...1800	Менее 500									
	не интерак-тивные, (часов)	Более 12	4...12	Менее 4									
Допустимый коэффициент ошибок		Более -5 10	-8 10... -5 10	Менее -8 10									

Время доступа к службе, секунд	Более 20	10... 20	Менее 10	Более 20	10... 20	Менее 10	Более 20	10... 20	Менее 10	Более 20	10... 20	Менее 10
Готовность сети, %	Менее 95	95... 98	Более 98	Менее 95	95... 98	Более 98	Менее 95	95... 98	Более 98	Менее 95	95... 98	Более 98

Таблица 1 Таблица классов инфокоммуникационных услуг



Условные графические обозначения на рисунке ① Подвижный куб ②

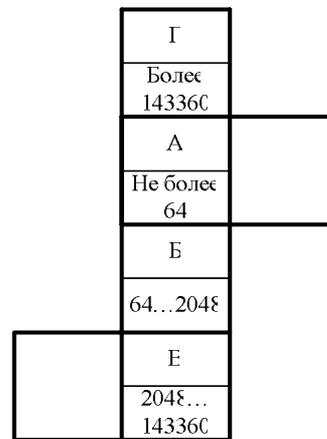


Рис 1 Модель представления классов ИКУ

Рис 2 Подвижный куб в развернутом виде

Результаты исследования

В данном исследовании была проведена классификация по различным параметрам, оценены возможные варианты, выбраны наиболее важные. В результате получены конкретные классы инфокоммуникационных услуг. Построена модель представления классов услуг.

Классы инфокоммуникационных услуг однозначно определяют требования, выдвигаемые услугами к сетям и оборудованию. Данные классы облегчают выбор либо замену участков сети или оборудования.

Список литературы.

1. Гук М. Аппаратные средства локальных сетей. Энциклопедия – СПб.: Питер, 2005. – 573 с.;
2. Сети и системы телекоммуникаций – М.В. Захарченко, Г.С. Гайворонская, А.И. Ещенко и др. К.: Техника, 2000. – 304 с.;
3. “Законы Украины”, официальное Интернет-представительство Украинского государственного центра радиочастот, Internet-ресурс: <http://www.ucrf.gov.ua/ru/doc/laws/>;

4. “Международный союз электросвязи”, Internet-ресурс: <http://www.itu.int>;
5. “Электронный журнал «Электросвязь»”, Internet-ресурс: <http://rtuis.miem.edu.ru>;
6. “Электронный журнал «СНПР»”, Internet-ресурс: <http://ichip.ru>;
7. “Электронный журнал «Сети и Системы Связи»”, Internet-ресурс: <http://www.ccc.ru>;

Поиск всех безызбыточных столбцовых покрытий булевых матриц в интеллектуальных распознающих системах³

Гедике А. И.,

*доцент Томского государственного
архитектурно-строительного университета
634003, г. Томск, пл. Соляная, д. 2
E-mail: gai@tsuab.ru*

Введение

Задача нахождения всех безызбыточных (неприводимых) столбцовых покрытий булевых матриц большой размерности весьма актуальна при решении ряда задач в различных проблемных областях.

В интеллектуальных распознающих системах, разрабатываемых в лаборатории интеллектуальных систем Томского государственного архитектурно-строительного университета (ЛИС ТГАСУ), к данной задаче сводится задача построения всех безызбыточных (тупиковых) безусловных диагностических тестов (ББДТ), используемых для построения решающих правил и принятия решений относительно исследуемых объектов из рассматриваемой проблемной области [1]. Отметим, что тупиковые тесты были введены С.В. Яблонским [2] при рассмотрении задачи логического контроля электрических схем, а в распознавании образов обоснование их применения совпадает с выходом работы Ю.И. Журавлева и И.Б. Гуревича [3].

Для нахождения всех безызбыточных столбцовых покрытий (БСП) булевой матрицы большой размерности предлагается оригинальный алгоритм, основанный на применении ряда правил и специальной разметки столбцов булевой матрицы, используемых в процессе построения и обхода дерева поиска, что позволяет существенно сократить число переборov путем отсечения неоптимальных путей в дереве поиска.

Алгоритм программно реализован и включен в состав интеллектуального инструментального средства (ИИС) ИМСЛОГ [4], являющегося интегрированной средой разработки прикладных интеллектуальных распознающих систем.

Постановка задачи

В соответствии с логико-комбинаторным подходом к тестовому распознаванию образов, предложенным А.Е. Янковской [1] и развиваемым в ЛИС ТГАСУ, считается, что одни и те же изучаемые объекты можно разбить различными способами (механизмами классификации) на классы эквивалентности. При этом под образом понимается подмножество объектов, принадлежащих одним и тем же классам эквивалентности по совокупности всех механизмов классификации.

Для представления данных и знаний об изучаемых объектах используется матричная модель, включающая матрицы описаний (Q) и различий (R).

Строки матрицы Q сопоставляются изучаемым объектам, столбцы – признакам, задающим описания объектов. Строки матрицы R сопоставляются строкам матрицы Q, столбцы – уровням различия (механизмам классификации), задающим разбиения изучаемых объектов на классы эквивалентности.

Описание каждого образа задается подмножеством строк матрицы Q, которым сопоставлены одинаковые строки матрицы R.

³ Работа выполнена при финансовой поддержке РФФИ (проект № 04-01-00144) и РГНФ (проект № 06-06-12603В).

Для представления условий различимости изучаемых объектов, принадлежащих разным образам, по матрицам Q и R строится безыбыточная (не содержащая поглощающих строк) булева матрица импликаций (U).

Столбцы матрицы U сопоставляются признакам, задающим описания объектов (столбцам матрицы Q), строки – результатам сравнения пар типа образ-образ, объект-образ и объект-объект из разных образов. Элемент строки матрицы U принимает единичное значение, если соответствующий признак является различающим для данной пары, иначе – нулевое значение. Таким образом, матрица U задает условия различимости любой пары объектов, принадлежащих разным образам, описания которых представлены матрицами Q и R .

Требуется найти все безыбыточные подмножества признаков, являющиеся ББДТ, каждое из которых задает необходимые и достаточные условия различимости любой пары объектов из разных образов, представленных матрицами Q и R .

Данная задача эквивалентна поиску всех БСП матрицы U , поскольку каждое подмножество признаков, соответствующее искомому БСП, задает необходимые и достаточные условия различимости любой пары строк матрицы Q , которым сопоставлены неодинаковые строки матрицы R , и является по сути ББДТ.

Описание алгоритма

Пусть задана булева матрица U размерности $n \times m$, где n – число строк, m – число столбцов. Дерево поиска всех БСП строится в виде иерархической системы подматриц матрицы U .

Правило выделения подматрицы по j -му столбцу ($j=1, \dots, m$) состоит в удалении j -го столбца и всех строк, содержащих единицы в j -ом столбце.

Корню дерева поиска (0-му уровню) сопоставляется матрица U ; вершинам 1-го уровня – подматрицы, выделяемые из матрицы U ; вершинам i -го уровня ($i=2, 3, \dots$) – подматрицы, выделяемые из подматриц ($i-1$)-го уровня. Концевым вершинам дерева поиска (листьям) соответствуют подматрицы, состоящие из одного столбца.

Каждой вершине i -го уровня ($i=1, 2, \dots$) присваивается значение, равное номеру столбца, по которому была выделена сопоставленная данной вершине подматрица, каждому листу – значение, равное номеру сопоставленного ему столбца.

Множество значений вершин, входящих в некоторый путь от корня к листу, задает столбцовое покрытие.

Если два или более пути состоят из вершин с одинаковыми значениями, но следующих в различном порядке, то они задают одни и те же столбцовые покрытия. В этом случае каждый путь, кроме найденного первым, считается повторяющимся.

Путь называется избыточным, если хотя бы одно подмножество значений его вершин равно множеству значений вершин другого пути. Каждый неповторяющийся безыбыточный путь в дереве поиска задает БСП матрицы U .

Предлагаемый алгоритм нахождения всех БСП булевой матрицы в процессе построения дерева поиска является развитием алгоритма, представленного в [5], и основан на применении ряда правил и специальной разметки столбцов, что позволяет (при существенном сокращении общего числа переборов) отсекал не приводящие к столбцовым покрытиям ветви дерева поиска, исключить построение повторяющихся путей, строить часть БСП без выделения подматриц, полностью исключить или свести к минимуму число построенных избыточных столбцовых покрытий.

При построении дерева поиска в каждой вершине, на каждом i -м ($i=0, 1, \dots$) уровне дерева поиска применяются следующие правила к соответствующим этим вершинам матрице U и текущим подматрицам,

обозначаемым далее U_i^T .

Правило 1. Нулевые столбцы удаляются.

Правило 2. Из каждой группы одинаковых столбцов удаляются (с запоминанием их номеров) все, кроме одного (левого).

Правило 3 (3'). Если есть строка (строки), содержащая одну единицу, то соответствующий этой единице столбец (столбцы) выбирается (выбираются) для построения вершины ($i+1$)-го уровня.

Правило 4. Если правило 3 (3') неприменимо и есть один столбец с наибольшим числом единиц, то данный столбец выбирается для построения вершины ($i+1$)-го уровня, так как в подматрице, соответствующей именно этой построенной вершине ($i+1$)-го уровня, остается наименьшее число непокрытых строк.

Правило 5. Если правило 3 (3') неприменимо и есть несколько столбцов с наибольшим числом единиц, то для каждого из них нужно строить вершины ($i+1$)-го уровня. К каждой из них следует последовательно

применить правила 1, 2, 3 (но не 3'), 4, 5 и выбрать тот столбец, при удалении которого остается наименьшее число строк в одной из подматриц среди всех подматриц, сопоставленных вершинам $(i+1)$ -го уровня.

Правило 6. Каждый столбец выбирается для выделения подматрицы при построении всех вершин $(i+1)$ -го уровня только один раз.

Правило возврата с $(i+1)$ на i -й уровень. Текущей становится вершина i -го уровня, по которой была выделена только что обработанная вершина $(i+1)$ -го уровня. В соответствующей данной вершине

подматрице U_i^T по номерам ранее удаленных равных столбцов строятся пути без выделения подматриц, и удаляется столбец (без удаления строк), по которому была построена вершина $(i+1)$ -го уровня.

Эффективное применение вышеперечисленных правил в вершинах дерева поиска обеспечивается специальной разметкой нулевых и равных столбцов, а также столбцов, по которым выделяются подматрицы.

Замечание 1. При применении правила возврата в очередной подматрице могут появиться нулевые строки, не покрываемые ни одним столбцом. Тогда все ветви, проходящие через сопоставленную данной подматрице вершину, подлежат отсечению.

Замечание 2. При рекурсивном применении правила 5 порождается ветвящийся процесс, реализация которого при глубине рекурсии больше двух требует параллельных вычислений.

Замечание 3. При ограничении глубины рекурсии для применения правила 5 числом два, становится невозможным гарантировать, что все построенные пути будут безыбыточными, поэтому необходима дополнительная процедура удаления избыточных покрытий. Однако, число избыточных покрытий, которые могут быть построены, как правило, оказывается относительно невелико.

Алгоритм поиска всех БСП состоит из следующих пунктов:

1. $P^T := \emptyset$ (P^T – текущее столбцовое покрытие), $i := 0$, $U_i^T := U$.
2. Если в U_i^T есть нулевые строки (замечание 1), то выполняется переход к п. 12.
3. Если к неотмеченным столбцам U_i^T применимо правило 1, то соответствующие столбцы отмечаются символом "-".
4. Если к неотмеченным столбцам U_i^T применимо правило 2, то в каждой группе равных столбцов левый из них отмечается символом "+", а остальные – символом "-" и их номера запоминаются.
5. Если к неотмеченным столбцам U_i^T применимо правило 3 (3'), то соответствующий столбец (столбцы) отмечается (отмечаются) символом "*", его номер (номера) включается (включаются) в P^T и выполняется переход к п. 8.
6. Если к неотмеченным столбцам U_i^T применимо правило 4, то соответствующий столбец отмечается символом "*", его номер включается в P^T и выполняется переход к п. 8.
7. К неотмеченным столбцам U_i^T применяется правило 5, соответствующий столбец отмечается символом "*" и его номер включается в P^T .
8. Если в U_i^T есть неотмеченные столбцы, то P^T запоминается и выполняется переход к п. 11.
9. P^T запоминается. Если $i = 0$, то для каждого столбца, отмеченного символом "+", по номерам равных ему столбцов строятся и запоминаются равномошные столбцовые покрытия и выполняется переход к п. 13.
10. Применяется правило возврата (при этом для каждого столбца, отмеченного символом "+", по номерам равных ему столбцов строятся равномошные столбцовые покрытия, а столбцы, отмеченные символом "*", и соответствующие им столбцы, отмеченные символом "+", переотмечаются символом "-") и $i := i - 1$. Восстанавливается P^T , соответствующее U_i^T . Если в U_i^T есть неотмеченные столбцы, то выполняется переход к п. 2, иначе, выполняется переход к п. 10.
11. Применяется правило выделения подматрицы по всем столбцам, отмеченным символом "*" с одновременным удалением столбцов, отмеченных символами "-" и "+", $i := i + 1$ и выполняется переход к п. 2.
12. Если $i > 0$, то выполняется переход к п. 10.
13. Конец алгоритма.

Заключение

Развитие алгоритма, представленного в [5], заключается в новой формулировке правил 3 (3'), 4 и 5, что позволяет при их применении полностью исключить или свести к минимуму число построенных избыточных столбцовых покрытий.

Алгоритм программно реализован на языке C++ с глубиной рекурсии для применения правила 5, равной двум, и включен в состав интеллектуального инструментального средства ИМСЛОГ в виде динамически загружаемого модуля (плагина). Апробация алгоритма на тестовых и реальных матрицах показала его более высокую эффективность в плане сокращения числа переборов и числа построенных избыточных покрытий, по сравнению с ранее разработанным алгоритмом.

В настоящее время в ЛИС ТГАСУ ведутся работы по созданию на базе ИИС ИМСЛОГ интеллектуальной информационной системы для диагностики состояний коммуникативного стресса у студентов и преподавателей (ДИАКОР-КС) с применением предлагаемого алгоритма.

Список литературы

1. Янковская А. Е. Логические тесты и средства когнитивной графики в интеллектуальной системе// Новые информационные технологии в исследовании дискретных структур. Докл. 3-ей Всеросс. конф. с междунар. уч. – Томск: Изд-во СО РАН, 2000. – С. 163-168.
2. Чегис И. А., Яблонский С. В. Логические способы контроля электрических схем// Сб. статей по математической логике и ее приложениям к некоторым вопросам кибернетики. Тр. Матем. ин-та им. В.А. Стеклова. – Т. 51. – М.: изд-во АН СССР, 1958. – С. 270-360.
3. Гуревич И. Б., Журавлев Ю. И. Минимизация булевых функций и эффективные алгоритмы распознавания// Кибернетика. – 1974. – № 3. – С. 16-20.
4. Yankovskaya A. E., Gedike A. I., Ametov R. V., Bleikher A. M. IMSLOG-2002 Software Tool for Supporting Information Technologies of Test Pattern Recognition// Pattern Recognition and Image Analysis. – 2003. – Vol. 13. – No. 4. – P. 650-657.
5. Гедике А. И., Янковская А. Е. Построение всех безыбыточных безусловных диагностических тестов в интеллектуальном инструментальном средстве ИМСЛОГ// Интеллектуальные системы (AIS'05), Интеллектуальные САПР (CAD-2005). Тр. Междунар. н.-т. конф. Том 1. – Москва: Физматлит, 2005. – С. 209-214.

Моделирование системы обеспечения информационной безопасности виртуальных интерфейсов пользователя

Гращенко Л. А.,

советник заместителя начальника отдела, в/ч 9998,

адрес: г. Орел, ул. Матвеева, д.17, кв.80,

e-mail: graschenko@mail.ru

Одним из основных направлений исследований в области обеспечения безопасности пользовательских интерфейсов (ПИ), и, в частности, визуальных интерфейсов пользователя (ВИП), является разработка моделей информационной безопасности (ИБ) ВИП [2, 6]. При этом задача обеспечения безопасности ВИП в широком смысле, адекватна задаче обеспечения ИБ ВИП, при условии комплексного учета информационных, функциональных, психофизиологических и экологических аспектов безопасности. Это связано, прежде всего, с включением информационного фактора в состав факторов среды систем человек-компьютер [6] и информационным характером почти всех происходящих в области распространения ВИП процессов, или их информационной обусловленностью.

Учет перечисленных аспектов безопасности был осуществлен на этапе моделирования множества известных или предполагаемых угроз ИБ ВИП, в результате чего была разработана обобщенная модель угроз (ОМУ) ИБ ВИП [7].

Следующим этапом исследований в области обеспечения ИБ ВИП является синтез моделей системы и подсистем обеспечения ИБ ВИП. Основанием этого этапа является анализ содержания существующих моделей защиты информации (ЗИ), отражающих различные составляющие безопасности ВИП, в том числе ОМУ ИБ ВИП, содержания понятийного базиса теории и практики информационной безопасности (ИБ) информационных систем различных классов, понятийного базиса теоретических и практических аспектов человеко-компьютерного взаимодействия (ЧКВ) и выработка системы требований к системе защиты информации [2, 6].

Проведенные исследования показали, что при решении проблемы обеспечения безопасности ВИП и взаимодействия человек-компьютер (ВЧК) в рамках системы человек-компьютер (СЧК) наибольшую разработанность получили вопросы защиты информации, циркулирующей непосредственно в технической компоненте СЧК, и наименьшую – информации, циркулирующей между компьютерной системой и пользователем, а также информации пользователя.

В ОМУ ИБ ВИП этим наименее разработанным областям проблематики защиты информации в СЧК соответствуют такие угрозы, как:

- искажение воспринимаемой пользователем информации за счет ее зашумления источниками среды на рабочем месте пользователя;
- потеря или искажение воспринимаемой пользователем информации из-за физической, семантической или синтаксической несогласованности ее представления пользователю;
- искажение представлений пользователя о реальном состоянии объекта управления за счет скрытых информационных воздействий и неадекватное принятие им решений в процессе решения задач в рамках СЧК.

Противодействие указанным угрозам непосредственно связано с учетом количественных и качественных показателей информационных потоков между компьютерной системой и пользователем, а также учетом характеристик пользователя. При этом основной характеристикой информационного потока от компьютера к пользователю можно считать информационную нагрузку на пользователя, которая во многом будет определять состояние пользователя, условия его деятельности и согласованность человеческой и технической компонент СЧК [1, 3, 4].

Общая информационная нагрузка на пользователя (IN_{SUM}), осуществляемая через пользовательский интерфейс определяется информационными нагрузками, осуществляющимися через его компоненты: визуальный (IN_{VIS}), акустический (IN_{AUD}) и кинестетический интерфейсы (IN_{KIN}):

$$\begin{cases} IN_{SUM} = f(IN_{VIS}, IN_{AUD}, IN_{KIN}), \\ IN_{VIS} \gg IN_{AUD} \gg IN_{KIN}. \end{cases} \quad (1).$$

Эмпирические соображения приводят к выводу, что для определения визуальной информационной нагрузки необходимо вычислить информативность визуализированной информации, разделить статическую и динамическую информацию, определить скорость смены динамической информации, учесть индивидуальные психофизиологические особенности пользователя, воспринимающего информацию. Таким образом, необходимо вычислить количественные характеристики визуального отображения информационной модели, с которой работает пользователь, а также определить пропускную способность визуального интерфейса, способность пользователя к обработке информации (временные параметры) по принятию им решений. Последняя характеристика определяется психофизиологическими особенностями каждого конкретного пользователя и его текущим состоянием. Таким образом, задача обеспечения безопасности ВИП подразумевает учет как объективных, так и ряда субъективных факторов.

Следовательно, применительно к рассматриваемой проблематике, средства защиты информации в рамках системы обеспечения ИБ ВИП должны обеспечивать непрерывное автоматическое вычисление и контроль визуальной информационной нагрузки, ряда объективных характеристик визуализированной информации, а также мониторинг состояния пользователя. При этом, ряд предлагаемых в известной литературе подходов к отслеживанию состояния пользователя [1, 3, 5] предполагает включение в состав технических средств СЧК дополнительных аппаратных компонент, что не всегда может быть приемлемо. Таким образом, средства ВИП должны обеспечивать сбор и анализ информации о пользователе в рамках типового набора средств ввода-вывода. При этом, контроль отдельных показателей визуального взаимодействия человека и компьютера целесообразно осуществлять отдельно реализованными подсистемами системы обеспечения ИБ ВИП. Таким образом, оправдано совмещение глобального (агентского) подхода к защите пользователя с электронным контролем деятельности пользователей [3]. Тогда система обеспечения ИБ ВИП может быть реализована в виде комплекса программных средств, объединенных интеллектуальным интерфейсом.

Помимо функции оперативного контроля совокупности характеристик, отражающих состояние ВЧК и пользователя, система обеспечения ИБ ВИП должна осуществлять их (характеристик) прогнозирование, что наиболее удобно при наличии в системе адекватной модели текущего пользователя. Такая модель может быть представлена в системе *отпечатком личности пользователя* (ОЛП) – специально структурированным файлом (базой знаний), хранящим динамически изменяющуюся информацию о всех возможных для восприятия технической компонентой СЧК атрибутах деятельности пользователя при его взаимодействии с вычислительной системой. В основе ОЛП должно лежать множество моделей отдельных сторон личности пользователя, на основе которых отдельные программы-агенты системы средств защиты информации ВИП получают возможность с высокой степенью вероятности моделировать и прогнозировать отдельные характеристики деятельности пользователя в СЧК.

В дальнейшем предлагается в рамках решения задачи синтеза модели системы обеспечения информационной безопасности визуальных интерфейсов пользователя решать задачу моделирования личности пользователя – разработки отпечатка личности пользователя.

Список литературы

1. Владимирский Б. М. Проблема человеко-компьютерного взаимодействия: психофизиологический аспект // Тезисы докладов 2-ой Международной научно-технической конференции «Интерактивные системы: Проблемы человеко-компьютерного взаимодействия», часть 2. – Ульяновск: Ульяновский государственный технический университет, 1997. – с. 69 – 70.
2. Граценко Л. А. Формирование понятийного базиса информационной безопасности визуальных интерфейсов пользователя // <http://www.ifap.ru/pi/07/sr18.doc> (01.03.2006).
3. Міщенко Н. М., Феліжанко О. Д., Щоголева Н. М. Програмні засоби захисту користувачів від шкідливого впливу комп'ютерного середовища // Математичні машини і системи. – 2003. – № 1. – с. 95-100.
4. Парфенова М. Я., Колганов С. К., Парфенов И. И., Иванов В. И. Человеческий фактор в интеллектуальных технологиях для профилактики, предупреждения и устранения аварийных ситуаций // Информационные технологии. – 2005. - №5. – с. 21-29.
5. Субботина Т. И., Новиков А. С., Царегородцев И. А., Яшин М. А., Яшин А. А. Автоматизированный анализ физиологического состояния организма // Автоматизация и современные технологии. – 2005. - №7. – с. 16-22.
6. Фисун А. П., Граценко Л. А. и др. Теоретические и практические основы человеко-компьютерного взаимодействия: базовые понятия человеко-компьютерных систем в информатике и информационной безопасности: Монография / Под ред. д.т.н. А.П.Фисуна; Орловский государственный университет. – Орел, 2004. – 169 с.: ил. – Библиогр.: 109 назв. – Рус. – Деп. в ВИНТИ 15.10.04. № 1624 – В2004.
7. Фисун А. П., Граценко Л. А. и др. Разработка модели угроз информационной безопасности визуального взаимодействия человек-компьютер для системы правовых, технических, программных и организационных методов и средств защиты информации // Информационные технологии и право: Материалы II Международной научно-практической конференции / Под ред. Ю. И. Кашинского. – Мн.: НЦПИ, 2004. – с. 221 – 223.

О вычислительной сложности двукритериальной задачи инвестора⁴

Гречкин В. А.,

*кафедра Организации и технологии защиты информации
Ставропольского государственного университета
355009, Россия, г. Ставрополь,
e-mail: i_live_in_russia@hotmail.ru*

При решении задачи обеспечения комплексной защиты информации на предприятии часто приходится сталкиваться с так называемой задачей распределения средств. Данная актуальная задача заключается в

⁴ Работа выполнена при поддержке гранта РФФИ, проект 06-01-00020а

определении оптимального порядка применения средств комплексной защиты информации в условиях ограниченных ресурсов (временных, материальных, человеческих), т.е. определение такого порядка применения средств по блокированию или ликвидации угроз, при котором обеспечивается минимально возможный ущерб, нанесенный предприятию. Математическая постановка рассматриваемой задачи – двукритериальная задача инвестора [1] определяется следующим образом. Производится оборудование предприятия средствами защиты информации. Определены n объектов защиты, пронумерованных индексом $j = 1, \dots, n$. Для каждого объекта j заданы: продолжительность ввода объекта в строй t_j , директивный срок d_j , после истечения которого, не введенный в строй объект каждую единицу времени увеличивает возможный ущерб на величину штрафа w_j . Пусть $X = \{x\}$ - множество допустимых решений (МДР) данной задачи. Всякое допустимое решение представляет собой одну из $n!$ перестановок $x = (j_1, j_2, \dots, j_n)$ чисел $1, 2, \dots, n$. На множестве X определена векторная целевая функция (ВЦФ)

$$F(x) = (F_1(x), F_2(x)), \quad (1)$$

состоящая из минимизируемых критериев, т.е. частных целевых функций:

$$F_\nu(x) \in \{\varphi_\nu(x), \bar{\varphi}_\nu(x), \psi_\nu(x), \bar{\psi}_\nu(x)\}, \quad (2)$$

$\nu = \overline{1, N}$, где каждый из критериев может иметь вид

$$\varphi_\nu(x) = \sum_{k=1}^n w_{j_k}^\nu \max(C_{j_k} - d_{j_k}^\nu, 0) = \sum_{k=1}^n w_{j_k}^\nu T_{j_k}^\nu \rightarrow \min, \quad (3)$$

где $C_{j_i} = \sum_{k=1}^i t_{j_k}$;

$$\bar{\varphi}_\nu(x) = \sum_{k=1}^n w_{j_k}^\nu \max(d_{j_k}^\nu - C_{j_k}, 0) = \sum_{k=1}^n w_{j_k}^\nu E_{j_k}^\nu \rightarrow \min, \quad (4)$$

$$\psi_\nu(x) = \max_{1 \leq k \leq n} w_{j_k}^\nu \max(C_{j_k} - d_{j_k}^\nu, 0) = \max_{1 \leq k \leq n} w_{j_k}^\nu T_{j_k}^\nu \rightarrow \min, \quad (5)$$

$$\bar{\psi}_\nu(x) = \max_{1 \leq k \leq n} w_{j_k}^\nu \max(d_{j_k}^\nu - C_{j_k}, 0) = \max_{1 \leq k \leq n} w_{j_k}^\nu E_{j_k}^\nu \rightarrow \min. \quad (6)$$

Будем называть такую задачу двукритериальной задачей распределения средств во времени или двукритериальной задачей инвестора [1]. Содержательный смысл рассмотренных критериев (3) – (6) состоит в следующем: $\varphi_\nu(x)$ - минимизация возможного суммарного ущерба, $\bar{\varphi}_\nu(x)$ - минимизация суммарного нерационального использования средств, $\psi_\nu(x)$ - минимизация возможного максимального ущерба (минимизация наихудшего исхода), $\bar{\psi}_\nu(x)$ - минимизация максимального нерационального использования средств. Заметим, что данные критерии наиболее точно отражают предпочтения инвестора (начальник службы безопасности, руководитель предприятия и т.д.), т.к. с точки зрения теории управления, наилучшим моментом использования ресурса является момент, когда он действительно востребован.

При решении задач с ВЦФ нескольких критериев поиск оптимума наталкивается на проблему выбора наиболее подходящего решения из множества несравнимых альтернатив, при этом искомым решением является множество Парето оптимальных решений [3]. Существуют следующие подходы к решению задач данного типа [4], [5]: лексикографическая оптимизация, одновременная оптимизация критериев.

При лексикографической оптимизации необходимо определить, какой из критериев $F_1(x)$ или $F_2(x)$ важнее (доминирующий) [4]. Далее необходимо минимизировать значение наиболее важного критерия, минимизацию второго критерия производится в соответствии с уже минимизированным первым критерием. Применение лексикографической оптимизации невозможно при отсутствии доминирующего критерия. В

таком случае применяют одновременную оптимизацию критериев, которая включает в себя линейную свертку критериев (ЛСК) и апостериорную оптимизацию. В случае ЛСК, оба критерия объединяют в одну целевую функцию F , после чего определяется оптимальное решение для данной задачи по F . При апостериорной оптимизации производится поиск всех Парето оптимальных решений, после чего они предъявляются ЛПР.

В результате проведенного анализа [5], [6], [7], [8] были составлены следующие таблицы сложности рассматриваемой задачи.

Таблица 1: Сложность двукритериальной зад. инвестора при лексикографической оптимизации

$F_1(x)$	$F_2(x)$				
	$\max_{1 \leq k \leq n} T_{jk}^v$	$\max_{1 \leq k \leq n} E_{jk}^v$	$\max_{1 \leq k \leq n} g_{jk}^v$	$\sum_{k=1}^n w_{jk}^v E_{jk}^v$	$\sum_{i=1}^n w_{ji}^v T_{jk}^v$
$\max_{1 \leq k \leq n} T_{jk}^v$	$O(n \log n)$	$O(n^2)$	$O(n^2)$!!	!!
$\max_{1 \leq k \leq n} E_{jk}^v$	$O(n \log n)$	$O(n^2)$	$O(n^2)$!!	!!
$\max_{1 \leq k \leq n} g_{jk}^v$	$O(n^2)$	$O(n \log n)$!!	!!	!!
$\sum_{k=1}^n w_{jk}^v E_{jk}^v$!!	!!	!!	!!	!!
$\sum_{i=1}^n w_{ji}^v T_{jk}^v$!!	!!	!!	!!	!!

Таблица 2: Сложность двукритериальной задачи инвестора при оптимизации в общем случае

$F_1(x)$	$F_2(x)$				
	$\max_{1 \leq k \leq n} T_{jk}^v$	$\max_{1 \leq k \leq n} E_{jk}^v$	$\max_{1 \leq k \leq n} g_{jk}^v$	$\sum_{k=1}^n w_{jk}^v E_{jk}^v$	$\sum_{i=1}^n w_{ji}^v T_{jk}^v$
$\max_{1 \leq k \leq n} T_{jk}^v$	$O(n^3 \log n)$!!	$O(n^3 \log n)$!!	!!
$\max_{1 \leq k \leq n} E_{jk}^v$!!	$O(n^3 \log n)$!!	!!	!!
$\max_{1 \leq k \leq n} g_{jk}^v$	$O(n^3 \log n)$!!	$O(n^4)$!!	!!
$\sum_{k=1}^n w_{jk}^v E_{jk}^v$!!	!!	!!	!!	!!
$\sum_{i=1}^n w_{ji}^v T_{jk}^v$!!	!!	!!	!!	!!

В таблицах условными знаками обозначены: «!!» - NP - трудные задачи в сильном смысле, «!» - NP - трудные задачи в обычном смысле. Если задача не является NP - трудной, то для нее указана сложность

алгоритма. В таблицах также представлены некоторые данные о сложности рассматриваемой задачи в случае, когда директивные сроки $d_{j_k} = 0$.

Как видно из представленных таблиц сложности, большинство постановок двукритериальной задачи инвестора являются *NP* - трудными задачами. Для *NP* - трудных задач не существует точных полиномиальных алгоритмов, однако решение таких задач возможно [9] путем применения псевдополиномиальных алгоритмов, алгоритмов полиномиальной трудоемкости нахождения приближенного решения без каких-либо гарантий качества получаемых решений, алгоритмов полиномиальной трудоемкости для нахождения приближенного решения с гарантированными оценками качества.

Список литературы

1. Перепелица В. А., Попова Е. В., Семенчин Е. А. Теория игр и исследование операций. – Ставрополь: Изд-во СГУ, 2004. – 182 с.
2. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416 с.
3. Ногин В. Д. Принятие решений в многокритериальной среде: количественный подход. – М: ФИЗМАТЛИТ, 2005. – 176 с.
4. Подиновский В. В., Ногин В. Д. Парето - оптимальные решения многокритериальных задач. – М.: Наука, 1982. – 256 с.
5. Hoogeveen H. Multicriteria scheduling // European Journal of Operational Research 167 (2005)
6. Utku D.H. Complexity of single machine hierarchical scheduling
7. В. А. Перепелица, Ф. Б. Тебуева, В. А. Гречкин Задача инвестора с интервальными данными // Вестник Ставропольского государственного университета 43 (2005)
8. Hoogeveen, J. A., 1992. Single-Machine Bicriteria Scheduling. PhD Thesis, CWI, The Netherlands Technology, Amsterdam.
9. Chen B., Potts C. N., Woeginger G. J. A Review of machine scheduling- Complexity, algorithms and approximability // Handbook of Combinatorial Optimization. Boston: Kluwer Acad. Publ. V.4, 1998

Система автоматического оценивания пользовательского интерфейса, управляемая базой знаний

Грибова В. В.,

Институт автоматизации и процессов управления ДВО РАН

Введение.

В настоящее время перед разработчиками программного обеспечения стоят две основные задачи: уменьшить стоимость разработки программного обеспечения и повысить его качество. Эти задачи, как правило, противоречат друг другу, поскольку разработка качественного программного обеспечения требует значительных затрат, в результате чего возрастает стоимость программного продукта. Поэтому актуальной является разработка таких методов, позволяющих, во-первых, свести ошибки разработчика к минимуму, во-вторых, наиболее более полно проверить программное средство до его поставки заказчику. Пользовательский интерфейс является неотъемлемым компонентом большинства программных средств, поэтому качество его разработки имеет большое значение. Помимо общих критериев качества пользовательский интерфейс имеет дополнительный критерий качества – юзабилити. Юзабилити – это критерий качества, который оценивает, насколько легко интерфейс в использовании. Данный атрибут имеет пять основных составляющих: простота обучения, эффективность работы, субъективное удовлетворение, запоминаемость (легкость возвращения к работе после перерыва), количество совершаемых ошибок[1].

Оценивание юзабилити является дорогой задачей в терминах времени и человеческих ресурсов. Одним из решений проблемы является либо увеличение числа команд для тестирования этого атрибута качества, либо автоматизация (насколько это возможно) данного процесса. Программные продукты, как отмечается в [ii], приобретают высокое качество не столько в результате комплексного тестирования конечного продукта, сколько в процессе его разработки. Целью данной работы является описание системы автоматического обнаружения дефектов юзабилити в пользовательском интерфейсе, управляемых базой знаний.

Инструментальное средство для проектирования и автоматической генерации пользовательского интерфейса.

В настоящее время для проектирования и реализации пользовательских интерфейсов используются специализированные средства: Построители интерфейсов и Моделеориентированные средства [iii]. Однако, они имеют следующие основные недостатки: в них отсутствуют средства для расширения инструментария; они поддерживают либо разработку только некоторых компонентов пользовательского интерфейса, либо требуют от разработчика изучения нескольких специальных языков для описания компонентов интерфейса, что снижает интеллектуальную поддержку разработчика, усложняет процесс разработки и увеличивает его время [iv].

Для устранения недостатков существующих подходов предложен онтологоориентированный подход к разработке интерфейса [v, vi]. Основными положениями концепции разработки интерфейса на основе онтологий являются: раздельное проектирование интерфейса и прикладной программы; объединение однородной по содержанию информации в компоненты модели интерфейса; формирование компонентов модели интерфейса на основе проблемно-независимых моделей онтологий, отражающих специфику каждого его компонента; автоматическая генерация пользовательского интерфейса по его модели.

Модель пользовательского интерфейса – это декларативное описание, по которому автоматически генерируется код пользовательского интерфейса. Модель интерфейса состоит из четырех основных компонентов:

- модели предметной области, которой описываются входные и выходные данные, интеллектуальная поддержка пользователя в процессе его взаимодействия с программным средством;
- модели представления - визуальной составляющей интерфейса, обеспечивающей поддержку диалога, основанного на экранных формах;
- модели связи с прикладной программой, определяющей переменные, типы их значений, общие для интерфейса и прикладной программы, а также протоколы, с помощью которых происходит коммуникация;
- модель сценария диалога, определяющая абстрактные термины для описания реакций на события.

Компонентом модели интерфейса, «отвечающим» за юзабилити, является модель представления, которая формируется на основе модели онтологии графического пользовательского интерфейса (ОГПИ). ОГПИ описывает все знания, связанные с представлением информации в интерфейсе на основе экранных форм. Элементы онтологии состоят из объектов, принадлежащих различным классам (кнопки, списки, меню, окна и т.д.). Классы могут находиться между собой в отношении обобщения (наследования) и агрегации (целое-часть).

Система автоматического обнаружения дефектов юзабилити.

При проектировании конкретного пользовательского интерфейса формируется его модель, одним из компонентов которой является модель представления. Она содержит описание подмножества интерфейсных элементов с заданным набором их свойств. На данном этапе проектирования разработчик может нарушить принципы юзабилити, что приводит к ошибкам проектирования (дефектам). Для обнаружения дефектов с каждым элементом ОГПИ (кнопкой, меню, списком, полем ввода и др.) связывается множество возможных нарушений при его проектировании, а также множество общих нарушений расположения интерфейсных элементов в окне (нарушение симметрии, баланса, сцепления, пропорциональности и др.). Если эти правила нарушаются в процессе проектирования модели представления, средство обнаружения дефектов информирует разработчика о дефекте и предлагает методы его устранения. Таким образом, средство автоматического обнаружения дефектов управляется базой знаний о дефектах, которая связывается с компонентами ОГПИ.

База знаний о дефектах формируется на основе онтологии дефектов, которая имеет следующую структуру.

1. Название дефекта (симптом, что наблюдается).

2. Тип дефекта: дефект элемента представления или дефект композиции. Если дефект относится к проектированию интерфейсного элемента, то такой дефект называется дефектом представления. Если дефект

связан с ошибкой расположения интерфейсных элементов в окне, то такой дефект называется дефектом композиции.

3. Имя класса(ов). Имя класса – метатермин ОГПИ. Обозначает имя класса, к которому относится интерфейсный элемент, дефект которого описывается. Данное поле может содержать несколько классов. С одной стороны, один элемент может характеризоваться разными классами, с другой, при описании дефекта композиции описываются все классы, которые участвуют в обнаружении дефекта.

4. Суперкласс - имя класса-родителя. Суперкласс - также метатермин ОГПИ.

5. Параметр(ы) - параметры, которые обнаруживают дефект. Названия параметров совпадают с соответствующими названиями параметров класса интерфейсного элемента из ОГПИ.

6. Метод обнаружения - вычисление дефекта.

7. Критика, совет - сообщение, выдаваемое пользователю.

База знаний о дефектах интерфейса содержит более 100 дефектов. Приведем пример одного из дефектов интерфейса.

Название дефекта: большое количество элементов главного меню

Тип дефекта: дефект элемента представления

Имя класса: элемент меню

Суперкласс: меню

Параметр(ы): уровень вложенности; индекс: (тип: целое)

Метод обнаружения: (количество(элемент меню типа: индекс) > 9) ^ (уровень вложенности = 1)

Сообщение. В меню больше 9 элементов. Такое количество элементов будет плохо запоминаться пользователем.

В настоящее время инструментальное средство реализовано и используется для проектирования, автоматического обнаружения дефектов и автоматической генерации пользовательских интерфейсов программных средств в различных предметных областях.

Список литературы.

- ⁱ Ivory, M. Y., Hearst, M. A.: State of the Art in Automating Usability Evaluation of User Interfaces// ACM Computing Surveys, 33 (December 2001). p.1–47. (Accessible at <http://webtango.berkeley.edu/papers/ue-survey/ue-survey.pdf>)
- ⁱⁱ Коммервил И. Инженерия программного обеспечения: Пер. с англ.-М.:Издательский дом «Вильямс», 2002.-624с.:с ил.
- ⁱⁱⁱ Szekely P., Sukaviriya P., Castells P., Muthukumarasamy J., Salcher E. Declarative Interface Models for User Interface Construction Tools: the Mastermind Approach. In Engineering for Human-Computer Interaction, L. Bass and C. Unger Eds. Chapman & Hall, 1996. <http://www.isi.edu/isd/Mastermind/mastermind-ia.htm>
- ^{iv} Puerta A. R. Supporting User-Centred Design of Adaptive User Interfaces via Interface Models. First Annual Workshop On Real-Time Intelligent User Interfaces for Decision Support and Information Visualization, San-Francisco, January 1998.
- ^v Gribova V., Kleshchev A. An ontology-oriented conception to user interface development// International Journal Information theories & applications. 2003. vol. 10, num.1, p. 87-94.
- ^{vi} В. В. Грибова, А. С. Клещев Использование методов искусственного интеллекта для проектирования пользовательского интерфейса// Информационные технологии №8.2005. с.58-62

Использование агрегирования в методах нелинейной динамики для анализа и прогнозирования временных рядов

Джашеева Ф. М.,

*ассистент каф. ПМ КЧГТА, аспирант;
369103 КЧР, Прикубанский р-н, с. Дружба, ул. Комсомольская, 217*

Мохамад-Богашева З. А.,

*ассистент каф. ПМ КЧГТА, аспирант;
369003, КЧР, г. Черкесск, ул. Октябрьская, №34, кв. 60
dzhashееva@rambler.ru*

В настоящей работе освещаются (рассматриваются) вопросы исследования потенциальной прогнозируемости временных рядов (ВР) на базе новых инструментариев нелинейной динамики, в частности, фрактального анализа и фазовых траекторий, их адаптации для получения предпрогнозных характеристик, выбора подходящего принципа агрегирования («декомпозиция») и его применения для улучшения предпрогнозных характеристик агрегированных ВР.

Объектом исследования является оптовая база. Исходные данные представляют подневные объёмы продаж за период 25.07.2005г. – 05.02.2006г. Фрактальный анализ представленного ВР $Z = \langle z_j \rangle, j = \overline{1, n}$ осуществлён путём использования метода последовательного R/S -анализа, вычислительная схема которого хорошо описана в [1]. В результате получено следующее нечёткое множество глубины памяти этого ВР: $N(Z) = \{(l; \mu(l))\} = \{(3; 0,9), (4; 0,66), (5; 0,55), (6; 0,35), (7; 0,29), (8; 0,16), (9; 0,05), (10; 0,04)\}$, которое гласит о весьма низкой трендоустойчивости исходного ВР в силу того, что его функция принадлежности $\mu(l)$ принимает максимальное значение 0,9 для минимально возможной глубины памяти $l = 3$.

С целью получения дополнительной прогнозной информации о динамике исходного ВР использован фазовый анализ [1], состоящий в построении фазовой траектории этого ВР с последующим её разложением на квазициклы. Построение фазовой траектории осуществлено в фазовом пространстве размерности $\rho = 2$, т.е. для ВР $Z = \langle z_j \rangle, j = \overline{1, n}$ фазовая траектория определена в декартовых координатах множеством точек $\Phi_2(Z) = \{(z_j, z_{j+1})\}, j = \overline{1, n-1}$. Далее для неё осуществлено разложение на квазициклы.

На рис.1 и рис.2 представлены соответственно фазовая траектория исходного ВР и типичные квазициклы, составляющие большинство в указанном разложении. Характерной особенностью этих квазициклов является то, что при малой их длине они содержат такие пары соседних звеньев, которые имеют противоположное направление вращения (против часовой стрелки). Вторая особенность рассматриваемой фазовой траектории состоит в том, что она содержит такие достаточно продолжительные отрезки, в которых отсутствует цикличность. Из анализа полученных результатов вытекает, что предпрогнозные характеристики исходного ВР весьма неудовлетворительны.

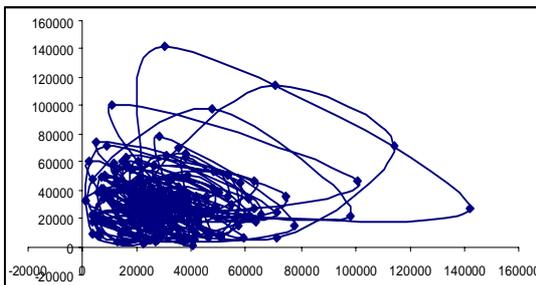
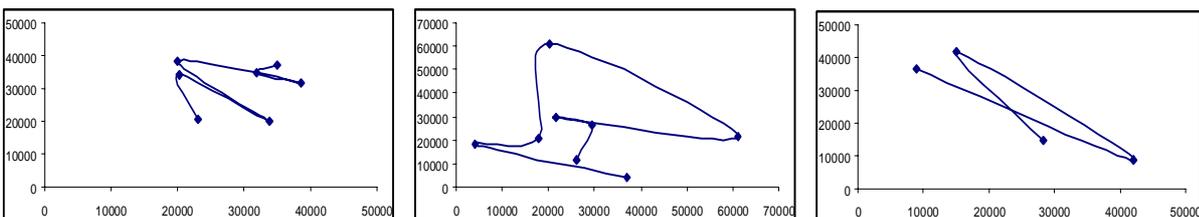


Рисунок 1. Фазовая траектория исходного временного ряда Z

Рисунок 2. Типичные квазициклы, полученные в результате разложения фазовой траектории исходного временного ряда Z



В качестве основной идеи улучшения предпрогнозных характеристик рассматриваемого ВР предлагается и реализуется метод агрегирования этого ВР, причём класс известных подходов к агрегированию (сложение показателей, представление группы агрегируемых показателей через их среднюю, взятие максимального элемента в каждом интервале агрегирования, использование различных взвешивающих коэффициентов, баллов и т.д.) авторы пополняют новым методом декомпозиции, который реализуется следующим образом.

Пусть M – множество элементов декомпозиции периодического ВР $Z = \langle z_j \rangle, j = \overline{1, n}$ с длиной периода в 7 дней (дни недели, циклически следующие друг за другом); ρ – ранг (длина) ρ -элемента $L_k^\rho = (k, k+1, \dots, k+\rho-1)_{\text{mod } 7}, k = \overline{1, 7}, \rho = \overline{1, 7}, k$ – номер элемента в упорядоченном множестве M_ρ (для каждого $\rho = \overline{1, 6}$ имеем $k = \overline{1, 7}$; для $\rho = 7$ имеем $k = 1$); $M_\rho = \{L_k^\rho\}$ – подмножество множества всех ρ -элементов:

$$M_1 = \{1, 2, \dots, 7\}, \quad M_\rho = \{(1, 2, \dots, \rho), (2, 3, \dots, \rho+1), \dots, (7, 1, 2, \dots, \rho-1)\}, \quad \rho = \overline{2, 6},$$

$$M_7 = \{(1, 2, \dots, 7)\}, \quad |L_k^\rho| = k, \quad k = \overline{1, 7}, \quad \text{количество элементов } |M_\rho| = 7, \quad \rho = \overline{1, 6}, \quad |M_7| = 1.$$

В допустимом решении $x = \{x_{\rho k}, x_{\rho k}^+\}$ ($k = \overline{1, 7}$ для $\rho = \overline{1, 6}$ и $k = 1$ для $\rho = 7$, $x_{\rho k}, x_{\rho k}^+ \in \{0, 1\}$) для всех пар $x_{\rho_1 k_1}, x_{\rho_2 k_2}$ таких, что при $x_{\rho_1 k_1} = x_{\rho_2 k_2} = 1$ пересечение элементов $L_{k_1}^{\rho_1}, L_{k_2}^{\rho_2}, 1 \leq k_1 < k_2 \leq 7, \rho_1, \rho_2 = \overline{1, 7}$ должно быть пусто ($L_{k_1}^{\rho_1} \cap L_{k_2}^{\rho_2} = \emptyset$), а объединение множества номеров $\{L_k^\rho\}$, составляющих элементы L_k^ρ , должно представляться равенством: $\bigcup_{x_{\rho k} \in x} \{L_k^\rho\} = \{1, 2, \dots, 7\}$. $X = \{x\}$ – множество всех допустимых решений.

Допустимое решение $x = \{x_{\rho k}, x_{\rho k}^+\}$ определяет собой множество агрегированных ВР $Z(x_{\rho k}), Z(x_{\rho k}^+)$ для единичных переменных ($x_{\rho k} = 1, x_{\rho k}^+ = 1$). ВР $Z(x_{\rho k}) = \langle z_j^{\rho k} \rangle, j = \overline{1, n_{\rho k}}$ состоит из следующих друг за другом отрезков исходного ВР Z ; в каждом таком отрезке уровни $z_j^{\rho k}$ соответствуют номерам, составляющим ρ -элемент L_k^ρ , для которого $x_{\rho k} = 1$. Например, для случая $x_{32} = 1$ в исходном ВР Z такими отрезками ВР $Z(x_{32})$ являются 3-элементы: $(z_2, z_3, z_4), (z_9, z_{10}, z_{11}), (z_{16}, z_{17}, z_{18})$ и т.д. В случае $x_{32} = 0$ и $x_{32}^+ = 1$ ВР $Z(x_{32}^+)$ состоит из уровней $z_s^{32+}, s = 1, 2, \dots$, где z_s^{32+} равен сумме элементов z_j из s -го отрезка исходного ВР Z .

Результаты применения метода последовательного R/S -анализа и фазового анализа к полученным таким образом агрегированным ВР оказались неоднозначными. Для одной группы агрегированных ВР (например, $Z(x_{12}), Z(x_{14}), Z(x_{16}), Z(x_{25}), Z(x_{41}), Z(x_{26}^+), Z(x_{41}^+)$) предпрогнозные характеристики остались столь же неудовлетворительными, как и соответствующие характеристики исходного ВР. Для другой группы (например, $Z(x_{13}), Z(x_{22}), Z(x_{23}), Z(x_{27}), Z(x_{32}), Z(x_{21}^+), Z(x_{22}^+), Z(x_{23}^+), Z(x_{24}^+), Z(x_{27}^+), Z(x_{25}^+), Z(x_{31}^+), Z(x_{33}^+), Z(x_{71}^+)$) выяснилось, что агрегированные ВР в сравнении с исходным ВР улучшили свои предпрогнозные свойства. В качестве примера, показывающего типичную картину для агрегированных ВР, на рис.3 и рис.4 представлены соответственно фазовая траектория агрегированного ВР $Z(x_{12})$ и типичные квазициклы, полученные в результате разложения этой траектории и составляющие большинство в указанном разложении. Нечёткое множество глубины памяти этого ВР имеет вид:

$$M(Z(x_{12})) = \{(t, \mu(t))\} = \{(3, 0, 9), (4, 0, 6), (5, 0, 4), (6, 0, 6), (7, 0), (8, 0, 1), (9, 0), (10, 0), (11, 0), (12, 0), (13, 0, 1)\}.$$

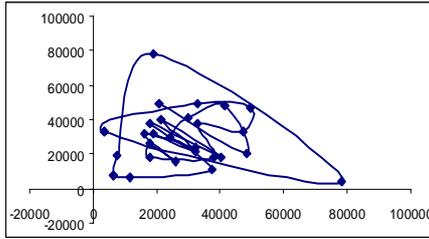


Рисунок 3. Фазовая траектория агрегированного ВР $Z(x_{12})$

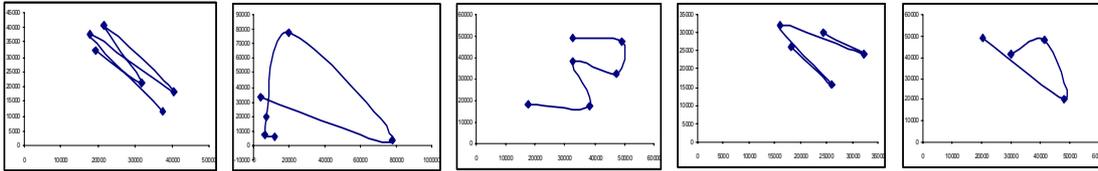


Рисунок 4. Типичные квазициклы, полученные в результате разложения фазовой траектории агрегированного ВР $Z(x_{12})$

Во множестве всех допустимых решений $X = \{x\}$ оптимальным по критерию глубины памяти является решение $x^0 = (x_{27}^+, x_{32}^+, x_{25}^+)$, состоящее из двух 2-элементов

$$L_7^2 = (7,1), L_5^2 = (5,6) \text{ и одного 3-элемента } L_2^3 = (2,3,4).$$

Для элемента $L_2^3 = (2,3,4)$ с переменной x_{32} нечёткое множество глубины памяти агрегированного ВР $Z(x_{32})$ имеет вид:

$$N(Z(x_{32})) = \{t; \mu(t)\} = \{(3; 0,56), (4; 0,41), (5; 0,86), (6; 0,46), (7; 0,41), (8; 0,41), (9; 0,25), (10; 0,05), (11; 0,10), (12; 0,25), (13; 0,05), (14; 0,15), (15; 0,05)\}.$$

Соответствующая ему фазовая траектория и типичные квазициклы, полученные в результате её разложения представлены на рис.5 и рис.6.

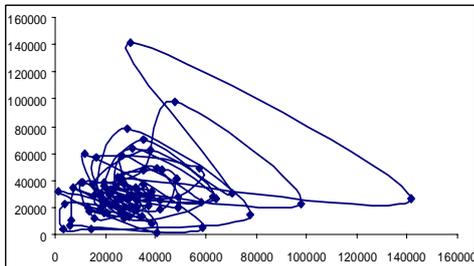


Рисунок 5. Фазовая траектория агрегированного ВР $Z(x_{32})$

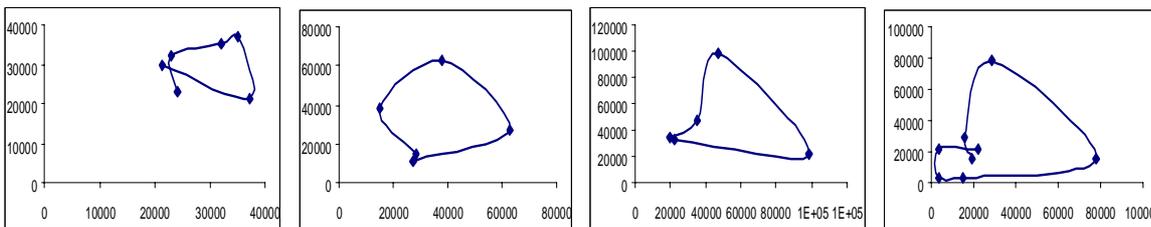


Рисунок 6. Типичные квазициклы, полученные в результате разложения фазовой траектории агрегированного ВР $Z(x_{32})$

Представленные результаты исследования дают основание для вывода о перспективности применения предложенного метода декомпозиции к ВР с целью улучшения их предпрогнозных характеристик.

Список литературы

1. Перепелица В. А., Тебуева Ф. Б., Темирова Л. Г. Структурирование данных методами нелинейной динамики для двухуровневого моделирования. – Ставрополь: Ставропольское книжное издательство, 2006. – 284 с.

Аппаратная поддержка процессором выполнения шифрованного кода

Дорошенко Андрей Николаевич

*Киевский Национальный Университет имени Т.Г. Шевченко,
Украина, 41100, г. Шостка, ул. Красная 1, кв. 26*

Украина, 03191, г. Киев, ул. Василя Касияна 6, кв. 144

e-mail: andrewd@bigmir.net

adoroshenko@svitonline.com

тел.: +38 (044) 201-72-14, +38 (068) 201-72-14.

Научный руководитель Анисимов Анатолий Васильевич

Введение

Одна из актуальных проблем, которая стоит перед разработчиками программного обеспечения (ПО) – как защитить свои программные продукты от нелегального использования. На сегодняшний день не существует такой эффективной защиты, которая гарантировала бы создателю ПО 100% защиту от пиратства.

Разработчики стараются всеми возможными методами затруднить анализ защиты, используя чаще всего или в комбинации с другими – антиотладочные методы и шифрование критических участков кода. Взломщику приходится иметь дело с шифром, шифрограммой и ключом, снимать дампы памяти, значительно модифицировать программный код. Но сам факт того, что ключ должен поставляться с программой приводит к тому, что он становится известным и взломщику.

Цель - «научить» процессор выполнять зашифрованные участки кода без предварительной расшифровки в память, не передавая ключ с программой, что гарантировало невозможность дизассемблирования, отладки и взлома (все зависело бы от криптостойкости шифрования).

Основные утверждения, тезисы:

1. В микропроцессор добавляются (рассматривается архитектура IA-32) регистры для хранения закрытого ключа для RSA:

D – 1024 бит, N – 1024 бит (модуль)

2. Аппаратно реализовываются алгоритмы RSA и ГОСТ.

3. Ключ KEY – состоящий из K1,...K32 - 32 бит x 32 расшифровывается алгоритмом RSA.

4. KEY – необходим для декодирования участков кода непосредственно при выполнении зашифрованной инструкции.

5. E, N – открытые ключи для шифрования KEY. Используются разработчиком ПО при компиляции.

6. Ключи D, N – одинаковые для всех процессоров.

7. tD, tN, tE, tP – уникальные для конкретного экземпляра. tD – закрытый недоступный для чтения и записи ключ; tN, tE – доступны для чтения, где tE – открытый ключ. tP – цифровая подпись для пары (tN, tE).

8. Инструкция, реализованная в микропроцессоре

DecryptKey { Key1 }, Key1 -1024 бит. Результат KEY (K1,...K32 - 32 бит x 32) – расшифровывает ключ для алгоритма потокового шифрования ГОСТ.

9. Инструкция

DecryptAndRun n, шифротекст - где N – длина шифротекста – выполняет зашифрованные участки кода. Используется алгоритм ГОСТ.

10. Инструкция tDecryptKey – аналогично инструкции DecryptKey – только используются ключи tD, tN.

11. CMF – регистр контекста 64 бит длина. Разрешается только запись, чтение – недопустимо.

12. Инструкции для работы с контекстом

- CLEAR_CMF
- MOV CMF, XXX
- XOR CMF, XXX

13. «Контекст» позволяет *замаскировать* любую инструкцию, например, MOV AX, BX на похожую XOR AX, BX, если установлен 0-бит флага CMF или, другой пример,

ADD BX, CX на

SUB BX, CX, если установлен 1-ий бит или на

SUB DX, CX, если установлен 1-ий и 2-ий бит, или

ADD DX, CX, если 1-ий не установлен, а 2-ой установлен.

Т.е. реализовать динамическое изменение системы команд в зависимости от контекста, задаваемого регистром контекста.

Заключение

Разработан алгоритм активации, использующий предлагаемые нововведения. Мы достигли цели, результатом усилий является принципиально новый микропроцессор, который умеет выполнять как обычный код, так и зашифрованные инструкции.

Преимущества – невозможность дизассемблировать и анализировать зашифрованные участки кода даже активированного ПО. Без одноразовой активации процессор не сможет аппаратно расшифровывать на лету и выполнять зашифрованные инструкции. Т.к. именно в момент активации он узнает ключ для выполнения зашифрованного участка. Только процессор может вычислить ключ дешифрования по возвращаемому ответному коду активации.

Список литературы

1. Handbook of Applied Cryptography. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. CRC Press. ISBN: 0-8493-8523-7. October 1996, 816 pages
2. А. В. Чернов. Анализ запутывающих преобразований программ //В сб. "Труды Института системного программирования", под. ред. В. П. Иванникова. М.: ИСП РАН, 2002.
3. Анісімов А. В, Алгоритмічна теорія великих чисел.
Модулярна арифметика великих чисел. Київ: Видавничий дім "Академперіодика", 2-001.- 153 с.
4. Скляров Д. В. Искусство защиты и взлома информации. – СПб.: БХВ-Петербург, 2004. – 288 с.: ил. ISBN 5-94157-331-6.
5. Устройство процессоров Intel архитектуры NetBurst. / Сергей Озеров. / 01.10.2005 <http://www.terralab.ru/system/236634/>

Разработка многофункциональной автоматизированной системы распознавания графических изображений

Дурденко В. А.,

*д.т.н., профессор Воронежского института
экономики и социального управления.*

Ильичев М. А.,

*к.т.н., доцент Воронежского института МВД России.
394049, г. Воронеж, ул. Лидии Рябцевой, д. 45 А, кв. 15.
e-mail: maip@rambler.ru*

Анализируя сложившиеся к настоящему времени подходы к решению задачи распознавания графических изображений, можно отметить, что вне зависимости от предметной области разработчики систем распознавания пользуются методами, которые в общем случае предполагают следующие этапы решения указанной задачи.

1. Проведение всестороннего анализа объекта исследования с целью получения максимально полных знаний о его характеристиках и свойствах для определения полного перечня признаков, характеризующих объект исследования.

Обоснование алфавита признаков является одним из важнейших этапов создания системы распознавания. Его важность обусловлена прежде всего тем, что работа блока принятия решений о принадлежности объекта определенному классу практически целиком основывается на анализе получаемых значений признаков. Ошибки, допущенные при обосновании алфавита признаков, на последующих этапах создания системы не только не исправляются, но даже не обнаруживаются, что может приводить к существенному снижению потенциальной эффективности системы распознавания (СР).

При обосновании алфавита признаков последовательно формируются априорный, а затем рабочий алфавит признаков. Основным требованием при формировании априорного алфавита является

обеспечение полноты рассматриваемых признаков, в силу чего априорный алфавит должен включать в себя максимально возможное количество признаков, безотносительно к их природе и без учета ограничений на возможности их измерения и формализации.

Формирование рабочего алфавита признаков основывается на снижении размерности априорного алфавита за счет исключения малоинформативных признаков и признаков, не удовлетворяющих ограничениям (стоимостным, точностным, временным), накладываемым на создание систем распознавания [1]. Получаемыми признаками будут являться вполне определенные формализованные свойства изучаемых объектов.

2. Решение задачи распознавания необходимо в общем случае для того, чтобы система управления, стоящая над системой распознавания, могла принимать правильные решения. Являясь, таким образом, частью системы управления, системы распознавания должны строиться с учетом обеспечения наиболее эффективного использования всего набора допустимых решений. Исходя из этого, следующей задачей построения СР является проведение классификации распознаваемых объектов в соответствии с задачами, решаемыми системой управления.

3. Выбор алгоритмов распознавания, обеспечивающих отнесение распознаваемого объекта к тому или другому классу или их совокупности. Этап реализуется на основе информации о типах признаков распознаваемых объектов, их количестве, особенностях измерения и т.п.

4. Разработка специальных алгоритмов управления работой системы с тем, чтобы процесс функционирования СР был в определенном смысле оптимальным и выбранный критерий качества данного процесса достигал экстремального значения. Достижение экстремальной величины критерия качества функционирования должно сопровождаться соблюдением некоторых ограничивающих условий (время, расходы, итерации и т.д.).

Рассмотренный традиционный подход к решению проблемы создания систем распознавания наряду со своей явной очевидностью обладает и рядом весомых недостатков, которые проявляются не только при разработке, но и на этапе функционирования СР. Рассмотрим основные недостатки.

1. Отсутствие формальной процедуры обоснования априорного словаря признаков.

Выбор признаков производится разработчиком на основании личного опыта и интуиции, что может привести к пропуску ряда важных признаков исследуемых объектов.

Таким образом, процедура создания алфавита признаков являющаяся одним из важнейших этапов создания СР, обладает существенным недостатком - субъективностью.

2. Жесткая привязка к объекту распознавания в рамках решаемой задачи. Каждая предметная область требует разработки и применения сложных алгоритмов анализа изображений для расчета признаков. Полученные результаты для одной предметной области фактически не применимы для других предметных областей. Это связано с тем, что методика преобразования входной информации, особенности создания признаков, их физические особенности, а также алгоритмы принятия решений являются, как правило, уникальными не только для каждой из предметных областей, но даже для задач, решаемых внутри одной предметной области. Естественно при этом, что каждая предметная область может потребовать полной или частичной переработки существующего алфавита признаков для достижения необходимой эффективности распознавания, а это фактически равносильно разработке новой системы.

Исходя из этого, представляется целесообразным совершенствование существующего подхода к проблеме распознавания образов за счет автоматизации решения задач создания систем распознавания, а также применения новых автоматизированных алгоритмов функционирования, адаптивных по отношению к объектам различных предметных областей и возникающим практическим потребностям. Решение указанной задачи позволит существенно сэкономить выделяемые материальные ресурсы, повысить скорость создания систем, а также эффективность их функционирования.

Результатом проводимых нами исследований является технология создания универсальной автоматизированной системы распознавания образов, способной функционировать в различных предметных областях. При этом подразумевается, что система может воспринимать только определенный тип исходных данных – графические изображения.

Как уже было отмечено, обеспечение полноты априорного алфавита является одним из важных требований, выполнения которого добиваются поиском максимально возможного количества признаков, исходя из возможностей разработчика. В виду того, что данное требование неформально, пропуск ряда признаков может привести к снижению потенциальной эффективности системы распознавания. Кроме того, универсальность системы, в смысле возможности работы с различными предметными областями в рамках выбранного способа получения исходных данных, может быть достигнута только благодаря наличию множества алгоритмов вычисления признаков. Однако, хранение в памяти всего множества алгоритмов и их применение для каждой предметной области представляется весьма затруднительным.

Исходя из этого, формирование алфавита признаков должно осуществляться автоматически с использованием некоторой формальной процедуры. Под формальной процедурой подразумевается алгоритм, позволяющий на основе имеющегося набора базовых правил строить различные комбинации описаний объектов.

Автоматический поиск признаков предполагает, что в рамках заложенного алгоритма будут выделены либо все признаки, удовлетворяющие заданному критерию полезности, либо их определенное количество, достаточное для решения поставленной задачи. Только в этом случае можно будет утверждать, что система использует в определенном смысле полный алфавит признаков и соответственно добивается оптимального (с точки зрения выбранной процедуры) уровня эффективности.

Таким образом, необходимость обеспечения возможности работы системы распознавания образов в различных предметных областях и с различными задачами этих областей вызывает потребность разработки процедуры автоматического формирования множества признаков, полезных для данной конкретной задачи.

Это, в свою очередь, позволит избавиться от ряда указанных выше недостатков СР и решать не только прямую задачу – распознавание объекта предметной области с максимальной точностью при заданных ресурсах, но и обратную – достижение определенного уровня точности при распознавании в пределах заданных ресурсов или исчерпание задаваемых ресурсов с достижением какого-либо уровня точности.

Указанная процедура должна обеспечивать возможность построения в автоматическом режиме множеств полезных признаков применительно к различным предметным областям без участия оператора, функция которого в данном случае сводится к предоставлению обучающей выборки.

Естественным условием функционирования процедуры является возможность генерации большого количества достаточно простых в вычислительном плане алгоритмов исходного преобразования поступающих изображений и ряда операторов, приводящих результаты вычислений к единому виду. Каждый признак при этом будет являться определенной комбинацией полученных результатов вычислений.

В настоящее время многофункциональная автоматизированная система распознавания графических изображений реализована нами в макетном варианте с использованием пакета Matlab. Выбор пакета обусловлен возможностью достаточно оперативно создавать сложные процедуры анализа графических изображений за счет наличия большого количества встроенных функций, существенно облегчающих процесс разработки.

Для успешного решения задачи построения процедуры в отношении алгоритмов исходных преобразований изображений должны быть выполнены следующие требования: отсутствие привязки алгоритмов к предметной области; максимальная простота алгоритмов, необходимая для облегчения их хранения; возможность получения большого количества алгоритмов.

Учитывая представленные требования к алгоритмам исходных преобразований, в системе реализован классический способ построения и считывания исходных данных и их последующая обработка. Получаемые многочисленные линейные характеристики строк изображения и являются исходными преобразованиями для построения признаков. В настоящее время система имеет возможность генерировать более 250 тысяч признаков и отбирать среди них полезные для решения требуемых задач. При этом, в отличие от классического подхода, каждый признак является отображением тех или иных свойств графических изображений представленных в неявном виде. Замечено, что выделяемые полезные признаки в зависимости от решаемых задач могут условно делиться на группы, каждая из которых относится к определенным преобразованиям исследуемого изображения, например, признаки, характеризующие геометрические фигуры в общем (без учета топологических особенностей), признаки характеризующие положение фигур (относительно своей оси или на изображении) и т.п.

Проведенная серия экспериментов показала, что система достаточно эффективно обучается и работает с различными классами изображений. При обучении одними из основных критериев отбора алфавита признаков являлись: инвариантность к любым преобразованиям изображения (плоскопараллельное смещение, поворот вокруг оси, масштабирование, перспективные преобразования, деформирование, произвольное изменение яркости, цвета или контрастности); постоянство характеристик признаков внутри классов и разнообразие характеристик для разных классов. Для сокращения времени и повышения эффективности обучения в обучающую выборку вводятся несколько дополнительных изображений, подвергнутых указанным преобразованиям.

В качестве алгоритмов принятия решений в системе используются секвенциальные алгоритмы [2] и алгоритм, основанный на вычислении оценок [1].

Важной характеристикой многофункциональной системы распознавания является возможность ее работы с различными предметными областями в режиме реального времени. Для реализации данной возможности в системе предложено использование так называемой физической иерархии исследуемого изображения. Равномерное сокращение частот дискретизации в вертикальном и горизонтальном направлениях приводит к уменьшению размеров кадра и соответственно объекта. За счет изменяемого уровня детализации неизвестный объект подразделяется на совокупность иерархически связанных слоев и чем ниже слой, тем менее детальным будет представление объекта и следовательно более обобщенными будут выводы.

Такой способ исследования объекта позволяет на ранних этапах распознавания "грубо" определить принадлежность объекта за счет выделения характерных особенностей без учета влияния мелких деталей и помех, сокращая область дальнейшего поиска. Для этого алфавит признаков должен состоять из двух частей – универсальный, предназначенный для первоначальной идентификации области принадлежности объекта на нижних слоях иерархии и рабочий, необходимый для принятия конкретных решений.

Используемый в системе способ получения исходных преобразований для построения признаков позволяет без существенных затрат переходить на требуемый уровень физической иерархии и проводить необходимые вычисления.

Таким образом, предложенный подход позволяет снизить общие затраты на создание систем распознавания и повысить эффективность их функционирования за счет автоматизации процессов создания системы, возможности эффективной настройки для работы в различных предметных областях и использования (при необходимости) физической иерархии объекта, что позволяет применять систему в динамически меняющейся обстановке.

Список литературы.

1. Горелик А. Л. Методы распознавания: Учеб. пособие для вузов/А. Л. Горелик, В. А. Скрипкин. – 4-е изд., испр. – М.: Высшая школа, 2004. – 261 с.: ил.
2. Ильичев М. А. Проверка непротиворечивости секвенциальных алгоритмов распознавания кирлиановских изображений. Сборник научных трудов ВВШ МВД РФ, 1997. - № 7. - с. 66 - 72.

Имитационное моделирование и оптимизация дискретного дозирования многокомпонентных смесей

Ермаков В. В.

*доцент кафедры высшей математики МАДИ(ГТУ), Москва
e-mail: av31125@comt.ru*

Рассмотрим имитационную модель приготовления многокомпонентной смеси с использованием дозаторов дискретного действия. Пусть требуется приготовить количество W смеси, состоящей из n компонент. Доля i -й компоненты составляет γ_i , т.е. ее должно быть в смеси $V_i = \gamma_i W$.

Предполагается, что дозаторы не имеют систематической погрешности, однако при выдаче дозы допускается случайная ошибка: если уставка i -го дозатора равна V_i , то выдается $X_i = V_i + \delta_i$, где δ_i – случайная величина. Из-за случайных погрешностей дозаторов соотношение компонент в приготовленной смеси будет отличаться от рекомендуемого рецептурой, причем на качество смеси определяющим образом будут влиять не абсолютные величины погрешностей δ_i , а отклонения отношений $X_1 : X_2 : \dots : X_n$. С математической точки зрения состояние системы описывается точкой $X = X_1 : X_2 : \dots : X_n$ в проективном пространстве. Отклонение от рецептуры описывается случайной величиной $R = \max |\alpha_i - \gamma_i|$, где $\alpha_i = X_i / \sum X_i$. Задача оптимизации заключается в минимизации математического ожидания величины R при условии $\sum X_i \geq W$ за счет выбора уставок дозаторов и порядка дозирования компонент.

При небольших n ($n = 2; 3; 4$) задача до конца решается аналитически. Если же n велико, целесообразно ограничиться квазиоптимальным решением, полученным в результате компьютерного моделирования. Установлено, что если нет дополнительных технологических ограничений на порядок дозирования компонент, то дозирование должно осуществляться в такой последовательности, когда $M \delta_i^2 / \gamma_i^2$ убывают. Для выбора уставки i -го дозатора прогнозируется ожидаемое количество готовой смеси на основе уже отдозированных компонент:

$$U = (X_1 + X_2 + \dots + X_{i-1}) / (\gamma_1 + \gamma_2 + \dots + \gamma_{i-1}).$$

Оптимальная уставка: $V_i = \gamma_i U$.

Использование нечеткой математики для анализа защищенности информационных систем

Ермаков С. А., Остапенко А. Г., Ермаков А. П.

Воронежский государственный технический университет,

394026, г. Воронеж, Московский пр.14,

ermakovs@voronezh.net

В последние годы возросла степень информатизации общества, что обусловило еще более быстрое развитие информационных технологий, а также внедрение их в производство и управление современных организаций. Одновременно наблюдается рост информационной инфраструктуры организаций, которая по мере приобретения средств вычислительной техники часто приобретает неструктурированный характер. Это приводит к неконтролируемому росту количества уязвимостей и увеличению возможностей доступа к информации со стороны внешних и внутренних нарушителей. Таким образом, информационные системы могут стать потенциально опасными.

Сейчас успешность деятельности предприятия на рынке, вне зависимости от отрасли, к которой оно принадлежит, все больше зависит от его способности накапливать, эффективно обрабатывать и анализировать информацию самого различного характера. Поэтому и убытки, которые связаны с нарушением работы автоматизированных систем и утратой ценной информации, способны мгновенно парализовать деятельность организации.

Очень быстро эти проблемы выходят на самый верхний уровень управления организацией, поскольку неизбежно затрагивают функции защиты организации в целом. Появление и осознание проблем информационной безопасности приводит к необходимости измерения величины информационного риска. Только на основе оценки риска можно определить необходимую степень защиты, выбрать стратегию развития информационной структуры организации и поддерживать на должном уровне безопасность организации.

Открытость информационной системы (объекта информатизации) и взаимодействие ее с внешней средой ставят множество проблем как при создании и анализе системы ее защиты, так и при построении соответствующих моделей. Эти проблемы связаны с неопределенностями, неизбежными при функционировании системы в изменяющейся внешней среде.

Анализируя безопасность информационной системы, мы, определяя ее границы, фактически рассматриваем выделенную нами часть реального мира. Само это выделение проводится на основании субъективных представлений о структурных связях (отношениях) этой системы, определяющих ее границы. На самом деле эти границы выражены весьма неопределенно и зависят от множества факторов, поэтому переход от исследуемой системы к окружающему ее внешнему миру происходит не скачкообразно через четкую границу, а непрерывно, образуя некоторый континуум.

Отличительной особенностью информационных систем является то, что помимо объективных законов в их функционировании существенную роль играют субъективные представления, суждения, поступки и даже эмоции людей. Действительно, при исследовании безопасности информационных систем значительное количество информации о системе может быть получено от различных групп людей: а) имеющих опыт управления данной системой и представляющих ее цели и задачи, но не знающих досконально особенностей ее функционирования; б) знающих особенности функционирования системы, но не имеющих полного представления о ее целях; в) знающих теорию и практику организации защиты, но не имеющих четких представлений о целях, задачах и особенностях функционирования системы в целом и т. п.

Получаемая информация, зачастую, носит субъективный характер, и ее представление на машинном языке не имеет аналогов в языке традиционной математики и содержит большое число неопределенностей типа: «много», «мало», если речь идет о вложениях денежных средств в совершенствование системы защиты или об изменении количества персонала, занятого вопросами защиты; «не выполнены частично», «почти выполнены» и т. д. – если речь, идет о выполнении требований руководящих документов по защите информации и т. д. Поэтому описание подобной информации на математическом языке обедняет модель исследуемой реальной системы.

Другая особенность заключается в том, что при декомпозиции объекта информатизации, представляющего собой информационную систему, на отдельные подсистемы (автоматизированные системы, системы связи и т. д.) в целях его аттестации по требованиям безопасности информации, практически невозможно точно описать связи (отношения) выделенных подсистем с остальной частью информационной системы. Информация же, полученная от экспертов, чаще всего бывает выражена в понятиях, которые имеют нечеткий смысл с точки зрения классической математики.

В реальности цели и задачи защиты часто субъективны и точно не определены преимущественно по причине, что замысел злоумышленника, от возможных действий которого строится защита, с одной

стороны, точно неизвестны, а с другой - противоположны действиям защищающейся стороны. Поэтому при построении моделей принятия решения по построению защиты таких систем возникает необходимость использования нечеткой логики (fuzzy logic), нечетких множеств и нечетких отношений.

При использовании нечеткой логики в области анализа защиты информации открывается возможность моделирования множества неопределенностей, в т.ч. неопределенности, выраженной в градациях новой информации, поступающей в систему во входных потоках данных, неопределенности, возникающей при анализе каналов утечки неопределенности, возникающей при определении уровней конфиденциальности сведений на объекте информатизации и ущерба от их разглашения и т. д., т.к. неотъемлемой частью информационной системы является человек, то применение нечеткой логики открывает возможность моделирования свойств целостности, диффузности психических образов и представлений, гибкости мышления, мотивации, присутствующих на всех уровнях отражения, регуляции и коммуникации. Здесь же затрагивается область лингвистики, заключающаяся в моделировании распространения сообщений по каналам утечки и анализа искажения смысла сообщений с помощью распределения возможностей их восприятия, описываемых функциями принадлежности.

В данной работе рассматривается вопрос применения теории нечетких множеств к оценке информационных рисков автоматизированных систем. Решается задача нахождения четких значений информационных рисков, в наибольшей степени отвечающих входным данным и базе продукционных правил, которая отражает логику взаимосвязи входных величин и риска, представленных в виде нечетких переменных. Анализируются известные методы оценки информационных рисков, рассматриваются их преимущества и недостатки.

Известны различные методики оценки информационных рисков. Их можно классифицировать по типу используемой в них процедуры принятия решения на:

- одноэтапные, в которых оценки риска выполняется с помощью одноразовой решающей процедуры;
- многоэтапные, с предварительным оцениванием ключевых параметров.

Одноэтапные методики, как правило, используются на начальной стадии развития информационной инфраструктуры организации, когда ключевые факторы, определяющие информационную безопасность, еще не выявлены.

Многоэтапные методики, с предварительным оцениванием ключевых параметров, являются более конструктивными.

Рассматривается механизм получения оценок рисков на основе нечеткой логики, который позволяет заменить приближенные табличные методы грубой оценки рисков современным математическим методом, адекватным рассматриваемой задаче. Базу знаний, при оценивании рисков на основе нечеткой логики, составляют правила, отражающие логику взаимосвязи входных величин и риска. В общем случае это логика, отражающая реальные взаимосвязи, которые могут быть формализованы с помощью продукционных правил вида «ЕСЛИ ..., ТО».

Механизм нечеткой логики требует формирования оценок ключевых параметров и представления их в виде нечетких переменных. При этом необходимо учитывать множество источников информации и качество самой информации. В общем случае это достаточно сложная задача, однако в каждом конкретном случае могут быть найдены и формально обоснованы достаточно убедительные ее решения. Механизм нечеткого вывода можно представить в виде следующей последовательности этапов (рис. 1).



Рис. 1 Основные этапы механизма нечеткого вывода

Приведенный механизм нечеткого вывода является общим. Каждая конкретная его реализация допускает некоторую свободу в выборе алгоритмов отдельных этапов обработки. Конкретные алгоритмы должны отражать специфику исследуемой системы, действующие взаимосвязи, а также вид и форму представления имеющихся априорных данных, на основе которых строится процедура вывода.

Очевидно, что риск тем больше, чем больше вероятность происшествия и тяжесть последствий. Риск в рассматриваемом случае вычисляется по формуле:

$$R = P_{ю} * C_{ю} , \quad (1)$$

где $P_{ю}$ - вероятность проявления угрозы, $C_{ю}$ - цена потери от действия угрозы.

Наиболее часто переменные $P_{ю}$ и $C_{ю}$ являются качественными величинами. В связи с этим операция умножения для них не определена и в явном виде формула (1) не может быть использована. Для качественных величин должны быть определены субъективные шкалы.

В работе реализована программа для оценки рисков с использованием теории нечетких множеств. Разработка программы «Fuzzy Logic» осуществлялась в среде Borland C++ Builder 6. Для решения поставленной задачи использован алгоритм Мамдани. Программа «Fuzzy Logic» позволяет выполнять оценку риска безопасности на основе двух входных переменных, например, вероятность осуществления угрозы и стоимостные потери. В программе учтены исключительные ситуации, которые могут возникнуть при ее выполнении (обработка некорректных входных данных).

В работе описан метод решения задачи по выполнению оценки риска безопасности. В качестве примера были проведены вычисления с входными переменными «Вероятность» и «Цена потери».

Управление интерфейсом приложений Microsoft Office на основе объектной модели

Ермолаев Б. А.,

*Пермский государственный университет
г. Пермь, 614036, ул. Мира д. 102А кв. 38
E-mail: yermboris@yandex.ru*

Плаксин М. А.,

*Пермский государственный университет,
Компьютерная школа ПГУ*

Изучение курса информатики включает в себя освоение таких общепринятых информационных технологий, как работа с текстами, электронными таблицами, базами данных, презентациями и т.п. При этом при выборе методики обучения учителю приходится выбирать, какими программными средствами пользоваться: стандартными производственными или специальными учебными. И то, и другое решение уязвимо.

С одной стороны, общепринятыми инструментами для выполнения вышеназванных работ являются программы пакета Microsoft Office (Word, Excel, Access, PowerPoint). Поэтому хотелось бы, чтобы учащиеся в процессе обучения сразу осваивали эти программы. Но для этого есть методическое препятствие. Программы Microsoft ориентированы на профессионального офисного работника. Они содержат огромное количество возможностей, полезных профессионалу, но слишком сложных для новичка. С этим еще можно мириться, если учащийся – взрослый (хотя и на взрослого огромное количество непонятных ему возможностей действует отрицательно, «давит»). Но это совершенно недопустимо, если учащийся – школьник, тем более, школьник из младших классов. Нельзя на первом же сеансе предъявлять начинающему все множество Word'овских меню и панелей. При этом неизбежно «разбегание глаз», переполнение возможностей памяти и шок, резко снижающий производительность.

Избежать всего этого можно, если использовать в учебном процессе не стандартные производственные, а специальные учебные программы. В качестве примера назовем учебный текстовый редактор Микрон, который входит в пакет «Роботландия». Учебные программы не содержат излишеств, необходимых лишь профессионалам, и позволяют обучающемуся освоить основные понятия обработки информации. Но после этого требуется дополнительное доучивание для того, чтобы перейти к работе на стандартных офисных программах.

Вообще говоря, в приложения Microsoft Office заложены средства для решения указанной проблемы. Все офисные программы имеют настраиваемый интерфейс. Пользователь может по своему усмотрению показать или спрятать те или иные панели, те или иные пункты меню и т.д. Отсюда идея: давайте продумаем для каждой офисной программы ряд конфигураций, соответствующих темам, изучаемым в школьном курсе. И при изучении каждой темы будем предоставлять пользователю только те возможности, которые нужны для изучения этой темы. Например, при первом сеансе работы с Word'ом в верхнем меню можно оставить единственную команду – «Файл», а в вертикальном меню – только команды «Открыть», «Сохранить» и «Выход». На следующем сеансе в вертикальное меню добавятся команды «Создать» и «Сохранить как» и т.д.

Такой подход является весьма эффективным, поскольку позволяет в учебном процессе сразу осваивать стандартные производственные программы, но при этом обеспечивает постепенное наращивание предъявляемых учащемуся возможностей. Подход позволяет выстроить систему изучения определенного приложения и легко интегрируется с другими подходами к обучению, так как не налагает никаких ограничений. Изучение приложения Microsoft Office таким образом более всего подходит тем, кто впервые начинает его осваивать.

К сожалению, стандартных средств настройки, предоставляемых пакетом Microsoft Office недостаточно для реализации вышеизложенной идеи. Стандартный процесс настройки слишком тяжел и медленен. Для учебного процесса нужна возможность заранее заготовить несколько конфигураций и быстро переключаться с одной на другую или вместо учебной конфигурации восстанавливать стандартную рабочую. Скорость переключения важна, поскольку в одном и том же кабинете на одних и тех же машинах, скорее всего, будут работать несколько разных групп обучающихся.

Для решения этой проблемы были разработаны специальные средства. Это программа-редактор для создания и редактирования настроек, программа-загрузчик для запуска уроков, а также программа для восстановления рабочей конфигурации Office в случае сбоев. Эти программы позволяют создавать конфигурации для основных программ Microsoft Office (Word, Excel, Access, Power Point). Они не зависят от версии Microsoft Office и Microsoft Windows.

Работа с программой-редактором осуществляется следующим образом. После запуска программы методист выбирает, с какой программой Microsoft Office он будет работать. Далее редактор запускает программу пакета Microsoft Office и запоминает ее текущую настройку.

Если создается новая конфигурация, то методист настраивает конфигурацию стандартными средствами Microsoft Office. После завершения настройки редактор запоминает новое состояние настроенной программы, возвращает программу в прежнее состояние и закрывает. Если уже существуют ранее подготовленные конфигурации, методист может выбрать одну из них для корректировки. Корректировка может быть выполнена как стандартными средствами Microsoft Office, так и собственными средствами редактора. Собственными средствами редактора представляют собой списки для выбора видимых пунктов меню (горизонтального и вертикальных), панелей и их элементов. Откорректированная конфигурация может быть сохранена под другим именем.

Кроме того, методист может указать «файл урока» – файл, который будет автоматически загружен при начале урока (при загрузке соответствующей конфигурации). Файл включается в конфигурацию. Имя приложения сохраняется в файле конфигурации. Это упрощает процесс запуска приложений и увеличивает его надежность. Невозможно запустить приложение с «чужой» конфигурацией.

Редактор позволяет загрузить конфигурацию для проверки и увидеть, как будет выглядеть готовый урок: состояние меню, панелей и текст загруженного файла. После завершения работы над учебными конфигурациями программа-редактор восстанавливает рабочую конфигурацию приложения.

Таким образом, может быть создан набор конфигураций (уроков) под темы, изучаемые в школьном курсе. Для каждого урока создается конфигурация, файл с заданиями урока, ярлык или bat – файл для загрузки конфигурации с помощью программы-загрузчика. Bat – файл для загрузки конфигурации создается автоматически.

При запуске урока программа-загрузчик извлекает из описания конфигурации имя программы Microsoft Office, запоминает её начальный вид, запускает приложение, настраивает по учебной конфигурации и открывает файл с информацией по уроку. После закрытия программы Microsoft Office загрузчик восстанавливает начальный вид приложения. В случае нормальной работы программа-загрузчик способна сама восстановить рабочую конфигурацию приложения Microsoft Office. Однако никто не гарантирован от сбоев во время работы приложения, в том числе, от зависаний операционной системы. В этом случае может приложение оказаться в учебной конфигурации. Для восстановления рабочей конфигурации предназначена специальная программа-восстановитель. Она может восстановить конфигурацию любых приложений Microsoft Office. Восстановлена может быть либо одна из ранее сохраненных конфигураций, либо стандартный вид приложения Microsoft Office.

Выбор нужной конфигурации может выполняться двумя способами. Программа-загрузчик может быть запущена без параметров. В этом случае пользователю будет предложено на выбор меню доступных конфигураций. Возможен запуск загрузчика с параметрами командной строки (например, с помощью bat – файла). В качестве таковых указываются имя файла-конфигурации и некоторые специальные уточнения, влияющие на скорость запуска загрузчика, его безопасность и способ отображения самого загрузчика на экране.

Способ вызова загрузчика зависит от категории пользователя и цели вызова. Вызов без параметров предназначен для пользователя-методиста, который знает о наличии загрузчика, файлов-конфигураций и т.п. Вызов с параметрами предназначен для учащихся. В этом случае вызов загрузчика с указанием нужной конфигурации может быть «спрятан» в bat-файл. Запуск такого bat-файла позволяет скрыть от пользователя само существование специальной программы-загрузчика.

Все программы используют общие принципы функционирования и общие структуры данных. Приложения взаимодействуют с программами Microsoft Office с помощью COM – технологии.

Управление интерфейсом приложений осуществляется на основе объектной модели. Конфигурации хранятся в виде наборов объектов, соответствующих панелям инструментов, меню, выпадающим меню, элементам панелей инструментов и элементам выпадающих меню. В конфигурацию включаются все видимые элементы интерфейса. При загрузке конфигурации эти элементы показываются, а все остальные прячутся.

Управление интерфейсом приложений Microsoft Office осуществляется на основе объектной модели. На этапе исследований была составлена объектная модель приложений, описывающая, какие объекты соответствуют панелям инструментов, меню, выпадающим меню, элементам панелей инструментов и элементам выпадающих меню. Было выяснено, как объекты связаны, какие свойства и методы имеют и на основе всех этих сведений организовано управление интерфейсом.

Формат данных и алгоритмы работы программ позволяют работать со всеми версиями Microsoft Office. Например, можно загрузить конфигурацию, созданную под одной версией пакета Microsoft Office, в любой другой его версии. Это касается также и разных локализаций пакета Microsoft Office.

При разработке комплекса была предусмотрена возможность расширения списка поддерживаемых приложений. При этом требовалось минимизировать изменения в коде программы, а в идеале – вообще обойтись без них. В настоящей версии программы допускается расширение списка поддерживаемых приложений без изменения кода программы. Для добавления приложения Microsoft Office в список поддерживаемых приложений достаточно отредактировать специальный текстовый файл с информацией о дополнительных приложениях. В файле находится информация о названии приложения, о соответствующем ему COM-объекте и сведения об объектной модели приложения. Если получить нужные сведения об объектной модели приложения, то можно добавить в этот файл описание нового приложения и программы будут работать с этим приложением. К сожалению, структура Windows-приложений в настоящее время не позволяет генерировать такие описания автоматически.

Разработанные программные средства настройки приложений Microsoft Office для учебных целей позволяют эффективно обучать новичков работе с пакетом. Разработанные программы были опробованы учителями и методистами пермских школ при подготовке преподавательского курса информатики для младших классов.

В перспективе опробованный подход обеспечивает возможность автоматического получения учебных версий программ из производственных. В настоящее время для этого есть препятствие: отсутствие единого стандарта на хранение информации, необходимой для реализации подхода. Информацию о приложениях Office, об их объектной модели приходилось получать вручную. Если появится возможность получать требуемую информацию о нужных приложениях автоматически, то появится возможность “на лету” превращать производственные программы в учебные и гибко настраивать их под производственные нужды. Было решено обратиться в Microsoft с вопросами и предложениями. В настоящее время ведется переписка с фирмой Microsoft о возможности получения и использования информации об объектной модели других ее продуктов. В первую очередь речь идет о системе программирования Visual Studio – основной инструментальной системе, продвигаемой Microsoft. В случае получения от фирмы необходимой информации и разрешения на ее использование можно будет создать на базе Visual Studio систему для обучения программированию. В настоящее время обучение, как правило, ведется на старинных программных средствах типа Турбо-Паскаль. Одна из основных причин этого – чрезмерная сложность современных программных систем с точки зрения процесса обучения. То есть картина здесь совершенно аналогична ситуации с Microsoft Office. В случае удачи это позволит перейти в обучении программированию от устаревших систем, типа Турбо-Паскаль к современным программным средствам типа Visual Studio.

Вейвлет-преобразование изображений с нецелым параметром сжатия

Жизняков А. Л., e-mail: lvovich@newmail.ru

доцент кафедры «Электроники и вычислительной техники»

Муромский институт Владимирского государственного университета

602264, г.Муром, Владимирской обл., ул.Орловская, 23

В статье рассматриваются возможности применения вейвлет преобразования с дробным коэффициентом сжатия в задачах обработки цифровых изображений. Предложен подход к определению дробных значений изменения масштаба, адаптируемых к частотным характеристикам изображения. Одной из проблем, возникающих при реализации алгоритмов вейвлет - обработки, является то, что в большинстве приложений используется вейвлет - преобразование с масштабным множителем равным 2. Главным их преимуществом является удобство численных расчетов. Однако можно показать [1-3], что в

рамках многомасштабного анализа этот множитель должен быть рациональным числом и никаких других требований не налагается. Поэтому можно построить схемы с другими целыми или дробными множителями. Их использование может привести к лучшей локализации по частоте. Для вейвлетов с масштабным множителем 2 их Фурье – образ сосредоточен в основном в пределах одной октавы между π и $\pi/2$, тогда как вейвлет – базисы с дробными множителями могут иметь ширину полосы пропускания, более узкую, чем октава. В связи с этим актуальна разработка алгоритмов, реализующих дробные значения изменения масштаба, адаптируемые к частотным характеристикам сигнала.

Также для анализа можно использовать непрерывное вейвлет – преобразование, так как возможности диадного преобразования для анализа сигналов значительно ниже. Непрерывной функции ставится в соответствие дискретная функция, т.е. не более чем счетное множество чисел. Однако при этом обеспечивается выполнение обратного преобразования, поэтому при обработке сигналов (сжатие, фильтрация и т.д.) диадное преобразование используется достаточно часто. Кроме того, реализованы быстрые алгоритмы его выполнения.

Обычно применяемое диадное вейвлет - преобразование представляет собой разложение по базису в рассматриваемом функциональном пространстве и не содержит избыточной информации[2]. При этом ее хватает для того, чтобы преобразование было обратимым. Однако этой информации может оказаться недостаточно для решения тех задач, где обратное преобразование может и не требоваться, зато необходим детальный анализ сигнала по его образу. Так, диадное преобразование не инвариантно сдвигу сигнала, что существенно ограничивает возможности преобразования в задачах распознавания, классификации, поиска по шаблону, сегментации и др. Для непрерывного преобразования образ сдвинутого на любое действительное число сигнала совпадает со сдвигом на то же число образа несдвинутого сигнала.

Основной недостаток непрерывного преобразования - содержание большого объема информации. Сигналу, определенному на R , ставится в соответствие функция, определенная на $R \times R$. Поэтому его целесообразно применять в задачах, где требуется анализ сигналов, выявление особенностей, локальных неоднородностей и т.д.

Пусть s – строка изображения, длиной N . Тогда вейвлет преобразование этой строки запишется как

$$W_{s,a,b} = \frac{1}{\sqrt{a}} \sum_{j=0}^{N-1} s_j |j+1| \psi\left(\frac{x-b}{a}\right) dx, \quad (1)$$

причем коэффициент сжатия a может быть любым вещественным числом (кроме нуля).

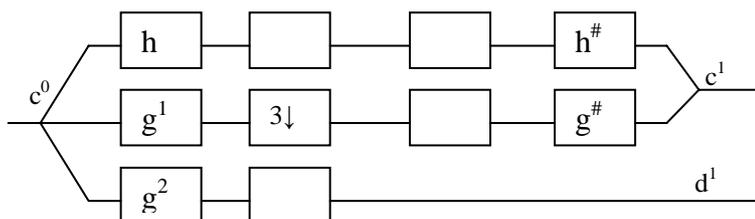
Применяя выражение (1) к строкам и столбцам исходного изображения и задавая на каждом шаге коэффициент сжатия, можно получить пирамиду с переменным масштабом изображений. Такой вариант, по сути, является численной реализацией непрерывного вейвлет – преобразования. При этом вопрос о его обратимости не ставится.

При вейвлет – обработке изображений часто используется адаптивный вариант вейвлет преобразования (пакетное преобразование). При этом на каждой итерации происходит разделение на аппроксимирующую и детализирующую составляющие не только НЧ-, но и ВЧ –компоненты на данном масштабе рассмотрения. Для выделения из пакета наиболее подходящего базиса разложения часто используют энтропийный критерий вероятностного распределения вейвлет – коэффициентов. Из полученного набора выбирается тот базис, который приводит к наименьшей энтропии.

Подобное преобразование находит широкое применение при фильтрации и сжатии, однако слабо подходит для решения задач, связанных с отслеживанием особенностей, выделением структурных признаков, анализом текстур и т.д. НЧ часть разложения (являющаяся уменьшенной копией исходного изображения), по-прежнему меняет масштаб только в 2 раза.

Другой подход предполагает формирование ортонормированных базисов вейвлетов с нецелыми показателями сжатия. В [3] рассматриваются конструкции для произвольных рациональных параметров сжатия и представлен пример преобразования с параметром сжатия $3/2$.

Согласно схеме рис. 1, начиная с уровня c_0 , производится разложение на три поддиапазона. Затем два диапазона с наименьшими частотами перегруппировываются с помощью фильтра восстановления с параметром сжатия 2. Результатом этой операции является уровень c_1 .



Рассмотренные выше подходы позволяют сформировать пирамиду изображений с переменным масштабным коэффициентом. Во всех этих случаях, стоит задача выбора конкретного значения коэффициента масштабирования.

Исходя из этого, предлагается подход, позволяющий адаптировать коэффициент масштабирования к частотным особенностям препарируемого изображения.

В основе подхода лежит тот факт, что при переходе к каждому следующему масштабу рассмотрения сигнала (в случае изображений - двумерного), точность воспроизведения его формы резко снижается.

Причина этого заключается в том, что энергия сигналов (по крайней мере, во многих конкретных приложениях), как правило, возрастает при переходе к более низким частотам. При кратномасштабном анализе возможно резкое отсечение части частотных составляющих, несущих значительную информацию. Именно во время такого «скачка» от масштаба j к $2*j$ и может произойти потеря определяющего признака[1].

Идея предлагаемого подхода заключается в адаптивном уменьшении кратности изменения масштаба рассмотрения по мере увеличения коэффициента k . При этом достигается более тонкое отслеживание поведения интересующих особенностей изображения.

В качестве адаптивного критерия здесь может быть использован, например, порог равный отношению мощности ВЧ- составляющей, к мощности НЧ – составляющей.

Кратность изменения масштаба определяется отношением полос.

Для двумерного перегруппированного (НЧ составляющие лежат в начале координат) спектра, вычисленного на основе БПФ (рис.2), коэффициент масштабирования

$$k = \frac{\Delta\omega_1 + \Delta\omega_2}{\Delta\omega_1} \text{ при условии, что } \frac{E(\Delta\omega_1) + E(\Delta\omega_2)}{E(\Delta\omega_1)} = P^0,$$

где $E(\Delta\omega)$ – мощность спектральных составляющих полосы $\Delta\omega$, P^0 - заданный порог.

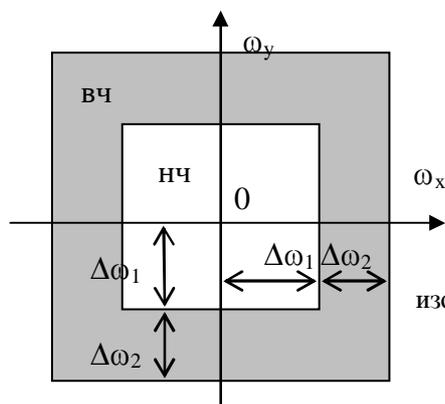


Рис. 2. Полосы «верхних» и «нижних» частот в спектре

Исследование предложенного подхода проводилось на тестовых изображениях.

Таблица 1 - Значения масштабных коэффициентов для тестового изображения

Шаг	P0=0.2	P0=0.1	P0=0.05	P0=0.02	P0=0.01
1	4.8	4.0	2.9	1.7	1.6
2	2.6	1.3	1.9	1.6	1.8
3	1.9	1.4	1.3	1.4	1.4
4	1.3	1.3	1.3	1.5	1.3
5	1.2	1.2	1.2	1.2	1.3
6	1.3	1.2	1.2	1.2	1.3
7	1.7	1.2	1.2	1.2	1.2

Полученная в результате подобной процедуры последовательность изображений $f^{(0)} \supset f^{(1)} \supset \dots \supset f^{(k)}$ используется для выполнения процедур обработки и анализа изображений, основанных на кратномасштабном представлении.

Список литературы

1. Жизняков А. Л., Вакунов Н. В. Вейвлет-преобразование в обработке и анализе изображений. – М.: Государственный научный центр Российской Федерации – ВНИИ Геосистем, 2004.-102 с.
2. Переберин А. В. О систематизации вейвлет – преобразований. Вычислительные методы и программирование// Вычислительные методы и программирование. – 2001.-Т.2.-С. 15-40

3. Добеши И. Десять лекций по вейвлетам. – Москва-Ижевск: НИЦ “Регулярная и хаотическая динамика”, 2004, 464 с.

Исследование алгоритмов измерения шума на основе вейвлет преобразования

Жизняков А. Л., докторант МИВЛГУ
Фомин А. А., МИВЛГУ,
602264, г. Муром, Владимирской обл., ул. Орловская, 23,
МИВЛГУ, кафедра "ИС"
verbovskiy@yandex.ru

В задаче автоматического шумоподавления на изображениях с разной степенью зашумленности может быть актуальна проблема корректного определения уровня шума. Часто шумы, присутствующие на изображениях, могут быть описаны моделью аддитивного белого гауссовского шума с нулевым математическим ожиданием и дисперсией σ^2 , то есть зашумленное изображение может быть представлено в виде $X = f + W$, где f – изображение, W – шум. В данном случае, измерение уровня шума можно свести к определению дисперсии шума на изображении.

Одним из наиболее часто применяемых подходов к оценке дисперсии шума на основе вейвлет преобразования является оценка по медиане вейвлет коэффициентов пренебрежением влияния f . Грубая оценка дисперсии шума $\tilde{\sigma}^2$ находится как $\tilde{\sigma}^2 = M / 0.6745$, где M – медиана диагональных коэффициентов вейвлет разложения. Данный подход применим в том случае, когда f – кусочно-гладкий сигнал, что приводит к небольшому количеству вейвлет коэффициентов с большой амплитудой, которые имеют малое влияние на медиану M [1]. В случае текстурных изображений или изображений с высокой степенью детализации этот подход не позволяет с достаточной степенью точности определять дисперсию шума, поскольку в высокочастотной составляющей вейвлет разложения присутствует большое количество коэффициентов большой амплитуды, что приводит к увеличению значения дисперсии $\tilde{\sigma}^2$.

Предлагаемый подход к оценке дисперсии шума основан на известном подходе, осуществляющем разбиение зашумленного изображения на небольшие блоки и рассчитывающем дисперсию элементов блоков, расположенных на гладких участках изображения [2]. Отличием предлагаемого подхода является то, что все вычисления ведутся в области высокочастотных коэффициентов вейвлет разложения. Возможность использования такого подхода обусловлена тем, что закон распределения шума в частотной области изображения (области вейвлет коэффициентов) совпадает с законом распределения в пространственной области, что является следствием свойства ортогональности вейвлет преобразования [3].

Подход к оценке дисперсии шума непосредственно по элементам изображения имеет ряд ограничений, связанных со средними значениями яркостей блоков, накладываемых на правила выбора блоков. В предлагаемом подходе эти ограничения отсутствуют, поскольку вся информация о яркости элементов изображения содержится в низкочастотной полосе вейвлет спектра, а в высокочастотной полосе содержатся лишь элементы, соответствующие резким перепадам яркости на изображении (границы объектов, детали, шум).

Вейвлет разложение изображения определяется как:

$$vw_i = \langle X | \varphi\psi_i \rangle, wv_i = \langle X | \psi\varphi_i \rangle, ww_i = \langle X | \psi\psi_i \rangle, i \in Z,$$

где $\varphi\psi$, $\psi\varphi$, $\psi\psi$ – вейвлеты, i – уровень разрешения, vw_i , wv_i , ww_i – высокочастотные горизонтальные, вертикальные и диагональные вейвлет коэффициенты.

Вейвлет коэффициенты подчеркивают, соответственно, горизонтальные, вертикальные и диагональные края предметов [4]. На изображениях, как правило, в большей мере присутствуют горизонтальные и вертикальные элементы, поэтому набор вейвлет коэффициентов ww_i содержит меньше элементов, соответствующих границам объектов или деталям. Вследствие этого, использование для расчетов набора коэффициентов ww_i повышает точность оценки дисперсии шума.

Повышение точности оценки дисперсии шума при использовании высокочастотных диагональных

коэффициентов вейвлет разложения W_i обусловлено еще и тем, что меньшее количество вейвлет коэффициентов, соответствующих границам объектов или деталям, позволяет использовать для расчетов больший размер блоков.

Тестирование алгоритма проводилось на изображениях с искусственно наложенным аддитивным белым гауссовским шумом с нулевым математическим ожиданием и дисперсией σ^2 . Часть тестовых изображений представлена на рисунке 1. В таблице 1 и на рисунке 2 приведены результаты расчетов дисперсии шума на основе трех подходов (приведены результаты расчетов для четырех изображений: портретного изображения – "Лена", пейзажа – "Море" и изображений, приведенных на рисунке 1). Первый подход оценивает дисперсию по медиане вейвлет коэффициентов, второй – основан на расчете дисперсии в блоках изображения, третий – рассчитывает дисперсию в блоках в области вейвлет коэффициентов. Предлагаемый подход (подход 3) превосходит два других по точности определения дисперсии. Первый подход становится неработоспособным в случае, когда изображение не является кусочно-гладким (рисунок 1б). Два других подхода позволяют получать удовлетворительные результаты, но при прочих равных условиях (размер блока) предлагаемый подход превосходит известный примерно на 10%.

При фильтрации шума на изображениях, методами вейвлет фильтрации, применение предлагаемого подхода к оценке дисперсии шума является более предпочтительным, поскольку это позволяет использовать один математический аппарат для решения обеих задач.

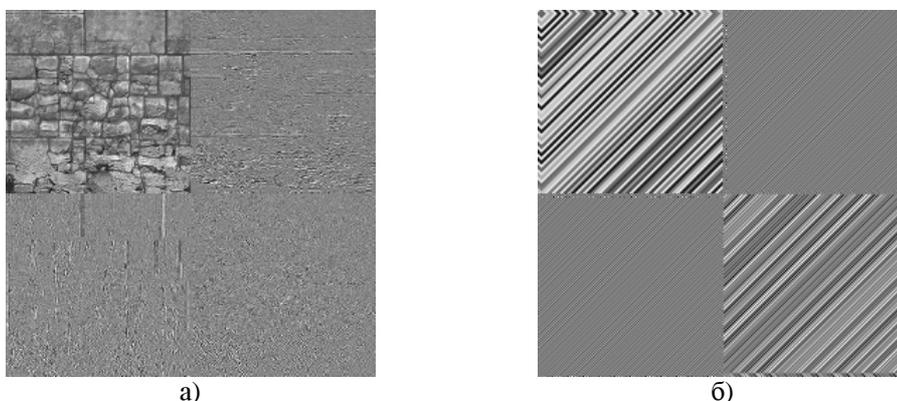


Рисунок 1 – Тестовые изображения: а – Изображение "Стена"; б – Изображение "Полосы";

Таблица 1 – Значения дисперсии шума

Подходы к оценке дисперсии шума	Изображение					
	"Лена"			"Море"		
	Фактическая дисперсия шума					
	0,0050	0,0150	0,0000	0,0100	0,0070	0,0300
	Результаты оценки дисперсии шума					
Подход 1	0,0053	0,0136	0,0003	0,0095	0,0067	0,0283
Подход 2	0,0059	0,0176	0,0007	0,0113	0,0081	0,0314
Подход 3	0,0050	0,0158	0,0002	0,0103	0,0071	0,0287
Подходы к оценке дисперсии шума	Изображение					
	"Стена"			"Полосы"		
	Фактическая дисперсия шума					
	0,0250	0,0150	0,0080	0,0180	0,0060	0,0030
	Результаты оценки дисперсии шума					
Подход 1	0,0235	0,0154	0,0091	0,0404	0,0330	0,0291
Подход 2	0,0302	0,0184	0,0124	0,0124	0,0037	0,0021
Подход 3	0,0246	0,0154	0,0099	0,0226	0,0079	0,0069

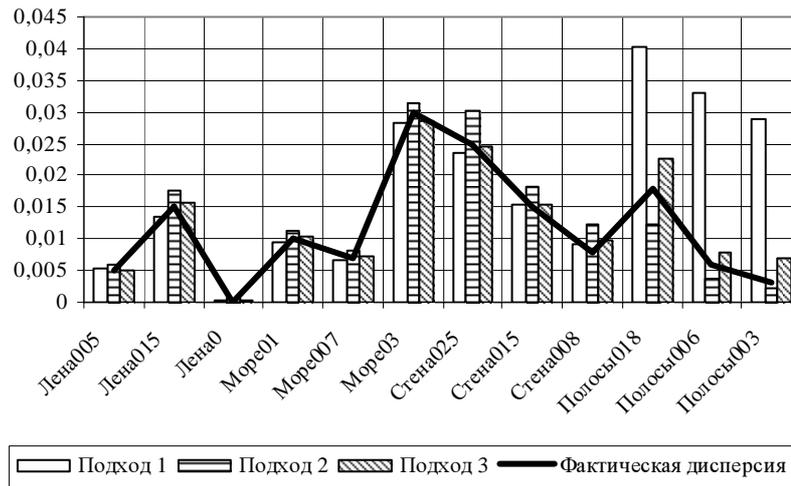


Рисунок 2 – Расчетные значения дисперсии шума

Список литературы

1. Малла С. Вэйвлеты в обработке сигналов: Пер. с англ. – М.: Мир, 2005. – 671 с., ил.
2. Жизняков А. Л., Вакунов Н. В. Вейвлет-преобразование в обработке и анализе изображений. – М.: Государственный научный центр Российской Федерации – ВНИИгеосистем, 2005. – 102 с., ил.
3. Калинкина Д. А. Определение уровня шума на изображении на основе усреднения дисперсии в блоках. Международная конференция студентов и аспирантов по фундаментальным наукам "Ломоносов 2005".
4. Добеши И. Десять лекций по вейвлетам. – Ижевск: НИЦ "Регулярная и хаотическая динамика", 2001. – 464 с.

О разработке и использовании программных агентов в распределенных интеллектуальных системах

Зайцев Е. И.,

*к.т.н., доц. Московского государственного университета
приборостроения и информатики
E-mail: zei@tsinet.ru*

Разработка распределенных интеллектуальных систем, состоящих из модулей, способных одновременно и независимо решать свои частные задачи, а также объединять свои локальные возможности для решения сложных проблем, связана с использованием программных агентов (software agents). Понимание того, что программные агенты имеют широкий спектр применений, а также осознание потенциала агентно-ориентированной технологии инвесторами привели к тому, что многие разрабатываемые лабораториями, университетами, фирмами и промышленными организациями системы позиционируются как агентные, при этом трактовка понятия программного агента вызывает разногласия.

Часто программные агенты определяют просто как автономные или полуавтономные процессы, которые действуют от имени пользователя в среде, в которой выполняются другие процессы, однако, под это определение попадают многие объектно-ориентированные компоненты. Базовое определение агента, данное международной организацией FIPA (Foundation for Intelligent Physical Agents), согласно которому агент – это главный исполнитель в домене, обладающий одной или несколькими сервисными возможностями, образующими единую и комплексную модель выполнения, которая может включать доступ к внешнему программному обеспечению, пользователям и средствам связи, также требует дополнительной детализации, так как оно применимо и к объектно-ориентированным программным серверам.

С позиций объектно-ориентированного программирования агент представляет собой особый тип объекта (smart object), который является экземпляром класса, содержащего не только типичные для

объектных классов методы инициализации, чтения и записи, конструкторы и деструкторы, обработчики исключений, атрибуты (члены-данные класса), но также когнитивные структуры данных (CDS - Cognitive Data Structures) и методы, реализующие достоверный (дедуктивный) или правдоподобный (индуктивный, абдуктивный) выводы. Синергетическая система, глобально демонстрирующая интеллектуальное поведение и способная решать слабо формализуемые задачи, может быть построена на базе обычных совместно функционирующих объектов. При этом ориентация на коллективный потенциал, а не на индивидуальные возможности объектов, не превращает объектно-ориентированную систему в агентную.

Отличительная особенность агентов состоит не только в способности предоставлять определенные услуги и наличии у них развитых средств для взаимодействия со средой и другими агентами, но также в их рациональности. В отличие от объектов в модели агентов присутствуют ментальные свойства (интенциональные характеристики): убеждения, желания и намерения (модель BDI – Belief, Desires, Intentions), которые направляют деятельность агента и которые могут меняться в процессе функционирования. В противоположность убеждениям (истинным или ложным) желания рассматриваются не с позиций их истинностного значения, а в контексте удовлетворения или выполнения. Если желания выполнимы, значит, агент в состоянии реализовать свои намерения и выполнить необходимые для достижения цели действия. Обычно для того, чтобы достичь цели агент осуществляет планирование, при этом подготовка агентом решения может рассматриваться как процесс уточнения иерархической структуры плана благодаря формированию всё более конкретных намерений.

Обычно используется регрессивное планирование, когда поиск последовательности действий, связанных с достижением цели осуществляется в обратном направлении (от цели). В данном случае основной задачей является поиск состояний, из которых, в результате выполнения определенных действий, можно достичь цели. Вычисление описаний таких состояний (регрессия цели через действие) в мультиагентной системе может осуществляться специальными когнитивными агентами, содержащими информацию о совместных целях (желаниях) и намерениях других агентов. В обязанности когнитивных агентов входит составление совместных планов, состоящих из действий, координирующих работу связанных с ними реактивных агентов. При этом совместные действия агентов группы могут определяться неявно путем описания того, как каждое отдельное действие сочетается с другими возможными действиями.

При решении задач агенты пользуются непроцедурной информацией о процессе – знаниями (истинными обоснованными убеждениями), которые определены в базе знаний и доступны с помощью механизмов логического вывода. В архитектуре агента кроме знаний о предметной области выделяют знания о взаимодействиях и управляющие знания (метазнания о применении знаний о взаимодействии к знаниям о предметной области). В распределенных интеллектуальных системах локальные базы знаний агентов независимы друг от друга, таким образом, проблема обеспечения непротиворечивости знаний, существенная для централизованных интеллектуальных систем, уступает место задачам обеспечения кооперации и коммуникации агентов.

Кооперация является основной формой организации взаимодействия между агентами распределенной системы. Она предполагает не только сотрудничество агентов при решении задач, но также координацию их совместных действий и возможность разрешения возникающих в процессе работы конфликтов. Агенты могут самостоятельно согласовывать и координировать свои действия, либо в роли координатора может выступать специальный когнитивный агент, который при этом понимается как находящийся на метаяуровне по отношению к остальным агентам. В свою очередь, работу группы метаагентов может координировать когнитивный агент вышележащего уровня. В распределенной интеллектуальной системе иерархическая организация взаимодействия агентов предполагает использование когнитивных агентов в каждом вычислительном узле.

Коммуникация агентов является не столько процессом передачи информации от одного агента к другому, сколько процессом, в котором агенты преследуют свои цели. Практическая реализация различных вариантов коммуникации агентов чаще всего сводится к использованию доски объявлений (blackboard), либо механизма передачи сообщений. Характерным способом реализации систем использующих доску объявлений является создание сообщества агентов, взаимодействующих путём согласования. Каждый агент может передавать на доску информацию о любых изменениях своего состояния, добавлять факты, основанные на специализированном знании, подтверждающие или опровергающие находящуюся на доске гипотезу, или выдвигать новую, тем самым, активизируя деятельность других агентов, использующих собственную базу знаний.

Даже в случае, когда объекты и агенты существуют в едином адресном пространстве инкапсуляция данных и методов их обработки, которая свойственна агентно-объектной модели (AOM – Agent Object Model), делает обоснованным использование механизма взаимодействия посредством передачи сообщений без использования разделяемой памяти. Естественным образом такое взаимодействие согласуется с концепцией распределенных вычислений в системах, построенных на основе кластерных решений или грид-технологии (Grid). В таких системах отсутствует центральное управление, вычислительные узлы имеют собственную локальную память, доступ к которой возможен только путем

обмена сообщениями с процессором, которому принадлежит адресуемая память, а параллельные вычислительные процессы реализуются на уровне крупных структурных единиц (программ), решающих некоторую часть сложной проблемы.

Действия агентов могут осуществляться в разных потоках одного процесса, что обеспечит простоту их взаимодействия путем простой передачи параметров и, возможно, использования глобальных переменных. Естественно, что параллельная работа агентов, разделяющих единое адресное пространство вычислительного узла, будет возможна только в случае многопроцессорной системы. При разработке интеллектуальной системы, в которой предполагается реализация агентов в различных узлах мультимедийной системы, для снижения общего уровня сложности задачи программирования можно использовать стандартные объектно-ориентированные технологии, обеспечивающие прозрачность доступа к удаленным ресурсам (CORBA, .NET, J2EE). В случае проектирования глобально распределенной системы, скоординированное пространство, необходимое для реализации открытых интеллектуальных многоагентных систем, в котором должны быть сосредоточены универсальные средства аутентификации, авторизации, доступа к ресурсам, а также поиска необходимых ресурсов, можно создать используя инструментальные средства Grid (Globus Toolkit или стандарты, развиваемые в контексте OGSA – Open Grid Services Architecture).

В зависимости от решаемых распределенными агентами задач могут применяться различные комбинации синхронных и асинхронных типов взаимодействия. Наиболее жесткая форма синхронного взаимодействия предполагает, что отправитель остается заблокированным на время обработки его сообщения получателем. Значительно повысить эффективность работы агентов позволяют асинхронные операции обмена. Для организации нерезидентной асинхронной связи можно использовать низкоуровневый набор примитивов служб транспортного уровня. В большинстве случаев для построения распределенных интеллектуальных систем вместо простой модели обмена сообщениями транспортного уровня, можно использовать высокоуровневые механизмы, предоставляемые системами промежуточного уровня. Для организации взаимодействия распределенных агентов удобно также воспользоваться коммуникационными протоколами прикладного уровня, называемыми языками взаимодействия агентов (Agent Communication Language – ACL).

В случаях, когда узлы должны динамически перестраиваться на решение различных задач используются мобильные агенты, поддерживающие либо слабую (weak mobility), либо сильную (strong mobility) модели мобильности. В случае слабой мобильности на удаленный узел переносится только сегмент кода (набор инструкций) возможно вместе с некоторыми данными инициализации. Характерной чертой слабой мобильности является то, что перенесённая программа всегда запускается из своего исходного состояния, как это происходит, например, с Java-апплетами. Интеллектуальные мобильные агенты часто требуют поддержки сильной мобильности, при которой вместе с сегментом кода переносится также сегмент исполнения, что позволяет работающему процессу после приостановки и перенесения на другую машину продолжить его выполнение с того места, на котором этот процесс был приостановлен. Примером агентов с сильной мобильностью являются Java-аглеты (aglets), которые пересылаются из одного контекста в другой по протоколу ATP (Agent Transfer Protocol) прикладного уровня, не зависящему от платформы и использующему URL для определения местоположения агентов и серверов.

Технология мобильных агентов MASIF (Mobile Agents System Interoperability Facility) представляет ещё один подход к разработке распределенных систем, при котором обеспечивается интероперабельность между различными существующими и разрабатываемыми агентскими системами. Стандарт MASIF, представляя собой надстройку над стандартом CORBA, позволяет объединить традиционную клиент-серверную парадигму и технологию мобильных агентов. Интероперабельность, обеспечивающая открытость создаваемых приложений относительно других сред, достигается при стандартизации таких аспектов, как передача агентов и служебных (используемых агентом) классов между агентскими системами, а также управление агентами. Примером инструментальной среды для разработки мобильных агентов, опирающейся на стандарт MASIF, является ИС Grasshopper (IKV++).

Способности использовать знания, самостоятельно формировать цели и осуществлять планирование, а также возможность обучаться (пополнять и модифицировать убеждения) выдвигают агентов на более высокий уровень сложности по отношению к традиционным объектам. При этом объектно-ориентированные методы и средства продолжают оставаться основным инструментом для моделирования и реализации агентно-ориентированных систем. При анализе и проектировании многоагентных систем могут применяться унифицированный язык моделирования (Unified Modeling Language – UML) и соответствующая ему спецификация MOF (Meta Object Facility) для описания метамodelей, а реализация агентов может осуществляться с использованием объектно-ориентированных языков программирования, которые либо имеют встроенные средства параллельной обработки, либо предполагают возможность использования системных и пользовательских библиотек, поддерживающих различные модели параллельных и распределенных вычислений (например библиотека MICO, соответствующая стандарту CORBA, инкапсулируемый в систему объектных классов интерфейс MPI – Message Passing Interface для высокопроизводительных параллельных систем или специальные

библиотеки агентов, такие как Intelligent Agent Library, Kafka, Agentx и др.). Для поддержки процессов проектирования агентов и мультиагентных систем могут использоваться специальные инструментальные средства, такие как Agent Builder Tool Kit (Acronymics, Inc.), JADE - Java Agent Development Framework (Telecom Italia Group), OAA – Open Agent Architecture (SRI International), ADE – Agent Development Environment (Gensym), Agent Development Kit (Tryllian), The ZEUS Agent Building Toolki, Toshiba Bee-agent и др.

Переход от строго упорядоченной последовательности задач к неупорядоченной коллекции слабосвязанных процессов, обменивающихся сообщениями естественным образом подходит для решения проблем, характеризующихся отсутствием или сложностью формальных алгоритмов решения, нечеткостью достигаемых целей. Агенты, реализованные в различных потоках и процессах одного или нескольких компьютеров, реализуют выводы параллельно, при этом программные агенты не обладают полным знанием о своем окружении и делают выводы на основании неполной и неопределенной информации. Для моделирования неопределенности вводятся дополнительные механизмы, использующие вероятностные, эмпирические или нечеткие методы учета неполноты информации.

Список литературы

1. Городецкий В. И., Грушинский М. С., Хабалов А. В. Многоагентные системы / Новости искусственного интеллекта, №2, 1998.
2. Зайцев Е. И. Проектирование и реализация распределенных интеллектуальных систем на основе агентов. Приборы и системы. Управление, контроль, диагностика. №11, 2004.
3. Рассел С., Норвиг П. Искусственный интеллект: современный подход. –М.: Вильямс, 2006.
4. Тарасов В. Б. От многоагентных систем к интеллектуальным организациям. –М.: Эдиториал УРСС, 2002.
5. Eiter T., Fink M., Sabbatini G., Tompits H. Using Methods of Declarative Logic Programming for Intelligent Information Agents / Theory and Practice of Logic Programming. 2002. - Vol. 2. - No. 6.
6. Jennings N. R. An agent-based approach for building complex software systems. Communications of the ACM, 44 (4). 2001.

Рост эффективности идентификации по модели Вольтерра при увеличении количества гармоник во входном сигнале

Зиновьев А. А.

*ООО «Digital Communication – Service», инженер-конструктор
Украина 65003, Одесса-03, ул. Черноморского казачества 38, а/я 11,
zinoval@mail.ru*

Рассматривается идентификация нелинейных инерционных систем по модели функционального ряда Вольтерра входными полигармоническими сигналами. Показано, что для фиксированного порядка идентифицируемой передаточной функции доля информативных гармоник растет с ростом числа гармоник во входном полигармоническом сигнале. Предлагается использовать указанный эффект для компенсации спада эффективности идентификации при возрастании порядка модели.

Одним из способов построения математической модели одновременно инерционной и нелинейной системы является ее представление в виде отрезка функционального ряда Вольтерра:

$$y[x(t)] = \sum_{n=1}^q y_n[x(t)] = \sum_{n=1}^q \underbrace{\int \dots \int}_{-\infty}^{\infty} w_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t - \tau_i) d\tau_i, \quad (1)$$

где $x(t)$ и $y[x(t)]$ – соответственно входные и выходные сигналы, называемые также тестами (стимулами) и откликами (реакциями); $y_n[x(t)]$ – парциальные составляющие отклика, которые соответствуют отдельным степеням нелинейности; $w_n(\tau_1, \dots, \tau_n)$ – определяемые в ходе идентификации ядра Вольтерра, которые можно рассматривать как многомерные обобщения весовой функции; и q – порядок модели Вольтерра. Модель Вольтерра находит применение в различных системах – от технических до физиологических [1]. Решение задачи идентификации состоит из двух этапов: разделения нелинейных свойств и определения из парциальных составляющих ядер Вольтерра (или их изображений). Для разделения нелинейных свойств известно достаточно много методов: разделения парциальных составляющих на четные и нечетные порядки [2]; кратного дифференцирования откликов по амплитуде тестов [2]; спектральном анализе отклика [3]; вычитания из отклика линейных комбинаций откликов от разных сочетаний компонент теста [4]; наименьших квадратов [5]; систем линейных алгебраических уравнений откликов от разноамплитудных тестов [6-9]. Определение ядер Вольтерра проводится во временной или частотной области. Во временной области определяются многомерные весовые или переходные функции численным решением интегральных уравнений Вольтерра [10]. В частотной области определяются многомерные передаточные функции (МПФ) – Фурье-образы ядер Вольтерра, рассматриваемые как многомерные обобщения передаточной функции:

$$W_n(j\omega_1, \dots, j\omega_n) = \int \dots \int_{-\infty}^{\infty} w_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n \exp(-j\omega_i \tau_i) d\tau_i \quad (2)$$

При определении МПФ – идентификации в частотной области – на вход системы подаются полигармонические сигналы, которые позволяют перейти от решения интегральных уравнений [10] с неизвестными переходными или весовыми функциями (при идентификации во временной области) к линейным алгебраическим уравнениям с неизвестными МПФ. Кроме этого, частотная идентификация устойчивее к белому шуму (за счет наблюдения нескольких периодов, что необходимо для фильтрации) и предпочтительнее, когда исследуемая система работает в периодическом режиме – например, физиологические системы [5]. При этом для полного описания всех возможных взаимных модуляций отдельных компонент входного сигнала для заданного порядка нелинейности входной полигармонический сигнал должен состоять из количества гармоник не меньшего, чем степень определяемой нелинейности. Согласно [11], парциальная составляющая отклика для входного полигармонического сигнала с нулевыми фазами и равными амплитудами, имеет вид:

$$y_n[x(t)] = \frac{a^n}{2^{n-1}} \sum_{m=0}^{\lfloor n/2 \rfloor} C_n^m \sum_{k_1=1}^n \dots \sum_{k_n=1}^n |W_n(-j\omega_{k_1}, \dots, -j\omega_{k_m}, j\omega_{k_{m+1}}, \dots, j\omega_{k_n})| \times \quad (3)$$

$$\times \cos \left(\left(-\sum_{l=0}^m \omega_{k_l} + \sum_{l=m+1}^n \omega_{k_l} \right) t + \arg W_n(-j\omega_{k_1}, \dots, -j\omega_{k_m}, j\omega_{k_{m+1}}, \dots, j\omega_{k_n}) \right)$$

где a – амплитуда входных гармоник, $\lfloor \]$ в индексе суммирования первой суммы означают целую часть; C_n^m – биномиальный коэффициент. Например, согласно [6], вторая парциальная составляющая отклика при входном полигармоническом сигнале

$$x(t) = \cos(\omega_1 t) + \cos(\omega_2 t) = \frac{1}{2} (\exp(j\omega_1 t) + \exp(-j\omega_1 t) + \exp(j\omega_2 t) + \exp(-j\omega_2 t)) \quad (4)$$

примет, с учетом настоящих обозначений, следующий вид:

$$y_2[x(t)] = \frac{a^2}{2} (|W_2(j\omega_1, j\omega_1)| \cos(2\omega_1 t + \arg W_2(j\omega_1, j\omega_1)) + \quad (5)$$

$$+ |W_2(j\omega_2, j\omega_2)| \cos(2\omega_2 t + \arg W_2(j\omega_2, j\omega_2)) +$$

$$+ 2|W_2(j\omega_1, j\omega_2)| \cos((\omega_1 + \omega_2)t + \arg W_2(j\omega_1, j\omega_2)) +$$

$$+ 2|W_2(j\omega_1, -j\omega_2)| \cos((\omega_1 - \omega_2)t + \arg W_2(j\omega_1, -j\omega_2)) +$$

$$+ |W_2(j\omega_1, -j\omega_1)| \cos(\arg W_2(j\omega_1, -j\omega_1)) + |W_2(j\omega_2, j\omega_2)| \cos(\arg W_2(j\omega_2, -j\omega_2)))$$

Видно, что в этой формуле последние два слагаемых фактически сливаются (на нулевой частоте) – поэтому амплитуда и фаза этой гармоники корректно не определяются. Такие сливающиеся гармоники были названы неинформативными [8]. Полагая, в первом приближении, что амплитуды всех гармоник в парциальных составляющих откликов примерно одного порядка можно сделать вывод, что доля

полезной информации в (5) равна примерно $\frac{1}{4}$. Парциальная составляющая третьего порядка для входного сигнала из трех гармоник (не приведенная здесь ввиду громоздкости) состоит уже из двадцати двух гармоник, из которых уже три неинформативны, а их доля составляет $\frac{5}{12}$. Видно, что с ростом порядка определяемой нелинейности доля полезной информации быстро падает. Было рассчитано, что уже для шестого порядка доля полезной информации будет меньше 1%.

Для компенсации такого снижения доли информативных гармоник предлагается использовать входные полигармонические сигналы, состоящие из гармоник в количестве, большим, чем порядок определяемой нелинейности. Например, парциальная составляющая второго порядка для входного сигнала, состоящего из трех гармоник, будет иметь вид:

$$\begin{aligned}
 y_2[x(t)] = & \frac{a^2}{2} \left(|W_2(j\omega_1, j\omega_1)| \cos(2\omega_1 t + \arg W_2(j\omega_1, j\omega_1)) + \right. \\
 & + |W_2(j\omega_2, j\omega_2)| \cos(2\omega_2 t + \arg W_2(j\omega_2, j\omega_2)) + |W_2(j\omega_3, j\omega_3)| \cos(2\omega_3 t + \arg W_2(j\omega_3, j\omega_3)) + \\
 & + 2|W_2(j\omega_1, j\omega_2)| \cos((\omega_1 + \omega_2)t + \arg W_2(j\omega_1, j\omega_2)) + 2|W_2(j\omega_1, -j\omega_2)| \cos((\omega_1 - \omega_2)t + \arg W_2(j\omega_1, -j\omega_2)) + \\
 & + 2|W_2(j\omega_1, j\omega_3)| \cos((\omega_1 + \omega_3)t + \arg W_2(j\omega_1, j\omega_3)) + 2|W_2(j\omega_1, -j\omega_3)| \cos((\omega_1 - \omega_3)t + \arg W_2(j\omega_1, -j\omega_3)) + \\
 & + 2|W_2(j\omega_2, j\omega_3)| \cos((\omega_2 + \omega_3)t + \arg W_2(j\omega_2, j\omega_3)) + 2|W_2(j\omega_2, -j\omega_3)| \cos((\omega_2 - \omega_3)t + \arg W_2(j\omega_2, -j\omega_3)) + \\
 & + |W_2(j\omega_1, -j\omega_1)| \cos(\arg W_2(j\omega_1, -j\omega_1)) + |W_2(j\omega_2, -j\omega_2)| \cos(\arg W_2(j\omega_2, -j\omega_2)) + \\
 & \left. + |W_2(j\omega_3, -j\omega_3)| \cos(\arg W_2(j\omega_3, -j\omega_3)) \right) \quad (6)
 \end{aligned}$$

Здесь уже только три слагаемых из 18 неинформативны, или только одна гармоника из десяти, т.е. доля неинформативных гармоник снижается по сравнению с (5) до $\frac{1}{6}$. По формуле перебора всех частот в парциальной составляющей из [8] были рассчитаны соотношения между информационными и неинформативными гармониками в парциальных составляющих разных порядков. Результаты приведены в таблицах 1 и 2.

Таблица 1. Соотношения между информативными и неинформативными гармониками в парциальной составляющей второго порядка

Число гармоник во входном сигнале	1	2	3	4	5
Количество неинформативных гармоник	0	1	1	1	1
Количество информативных гармоник	2	4	9	16	25
Доля амплитудных коэффициентов неинформативных гармоник в парциальной составляющей	0	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{8}$	$\frac{1}{10}$

Таблица 2. Соотношения между информативными и неинформативными гармониками в парциальной составляющей третьего порядка

Число гармоник во входном сигнале	2	3	4	5	6
Количество неинформативных гармоник	2	3	4	5	6
Количество информативных гармоник	6	19	44	85	146
Доля амплитудных коэффициентов неинформативных гармоник в парциальной составляющей	$\frac{9}{16}$	$\frac{5}{12}$	$\frac{21}{64}$	$\frac{27}{100}$	$\frac{99}{432}$

Из сравнения численных значений доли информативных и неинформативных гармоник в парциальных составляющих – в последних строках табл.1,2 можно сделать вывод, что для каждого в отдельности определяемого порядка нелинейности доля неинформативных гармоник в парциальной составляющей снижается с ростом числа гармоник во входном сигнале. Это свойство позволяет не только сохранять заданную эффективность идентификации, но и использовать его даже для повышения эффективности всей частотной идентификации. Например, если от идентификации передаточной функции второго порядка при использовании двухкомпонентного входного полигармонического сигнала, когда эффективность идентификации, согласно табл.1, равна $\frac{1}{4}$, перейти к идентификации передаточной функции третьего порядка с использованием шестикомпонентного входного полигармонического сигнала, когда эффективность идентификации, согласно табл.2, равна $\frac{99}{432}$, то эффективность идентификации даже возрастет. Фактически предложенный способ повышения эффективности частотной идентификации может рассматриваться как обобщение метода

многочастотной линейной идентификации на многомерный случай. Этот способ открывает перспективы использования частотной идентификации в тех случаях, когда стоимость наблюдения входных и выходных сигналов достаточно высока, как, например, в некоторых физиологических системах – как ввиду быстрой усталости системы (например, системы взаимодействия зрачка и нервов), так и ввиду долгого переходного процесса (поскольку отклик используется в установившемся режиме). Кроме этого, рассмотренный способ может быть использован там, где ранее использовалась идентификация только во временной области по причине низкой эффективности частотной идентификации.

Список литературы

1. Giannakis G. B., Serpedin E., A bibliography on nonlinear system identification and its applications in signal processing, communications and biomedical engineering, Signal Processing – EURASIP, Elsevier Science B.V., vol. 81, no. 3, pp. 533–580, march 2001.
2. Пупков К. А., Капалин В. И., Ющенко А. С. Функциональные ряды в теории нелинейных систем. – М.: Наука, 1976, 448 с.
3. Farina A., Bellini A., Armelloni E., Non-linear convolution: a new approach for the auralization of distorting systems, Preprints of the 110-th AES Convention, Amsterdam, Netherlands, 12–15 may 2001
4. Schetzen M., The Volterra and Wiener Theories of Nonlinear Systems”, Malabar, Fla., 1989.
5. Zhang Q., Suki B., Westwick D. T., Lutchen K. R., Factors Affecting Volterra Kernel Estimation: Emphasis on Lung Tissue Viscoelasticity, Annals of Biomedical Engineering, vol. 26, pp. 103–116, 1998.
6. Данилов Л. В., Матханов Л. Н., Филиппов В.С. Теория нелинейных электрических цепей. – М.: Энергоатомиздат, 1990, 454 с.
7. Зиновьев А. А. Точность представления нелинейных свойств в непараметрической идентификации нелинейных систем функциональными полиномами. // Электронное моделирование, 2004, №6, с.29–36.
8. Зиновьев А. А., Павленко В. Д. Повышение точности идентификации нелинейных систем в виде интегральных рядов. В сб.: Труды III Международной конференции "Идентификация систем и задачи управления" SICPRO'04. Москва, 28–30 января 2004 г. Институт проблем управления им. В. А. Трапезникова РАН, с.884–898.
9. Павленко В. Д., Зиновьев А. А. Амплитуды тестовых сигналов для идентификации ядер Вольтерра // Труды Одесского политехнического университета. – Одесса, 2001. – Вып. 1(13). – С.198–201.
10. Верлань А. Ф., Сизиков В. С. Интегральные уравнения: методы, алгоритмы, программы. – Киев: Наук. думка, 1986, 252 с.
11. Павленко В. Д., Зиновьев А. А. Выбор тестовых частот при идентификации нелинейной системы рядом Вольтерра. // Электронное моделирование, 2002, №1, с.17–25.
12. Данилов Л. В. Ряды Вольтерра – Пикара в теории нелинейных электрических цепей. – М.: Радио и связь, 1987, 224 с.

Формирование экстенционала базы знаний и выработка решений в интеллектуальных автоматизированных обучающих комплексах для подготовки операторов АСУТП

Иващенко М. В.,

кандидат технических наук,

с.н.с. НИИ Серпуховского военного института РВ

142207, г. Серпухов Московской области, Борисовское шоссе, д. 40, кв. 35.

E-mail: imv1@yandex.ru

Рассматривается процесс представления онтологических знаний об автоматизированных системах управления технологическими процессами, как объекте контроля и управления при создании автоматизированных обучающих комплексов для подготовки операторов АСУТП.

Приводятся процедуры формирования экстенционала базы знаний и выработки решений для таких комплексов.

Ключевые слова: интеллектуальная обучающая система, структурно-функциональное моделирование автоматизированных систем управления технологическими процессами, динамическая экспертная система, онтологические знания, семантико-продукционная сеть.

Непрерывное развитие общества сделало неизбежным совершенствование педагогических теорий и технологий. В наши дни происходит формирование новых методических подходов, основанных на использовании информационных технологий в обучении. Собственно говоря, любая педагогическая методика информационна по своей сути, т.к. связана с передачей обучаемому какой-либо информации. Характерной особенностью нынешнего периода является то, что необходимо не только передать обучаемому набор сведений, но и научить способам самостоятельного получения и обработки информации, перейти к непрерывному образованию. В настоящее время, когда происходит заметная интеллектуализация вычислительной техники, особенно остро встает проблема исследования систем "человек-машина", в частности, интеллектуальных обучающих систем и комплексов.

Для реализации образовательного процесса с требуемым уровнем качества интеллектуальные обучающие комплексы должны интегрировать знания трех типов:

- знания о педагогической технологии, которые включаются в систему на этапе ее проектирования;
- знания об изучаемой предметной области, которые включаются в уже готовую программную оболочку;
- знания о психологических особенностях обучаемого и его учебных достижениях, которые приобретаются системой в процессе работы с конкретным пользователем.

Остановимся на изучаемой предметной области, которая определяет специфические особенности разрабатываемого обучающего комплекса. В нашем случае предметной областью является процесс функционирования автоматизированных систем управления технологическими процессами (АСУТП).

Автоматизированная система управления технологическим процессом является сложным аппаратно-программным комплексом и предназначена для автоматизации управления техническими системами и технологическими процессами [3]. Для обучения операторов таких систем необходима специальная подготовка, в том числе и в системе дополнительного профессионального образования. Повышение оперативности такой подготовки возможно лишь при условии наличия в системе обучения специальных интеллектуальных автоматизированных обучающих комплексов, которые в свою очередь, по функциям схожи с самими АСУТП.

Основным средством формирования информационного ресурса интеллектуальных автоматизированных обучающих комплексов (ИАОК) является динамическая экспертная система (ЭС) [3], функционирующая в составе автоматизированной системы управления процессом обучения. При этом необходимо отметить, что главное отличие динамической ЭС от статической ЭС заключается в том, что для динамической ЭС необходимы знания о модели внешнего мира и методах взаимодействия с внешним миром [1, 8]. Это знания об АСУТП как объекте мониторинга и о методах контроля и управления состоянием ее элементов с целью поддержания их характеристик на требуемом уровне.

В основе формирования проблемно-ориентированных знаний ИАОК лежит процесс представления онтологических знаний о предметной области. Важнейшей составляющей этого процесса считается выявление сущностей, то есть обнаружение устойчивых образований признаков. К сущностям могут быть отнесены процессы и объекты реального мира. Под признаками сущностей понимают свойства и отношения. Изменение свойств и отношений во времени проявляется в процессе функционирования объектов исследований [1, 4, 6, 11]. Обнаружение сущностей и формирование перцептивной составляющей исследуемого объекта может быть осуществлено через процесс обобщения на основе перехода от конкретного объекта к классу объектов и создания на базе обобщенного объекта его отображения в виде математической модели.

При формировании отношений и свойств может быть использован объединенный метод сходства и различия с абдуктивным механизмом логического вывода [2, 14].

В общем случае для получения математической модели АСУТП как объекта мониторинга при известных математических моделях отдельных элементов системы и связей между ними могут быть использованы формальные методы. Чтобы представить формализованную структурно-функциональную модель АСУТП необходимо, на основе анализа эволюции во времени информационных параметров ее элементов, конфигурации самой системы, ввести базис переменных, отражающих множество параметров элементов системы и установить отношения между ними [5, 7].

Структурно-функциональное моделирование АСУТП как объекта мониторинга представляет собой часть информационного процесса по формированию онтологических знаний в ИАОК. При этом под структурным моделированием АСУТП понимается процесс представления в виде определенной структуры множества элементов системы и связей между ними. Функциональное моделирование

представляет собой математическое описание параметров и характеристик АСУТП и ее элементов [10, 12, 13].

Определенную таким образом структурно-функциональную модель АСУТП в обобщенном виде можно представить следующим соотношением

$$M_{\text{CFACU}} = \langle K_{\text{ACU}}, R_{\text{Э}}, Z_{\text{ACU}}, П_{\text{Э}} \rangle, \quad (1)$$

где K_{ACU} - конфигурация АСУТП;

$R_{\text{Э}}$ - множество информационных параметров элементов АСУТП;

Z_{ACU} - характеристики АСУТП;

$П_{\text{Э}}$ - множество режимов функционирования элементов АСУТП.

При формировании проблемно-ориентированных знаний в ИАОК важное значение имеет процесс приобретения и накопления знаний экспертов. Суть его заключается в формировании экстенционала знаний на основе модели семантико-продукционной сети [3]. При этом под экстенционалом понимается набор конкретных знаний, заданных в декларативной форме [13].

Учитывая особенности построения и функционирования АСУТП и следуя выводам, сделанным в ходе анализа процессов представления и структуризации знаний в динамических ЭС, процесс формирования экстенционала знаний об информационных параметрах элементов, входящих в состав АСУТП, основывающихся на семантико-продукционной сети (СПС), следует организовывать в два этапа. На первом этапе формируется часть экстенционала СПС, характеризующая особенности структурных отношений между элементами в АСУТП. На втором этапе формируется часть экстенционала, характеризующая функциональные особенности самих элементов. При этом экстенционал СПС должен быть представлен фиксированными понятиями свойств и взаимосвязей, соответствующих понятию реляции, лежащей в основе реляционной модели Е.Ф. Кодда. Реляция – это отношение в реляционной алгебре, в основе которой лежит математический аппарат для манипулирования знаниями, включающих операции объединения, пересечения, вычитания, декартова произведения, селекции, проекции, соединения и деления отношений [4, 13].

Отображение СПС в виде набора реляций производится путем представления структуры понятий, где отношения, выступающие в роли указателей на соответствующие свойства понятий, играют роль атрибутов, отражающих содержание, кортежей реляций. Экстенционалы понятий и их свойств практически представляют собой подмножества декартовых произведений определенных наборов, которые и составляют соответствующие реляции. Каждому примеру понятия в реляции соответствует определенный кортеж. Атрибут, определяющий идентификатор понятия, является ключевым атрибутом. В теории информационных систем под атрибутом понимается признак, описатель данных и знаний, содержащий одну из основных характеристик знания: имя, тип, количество, форму представления, систему исчисления.

На первом этапе формирования экстенционала знаний производится заполнение следующего набора реляций:

$$R_{\text{СПС}} : \{ \text{TPART}, \text{TPARD}, \text{TPK}, \text{TOP}, \text{TAGR}, \text{BAR} \}, \quad (2)$$

где TPARD - определяет экстенционал понятия информационного параметра дискретного типа;

TPART - определяет экстенционал понятия информационного параметра непрерывного типа;

TPK - определяет экстенционал понятия качественного параметра;

TOP - определяет экстенционал понятия режима работы;

TAGR - определяет экстенционал понятия АСУТП;

BAR - определяет экстенционал понятия элемента АСУТП.

Структура каждой реляции практически идентична структуре соответствующего понятия, за исключением того, что значения свойств, описывающих фактическое состояние информационных параметров, будут храниться в отдельных информационных массивах, включенных в базу знаний динамической ЭС, отображающей и накапливающей информацию, определяющую фактическое состояние АСУТП за фиксированный интервал времени.

Таким образом, представленные выше реляции имеют следующие схемы:

TPART: < ID, LOC, r1, r2, MAX, NOM, MIN >;

TPARD: < ID, LOC, r1, r2, SVAL >;

TPK: < ID, LOC, IS, LIM, KONTS >;

TOP: < ID, AGR, PROC, FOL, TPAR, TPKS >;

TAGR: < SD, AGR, PART, PROC, FOL, TPAR, TPKS >;

BAR: < SD, AGR, PART, PROC, FOL, TPAR, TPKS >;

где ID, LOC - наименование атрибутов, соответствующие идентификаторам отношений СПС, указывающих на определенные свойства и взаимосвязи понятий.

Следует отметить, что значение атрибута KONTS реляции TPK, определяющего множество КПК, способствующих к конфликтному отклонению соответствующего информационного параметра, будет

формироваться на втором этапе. На рис. 1 представлена обобщенная процедура формирования структурных отношений в СПС.

При формировании атрибутов реляций информационных параметров АСУТП в первую очередь загружаются реляции, описывающие экстенсионалы понятий TPART, TPARD и TPK. При этом, формируется содержание атрибутов ID, r1, r2, SVAL, IS, LIM, за исключением атрибута LOC, загрузка которого производится после заполнения реляций, определяющих структурную иерархию СПС.

Далее следует загрузка реляций TOP, TAGR, BAR, которая производится путем последовательного продвижения по оси обобщения абстрактно-топологического пространства, определяемого СПС. В ходе последовательного прохождения оси обобщения производится циклическое прохождение временной оси для каждого уровня структурной иерархии, т.е. при заполнении каждой из реляций. Продвижение по оси агрегации производится для каждого элемента (кортежа) заполняемой реляции, поскольку агрегируемые составляющие фиксируются для каждого примера экстенсионала понятий.

Отметим, что при начальной загрузке реляций TOP, TAGR, BAR атрибуты AGR не заполняются.

Последним шагом загрузки структурной составляющей экстенсионала СПС является заполнение атрибутов LOC в реляциях, характеризующих информационные параметры, и атрибутов AGR в реляциях, отображающих структурную иерархию СПС.

В ходе загрузки и при ее окончании производится синтаксический - семантический контроль введенной информации. В первую очередь контролируется уникальность формируемых идентификаторов (атрибут ID), производится сравнительный анализ нормативных значений информационных параметров (атрибуты MAX, NOM, MIN), контролируются вводимые значения признаков измеряемости и управляемости.

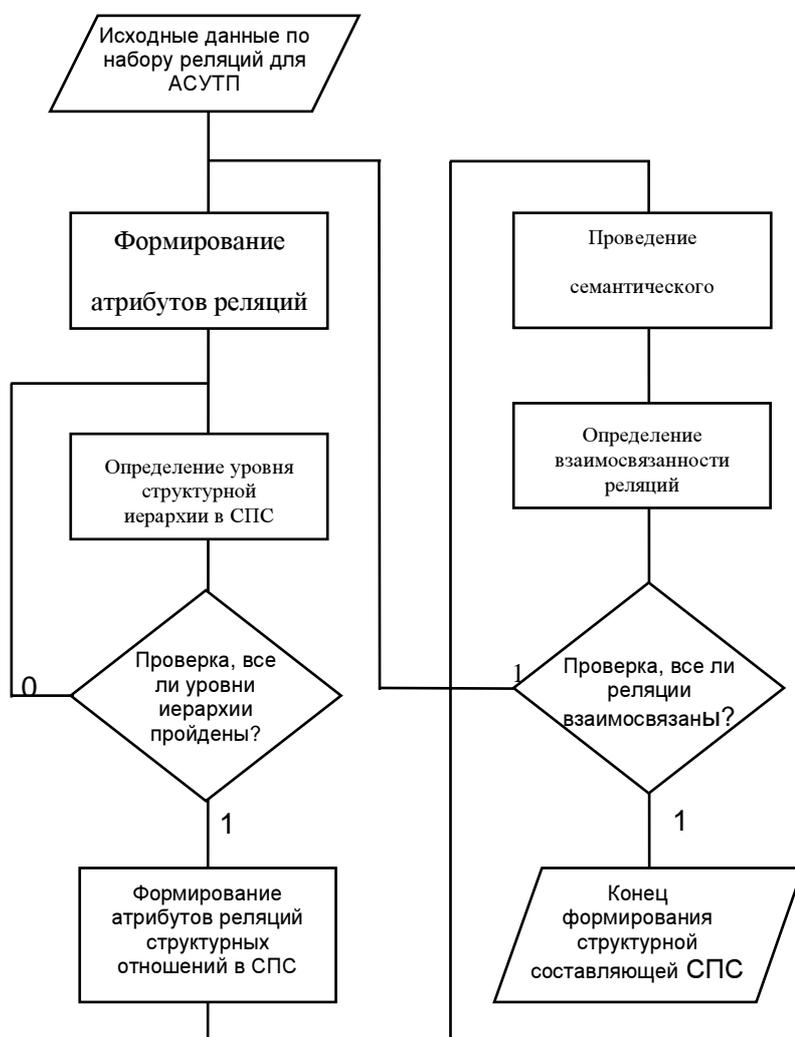


Рис. 1. Обобщенная структура процедуры формирования структурных отношений в семантико-продукционной сети.

В конце загрузки производится проверка сформированной части экстенсионала на взаимосвязанность. При этом не вошедшие в общую сеть составляющие реляций (кортежи, в которых

атрибуты LOC или AGR остаются пустыми) выводятся на печать. Кроме этого фиксируется противоречивость значений атрибутов PROC и FOL в кортежах, определяющих составляющие соответствующего уровня структурной иерархии, расположенные в непосредственной близости на пространственно-временной оси. После соответствующей коррекции содержания реляций формирование структурной составляющей экстенционала СПС считается законченным.

Этап формирования части экстенционала СПС, характеризующей функциональные особенности АСУТП, является более трудоемким, поскольку требует проведения опроса экспертов.

Накопление эвристических экспертных знаний следует организовывать, основываясь на априорно построенной части экстенционала СПС, отображающей структурные особенности АСУТП.

На втором этапе формирования экстенционала знаний об информационных параметрах элементов АСУТП производится заполнение реляции PR и KONT, отражающих понятия коэффициентов воздействия K1 и присутствия K2 и характеризующих возможность появления конфликтности информационных параметров:

PR: < ID, TRF, P, VAL, K1 >;

KONT: < ID, PRS, K2 >, (3)

где ID, TRF, P, VAL – наименование атрибутов, определяющих структуру предикатов, отображаемых примерами понятий;

ID, PRS - наименование атрибутов, указывающих на определенные свойства и взаимосвязи понятий.

В процессе опроса экспертов необходимо зафиксировать определенные причинно-следственные отношения между информационными и качественными параметрами АСУТП, выбрать определенный элемент из состава АСУТП и задать режим его работы.

Для различных условий функционирования элементов АСУТП необходимо сформировать набор контекстов порождения конфликтности (КПК). Формирование элементарных конфликтностей включает процесс построения antecedentной части продукций, представляющих предикатные условия в виде пороговых значений информационных параметров Γ_{Π} . В случае информационного параметра дискретного типа значение $\Gamma_{\Pi d}$ определяется путем последовательного перебора значений множества, получаемого в процессе операции:

$$r_{\Pi d} = R_{SVAL} [TPARD]. \quad (4)$$

Поскольку конкретным состоянием информационного параметра непрерывного типа $\Gamma_{\Pi n}$ считается превышение или нормативного максимального значения или понижение его фактического значения ниже нормативного минимума, то при формировании элементарных конфликтностей выполняется операция:

$$r_{\Pi n} = R_{\max} [TPART] \vee R_{\min} [TPART]. \quad (5)$$

С целью сопоставления элементарной конфликтности с качественными параметрами определяется их множество, принадлежащее к рассматриваемому режиму работы элемента АСУТП:

$$q_k = Q_{TRK} [TOP]. \quad (6)$$

Соответственно выбор информационных параметров осуществляется при выполнении операций:

$$\begin{aligned} r_{и1} &= R_{TRK} [TAGR]; \\ r_{и2} &= R_{TRK} [BAR]. \end{aligned} \quad (7)$$

В процессе формирования экстенционала знаний конечной целью является выработка рекомендации для разрешения конфликтной ситуации. Для этого необходимо разработать процедуры обработки причинно-следственных зависимостей, обладающих ситуативностью и отсутствием четкой детерминированности [6, 9].

Конфликтная ситуация S_{ACU} в процессе функционирования АКР может быть охарактеризована контекстом конфликтной ситуации КО, который включает набор качественных параметров Q_d АСУТП, определяющих конфликтность ситуации. В общем случае конфликтную ситуацию определяет некоторый набор конфликтностей качественных параметров, каждая из которых участвует в формировании контекста конфликтной ситуации КО в АСУТП

$$KO = \bigcup_{j=1}^k KO_j, \quad (8)$$

где KO_j - набор КПК, определяющий конфликтность качественного параметра q_j ; k – количество параметров наблюдаемой ситуации S_{ACU} , находящихся в конфликтном состоянии.

На рис. 2 представлена обобщенная структура процедуры формирования функциональных отношений в СПС, основывающаяся на операциях выделения контекста конфликтной ситуации,

разрешения каждой составляющей конфликтности для формирования рекомендаций по обеспечению заданных требований к качеству функционирования АСУТП.

В качестве исходных данных для процедуры формирования функциональных отношений используются: множество информационных параметров R_{\ominus} , множество режимов функционирования элементов АСУТП Π_{\ominus} , качественные характеристики АСУТП Q_f и качественные параметры ее элементов q_d , контролируемые характеристики АСУТП N_1 и контролируемые параметры ее элементов n_b .

Определение конфликтности ситуации S_{ACU} заключается в определении конфликтного состояния качественного параметра q_i , превышающего заданное пороговое значение q_{i0} . Значение q_i задается в ходе загрузки исходных данных на основе заданного норматива качественного параметра (содержание атрибута LIM ТРК на семантико-продукционной сети)

$$q_i = Q_{LIM} [ТРК]. \quad (9)$$

Фиксация набора КПК основывается на использовании реляции KONT с определением коэффициента присутствия K_2 соответствующего КПК

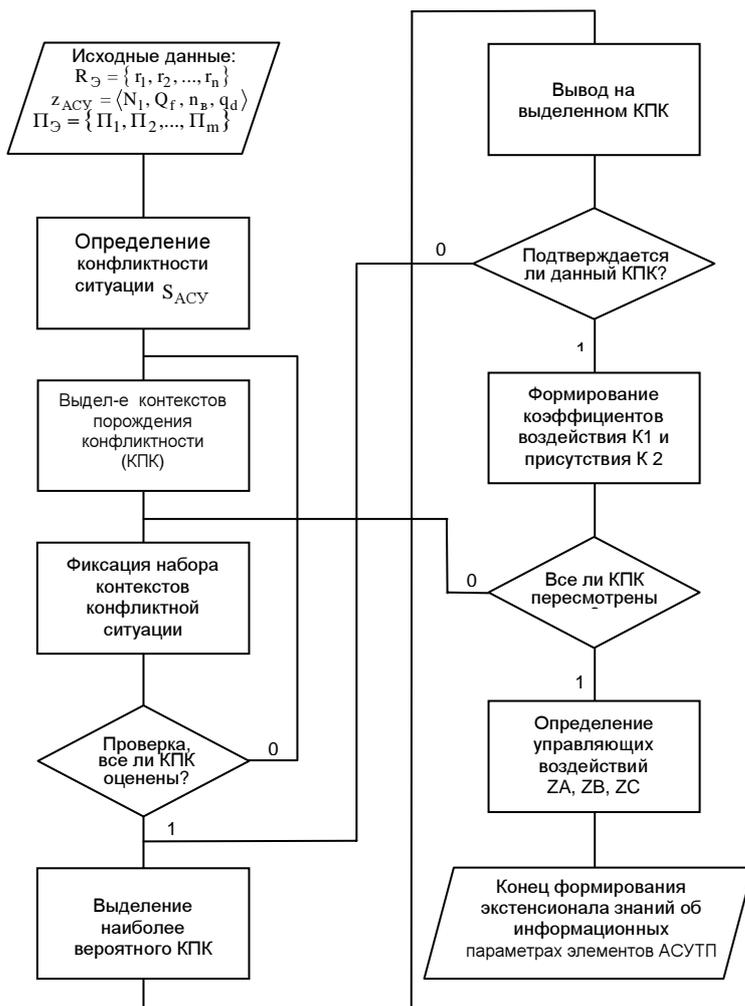
$$K_{2_1} = KO_{PRS}(KONT),$$

$$K_{2_2} = KO_{PR}(KONT). \quad (10)$$

Выделение наиболее вероятного КПК производится путем оценки коэффициента присутствия K_2 заданного КПК. Данная операция проводится последовательно путем выбора первого из представленных КПК с последующим раскрытием содержания выбранного КПК. Далее, на основе проверки предиката PR, представляющего элементарную составляющую КПК, проводится операция по проверке фактического значения анализируемого параметра. После каждого обработанного предиката производится оценка сопричастности данного КПК в появлении конфликтности качественного параметра. Сопричастность контекста порождения конфликтности оценивается на основе анализа множества коэффициентов воздействия

$$K_1 = \{ K_{1_1}, K_{1_2}, \dots, K_{1_v} \}, \quad (11)$$

где $K_{1_f}, f = \overline{1, v}$ соответствует значениям коэффициента воздействия неудовлетворенных признаков КПК.



При формировании значений коэффициентов K_1 и K_2 производится трансформация лингвистических экспертных оценок в соответствующие числовые значения на интервале $[0, 1]$. В роли экспертных оценок выступает множество лингвистических значений: никогда (н), очень редко (ор), редко (р), часто (ч), очень часто (оч), всегда (в)

$$M_0 = \{ н, ор, р, ч, оч, в \}. \quad (12)$$

В зависимости от значений оценок могут быть применены такие правила трансформации, которые позволяют исключить несопричастные составляющие КПК.

Разрешение конфликтных ситуаций, возникающих в процессе функционирования АСУТП, производится путем формирования управляющих воздействий на ее элементы с целью коррекции информационных параметров, включенных в актуализированную часть фиксированного КПК.

Процесс разрешения конфликтной ситуации основывается на формировании вспомогательных реляций PR1, TPK1, KONT1, имеющих следующую структуру:

$$\begin{aligned} TPK1 &: \langle ZA, ID, KONTS \rangle, \\ KONT1 &: \langle ZB, ID, PRS, K2 \rangle, \\ PR1 &: \langle ZC, ID, TPF, r1, P, VAL, r2, K1 \rangle. \end{aligned} \quad (13)$$

Атрибут ZA приобретает значение из множества положительных и отрицательных значений, соответственно характеризующий то, что КПК либо определен, либо не определен. Атрибут ZB приобретает значение в диапазоне $[0; 1]$. Он указывает на сопричастность данного КПК к появлению конфликтности. Атрибут ZC приобретает численные значения с определенным знаком. При положительном значении величину параметра следует повысить, при отрицательном – понизить. Фактические значения информационных параметров $r1$ и $r2$ определяются базовыми атрибутами TPF и VAL.

Таким образом, представленные процедуры формирования экстенционала знаний об информационных параметрах элементов АСУТП характеризуют структурные и функциональные отношения в семантико-продукционной сети и могут быть положены в основу управления информационными параметрами элементов АСУТП в процессе мониторинга при создании ИАОК для подготовки операторов АСУТП.

Список литературы

1. Ерофеев А. А., Поляков А. О. Интеллектуальные системы управления. СПб.: Изд-во СПб ГТУ, 1999. - 263 с.
2. Ивашко В. Г., Финн В. В. Экспертные системы и некоторые проблемы их интеллектуализации / Семиотика и информатика. - М.: ВИНТИ, 1987. - № 27. - С. 25-61.
3. Интеллектуальные адаптивные системы и комплексы в связи и управлении: Монография. / Злобин В. И., Иващенко М. В., Иванова Г. В. - МО РФ, 2005. - 276 с.
4. Интеллектуальные системы // Труды IV Международного симпозиума «Интелс - 2000». – М.: МГТУ им. Н. Э. Баумана, 2000. - 296 с.
5. Йордан Э., Артила К. Структурные модели в объектно-ориентированном анализе и проектировании. - М.: Лори, 1999. - 264 с.
6. Каргин А. А., Петренко Т. П. Модели динамических ситуационных интеллектуальных машин // Искусственный интеллект. - 2000.- № 1. - С. 59-68
7. Куракин А. П. Обработка сигналов обнаружения при непрерывном мониторинге // Труды LVII научной сессии РНТО РЭС им. А.С. Попова, 2002. – С. 148-151.
8. Попов Э. В. Динамические интеллектуальные системы в управлении и моделировании. – М.: МИФИ, 1996. – 102 с.
9. Построение экспертных систем: Пер. с англ. / Под ред. Ф. Хайеса-Рота. - М.: Мир, 1987. - 441 с.
10. Поспелов Г. С. Искусственный интеллект - основа информационной технологии. – М.: Наука, 1988. – 280 с.
11. Пупков К. А. Интеллектуальные системы: проблемы теории и практики // Изв. вузов. Приборостроение, 1994. Т. 34. № 9. С. 5-7.
12. Системы автоматизированного проектирования. В девяти книгах / Под ред. И. П. Норенкова. – Мн.: Выш. шк., 1988.

13. Толковый словарь по искусственному интеллекту. / Авторы-составители А. Н. Аверкин и др. - М.: Радио и связь, 1992. - 256 с.
14. Zadeh L. Fuzzy Logic, neural networks and soft computing // CACM, Vol.37, No 3, 1994.

Объектно-структурный подход в проектировании программных систем

Исмагилов Н. А.,

*доцент кафедры прикладной математики
Казанского государственного университета,
420008, Казань, ул. Кремлевская, 18
E-mail: Nail.Ismagilov@ksu.ru*

В данной статье описывается подход в проектировании программных систем, основанный на представлении программ в виде определенной структуры, объектами которой являются специальные функциональные компоненты. Подход также основан на принципе декомпозиции программной системы на множество элементарных программ и инвариантной управляющей части с возможностью управления ее выполнением с применением дескриптора – описателя сценария. Разделение управляющей части и описателя выполнения программы позволяет существенно сократить временные и материальные затраты на создание программных систем.

Ключевые слова: программная система, программа сильной структуры, множество элементарных программ, дескриптор, описатель сценария выполнения, UML – унифицированный язык моделирования, модель требований, модель реализации, MDA - model driven architect, компьютерные обучающие средства, обучающие программы, инструментарии для создания систем тестирования и компьютерных средств обучения.

Сложность создания программных систем

Всем известен тот факт, что применение программных систем требует немалых капиталовложений в индустрию разработки. По статистике всего 26% разработок ПС доводятся до успешного завершения, т.е. доводится до логического завершения, внедряется и успешно эксплуатируется одна из четырех программных систем, созданная в рамках первоначально определенных временных и бюджетных ресурсов.

Структура затрат на создание программного обеспечения (ПО) зависит от процессов, используемых при разработке ПО, а также от типа разрабатываемого программного продукта [1]. Существующие современные методологии создания программных систем (ПС), в которых используются различные модели жизненного цикла ПО, позволяют снижать затраты на создание ПС, сокращать сроки разработки, добиваться определенного качества программного обеспечения. Однако постоянно происходит совершенствование процессов разработки ПС. Примерами такого развития являются методологии системного анализа и проектирования, с использованием языка UML (универсального языка моделирования). В последнее десятилетие совершенствовался язык UML, в 1997 году была опубликована первая версия спецификации языка. В 1997 г. международный консорциум OMG (Object Management Group) объявил UML стандартным языком объектно-ориентированного моделирования. В 2004 году разрабатывается спецификация языка UML 2.0. К настоящему времени фирмы, занимающиеся разработкой программных инструментариев, создали множество программных систем, в которых уже реализован язык UML спецификации 2.0. Совершенствование методологии, основанной на языке UML, явно прослеживается.

Применение средств визуального моделирования таких, как язык UML, по мнению специалистов, перспективно. Следует отметить, что с другой стороны использование языка UML при создании больших, серьезных программных систем вызывает определенную сложность.

Создание программной системы начинается с исследования бизнес-процесса (на предприятии, в организации и т.п.), который должен быть полностью или частично автоматизирован. Результаты исследования в виде описания бизнес-процессов влияют на дальнейшую систему действий по созданию ПС. Описание бизнес-процесса связано с постановкой задачи, с требованиями к разрабатываемой системе, с моделью требований. Полнота требований к разрабатываемой программной системе влияет на качество решений по автоматизации бизнес-процесса.

Статистика показывает, что наиболее часто встречающиеся серьезные проблемы при разработке систем связаны с неполными требованиями и спецификациями проектов, а также с управлением изменениями требований клиента. Исследования групп Стендиша и организации ESPITТ показывают, что проблемы требований, судя по всему, превосходят остальные в плане риска неполадок, которые они вызывают при разработке систем [2]. Ошибки требований занимают первое место среди оставшихся недоработок и составляют примерно одну треть всех не устранённых дефектов.

По требованиям к программной системы формируется модель спецификации. Результаты действий на этапе проектирования влияют на качество реализации, тестирования и отладки разрабатываемой программной системы. Полнота множества требований позволяет надеяться, что разрабатываемая программная система в итоге будет решать задачи конечных пользователей, выполнять в виде определенной совокупности последовательностей элементарных действий все требования пользователя и тем самым двигаться к цели – к решению задач конечных пользователей.

Аналитическое обоснование объектно-структурного подхода

В данной статье описывается объектно-структурный подход в проектировании программных систем, основанный на представлении программной системы в виде совокупности структурных программных компонентов – объектов. Программная система – контейнер, в котором используются программные объекты.

При построении программных систем с использованием данного подхода используется определенная программная структура.

Программную систему можно представить в виде $P = \langle D, V, C, A \rangle$,

(для простоты будем использовать термин программа)

где P - программа,

D – объект дескриптор - описатель сценария выполнения, состоит из записей дескриптора;

V - программа выбора, выделения и передачи D_i ($i=1,2,..n$);

C - объект множественного ветвления, используется определения типов программ, реализован с помощью CASE оператора;

A – множество программ, называемых элементарными.

Ниже приведем аналитическое обоснование применимости программы сильной структуры. С помощью теории программной меры программ [3] можно получить оценки метрических характеристик, которые показывают наименьшие значения оценки трудоемкости создания программных систем с использованием данного подхода.

Традиционной формой представления алгоритмов функционирования программ является блок-схема, которую можно рассматривать как ориентированный помеченный граф, указывающий порядок выполнения операторов программы. Каждый оператор представляется помеченной вершиной (помеченным узлом) графа, а каждое возможное направление передачи управления, в общем случае, - помеченным ребром (помеченной линией).

В зависимости от количества входящих и исходящих линий узлы классифицируются на три типа - функциональные, предикатные узлы и узлы слияния.

Функциональным называется узел, имеющий только по одной входной и выходной линии (по одному входу и выходу) (рис. 1)

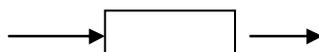


Рис. 1. Функциональный узел

Предикатным называется узел, имеющий только один вход и два выхода (рис. 2)

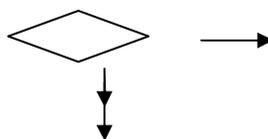


Рис. 2. Предикатный узел

Узлом слияния называется узел только с двумя входами и одним выходом (рис.3).

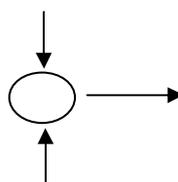


Рис. 3. Узел слияния

Разметка функционального узла производится именем оператора - функции, исполнение которого обозначает этот узел. Предикатный узел размечается предикатом обозначаемого условного оператора. Из множества линий графа размечаются только исходящие из предикатных узлов: указания "И" обозначают истинные, а "Л" - ложные значения соответствующих предикатов.

Неразмеченный граф будем называть структурой алгоритма или, просто, структурой; а размеченный граф с последующим его кодированием на языке программирования - программой.

В зависимости от количества и типов узлов, а также порядка исполнения обозначаемых ими операторов, структуры и программы классифицируются на элементарные и составные.

Элементарными являются следующие структуры:

1) <функция> - состоит только из одного функционального узла (рис. 4)

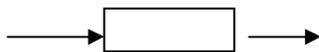


Рис. 4. Элементарная структура <функция>

2) <последовательность> - представляет собой цепочку из двух последовательно соединенных функциональных узлов (рис. 5)

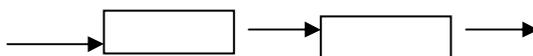


Рис.5. Элементарная структура <последовательность>

3) ветвление (обозначается сочетанием <if_then_else>) - состоит из предикатного узла, вход в который является входом структуры; двух функциональных узлов, входами которых являются выходы предикатного узла; узла слияния, входами которого являются выходы функциональных узлов, а выходом - выход структуры (рис. 6)

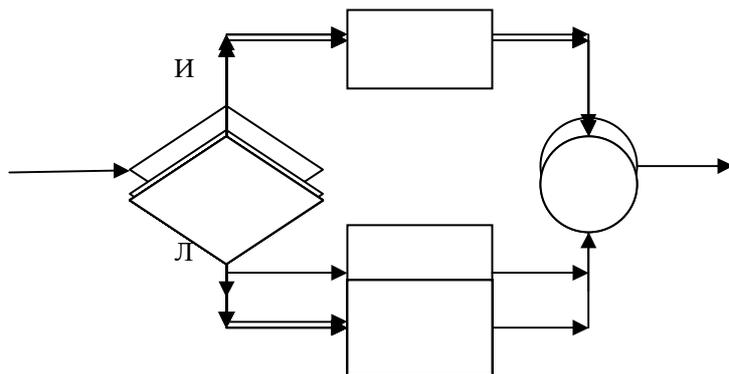


Рис. 6. Элементарная структура <if_then_else>

4) частичное ветвление (обозначается <if_then>) - в отличие от структуры ветвления содержит только один функциональный узел, входом которого является выход предикатного узла, помеченный указанием "И", (рис. 7)

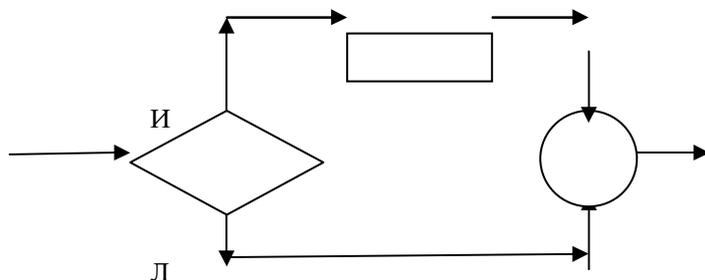


Рис.7. Элементарная структура <if_then>

5) циклическое выполнение с предусловием (обозначается <while_do>) - состоит из узла слияния, один из входов которого является входом структуры; предикатного узла, входом которого является выход узла слияния, выход, помеченный указанием "Л", является выходом структуры; и функционального узла, входом которого является выход предикатного узла, помеченный указанием "И", а выход является вторым входом узла слияния (рис. 8)

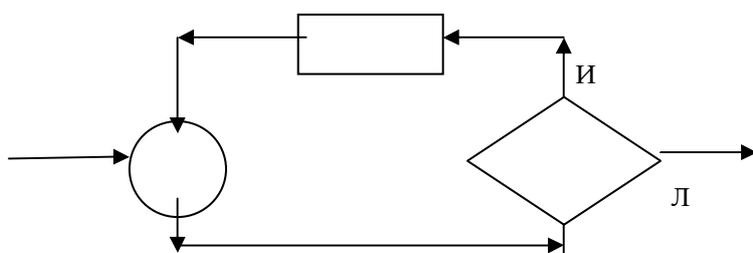


Рис. 8. Элементарная структура <while_do>

б) циклическое выполнение с постусловием (обозначается <do_until>) - состоит из узла слияния, один из входов которого является входом структуры; функционального узла, входом которого является выход узла слияния; и предикатного узла, входом которого является выход функционального узла, выход помеченный указанием "И", является вторым входом узла слияния, а выход, помеченный указанием "Л", является выходом структуры (рис. 9)

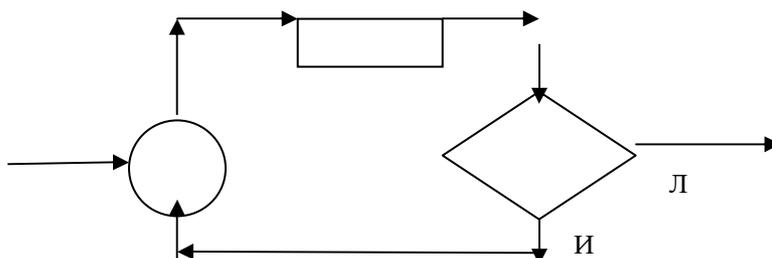


Рис. 9. Элементарная структура <do_until>

7) циклическое выполнение с пред- и постусловием (обозначается <do_while_do>) - состоит из узла слияния, двух функциональных узлов и одного предикатного узла. Входом структуры является один из входов узла слияния, а выходом - выход предикатного узла, помеченный указанием "Л". Входом одного из функциональных узлов является выход узла слияния, а выходом - вход предикатного узла. Входом второго функционального узла является выход предикатного узла, помеченный указанием "И", а выходом - второй вход узла слияния (рис. 10).

Реализованные на языке программирования размеченные элементарные структуры являются элементарными программами.

Элементарные программы в общем случае содержат два типа компонент - элементарные подпрограммы и исполняемые подпрограммы. Элементарными подпрограммами являются реализации различных предикатных узлов и узлов слияния.

Исполняемые подпрограммы являются реализации размеченных функциональных узлов. Таким образом, элементарная программа типа:

- 1) <функция> состоит из одной исполняемой подпрограммы;
- 2) <последовательность> из двух исполняемых подпрограмм;
- 3) <if_then_else> состоит из элементарной подпрограммы и двух исполняемых подпрограмм;
- 4) <if_then> состоит из элементарной подпрограммы и одной исполняемой подпрограммы;

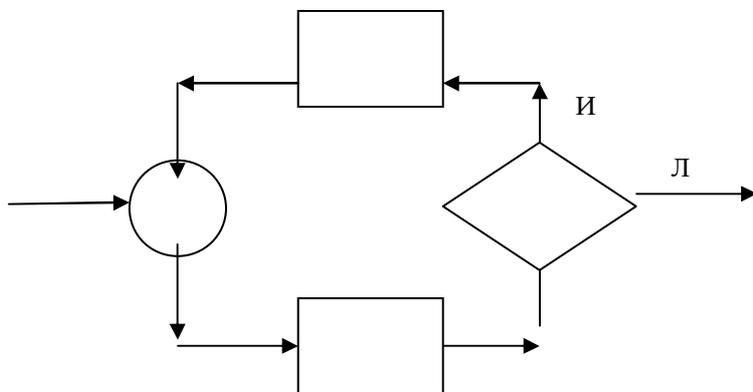


Рис. 10. Элементарная структура <do_while_do>

- 5) <while_do> состоит из элементарной подпрограммы и одной исполняемой подпрограммы;
- б) <do_until> состоит из элементарной подпрограммы и одной исполняемой подпрограммы;

7) `<do_while_do>` состоит из элементарной подпрограммы и двух исполняемых подпрограмм.

Пусть A_i - элементарная программа i -го типа ($i=0,1, \dots,6$);

$i=0$ - для элементарной программы типа `<функция>`;

$i=1$ - для элементарной программы типа `<последовательность>`;

$i=2$ - для элементарной программы типа `<if_then_else>`;

$i=3$ - для элементарной программы типа `<if_then>`;

$i=4$ - для элементарной программы типа `<while_do>`;

$i=5$ - для элементарной программы типа `<do_intil>`;

$i=6$ - для элементарной программы типа `<do_while_do>`.

F_i - элементарная подпрограмма элементарной программы i -го типа;

E_{is} - s -ая элементарная подпрограмма элементарной программы i -го типа ($s=0,1,2$), при $s=0$

E_{i0} указывает на отсутствие исполняемой подпрограммы в элементарной программе i -го типа;

тогда $A_i = (F_i; E_{i0}, E_{i1}, E_{i2})$.

Составной структурой называется структура, получаемая заменой произвольного функционального узла элементарной структуры элементарной структурой. Если в составной структуре заменить произвольный функциональный узел элементарной либо составной структурой, то получится новая составная структура и т.д. Размеченная составная структура, закодированная на языке программирования является составной программой.

Все множество элементарных структур позволяет создавать всевозможные составные программы. Каждое конкретное подмножество элементарных структур позволяет создавать конкретный класс составных программ. Так подмножество `{<последовательность>, <if_then_else>}` обеспечивает формирование класса ациклических программ, а подмножество `{<if_then_else>, <while_do>}` приводит к классу программ, при выполнении которых исполняется не более одного оператора функции.

Подмножество элементарных структур называется фиксированным базисным множеством элементарных структур.

Множество элементарных программ, формируемых на основе фиксированного базисного множества элементарных структур, называется фиксированным множеством элементарных программ.

Составная программа, сформированная на основе фиксированного базисного множества элементарных программ, называется структурированной программой.

В соответствии с теоремой о структурировании программ [4] любая программа, имеющая только по одному входу и выходу, и через каждый узел структуры которой проходит путь от входа к выходу, функционально - эквивалентна структурированной программе, составленной из элементов базисного множества `{<последовательность>, <if_then_else>, <while_do>}` и размеченной функциями и предикатами исходной программы.

Программа, имеющая только по одному входу и выходу, и через каждый узел структуры которой проходит путь от входа к выходу, называется простой программой.

Таким образом, любая простая программа может быть представлена в виде совокупности трех видов элементарных программ `<последовательность>`, `<if_then_else>`, `<while_do>`, указываемых, в общем случае, в произвольном порядке и количественном составе.

Чем сложнее алгоритм реализуется программой, тем больше в ней элементарных программ одного и того же вида и различных видов. Однако, можно сформировать составную структуру, в которой каждый вид элементарных структур указывается лишь один раз, а разметка структуры производится динамически после выполнения очередной элементарной программы. Выбор информации для разметки производится из информационного массива, который назовем массивом дескрипторов.

Такая структура называется сильной структурой и представлена на рис. 11.

Компонентом сильной структуры является составная структура `<case>`, которая является широкоизвестным сокращенным представлением последовательности структур ветвления. В структуре `<case>` вводится обобщенное представление последовательности предикатных узлов - узел выбора. Узел выбора имеет один вход, являющийся входом первого предикатного узла последовательности, и столько выходов, сколько предикатных узлов в последовательности. Каждый выход размечается номером выбранного варианта, а узел выбора размечается многоместным предикатом и обозначается `<case_test>`.

Входом сильной структуры является вход функционального узла, который размечается именем оператора-функции выбора очередного дескриптора из массива дескрипторов и разметки соответствующей элементарной структуры.

Динамически размечаемая сильная структура, закодированная на языке программирования, называется P - программой сильной структуры (ПСС).

P - программа сильной структуры (ПСС) может быть представлена в виде:

$$P = \langle D, V, C, A \rangle,$$

где $D = (D_1, D_2, \dots, D_n)$, V-программа выбора, выделения и передачи D_i ($i=1,2,\dots,n$); C - case оператор; $A = \{(F_j, E_j)\}$ ($j=1,2,\dots,8$); F_j -прототип функциональной подпрограммы

A_j ; E_j -прототип исполняемой подпрограммы A_j , или $P = \langle V, C, F, E \mid D \rangle$, здесь $\langle V, C, F, E \rangle$ - унифицированная программная часть ПСС; $\langle D \rangle$ - параметр настройки.

Каждый дескриптор D_i содержит информацию, характеризующую i -ую элементарную программу в виде восьмерки: $D_i = \{ N_i, S_i, T_i, R_i, P_i, U_{1i}, U_{2i}, G_i \}$, ($i=1, 2, \dots, n$).

Здесь N_i - имя i -ой элементарной программы; S_i -имя первой исполняемой функциональной подпрограммы i -ой программы; T_i -числовой код типа i -ой программы (1-<функция>, 2-<последовательность>, 3-<if_then_else>, ..., 8-<goto>); R_i - код завершения i -ой программы; P_i -предикат предикатного узла i -ой программы; U_{1i} , U_{2i} -имена подпрограмм, исполняемых при условии ИСТИНА, ЛОЖЬ предикатного узла i -ой программы; G_i -имя программы, выполняемой после i -ой программы; n - общее количество элементарных программ в программе.

Таким образом, информации в дескрипторе достаточно для разметки элементарных структур всех типов, т.е. для исполнения любой элементарной программы, а значит, массив дескрипторов позволяет описывать исполнение программ сильной структуры.

Элементы в информационном массиве дескрипторов D упорядочиваются в последовательности вызовов элементарных программ. Каждая элементарная программа проектируется так, что при выходе из нее формируется код завершения R_i ($i=1, n$), который используется, как правило, при вычислении значений предикатов P_j ($j=1, n$) последующих программ.

Первой исполняемой элементарной программой в программе сильной структуры является программа, информация о которой располагается в первом дескрипторе массива D . Для определения каждой последующей выполняемой элементарной программы используются следующие схемы продвижения:

1) явное указание в дескрипторе выполняемой элементарной программы имени очередной элементарной программы (параметр G_i , $i=1, n$);

2) при умолчании явного указания выполняется элементарная программа, информация о которой располагается в следующем по порядку элементе массива дескрипторов.

Завершение выполнения программы сильной структуры осуществляется после выполнения элементарной программы с индексом $i=n$.

Взаимодействие и обобщенное содержание компонентов V и C программ сильной структуры можно представить следующим метаописанием:

```
while (1)
  < V - выбор, выделение и передача значений дескриптора
    очередной элементарной программы >
  case 1:
    <выполнение очередной элементарной программы типа 1>
  case 2:
    <выполнение очередной элементарной программы типа 2>
  case 3:
    <выполнение очередной элементарной программы типа 3>
  ...
  case 8:
    <выполнение очередной элементарной программы типа 8>
endwhile
```

Таким образом, программа сильной структуры - это параметрически настраиваемая программа, управление выполнением которой происходит путем выбора, выделения передачи значений дескриптора очередной элементарной программы, с последующим исполнением этой программы.

Программные системы могут быть организованы по такой же схеме, с использованием аналогичной программной структуры.

Объектами являются:

- D - дескриптор – описатель сценария выполнения элементарных программ. Реализация дескриптора обычно происходит в виде таблицы.

- V – объект – программа выбора, выделения и передачи очередной записи дескриптора.

- C – объект множественного ветвления. Реализация в управляющей инвариантной части в виде оператора CASE.

- A – множество элементарных программ. Реализация в виде DLL библиотек. DLL файл является хранилищем для элементарных программ всех типов.

Данный подход позволяет на основе разделения в программной системе инвариантной управляющей части и описательной части – дескриптора – сценария выполнения элементарных программ получить программную конструкцию, в которой в качестве элементарных программ могут использоваться элементарные составляющие требований к программной системе. В случае представления требований к

программной системе в виде совокупности последовательностей элементарных действий реализация системы возможна с использованием программы сильной структуры.

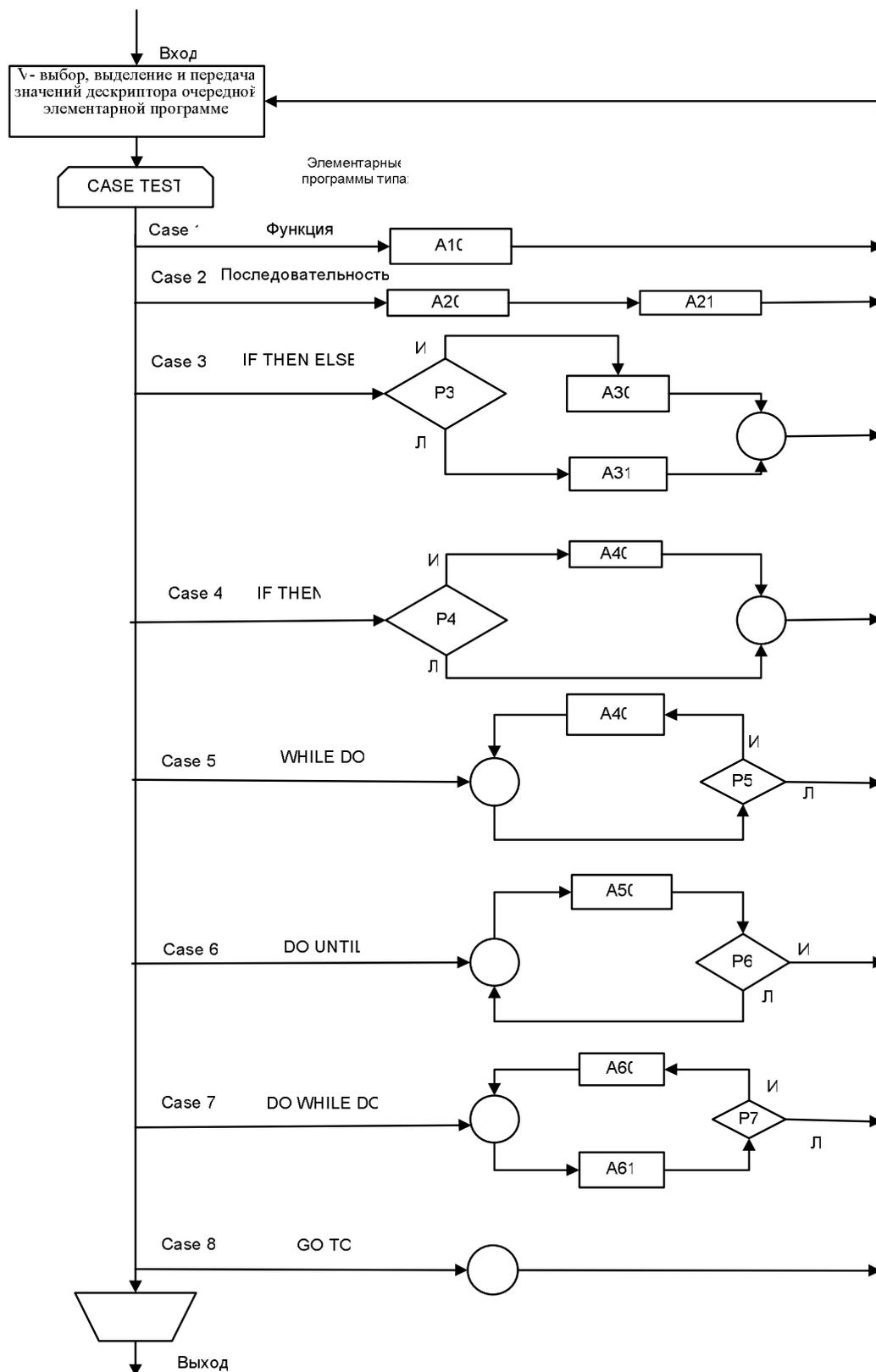


Рис. 11. Сильная структура

Данный подход удачно сочетается с требованиями по тестированию программных систем. Вместо элементарной программы можно выполнить элементарную программу отладочного типа, с помощью которой можно фиксировать все действия работы программной системы. В этом случае в записи дескриптора будет обозначена программа для отладки. Периодически выполняя отладочные

элементарные программы можно максимально отладить программную систему. Данная программная структура позволяет также доказывать правильность выполнения программ и получения точных и правильных результатов.

Применение объектно-структурного подхода в проектировании программных систем

Объектно-структурный подход в проектировании программных систем был применен при создании следующих типов программных систем:

1. Реализация концепции MODEL DRIVEN ARCHITECT (MDA) для задач внедрения информационных систем.
2. Компьютерные средства обучения.
3. Инструментарий для создания систем тестирования.

По реализации концепции MODEL DRIVEN ARCHITECT (MDA) для задач внедрения информационных систем разработан программный инструментарий, позволяющий формировать модели бизнес-процессов на предприятии с использованием языка UML. Модель бизнес-процессов включает вариант использования (*UseCase*), исполнитель (*Actor*) и отношения ассоциации и упорядоченности (*Link*). Отдельный бизнес-процесс, представленный в виде элемента варианта использования (*UseCase*), может быть специфицирован диаграммой такого же вида. Если бизнес-процесс является элементарным, то возможна его детализация путем сведения к заданию алгоритма, т. е. последовательности действий. В соответствии с этим также были определены элементы языка UML для описания алгоритма элементарных бизнес-процессов, а именно, точки начала алгоритма «StartState» и выхода из алгоритма «EndState», действие «Activity», условный переход, цикл. Элементарный бизнес-процесс описывается диаграммой метода. Каждый метод имеет список используемых в нем переменных. Далее методы регистрируются в некотором классе системы, где происходит подробное описание алгоритма метода. Элементы диаграммы алгоритма также могут иметь подробное описание, схему выполнения в виде диаграммы действия.

Реализован редактор моделей, состоящий из элементов перечисленных типов диаграмм и описания алгоритмов, в виде программы сильной структуры.

На рис.12 приводятся фрагменты разработанного инструментария. Подробное описание программного инструментария можно найти в [5].

За определенный период времени были созданы цикл компьютерных обучающих программ и инструментарий для их создания. Цикл обучающих программ охватывал двенадцать тем по курсу «Математическая статистика». Разработанные обучающие программы успешно использовались в учебном процессе в Казанском государственном техническом университете (КАИ). При создании компьютерных средств обучения был применен объектно-структурный подход. При реализации компьютерных обучающих программ и инструментария для их создания был использован подход с применением программы сильной структуры. При внедрении данного подхода были сокращены в 6-10 раз сроки и трудоемкость создания компьютерных обучающих программ.

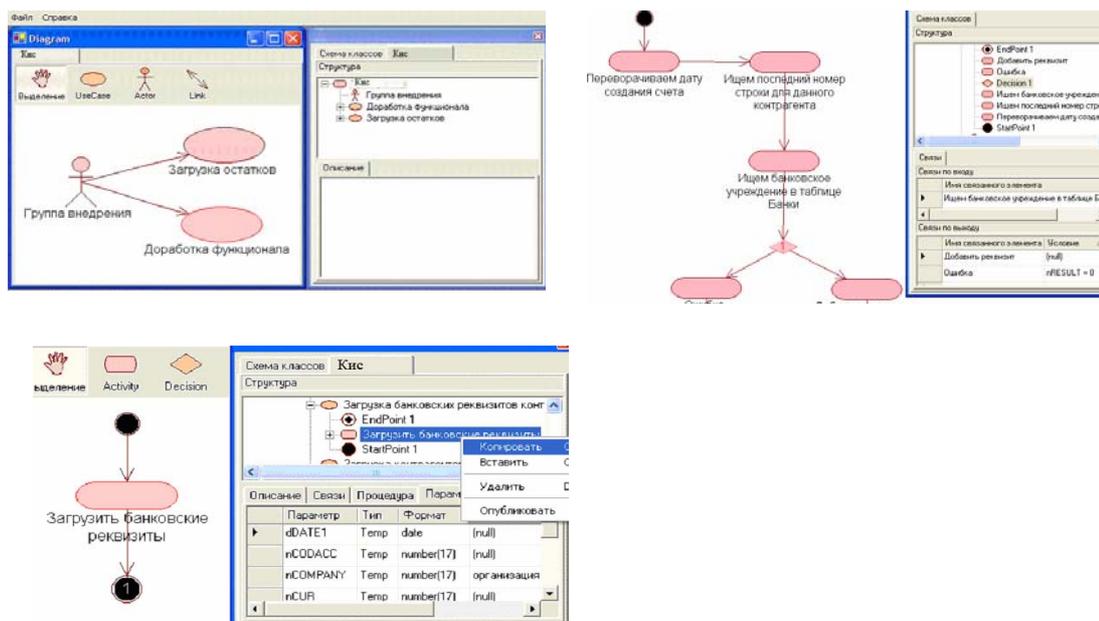


Рис. 12. Фрагменты программного инструментария для реализации концепции MDA для задач внедрения информационных систем

При разработке инструментария для создания систем тестирования также использовался вышеописанный подход. В программных реализациях была использована программа сильной структуры. На рис. 13, 14. показаны отдельные фрагменты программного инструментария для создания систем тестирования.

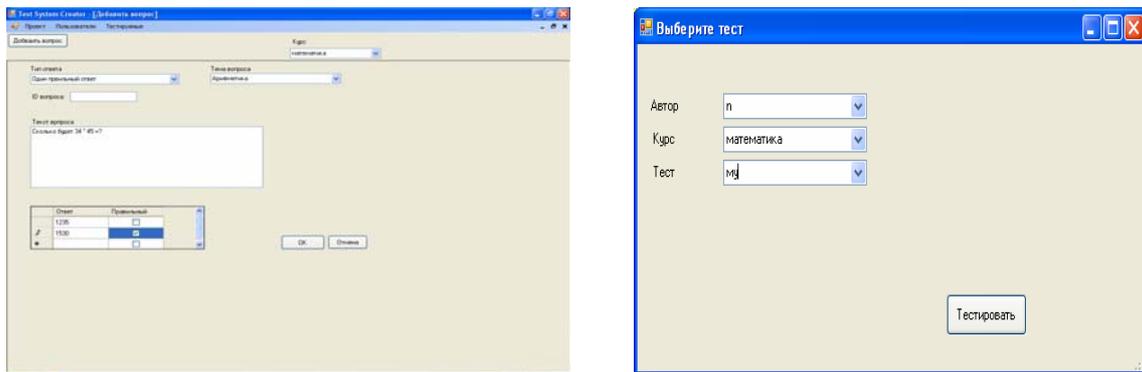
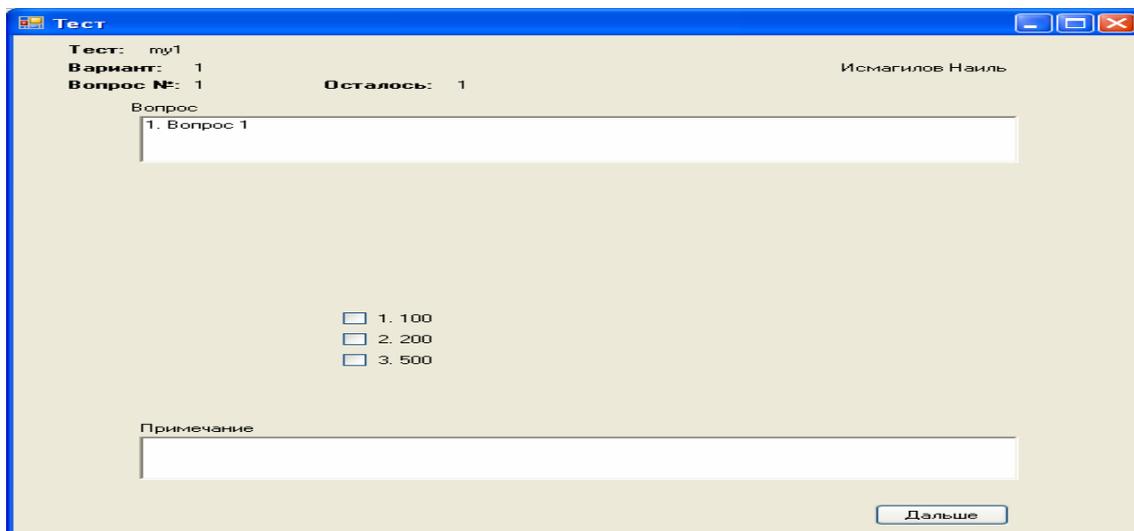
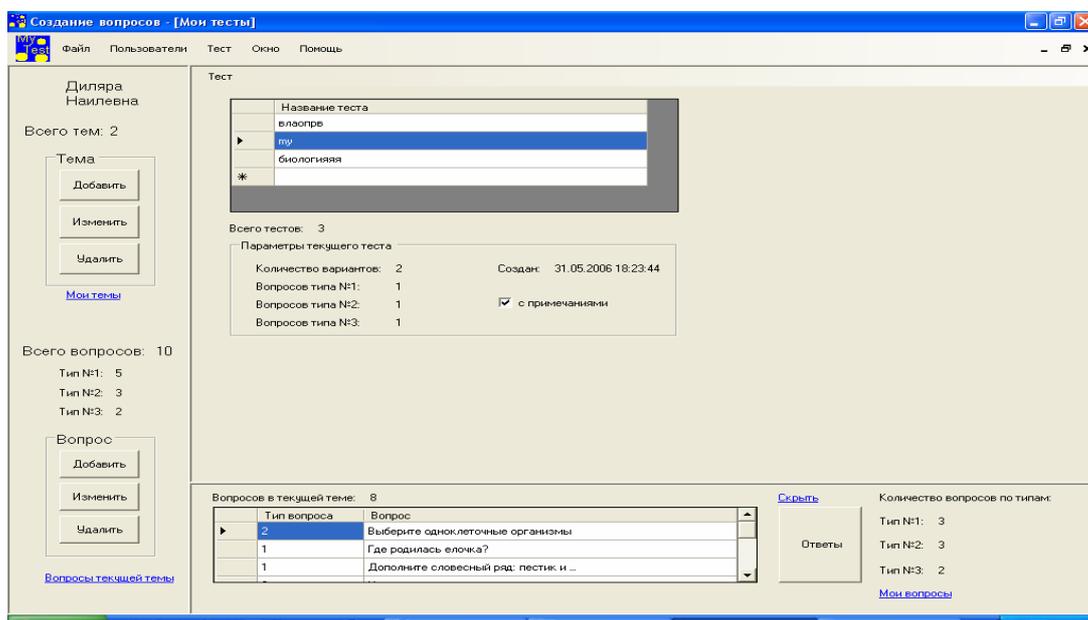


Рис. 13. Фрагменты программного инструментария для создания систем тестирования



Закключение

Описанный объектно-структурный подход может применяться в проектировании программных систем. Приведенные примеры реализации программных систем демонстрируют возможность создания с использованием программ сильной структуры.

Дальнейшее совершенствование данного подхода возможно:

- 1) В направлении представления требований к программным системам в виде совокупности последовательностей элементарных составляющих и представление их в виде множества элементарных программ.
- 2) Применение модели требований для описания связей между элементарными программами.
- 3) Преобразование модели требований в модель реализации элементарных программ и управляющей инвариантной программы. Для этих целей возможно использование языка UML.

В общем случае для успешного создания программной системы с использованием объектно-структурного подхода, необходимо выполнение следующих условий:

1. Полное описание бизнес-процесса. Описание должно быть достаточно полным и подробным.
2. Необходимое и достаточное множество требований к разрабатываемой программной системе, т.е. должно быть достаточно полное множество требований.
3. Разложимость отдельного требования на элементарные составляющие, т.е. представление требования в виде определенной совокупности последовательностей элементарных действий – элементарными программами.
4. Связь между элементарными программами. Должно быть описание сценария выполнения элементарных программ.

Список литературы.

1. Соммервилл И. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 624 с.
2. Леффенгуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению Унифицированный подход.: Пер. с англ.- М.: Издательский дом «Вильямс», 2002. – 448 с.
3. Холстед Х. М. Начала науки о программах. - М.: Финансы и статистика, 1981. - 126 с.
4. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования: Пер. с англ. М.: Мир, 1982. - 406 с.
5. Исмагилов Н. А., Рыжиков Д. Д. Модель автоматизации процесса внедрения информационных систем. // Сеточные методы для краевых задач и приложения. Материалы Шестого Всероссийского Семинара. Казань - 2005, 1-4 октября 2005 года. Издательство «Казанский государственный университет»- с. 110-122.

Распознавание и интерпретация изображений в условиях априорной неопределенности

Исмагова Х. Р.

Начальник сектора ИКИПП, 1106, Баку, пр. Азадлыг, 159

E-mail: khasiyat@box.az

Введение.

В настоящее время уже сложилась технология обработки данных дистанционного зондирования для широкого круга пользователей.

Анализ новых возможностей, привносимых децентрализацией средств ведения космического мониторинга в региональные, территориальные информационные структуры, позволил выявить

преимущества этого процесса в формировании национальных архивов изображений, реальное вовлечение технологий дистанционного зондирования (ДЗ) в решение хозяйственных проблем, систем мониторинга, сокращение времени от приема информации до принятия решения по результатам ее анализа.

Но вместе с тем возникают проблемы в выборе и решении необходимых для конкретных регионов задач, которые уже методически и технологически решены, или новых в аспекте сугубо региональных задач, актуальность которых сильно зависит от географического положения и социально-экономической специфики конкретной территории.

В этой связи для широкого использования данных ДЗ актуальна проблема выработки научно-обоснованных решений на базе привлечения материалов и технологий их обработки, интеграции с другими технологиями и данными. В этой области, решая конкретные задачи по проектам использования данных ДЗ, несмотря на наличие специализированных комплексов обработки изображений (ERDAS, ENVI), мы сталкиваемся с проблемой эффективного распознавания, интерпретации изображений и принятия решений на базе результатов их обработки.

Естественно, эта проблема касается также и задачи корректного включения в систему распознавания субъективных знаний пользователей, опыта и знаний оператора и объективных данных обработки результатов мониторинга исследуемой территории.

В настоящее время такого рода системы, позволяющие включать интуицию и неформальные знания оператора-исследователя в процесс мониторинга и количественно описывать его результат, выделяются в специальный подкласс информационно-производящих человеко-машинных систем [1].

В данной статье будет представлена методология создания и использования информационно-производящих человеко-машинных систем в проектах по экологическому мониторингу загрязнённых территорий промышленных зон, мест свалок отходов предприятий.

Описание задачи.

Основным элементом любой задачи распознавания изображений является ответ на вопрос: относятся ли данные (входные) изображения к классу изображений, который представляет данный эталон? Казалось бы, ответ можно получить, сравнивая непосредственно изображение с эталонами (или их признаки). Однако возникает ряд трудностей и проблем:

1. Изображения предъявляются на сложном фоне;
2. Изображения эталона и входные изображения отличаются положением в поле зрения;
3. Входные изображения не совпадают с эталонами за счет случайных помех;
4. Отличия входных и эталонных изображений возникает за счет изменения освещенности, подсветки, локальных помех и т.д.;
5. Эталоны и изображения могут отличать геометрические преобразования, включая такие сложные как аффинные и проективные;
6. Интерпретация изображений зависит от опыта дешифровщика, полноты дополнительной информации по конкретной задаче.

Перечисленные обстоятельства обуславливают появление неопределённости в условиях задачи распознавания изображений и неоднозначных высказываний при их интерпретации. Наблюдается значительное влияние субъективного принятия решения оператором-дешифровщиком при отсутствии полной объективной информации по непосредственным синхронным наземным и спутниковым измерениям. Очевидно, необходимо развивать комплексный анализ изображений на основе привлечения спектральных и пространственных характеристик, использования прямых и косвенных дешифровочных признаков, но в то же время, для поддержки принятия точных решений необходимо иметь достаточно полную систему дополнительных наземных данных по области исследований.

При этих обстоятельствах традиционный подход к решению задач распознавания и интерпретации не даёт желаемого эффекта, вследствие того, что их классификационные правила базируются на формальной модели распознавания, не ориентированы на использование экспертной информации, логических правил принятия решений и не позволяют осуществить выбор принадлежности при разнородной, неполной, нечёткой косвенной информации.

Исследуя эту проблему, мы пришли к выводу, что при любых способах обработки аэрокосмической информации (визуальном, интерактивном или с помощью нейросетевых технологий) все они в конечном итоге неизбежно сводятся к экспертной интерпретации [2].

Наилучший выход в этой ситуации видится в интеграции с данными дистанционного зондирования экспертных знаний, геоинформационных систем (ГИС) и технологий искусственных нейронных сетей (ИНС) [2]. То есть путём интеграции знаний в единую систему можно создать достаточно полную информационную среду для принятия правильного решения

Эти обстоятельства обуславливают обращение к теории экспертных систем, где одним из важных ограничений при их использовании является трудность формулировки правил для машинной обработки. А при использовании совместно с геоинформационной системой - это трудность формулировки правил передачи и преобразования экспертизы оценки территорий от источника знания

до программы. В этом виде в экспертной системе для оперативной выборки знаний должна быть разработана эффективная методология фиксации, сохранения и использования экспертных знаний [3].

Альтернативный метод фиксирования экспертных знаний, не используя правила, заключается в применении искусственных нейронных сетей, используя их способность к обобщению, самообучению и переобучению. Их преимущество также заключается в возможности работы в «on-line» режиме, адаптивности и быстрой приспособляемости в конкретных ситуациях.

Методы решений.

Решение, как уже отмечалось выше, находим в интеграции данных дистанционного зондирования, экспертных систем и нейросетевых технологий. Вспомогательным и организующим элементом будет ГИС.

Таким образом, перед выполнением проекта по экологическому мониторингу исследуемой территории ставится задача оценки качества земель с учётом фактор-условий: климатических (K_c), топографических (K_t), по влажности (K_w), физических и химических свойств почв (K_f), солёность и щёлочность почв (K_n) [2,3].

Здесь решающую роль играет сбор и организация адекватной задаче структуры экспертно-информационной базы данных, сопоставимой с данными дистанционного зондирования.

. Пусть в результате классификации изображения по спектральным характеристикам получена векторная модель, определяющая контура локальных ландшафтов исследуемой местности, то есть исследуемая местность разделена на элементарные участки по типам ландшафтов, составляющих множество V элементарных участков территории. Для того, чтобы отобразить в необходимой форме исходную пространственную информацию на данную территорию предполагается, что пространственная информация, хранимая в памяти компьютера в цифровом виде, будет представлена в виде матрицы R размерности $(M \times N)$, имеющий вид [3]:

$$R = \left\| r_{i,j} \right\|, \text{ где } i = 1, \dots, M; j = 1, \dots, N. \quad (1)$$

Каждый элемент матрицы R является вектором параметров, характеризующих каждый (i,j) -й элементарный участок листа карты по некоторому m -му ($m = 1, \dots, T$) набору тематических свойств:

$$r_{ij} = (r^1_{ij}, \dots, r^T_{ij}), \quad (2)$$

природа компонентов векторов $r_{i,j}$ в общем случае принципиальной роли не играет.

Тогда, если задаётся набор тематических свойств $\{P_n\}$, ($n = 1, \dots, K$), по которым необходимо районировать исследуемую территорию и пороговые границы ограничений по всему набору тематических свойств, то требуется для каждого P_n по их пороговым ограничениям поставить в соответствие множество $V_t \in V$, ($1 \leq t \leq (M \times N)$) элементарных участков земной поверхности (ЛЛС), которые обладают свойством P_n .

В нашем случае вектор P_n задан следующим образом: $P_n = (P_{K_c}, P_{K_t}, P_{K_w}, P_{K_f}, P_{K_{ch}}, P_{K_n})$, где каждый P_{K_i} - одно из значений из набора признаков, по которым известно, что каждый признак ранжирован по порогам ограничений.

В НЭС каждые из соответствующих наборов признаков представлены эталонными таблицами ранжирования по порогам ограничений, следуя системе частных шкал (таблицы), сгруппированных по типам физико-химических свойств K_c , K_t , K_w , K_f , K_{ch} , K_n . Соответственно, согласно условиям ранжирования экспертами вводится оценка отнесения к одному из классов пригодности земли (S_1, \dots, S_5), (S_1 - лучшие; S_2 - хорошие; S_3 - средние; S_4 - низкие; S_5 - условно непригодные).

При такой постановке задачи в статье описывается итеративная процедура распознавания и интерпретации изображений на базе интеграции данных дистанционного зондирования, НЭС и ГИС.

Список литературы.

1. Antonio Di Gregorio. The Land Cover Classification System (LCCS). //Classification Concepts and User: FAO UN, Rome, 1998, 95с.

2. Мамедов Г. Ш. Карта экологической оценки почв Азербайджана и её значения. //Обзорная информация АЗНИИНТИ, 1992, серия «Сельское хозяйство», С.1-25. 3. Исмаилова Х. Р., Тальбова С. С., Абдуллаева С. Н. Один способ интеграции экспертной базы знаний в систему тематического дешифрирования изображений // Сб. научных трудов Национальной Авиационной Академии, посвящённый 70-летию ак. А. М-Дж. Пашаеву, Баку, 2004 февраль, 2004, «Cild 6», №1, С.138-145.

Математические модели и алгоритмы клиринга межбанковских платежей

Карпук А. А., Шейнкман Е. С.

*Расчетный центр Национального банка Республики Беларусь,
220048, ул. Кальварийская, 7, Минск
E-mail: a_karpuk@mail.ru*

Математическую модель задачи клиринга межбанковских платежей в терминах теории исследования операций можно представить в виде:

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{l=1}^{P_{ij}} k_{ijl} a_{ijl} y_{ijl} \rightarrow \max, \quad (1)$$

$$\sum_{j=1}^n \sum_{l=1}^{P_{ij}} a_{ijl} y_{ijl} - \sum_{j=1}^n \sum_{l=1}^{P_{ji}} a_{jil} y_{jil} \leq r_i, i = \overline{1, n}, \quad (2)$$

$$y_{ijl} \in \{0,1\}, \quad (3)$$

где

$P_{ij} \geq 0, i = \overline{1, n}, j = \overline{1, n}$ – число платежей из банка с индексом i в банк с индексом j , если $i = j$, то $P_{ij} = 0$;

$a_{ijl} > 0, i = \overline{1, n}, j = \overline{1, n}, l = \overline{1, P_{ij}}$ – величина платежа с индексом l из банка с индексом i в банк с индексом j

$k_{ijl} > 0, i = \overline{1, n}, j = \overline{1, n}, l = \overline{1, P_{ij}}$ – весовые коэффициенты, учитывающие приоритеты платежей;

$r_i \geq 0, i = \overline{1, n}$ – величина резерва, установленного банком с индексом i для проведения клирингового сеанса;

$y_{ijl}, i = \overline{1, n}, j = \overline{1, n}, l = \overline{1, P_{ij}}$ – искомые переменные, доставляющие максимум целевой функции (1) при выполнении условий (2).

В терминах теории графов задачу клиринга межбанковских платежей можно сформулировать следующим образом. Пусть задан ориентированный граф $G=(B,P)$ с кратными дугами без петель с помеченными вершинами и помеченными дугами. Множество вершин графа $B = \{B_i\}, i = \overline{1, n}$, множество дуг графа $P = \{(i, j, l)\}, i = \overline{1, n}, j = \overline{1, n}, l = \overline{1, P_{ij}}$. Каждая вершина $B_i, i = \overline{1, n}$, графа соответствует банку, участвующему в клиринговом сеансе. Меткой вершины $r_i > 0$ является величина резерва банка на проведение клирингового сеанса. Каждая дуга графа $(i, j, l), i = \overline{1, n}, j = \overline{1, n}, l = \overline{1, P_{ij}}$, соответствует одному платежу и имеет две метки: величину платежа $a_{ijl} > 0$ и весовой коэффициент платежа $k_{ijl} > 0$. Требуется найти множество дуг графа, имеющих максимальную сумму произведений меток и удовлетворяющих следующему условию: для любой вершины графа разность между суммой платежей исходящих дуг и суммой платежей входящих дуг не превосходит метку вершины.

Задача (1) – (3) есть задача линейного программирования с булевыми переменными, число которых измеряется сотнями тысяч, что исключает непосредственное применение существующих методов поиска оптимального решения для задач этого типа. Известны два подхода к решению задач оптимизации при невозможности использования точных методов: использование эвристических алгоритмов для решения задачи и упрощение исходной задачи за счет ослабления некоторых условий, снятия ограничений или введения новых ограничений. Цель упрощения – получить задачу или совокупность задач, которую можно решить точными методами. Если полученное решение не удовлетворяет некоторым ограничениям из (2), (3), то производится его модификация. В автоматизированной системе «Клиринг прочих платежей», входящей в состав автоматизированной системы межбанковских расчетов Национального банка Республики Беларусь, применяются оба названных подхода.

Три основных эвристических алгоритма клиринга межбанковских платежей описаны в материалах Немецкого федерального банка [1] под названиями BILATERAL, REMAINDER и MULTILATERAL. Простейший эвристический алгоритм BILATERAL учитывает только встречные двусторонние платежи каждой пары банков, участвующих в клиринговом сеансе, и может применяться только для урегулирования двусторонних отношений банков при нежелании хотя бы одного из них участвовать в клиринге на многосторонней основе с участием другого банка. В Национальном банке Республики Беларусь алгоритм BILATERAL не применяется. Эвристический алгоритм REMAINDER, учитывающий для каждого банка все входящие и исходящие платежи, состоит из следующих шагов.

1. Для каждого банка построить список всех исходящих платежей, упорядочив их по не убыванию весовых коэффициентов и не убыванию величин платежей.

2. Для каждого банка с индексом $i \in \overline{1, n}$ вычислить текущее сальдо S_i по формуле:

$$S_i = \sum_{j=1}^n \sum_{l=1}^{P_{ji}} a_{jil} - \sum_{j=1}^n \sum_{l=1}^{P_{ij}} a_{ijl} + r_i$$

3. Для всех $i \in \overline{1, n}$, $j \in \overline{1, n}$, $l \in \overline{1, P_{ij}}$ положить $y_{ijl} = 1$.

4. Если для всех $i \in \overline{1, n}$ выполняется условие $S_i \geq 0$, то закончить работу.

5. Организовать цикл по числу банков, для которых $S_i < 0$, и выполнить для каждого из этих банков шаги 6 – 9, просматривая банки в порядке не убывания значений S_i .

6. Для рассматриваемого банка с индексом $i' \in \overline{1, n}$ просматривать упорядоченный список исходящих платежей, пока не будет найден платеж (i', j', l') , такой, что $y_{i'j'l'} = 1$ и $S_{j'} - a_{i'j'l'} \geq 0$. Если просмотрен весь список, и такой платеж не найден, то перейти к выполнению шагов 6 – 9 для следующего банка.

7. Удержать найденный платеж, положив $y_{i'j'l'} = 0$, $S_{i'} = S_{i'} + a_{i'j'l'}$, $S_{j'} = S_{j'} - a_{i'j'l'}$. Если $S_{i'} < 0$, то перейти к шагу 6.

8. Просматривать все удержанные ранее исходящие платежи из банка с индексом i' в порядке не возрастания величин платежей и не возрастания весовых коэффициентов платежей, и для каждого очередного удержанного платежа (i', j', l') выполнить действия шага 9.

9. Если $S_{i'} - a_{i'j'l'} < 0$, то перейти к рассмотрению следующего удержанного платежа, иначе удалить платеж (i', j', l') из числа удержанных платежей, положив $x_{i'j'l'} = 1$, $S_{i'} = S_{i'} - a_{i'j'l'}$, $S_{j'} = S_{j'} + a_{i'j'l'}$, и перейти к рассмотрению следующего удержанного платежа. Если просмотрены все удержанные платежи, то перейти к выполнению шагов 6 – 9 для следующего банка.

10. Если для всех банков, для которых выполнялись шаги 6 – 9, выполняется условие $S_i \geq 0$, то закончить работу.

11. Организовать цикл по числу банков, для которых $S_i < 0$, и выполнить для каждого из этих банков шаги 12 – 15, просматривая банки в порядке не убывания значений S_i .

12. Для рассматриваемого банка с индексом $i' \in \overline{1, n}$ просматривать упорядоченный список исходящих платежей пока не будет найден платеж (i', j', l') , такой, что $y_{i'j'l'} = 1$. В отличие от шага 6, выполнение условия $S_{j'} - a_{i'j'l'} \geq 0$ здесь не требуется. Если просмотрен весь список, и такой платеж не найден, то перейти к выполнению шагов 12 – 15 для следующего банка.

13. Удержать найденный платеж, положив $y_{i'j'l'} = 0$, $S_{i'} = S_{i'} + a_{i'j'l'}$, $S_{j'} = S_{j'} - a_{i'j'l'}$. Если $S_{i'} < 0$, то перейти к шагу 12.

14. Просматривать все удержанные ранее исходящие платежи из банка с индексом i' в порядке не возрастания величин платежей и не возрастания весовых коэффициентов платежей, и для каждого очередного удержанного платежа (i', j', l') выполнить действия шага 15.

15. Если $S_{i'} - a_{i'j'l'} < 0$, то перейти к рассмотрению следующего удержанного платежа, иначе удалить платеж (i', j', l') из числа удержанных платежей, положив $x_{i'j'l'} = 1$, $S_{i'} = S_{i'} - a_{i'j'l'}$, $S_{j'} = S_{j'} + a_{i'j'l'}$, и перейти к рассмотрению следующего удержанного платежа. Если просмотрены все удержанные платежи, то перейти к выполнению шагов 12 – 15 для следующего банка.

16. Если для всех банков с номерами $i = \overline{1, n}$ выполняется условие $S_i \geq 0$, то закончить работу, иначе перейти к шагу 5.

При использовании алгоритма REMAINDER возможна ситуация, когда алгоритм не заканчивает работу за заданный промежуток времени. В этом случае для завершения процесса решения задачи клиринга межбанковских платежей можно использовать эвристический алгоритм MULTILATERAL, который незначительно отличается от алгоритма REMAINDER. В алгоритмах REMAINDER и MULTILATERAL используется достаточно эффективное эвристическое правило выбора очередного обрабатываемого банка. Действительно, если для некоторого банка текущее сальдо отрицательно, то в оптимальном решении задачи (1) – (3) не все платежи этого банка будут проведены. Чем раньше такой банк обрабатывается, тем больше вероятность того, что будут найдены платежи, удержание которых не приведет к возникновению отрицательного сальдо у другого банка. Поэтому в алгоритмах REMAINDER и MULTILATERAL банки с текущим отрицательным сальдо обрабатываются в порядке его не убывания. Проведенные вычислительные эксперименты показали, что при наличии хотя бы небольшого резерва у каждого банка алгоритмы REMAINDER и MULTILATERAL дают решение, отличающееся не более, чем на 5 % от оценки оптимального решения сверху. Однако, при наличии банков с нулевым резервом качество решения алгоритмов REMAINDER и MULTILATERAL резко ухудшается, а когда все банки имеют нулевые резервы, алгоритмы REMAINDER и MULTILATERAL после длительных вычислений чаще всего удерживают все платежи.

Традиционный приемом упрощения задач дискретной оптимизации является переход к непрерывным переменным. В нашем случае этот прием состоит в замене ограничений (3) на ограничения

$$0 \leq y_{ijl} \leq 1, \quad i = \overline{1, n}, \quad j = \overline{1, n}, \quad l = \overline{1, P_{ij}}. \quad (4)$$

После такой замены становится несущественным, какие конкретные платежи с равными весовыми коэффициентами будут проводиться в оптимальном решении задачи (1), (2), (4). Действительно, обозначив

$$x_{ij} = \sum_{l=1}^{P_{ij}} k_{ijl} a_{ijl} y_{ijl}, \quad i = \overline{1, n}, \quad j = \overline{1, n}, \quad (5)$$

$$S_{ij} = \sum_{l=1}^{P_{ij}} a_{ijl}, \quad i = \overline{1, n}, \quad j = \overline{1, n}, \quad (6)$$

получим следующую задачу линейного программирования:

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} \rightarrow \max, \quad (7)$$

$$\sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} \leq r_i \quad \text{для всех } i = \overline{1, n}, \quad (8)$$

$$0 \leq x_{ij} \leq S_{ij} \quad \text{при } i \neq j, \quad x_{ij} = 0 \quad \text{при } i = j. \quad (9)$$

Число искоемых переменных в задаче (7) – (9) равно $n(n-1)$, в клиринговом сеансе в Республике Беларусь участвуют не более 30 головных банков, поэтому задачу можно решить известными методами решения задач линейного программирования. Для перехода от найденного оптимального решения задачи

(7) – (9) к решению задачи (1) – (3) для каждой пары банков, для которой $x_{ij} > 0$, требуется решить задачу о рюкзаке [2]. В отличие от классической задачи о рюкзаке, здесь незаполненное место в рюкзаке оставлять нельзя, т. е. реальная сумма величин платежей из банка с индексом i в банк с индексом j

может превосходить x_{ij} . Недостающую величину платежа можно взять только из резерва, но даже если этот резерв был, у многих банков он будет израсходован при решении задачи (7) – (9). Поэтому такой подход можно использовать только при управляемом клиринге, когда после выполнения расчетов каждому банку сообщается требуемая величина дополнительного резерва, и банк сразу осуществляет резервирование средств.

Чтобы сохранить максимальную величину резерва до решения задачи о рюкзаке, положим сначала $r_i = 0$, для всех $i = \overline{1, n}$. Можно показать, что в этом случае все ограничения (8) становятся равенствами и имеют вид

$$\sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = 0 \quad \text{для всех } i = \overline{1, n}. \quad (10)$$

Полученная задача (7), (10), (9) относится к классу транспортных задач, для решения которых разработаны эффективные алгоритмы. В частности, оптимальное решение задачи (7), (10), (9) можно найти путем выделения и анализа всех циклов в графе, полученном из графа $G=(B, P)$ в результате объединения всех дуг, идущих из вершины с индексом $i = \overline{1, n}$ в вершину с индексом $j = \overline{1, n}$, в одну дугу с пропускной способностью S_{ij} .

В автоматизированной системе «Клиринг прочих платежей» Национального банка Республики Беларусь при проведении клиринговых сеансов применяется комбинированный алгоритм решения задачи (1) – (3), состоящий из следующих шагов.

1. Задача (7), (10), (9) решается методом выделения и анализа циклов в ориентированном графе. Каждый цикл, входящий в решение задачи, характеризуется пропускной способностью, равной минимальной пропускной способности дуги цикла, и стоимостью, равной произведению пропускной способности на число дуг цикла.

2. Для каждой пары банков, для которой $x_{ij} > 0$, решается задача о рюкзаке с минимальным перекрытием объема рюкзака за счет выделенного резерва банка. Если выделенного резерва недостаточно, то для этой пары банков решается задача о рюкзаке в классической постановке и определяется невязка, равная разности между x_{ij} и полученным объемом рюкзака. Среди циклов, являющихся решением задачи (7), (10), (9) и проходящих через дугу (i, j) , находится цикл с минимальным числом дуг, пропускная способность которого не меньше невязки. Пропускная способность найденного цикла уменьшается на величину невязки, и соответствующим образом изменяется решение задачи (7), (10), (9). Платежи, входящие в решение задачи о рюкзаке, включаются в искомое решение задачи (1) – (3).

3. К оставшимся платежам и оставшимся величинам резерва банков применяется алгоритм REMAINDER. Если за отведенное время этот алгоритм не закончил работу, то решение задачи завершается с использованием алгоритма MULTILATERAL.

Список литературы.

1. Электронный доступ в Немецком федеральном банке. Внешние спецификации. Глава 3. Система EAF-2. 7 сентября 1998 г. Версия 4.1. – Франкфурт: Немецкий федеральный банк, 1998. – 137 с.
2. Ковалев М. М. Дискретная оптимизация (целочисленное программирование). – Минск: БГУ, 1977. – 192 с.

Особенности информационных технологий в сфере образования российского общества

Климов В. Г.,

*кандидат педагогических наук, зав. учебно-исследовательской лабораторией
информационной безопасности, механико-математический факультет,*

Т. 8 (342) 2-39-62-98, E-mail: vklimov@perm.ru

Яковлев В. И.,

*доктор физико-математических наук, профессор,
декан механико-математического факультета,
Пермский государственный университет*

Для российского общества первостепенными являются вопросы не о том, что будет с системой образования через 50-70 лет, хотя это тоже очень важно, а о том, что необходимо сделать сегодня или в ближайшие годы, чтобы выйти из кризисного состояния, стабилизировать обстановку в сфере

образования и направить его в русло общемировых тенденций с учетом национальных и региональных особенностей подготовки конкурентоспособных специалистов. По нашему мнению, на новом этапе экономической реформы в российском обществе необходимо обеспечить системное реформирование содержания образования на основе информационно-коммуникационных педагогических технологий, создать действующий механизм его постоянного обновления. Предстоит переориентировать учебно-воспитательный процесс с воспроизводства только образцов прошлого опыта человечества на освоение современных интеллектуальных возможностей компьютерных средств обучения, овладение способами и методами самообразования, умением учиться с использованием дистанционных форм обучения.

Уместно перечислить особенности информационных и коммуникационных педагогических технологий в области образования:

1. Компьютерные средства обучения нельзя вывести из себя. Они могут сколько угодно раз повторять один и тот же цикл лекции или подбирать задачи на заданную тему. Педагогам известно, что достаточно простые темы, которые без труда осваивает основная масса студентов, некоторые из них даются с трудом. Представить себе преподавателя, помогал бы одному и тому же студенту решать большое количество однотипных задач, не испытывая при этом раздражения и нетерпения, затруднительно.

2. Интеллектуальные способности компьютерных систем отличаются от субъекта образования. Студент прекрасно понимает, что его неспособность быстро усвоить учебный материал, с точки зрения других – очень простой, выставлять его в неприглядном свете в глазах преподавателя и сокурсников. Как правило, студенты, даже с сильной мотивацией, после неудачных попыток стараются избежать решение задач на эту тему. Подобная ситуация может быть эффективно разрешена с помощью самостоятельных занятий на компьютерных средствах обучения, которого можно не стесняться и который позволит студенту тренироваться в решении задач данного типа до тех пор, пока не будет достигнуто необходимое усвоение темы.

3. Компьютерный интеллект способен практически без задержки генерировать стандартные задачи, для решения которых существуют алгоритмизированные методики.

4. Программы компьютерных систем хорошо справляются с оценкой только тех ответов и решений, которые подразумевают однозначный результат.

Выводы.

Преимущества информационных технологий с использованием компьютерных средств обучения очевидны: сокращение времени сложных математических расчетов, сведение к минимуму рутинной работы, возможность анализа и синтеза большого количества вариантов, простота решения оптимизированных задач. Однако следует отметить, что при наличии готовых программ студенты не усваивают методику математического расчета и не понимают физической сущности задачи, поскольку их роль ограничивается нажатием нескольких клавиш и вводом исходных данных.

Список литературы

1. Роберт И. В. Влияние тенденций информатизации, массовой, глобальной коммуникации современного общества на образование. Материалы Международной научно-практической конференции «Информатизация образования - 2005». – Елец: Елецкий государственный университет им. И. А. Бунина, 2005. – С. 5 – 55.

2. Климов В. Г. Повышение качества и доступности профессионального образования на основе информационных и коммуникационных технологий обучения. Материалы Международной конференции (9-11 сентября 2005) «Болонский процесс в математическом и естественнонаучном педагогическом образовании: тенденции, перспективы, проблемы». – Петрозаводск: Изд-во Карельского государственного педагогического университета, 2005. – С. 86 – 92.

3. Климов В. Г. Информационно-коммуникационные технологии как основа эффективного решения широких классов задач российского общества. Материалы Международной научно-практической конференции «Валихановские чтения - 10». – Кокшетау: Изд-во Кокш. ун-та им. Ш. Уалиханова, 2005. – С. 86 – 92.

Интернет-система, моделирующая процесс преобразования программ, управляемый знаниями

Князева М. А., *с.н.с.*, mak@imcs.dvgu.ru,
Бердник А. Н., non-alexandro@yandex.ru,
Волков Д. А., vd@ccimb.dvgu.ru,
Жеравин М. В., zheravin@mail.ru,
Зотов И. Ю., nirwander@gmail.com,
Маевский М. С., egolf@nm.ru,
Плюхих С. А., Neo_555@mail.ru,
Тимченко В. А., rakot2k@mail.primorye.ru

690041, г. Владивосток, ул. Радио, 5,
Институт Автоматики и Процессов Управления ДВО РАН

Введение

В настоящее время задача увеличения производительности компьютеров по-прежнему остаётся актуальной [1,2]. С усложнением архитектуры компьютеров усложняются и языки программирования, что влечет за собой снижение качества как исходных, так и объектных программ. Поэтому, чтобы не потерять выигрыш, полученный за счет возможностей новых архитектур компьютеров, необходимо улучшать исходные программы и совершенствовать компиляторы языков программирования.

Несмотря на существенные достижения в сферах науки, профессиональной деятельности и образования, связанные с преобразованиями программ, существует ряд проблем, тормозящих развитие этой области: разнородность систем понятий и моделей в области преобразования программ, невозможность моделирования работы оптимизирующих компиляторов и их быстрое моральное старение. Поэтому актуальной является разработка подходов к решению вышеперечисленных проблем.

В настоящей работе рассмотрена Интернет-система, моделирующая процесс преобразования программ, управляемый знаниями. Наборы преобразований представленных в базе знаний, а также наборы языков представления программ не являются фиксированными. Работа содержит разделы, в которых описана концепция системы, описание ее подсистем и предложен подход к реализации рассматриваемой системы. Рассмотренная система является ядром оптимизирующего компилятора, управляемого знаниями. Информационное обеспечение и доступ к системе осуществляется специализированным банком знаний о преобразованиях компьютерных программ (СБкЗ_ПП) [3-6].

Работа выполнена при финансовой поддержке ДВО РАН, инициативный научный проект "Интернет-система управления информацией о преобразованиях программ".

1. Концепция Интернет – системы, моделирующей процесс преобразования программ, управляемый знаниями

Основное назначение данной системы состоит в предоставлении возможности проведения экспериментов для исследования эффективности преобразований, зависимостей между преобразованиями и их влияния на результаты друг друга. Пользователю, наряду с возможностью применения набора выбранных преобразований к выбранной программе, задания стратегии применения этих преобразований, предоставляется возможность задавать собственные преобразования. Пользователь может добавлять описанные преобразования к уже существующим, расширять набор преобразований, с которыми может работать система, а также модифицировать уже существующие в системе преобразования. Система позволяет генерировать низкоуровневый код на языке ассемблера для платформы x86. В системе присутствуют средства оценивания эффективности применения преобразований: есть возможность исследовать влияние как одного преобразования, так и целого набора преобразований, примененных в заданной последовательности к одной программе. Таким образом, можно выделить следующие основные функции системы:

- создание, редактирование, а также сохранение программ на языке программирования высокого уровня в базе программ;
- генерация программы на языке моделей структурных программ (ЯМСП) [3] по исходному представлению программы на языке программирования высокого уровня;
- описание, редактирование знаний о преобразованиях программ и их занесение в базу знаний о преобразованиях программ;
- расширение программы на ЯМСП терминами потокового анализа программ;
- поиск участков экономии в программе на ЯМСП в процессе интерпретации знаний о преобразованиях программ;

- преобразование найденных участков экономии в программе на ЯМСП в процессе интерпретации знаний о преобразованиях программ;
- генерация низкоуровневого кода на языке ассемблера для платформы x86 по преобразованным программам на ЯМСП;
- сбор различных параметров исходной программы и оценка на их основе эффективности применяемых преобразований;
- отображение информации, накопленной подсистемой измерения эффективности, об историях применения преобразований к конкретным программам;
- создание графических представлений программ на ЯМСП;
- генерация преобразованной программы на языке программирования высокого уровня по представлению программы на ЯМСП.

2. Основные компоненты Интернет – системы, моделирующей процесс преобразования программ, управляемый знаниями

Подсистема управления.

Пользователь взаимодействует с подсистемой управления, которая управляет проведением экспериментов. В ходе эксперимента она дает пользователю возможности:

- выбрать исходную программу на языке программирования высокого уровня из набора доступных программ, хранящихся в базе программ, с которой будут осуществляться преобразования, или создать для этой цели новую программу, вызвав соответствующее средство редактирования, и сохранить ее в базе программ;
- выбрать набор преобразований из множества доступных преобразований, хранящихся в базе знаний о преобразованиях программ, и указать стратегию их применения к исходной программе или описать новое/модифицировать существующее преобразование, вызвав соответствующее средство редактирования, и сохранить его в базе знаний о преобразованиях программ;
- после применения набора преобразований к выбранной программе просмотреть историю применения преобразований и получить отчет по качеству проведенных преобразований.

Редакторы программ и МСП – генератор

Создание, редактирование, а также сохранение программ на различных языках программирования в базе программ осуществляется соответствующими структурными редакторами программ, каждый из которых ориентирован на определенный язык программирования в рамках модели онтологии соответствующего языка.

Для получения внутреннего представления программы, разработано программное средство – генератор моделей структурных программ (МСП - генератор). Данное средство по представлению программы, сформированному соответствующим редактором программ генерирует представление этой программы на ЯМСП.

После того, как на основе выбранной пользователем исходной программы будет сформирована программа на ЯМСП, подсистема управления вызовет преобразователь программ, передав ее, в числе прочего, в качестве входной информации.

Преобразователь программ

Преобразователь программ – совокупность подсистем, управляемых знаниями, общей задачей которых является непосредственное проведение преобразований над исходной программой. Преобразователь программ состоит из четырех подсистем: потокового анализа, поиска участков экономии, трансформации и измерения эффективности. Исходными данными для преобразователя программ является МСП и набор преобразований, выбранных пользователем через подсистему управления. Каждое преобразование состоит из формулы контекстного условия и формулы трансформации. Результатом работы преобразователя программ является исходная МСП, трансформированная согласно указанным преобразованиям.

Процесс работы преобразователя программ происходит следующим образом. Первой работает подсистема потокового анализа программ. Для каждого преобразования (если их применяется несколько) до тех пор, пока оно применимо, подсистема потокового анализа, расширяет МСП дополнительной информацией об исходной программе, используя метод транзитивного замыкания. Затем подсистема поиска участков экономии, согласно формуле контекстного условия, находит первый участок экономии и передает информацию о нём подсистеме трансформации. Эта подсистема, перестраивает указанный фрагмент программы, используя формулу трансформации. После чего цикл повторяется для трансформированной МСП до тех пор, пока есть участки экономии. Таким образом, в результате работы подсистем, исходная МСП будет преобразована.

Подсистема измерения эффективности организует сбор различных параметров исходной программы, что позволяет оценить эффективность применяемых преобразований. В оценку включаются такие

показатели, как общая “стоимость” программы, “цена” отдельных участков экономии, время, потраченное на преобразование, количество переменных и констант, используемых на каждом шаге трансформации и др. Эти данные централизованно хранятся в архиве историй, что позволяет сравнивать отдельные эксперименты между собой.

Множество преобразований, с которым может работать преобразователь программ, не является фиксированным, а может изменяться следующим образом: к нему могут добавляться новые преобразования, из него могут при необходимости удаляться преобразования, а также могут быть модифицированы уже существующие преобразования.

Все преобразования хранятся в базе знаний о преобразованиях программ и описаны в терминах онтологии знаний о преобразованиях программ. Для осуществления всех вышеперечисленных операций над множеством преобразований используется редактор знаний о преобразованиях программ. Процесс редактирования знаний в редакторе осуществляется в терминах онтологии знаний о преобразованиях программ.

После того, как процесс преобразования МСП закончился, подсистема управления вызывает подсистему генерации низкоуровневого кода, на вход которой подается преобразованная МСП.

Подсистема генерации низкоуровневого кода.

Подсистема генерации позволяет генерировать низкоуровневый код на языке ассемблера платформы x86 по МСП.

Средства визуализации и отчетов по качеству оптимизации программы.

Средства визуализации позволяют отображать информацию, накопленную подсистемой измерения, об историях применения преобразований к конкретным программам. Данная компонента также позволяет создавать графические представления моделей структурных программ.

3. Подход к реализации системы в среде Интернет

Одним из путей привлечения к согласованному решению комплекса задач в какой-либо предметной области различных специалистов и других людей, так или иначе занятых в этой области, является объединение всей используемой информации в единый ресурс и обеспечение возможности его использования.

В качестве единого информационного ресурса был использован Специализированный банк знаний преобразований программ (СБкЗ_ПП), в рамках Многоцелевого банка знаний (МБкЗ), концепция которого предложена в работе [6].

Архитектура банка знаний предполагает реализацию системы в соответствии с технологией “клиент-сервер”. При этом на сторону клиента переносятся: подсистема генерации низкоуровневого кода, преобразователь программ, МСП-генератор, подсистема управления, а также интерфейсы средств редактирования и редактор ИРУО [7].

На сервере остаётся, хранящиеся в информационном наполнении (ИН) банка знаний: модель онтологии знаний о преобразованиях программ, модели онтологий языков программирования, модель онтологии МСП, модель онтологии знаний о потоковом анализе программ, модель онтологии знаний о целевых платформах. Также в состав ИН входят: база знаний о преобразованиях программ, база программ в терминах онтологий языков программирования высокого уровня, база знаний о потоковом анализе программ, база знаний о целевых платформах, а также архив историй преобразований программ и отчетов по качеству процесса оптимизации программы.

Для поддержки ИН, на стороне сервера, в программном наполнении (ПН) банка знаний содержится функциональная часть средств редактирования, доступ к которым организован через соответствующие интерфейсы клиентской части. К средствам редактирования относится редактор ИРУО, а также редактор знаний о преобразованиях программ, редактор знаний о целевых платформах, редактор знаний о потоковом анализе программ и редакторы программ на языках программирования высокого уровня. Каждый специализированный редактор – это редактор ИРУО, управляемый соответствующей моделью онтологии.

Список литературы

1. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. - СПб.: БХВ_Петербург, 2002.-608 с.
2. Касьянов В. Н. Оптимизирующие преобразования программ. - М.: Наука, 1988.- 336 с.
3. Князева М. А., Купневич О. А. Модель онтологии предметной области «Оптимизация последовательных программ». Определение языка модели структурных программ. // НТИ. Сер. 2.-2005.- № 2.-С. 17-21.
4. Клещев А. С., Князева М. А. Управление информацией о преобразованиях программ. I. Анализ проблем и пути их решения на основе методов искусственного интеллекта // Изв. РАН. ТИСУ. 2005. № 5.

5. Клещев А. С., Князева М. А. Управление информацией о преобразованиях программ. II. Внутреннее устройство специализированного банка знаний о преобразованиях программ // Изв. РАН. ТиСУ. 2005. № 6, с. 101-110.

6. Орлов В. А., Клещев А. С. Компьютерные банки знаний. Многоцелевой банк знаний // Информационные технологии. 2006. №2. С.2-8.

7. Орлов В. А., Клещев А. С. Многоцелевой банк знаний. Часть 3. Концепция универсального редактора ИРУО. Владивосток: ИАПУ ДВО РАН, 2003. 28 с.

Математическая модель и комплекс программ для исследования процессов пластической деформации в ГЦК материалах

Колупаева С. Н.,

профессор, Томский государственный архитектурно-строительный университет, 634003, Россия, г. Томск, пл. Соляная, 2, vir@mail.tomsknet.ru

Семенов М. Е.,

программист, информационный научно-правовой центр "Томветцентр", 634059, г. Томск, ул. Фестивальная, 9, isthis@yandex.ru

Вихорь Н. А.,

доцент, Томский государственный архитектурно-строительный университет, 634003, Россия, г. Томск, пл. Соляная, 2, vihor@yandex.ru

Исследование механизмов и процессов пластичности кристаллических материалов является актуальной и сложной задачей, как для экспериментального изучения, так и для математического моделирования, поскольку пластическая деформация металлов и сплавов - сложный динамический процесс, определяемый как типом, свойствами и структурой деформируемого материала, так и способом воздействия на него. При исследовании процессов пластической деформации в кристаллических материалах эффективно используются математические модели, основанные на обыкновенных дифференциальных уравнениях (ОДУ) баланса деформационных дефектов [1-9]. Различные модели пластической деформации отличаются, прежде всего, набором учитываемых деформационных дефектов и механизмов их образования и аннигиляции.

В работах [4-9] на основе единой концептуальной модели разработана система математических моделей пластической деформации механизмами кристаллографического скольжения, основанных на уравнениях баланса деформационных дефектов, для различных кристаллических материалов и условий деформирования. Математические модели сформулированы для материалов с ГЦК структурой, к которым относятся широко используемые на практике в чистом виде либо в сплавах медь, алюминий, никель, свинец, золото и другие.

Математическая модель пластической деформации для ГЦК металлов и дисперсно-упрочненных материалов на их основе [7-9] включает систему ОДУ баланса деформационных дефектов, которая имеет следующий общий вид:

$$\frac{d\bar{X}}{da} = G(\bar{X}, \bar{Y}, a, t) - A(\bar{X}, \bar{Y}, a, t) - R(\bar{X}, \bar{Y}, a, t), \quad (1)$$

где \bar{X} - вектор переменных, характеризующих дефектную среду, \bar{Y} - вектор переменных, характеризующих приложенное воздействие, a - степень деформации сдвига, t - время, $G(\bar{X}, \bar{Y}, a, t)$, $A(\bar{X}, \bar{Y}, a, t)$, $R(\bar{X}, \bar{Y}, a, t)$ - функции генерации, аннигиляции и релаксационной трансформации деформационных дефектов соответственно.

При пластической деформации скольжения образуются, преимущественно, дислокации различного типа и точечные дефекты. В качестве переменных модели (1), характеризующих деформационную дефектную среду, в работах [7-9] учтены основные типы линейных и точечных дефектов: для металлов с ГЦК структурой - сдвигообразующие дислокации, дислокации в дипольных конфигурациях вакансионного и межузельного типа, межузельные атомы, моно- и бивакансии, для дисперсно-упрочненных материалов, дополнительно, дислокационные призматические петли вакансионного и

межузельного типа. Учитывается полный набор взаимодействий между точечными дефектами. Система уравнений баланса деформационных дефектов дополнена уравнением для скорости деформации и уравнением, описывающим физическое воздействие, которое является причиной деформации.

Уравнения модели [4-9] построены в результате рассмотрения процессов, происходящих при формировании элементарного скольжения и зоны кристаллографического сдвига. Именно при формировании элементарного скольжения образуются основные дефекты (прежде всего сдвигообразующие дислокации), определяющие процессы пластичности. Области элементарных скольжений являются также источниками точечных деформационных дефектов, причем имеют место кристаллогеометрические соотношения, связывающие величины, которые характеризуют количество точечных деформационных дефектов, производимых при скольжении, с вкладом, который это скольжение вносит в деформацию кристалла [4-9]. Частные модели механизмов генерации и аннигиляции деформационных дефектов для ГЦК металлов и дисперсно-упрочненных материалов на их основе записаны на основе единых предположений и базируются на фундаментальных физических и топологических свойствах дефектов строения решетки [4-9]. Все параметры уравнений имеют физический или кристаллогеометрический смысл и могут быть вычислены (или указаны пределы их изменения).

При численном решении системы ОДУ модели пластической деформации скольжения возникают сложности, связанные с тем, что процессы генерации и аннигиляции деформационных дефектов являются существенно разноразмерными, переменные системы являются разнопорядковыми величинами и изменяются на интервале интегрирования на порядки величины. В этом случае, как правило, приходится иметь дело с жесткими системами ОДУ, поэтому используемые вычислительные методы должны быть пригодными для их решения. При этом следует учитывать трудности, связанные с вычислением начального приближения, выбором начального шага интегрирования, управлением размером шага при сохранении необходимой точности решения; кроме того, для различных деформирующих воздействий и материалов может отличаться число уравнений, сами уравнения могут быть кусочно-сшитыми. В результате, работа с математической моделью доступна только исследователю, владеющему навыками программирования и решения систем ОДУ.

Для того чтобы модель стала доступной как инструмент исследования специалисту (например, экспериментатору, металлофизику или материаловеду), имеющему минимальный опыт работы с вычислительной техникой, для реализации математической модели пластической деформации [4-9] разработан программный комплекс [10-11] Slip Plasticity of Face-Centered Cubic v1.0 (SPFCC). Программный продукт имеет интуитивно понятный, «дружественный» интерфейс пользователя; результаты, полученные в вычислительных экспериментах, автоматически сохраняются в базе данных. Комплекс программ SPFCC позволяет проводить расчеты для случая деформации кристалла с постоянной скоростью деформирования, постоянного приложенного напряжения, постоянной нагрузки при растяжении и сжатии.

Пользователь может провести моделирование, используя как полную модель, так и сформировать в диалоговом режиме редуцированную модель для различных приложенных воздействий и материалов, в которой будут учтены указанные им типы дефектов и механизмы их образования и аннигиляции. Для каждого компьютерного эксперимента в базе данных кроме полного набора значений переменных системы с заданным пользователем шагом, хранятся значения параметров модели, описание типов дефектов и механизмов, учтенных пользователем при формировании модели, а также возможные комментарии.

Для различных деформирующих воздействий и материалов явный вид системы ОДУ пользователь может сформировать, используя иерархическую группу переключателей «Дефекты», которые позволяют пользователю в рамках предложенной модели определить набор учитываемых дефектов и механизмов их образования и аннигиляции, или воспользоваться выбором по умолчанию (в этом случае учитывается максимальный возможный набор дефектов и механизмов). Такая возможность является существенным преимуществом созданного комплекса программ перед программами символьной алгебры. Практически исследователь имеет дело с системой моделей, которая может быть получена из базовой модели пластической деформации для разных материалов и приложенных воздействий.

Ввод данных в диалоговых окнах осуществляется с использованием стандартных элементов ввода информации (поле, поле со списком, поле с прокруткой). Для удобства пользователя на панели инструментов, расположенной непосредственно под системным меню, содержатся кнопки и поля выбора, с помощью которых можно выполнить основные команды, не используя меню (например, задать тип материала, материал, условия деформирования и др.). Существует возможность вводить данные вручную либо загружать информацию из файла. Кроме того, есть функция сохранения введенных данных для последующего использования. «Менеджер входных параметров» обладает удобным интерфейсом пользователя и гибкой системой настройки более чем 20 параметров. Набор параметров модели зависит от типа деформирующего воздействия и материала.

Основными являются:

- *начальные плотности дислокаций различного типа и концентрации точечных дефектов;*
- *тип материала* (ГЦК металлы, дисперсно-упрочненные материалы);
- *значения параметров, характеризующих материал* (модуль сдвига, энергия образования и миграции точечных дефектов, коэффициент Пуассона, модуль вектора Бюргерса и т.п.);
- *условия деформирования* (постоянная скорость деформирования, постоянное напряжение, постоянная нагрузка для растяжения и сжатия).

Значения параметров, характеризующих материал, заложены в комплекс программ из справочной литературы либо из результатов независимых исследований и оценок. При выборе материала в поля для задания параметров автоматически вводятся значения, представленные в базисном наборе. Кроме того, пользователь может выбрать значения из некоторого диапазона физически реализуемых значений параметра либо ввести их. Если пользователь вводит значение параметра, не имеющее физического смысла, программа SPFCSS выведет диалоговое окно с сообщением об ошибке и неверное значение будет автоматически скорректировано на допустимое. После ввода необходимой информации осуществляются вычисления, и пользователь может в режиме реального времени расчетов получать графическое отображение результатов. Программа позволяет проводить серии расчетов, для этого необходимо выбрать варьируемый параметр модели (например, температуру, скорость деформирования или начальную плотность сдвигобразующих дислокаций), нижний и верхний предел его изменения и шаг. В этом случае можно построить, например, температурную или скоростную зависимости. При работе с графикой пользователь может выбрать систему координат (декартова или полулогарифмическая), зависимые и независимую переменные, сохранить график в формате .wmf. У пользователя есть возможность вернуться к ранее сохраненным в базе результатам вычислений и провести сравнение, совместный анализ результатов различных экспериментов, экспортировать данные, относящиеся к различным экспериментам в текстовый файл.

Для решения системы ОДУ в программе использованы: явный линейный многошаговый метод Адамса (нахождение точек разгона) и неявный многошаговый метод Гира переменного порядка, который устойчив при любой величине шаге интегрирования, и поэтому шаг интегрирования можно выбирать, руководствуясь лишь соображениями точности, а не устойчивости. Для эффективного (в смысле времени вычислений и использования оперативной памяти ЭВМ) управления шагом и порядком метода использовано представление Нордсика [12].

Результаты тестирования расчетного блока на жестких задачах, аналитическое или табличное решения которых известны из литературы (задача Ван дер Поля, модель Робертсона, задача Круга), свидетельствуют о надежности методов и алгоритмов, используемых в программе SPFCSS.

Программный комплекс реализован в среде Delphi 6.0 и предназначен для использования на IBM-совместимых компьютерах в операционных системах Microsoft Windows 2000/XP и старше. Он не предъявляет особо серьезных требований к объемам жесткого диска или оперативной памяти. Для хранения результатов компьютерных экспериментов использована база Microsoft Access 2002 (наличия установленного в системе Microsoft Access не требуется). Комплекс снабжен контекстно-зависимой справкой на русском языке, доступной на любом этапе работы с программой. Возможна модификация и расширение программного комплекса за счет добавления (включения) новых типов материалов и воздействий (в настоящее время для этих целей в пакет заложено 10 модулей) или альтернативных численных методов решения задач. Система может быть интегрирована с другими программными продуктами (например, OriginPro, Microsoft Office).

Созданный комплекс программ позволяет проводить системное изучение процессов и механизмов пластической деформации исследователю, экспериментатору, технологу, физику не обращаясь за помощью к специалистам в области прикладной математики.

Список литературы

1. Orlov A. N. Kinetics of Dislocation / Theory of Crystal Defects. Prague Academia. - 1966. - P. 317-338.
2. Lagneborg R. // Intern. Metals. Rev. - 1972. - Vol. 17. - P. 130-146.
3. Гилман Дж. / Микропластичность. – М.: Металлургия, 1972. - С. 18-37.
4. Попов Л. Е., Кобытев В. С., Ковалевская Т. А. Пластическая деформация сплавов. М.: Металлургия, 1984. 182 с.
5. Попов Л. Е., Пудан Л. Я., Колупаева С. Н. и др. Математическое моделирование пластической деформации. Томск: Изд-во Том. ун-та, 1990 – . 185 с.

6. Колупаева С. Н., Старенченко В. А., Попов Л. Е. Неустойчивости пластической деформации кристаллов. – Томск: Изд-во Том. ун-та, 1994. – 300 с.
7. Колупаева С.Н., Комарь Е.В., Ковалевская Т.А. // Физ. мезомеханика. – 2004. – Т. 7. – Спец. выпуск. – Ч. 1. – С. 23-26.
8. Ковалевская Т. А., Колупаева С.Н., Попов Л.Е. / Структурно-фазовые состояния и свойства металлических систем. – Томск: Изд-во НТЛ, 2004. – С. 135-163.
9. Колупаева С. Н., Семенов М. Е., Пуспешева С. И. // Деформация и разрушение материалов. – 2006. - № 4. – С. 40-46.
10. Колупаева С. Н., Семенов М.Е. Пакет прикладных программ для исследования пластической деформации скольжения в г.ц.к. материалах // Вестник ТГАСУ. 2005. № 1. С. 36–46.
11. Семенов М. Е., Колупаева, С.Н. Свидетельство об официальной регистрации программы для ЭВМ № 20055612381 Slip Plasticity of Face-Centered Cubic v1.0 (SPFCC). Зарегистрировано в Реестре программ для ЭВМ 12 сентября 2005 г. Федеральная служба по интеллектуальной собственности, патентам и товарным знакам.
12. Брайтон Р., Густавсон Ф. Новый эффективный алгоритм для решения дифференциальных систем // Труды института электро- и радиоинженеров. Т. 60. - № 1. - 1972. – С. 124-129.

Проектирование иерархии классов для построения реалистичных изображений с использованием алгоритма трассировки луча

Королёв С. Ю.,

*механико-математический факультет
Пермского государственного университета.
Г.Пермь ул. Коммунистическая 115-326.
email: Korolyovs@mail.ru*

Алгоритм трассировки луча является одним из самых распространённых методов построения реалистичных изображений. Основная идея алгоритма заключается в слежении за распространением световых лучей от источников света до точки наблюдения. Работа алгоритма опирается на законы геометрической оптики. Самым распространённым вариантом этого алгоритма является алгоритм обратной трассировки луча, в котором световые лучи движутся в обратном направлении – от камеры к источникам. С его помощью можно создавать изображения зеркальных и прозрачных поверхностей.

Построение изображений методом трассировки луча позволяет визуализировать сцены с достаточно сложным освещением. Однако этот алгоритм является трудоёмким как с точки зрения использования аппаратных ресурсов, так и с точки зрения программирования. В алгоритме цвет каждого пикселя изображения вычисляется отдельно и только программными средствами. Более того, кроме самого алгоритма, требуется программирование дополнительных возможностей и спецэффектов. Чтобы написать программу, использующую алгоритм трассировки луча, необходимо запрограммировать:

1. алгоритм трассировки луча
2. управление объектами сцены
3. управление источниками света
4. наложение текстур
5. вывод изображения на дисплей
6. различные спецэффекты.

Особую сложность представляет реализация и добавление спецэффектов в уже существующий проект. Под спецэффектом понимается возможность визуализации тех особенностей объектов и источников света, которые не учитываются классическим алгоритмом трассировки луча. К таким эффектам можно отнести создание мягких теней, изображение глянцевых поверхностей, сглаживание, а также эффект ослабления света при распространении в плотной среде. Для каждой сцены важно обеспечить совместимость различных эффектов. Один эффект не должен влиять на работу другого.

Реализация одновременно всех спецэффектов замедляет создание изображения. Поэтому для каждого типа сцен желательно добавлять только определённые эффекты.

Для построения сцен, содержащих различные объекты, текстуры, источники света удобно использовать некоторую заготовку или шаблон, который легко реализуется применением объектно-ориентированного подхода. Использование такого шаблона позволяет с минимальными затратами реализовать построение изображения и с такой же лёгкостью добавлять различного рода улучшения и новые спецэффекты.

Разработана система классов, облегчающая написание приложений, использующих алгоритм трассировки луча как основу для визуализации трёхмерных реалистических сцен. В систему входят следующие классы:

- **Класс описания геометрических объектов.** Используется для описания сцены. Эффективное описание сцены очень важно при реализации алгоритма трассировки луча, т.к. основная вычислительная нагрузка приходится на различные операции с объектами. К каждому объекту во время работы алгоритма применяется три операции:

- ✓ Получить точку пересечения с лучом
- ✓ Получить нормаль в точке пересечения
- ✓ Получить цвет в точке пересечения

Этот класс является базовым для описания других объектов сцены и предоставляет интерфейс для выполнения всех этих операций. Каждый экземпляр класса обладает собственным материалом, текстурой и положением в пространстве. Класс – абстрактный, но на его основе создаётся иерархия классов, которую можно использовать для описания сцены. Для функционирования приложения, при добавлении нового типа объекта не придётся переписывать программный код.

- **Класс источника света.** Описывает источники света сцены, которые могут быть разнообразной формы и различными способами распространять свет. На основе этого класса можно создать разные виды источников и использовать их в приложении без внесения изменений. Каждый источник характеризуется некоторой формой, которая описывается объектом сцены, и цветом. На основе этого класса реализован направленный источник света и конический источник.

- **Класс текстур.** Наложение текстур производится не только на различные геометрические объекты, но и разнообразными способами. Для наложения различными способами к текстуре привязывается класс примитивного объекта, через который происходит отображение текстуры на поверхность объекта. Существует возможность определить свой класс текстур, наследуемый от данного и использовать его в приложении. Описаны классы проективных и процедурных текстур. На основе класса текстур реализуется класс для наложения микрорельефа. Механизмы их функционирования очень похожи. В каждой точке объекта задаётся отклонение нормали. Это позволяет моделировать шероховатые поверхности.

- **Классы трёхмерного вектора и цвета в формате RGB.** Эти классы очень полезны, так как при вычислении формул, используемых в алгоритме трассировки луча, часто применяются операции над векторами и цветом. Классы трёхмерного вектора и цвета имеют общего предка, что позволяет оперировать ими одинаковым образом. Благодаря этому стало возможным построить класс для наложения микрорельефа на основе класса текстур. Классы поддерживают выполнение всех операций над векторами, необходимых при реализации алгоритма трассировки луча. Их использование намного облегчает программирование алгоритмов.

- **Класс камеры.** Используется для определения направления взгляда. Это определяет часть сцены, которая будет визуализироваться. Класс отвечает за генерацию первичных лучей. Наследованием от этого класса можно определить другой класс камеры, обладающей какими-нибудь особенностями. Камера отвечает также за обработку изображения перед выводом на устройство. Например, антиалиасинг, глубина резкости (depth of field) и другие эффекты, получающиеся благодаря оптической системе камеры, могут быть реализованы в этом классе. После обработки изображение передаётся буферу для вывода его на устройство.

- **Класс буфера.** Применяется для вывода изображения на некоторое устройство, например на дисплей или в файл. Он может быть заменен потомком для реализации вывода на другие устройства или вывода другими механизмами. Класс сохраняет выведенное изображение, и оно может быть использовано для вывода на другое устройство. Изображение буфер получает от камеры. Оно может быть передано либо всё целиком, либо построчно.

- **Класс мира.** Управляет источниками и объектами сцены. Он отвечает за вычисление затененности и освещенности объектов и за поиск объекта, который пересекает луч. Эффективный поиск пересечения луча с объектом важен для реализации алгоритма трассировки луча, т.к. это занимает основную часть алгоритма. Для реализации оптимального способа поиска можно переопределить соответствующие методы класса.

- **Класс движка.** Отвечает за работу самого алгоритма трассировки луча и управляет камерой, буфером и миром. Класс содержит основную часть алгоритма трассировки луча. Переопределение

этого класса позволит добавить различные спецэффекты или изменить работу алгоритма трассировки луча.

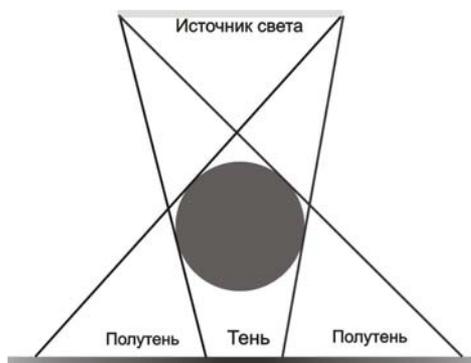
Описанные классы позволяют эффективно и за небольшие сроки создать приложение, реализующее построение реалистичных изображений.

Процесс создания изображения состоит в следующем:

- движку передаётся количество строк, которые необходимо отобразить
- движок запрашивает массив первичных лучей у камеры и находит цвета, соответствующие лучам, вызывая метод RayTrace (этот метод реализует основную часть работы алгоритма трассировки луча)
- полученные цвета передаются камере, где они обрабатываются
- и, наконец, обработанные цвета передаются буферу, который выводит изображение на устройство.

На основе описанных классов были добавлены некоторые спецэффекты, такие как мягкие тени, глянцевые поверхности, сглаживание, а также эффект ослабления света при распространении в плотной среде.

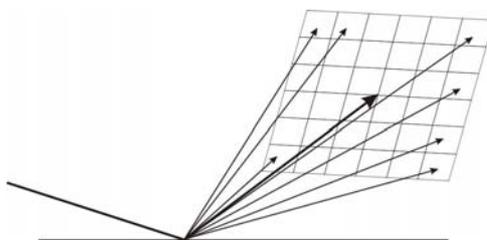
Мягкие тени возникают при наличии объёмных источников света. Чёткая граница тени отсутствует, вместо неё – плавный переход, полутень. При построении теней от точечных источников возможны только две ситуации – источник света полностью виден из точки или источник полностью не виден. Из-за этого тени получаются резкими. Если источник объёмный, то возможна ещё одна ситуация – источник виден частично. В этом случае интенсивность освещения точки неполная. Образуется полутень.



Для создания мягких теней источник света рассматривается как состоящий из нескольких источников. Интенсивность источника распределяется равномерно между всеми частями. Каждая часть обрабатывается как точечный источник.

Мягкие тени встречаются повсюду. Резкие тени можно увидеть редко. Но для того, чтобы существовала возможность создавать такие тени, описан специальный класс точки. Этот класс позволяет описывать точечные источники, от которых и получаются резкие тени.

Глянцевая поверхность – это поверхность, от которой лучи света отражаются не идеально зеркально, а с небольшим отклонением. Отражение окружающих объектов при этом становится немного размытым. Причиной отклонения лучей от направления идеального отражения являются неровности поверхности, которые незаметны невооружённым глазом. Для построения отклонённых лучей используется сетка, перпендикулярная идеально отражённому лучу. Через каждую ячейку сетки проходит луч. Вычисленные интенсивности этих лучей образуют интенсивность отраженного луча.



При распространении света в плотной прозрачной среде, например воде, его интенсивность уменьшается, а цвет приобретает оттенок среды. Если взглянуть на дно кружки с подкрашенной чернилами водой, то дно будет казаться голубым. А если это будет не кружка, а бочка, то дна не будет

видно вообще. Интенсивность луча обратно пропорциональна экспоненте длины пути, пройденного им в среде. Результирующий цвет формируется из цвета луча, вошедшего в среду, и цвета среды.

С помощью описанных классов построены разнообразные сцены с различными объектами, с наложенными на них проективными и процедурными текстурами, а также с различными эффектами. Эти классы позволяют создавать сложные сцены и на их основе можно реализовывать новые классы для построения более сложных сцен.

Реализация выполнена на языке C++ с использованием Microsoft Visual C++.

Список литературы

1. Д. Роджерс. Алгоритмические основы машинной графики. Перевод с англ. М.: Мир, 1989.-512 с.
2. Bram de Greve. Reflections and Refractions in Ray Tracing <http://www.flipcode.com>
3. Zack Waters. Realistic Raytracing. <http://www.cs.wpi.edu>
4. Jacco Bikker. Raytracing Topics & Techniques. <http://www.flipcode.com>
5. John C. Hart. Procedural Solid Texturing. Eurographics/SIGGRAPH Graphics Hardware Workshop, Aug. 1999, pp. 45-53.

Разработка классификационного куба знаний для нового класса информационных систем – электронных информотек

Котиков В. И.,

проф., зам. начальника ИВЦ,

Московский государственный технический университет

гражданской авиации

e-mail: kotikov@mstuca.ru

Разработка нового класса информационных систем – электронных информотек (ЭИ), в которых должно осуществляться одновременное формирование электронного каталога и базы полнотекстовых интеллектуальных информационных ресурсов непосредственно их создателями с использованием сетевых технологий, требует определения их сущностной модели и классификации.

В зависимости от вида носителя интеллектуальной информации, под которой в дальнейшем будем понимать информацию, непосредственно создаваемую человеком на основе его знаний, была проведена классификация информотек: традиционные и гибридные библиотеки, электронные и электронно-биологические информотеки, знанияеки [1]. Несмотря на множественность классов и видов информотек, все они представляют систему, обеспечивающую сбор, накопление, хранение, обработку и передачу информационных ресурсов пользователям по их запросам. Под электронной информотекой будем понимать систему, в которой авторы интеллектуальных информационных ресурсов сами создают полнотекстовый электронный документ и формируют в ней фонды и электронные каталоги на электронных носителях, а для их доставки в базы электронной информотеки и пользователям по их запросам применяются высокоскоростные электронные транспортные артерии. В этом случае сущностная модель ЭИ включает в себя четыре важнейших элемента [2]:

- авторы и пользователи информационных ресурсов;
- информационные транспортные артерии;
- программно-аппаратный комплекс управления информотекой;
- база информационных ресурсов.

Проведенный анализ существующих классификационных систем позволил сформулировать требования к новой классификационной системе: система должна быть построена по иерархическому признаку, обеспечивать эффективный поиск пользователю релевантных ЭД в ЭИ с учетом ее динамического развития и не требовать от авторов и пользователей специальных знаний в области описания ЭД, принадлежащих к различным областям науки, техники и образования. Такая классификационная система должна содержать прежде всего вербальное описание классификационных единиц, объединенных в единый рубрикатор, на всех иерархических уровнях которого применяется

алфавитный принцип размещения единиц классификации, что отвечает главному постулату воспроизводства знаний человечеством: независимо от развития науки, техники, культуры и образования все виды человеческой деятельности носят равноправный характер.

Выбор числа уровней классификации в информотеках определяется объемом накапливаемых информационных ресурсов и их влиянием на эффективность работы информационно-поисковых систем. На рис. 1 представлено ранжирование информотек в зависимости от объема накопленных документов. Информотека личных коллекций, насчитывающая в среднем 900 – 1000 томов печатных изданий, не требует многоуровневой классификации. Для классификации библиотечных ресурсов в 10-15 тысяч изданий одноуровневая классификационная система приведет к большим временным затратам на поиск необходимого издания. Переход к информотекам, в которых должны накапливаться сотни тысяч единиц хранения документов, требует построения классификационных систем с большим числом иерархических уровней. Эта область университетских библиотек и здесь широко используется десятичная классификационная система Дьюи и ГРНТИ ВИНТИ (рис. 1). Информационные ресурсы национальных библиотек и Интернет-ресурсы, как показали наши исследования, не вписываются в существующие классификационные системы.

Многоуровневые классификационные структуры тоже обладают недостатками. При слишком глубокой иерархической классификации снижается эффективность работы информационно-поисковой системы, так как пользователь на свой запрос не получает необходимой полноты информационных ресурсов и требуется многократное обращение к системе для достижения желаемого результата. Поэтому существует оптимальная структура построения классификационной системы, которая обеспечивает эффективность работы информационно-поисковых систем электронных информотек и получение релевантных документов пользователем на свой запрос при заданных значениях полноты и точности.

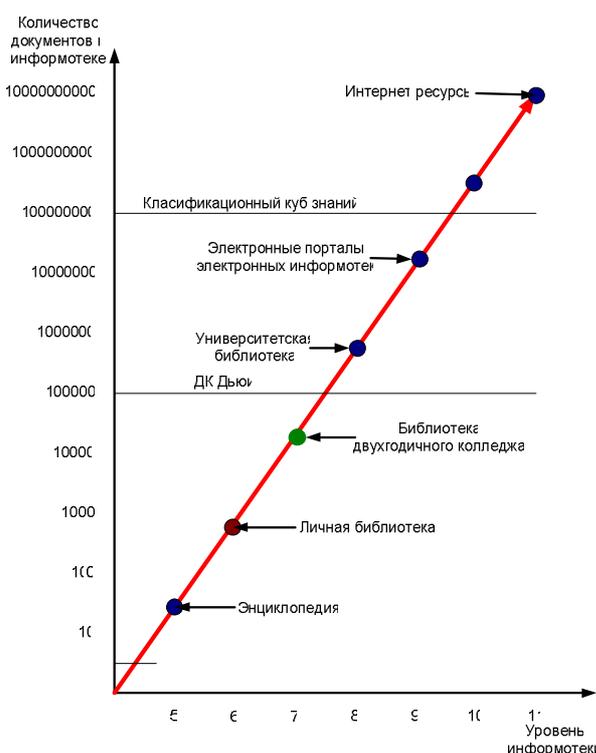


Рис 1. Ранжирование информотек в зависимости от объема документов

Для выбора оптимальной структуры классификационной системы ЭИ был проведен анализ рубрикаторов [4], выпущенных ВИНТИ и предназначенных для систематизации всего потока научно-технической информации, создаваемой в нашей стране. Статистический анализ рубрикаторов ВИНТИ показал, что первый классификационный уровень рубрикатора содержал не более 73 классификационных единиц. На втором классификационном уровне распределение элементов классификации носило случайный характер и колебалось в пределах от 0 до 39. На третьем уровне классификационной системы максимальное значение числа классификационных параметров достигало 70. Среднее значение классификационных параметров для всех выпусков ГРНТИ лежало около 11

независимо от развития науки и техники, что подчеркивает определенную устойчивость классификационной системы.

Полученные результаты позволили определиться и с оптимальным количеством иерархических уровней классификационной системы рубрикатора ЭИ. Она должна быть построена на основе трехуровневой универсальной динамической классификационной (УДК) модели. Каждый ее уровень должен содержать не более ста вербальных элементов деления, что соответствует двум разрядам десятичного кода. В результате такой оптимальной структуры возможно сформировать 10^6 элементов классификации по всем областям человеческой деятельности, что на три порядка превышает существующую десятичную классификационную систему, предложенную Дьюи [5]. В таком рубрикаторе создается определенная избыточность, что отвечает обязательному требованию при проектировании любых электронных информотек, работающих в едином телекоммуникационном пространстве обмена информацией, и обеспечивается эффективная работа информационно-поисковых систем. Такая классификационная система отражает знания, которые создает человек в процессе своей познавательной деятельности. Поэтому в дальнейшем такую классификационную систему назовем классификационным кубом знаний. Сопоставление модели классификационного куба знаний и отраженных в его структуре результатов классификации научно-технической информации на примере 5-го выпуска ГРНТИ ВИНТИ показывает, что весь классификационный потенциал ГРНТИ ВИНТИ 5-го выпуска занимает в классификационном кубе знаний всего 0,75%.

Если в какой-то области человеческой деятельности число классификационных единиц и превысит значение 100, то это не меняет общей структуры классификационного куба знаний, так как в основе его построения лежит вербальный способ классификации. Такой способ классификации для живой природы использовал Линней и благодаря этому она обладает большим динамизмом в случае формирования новых классификационных единиц на каждом иерархическом уровне в случае открытия новых видов. При таком подходе каждому вербальному элементу классификации может ставиться в соответствие некоторый числовой код, который можно использовать при кластеризации накапливаемых информационных ресурсов и осуществлении поиска с использованием как вербального, так и цифрового запроса. Для информационно-поисковых систем ЭИ таким кодом является двоичное исчисление, для описания интеллектуальных информационных ресурсов, создаваемых на бумажных носителях, - это десятичный код.

Оценку эффективности алгоритмов поиска, которые закладываются в работу информационно-поисковых систем (ИПС) ЭИ, целесообразно проводить путем сравнительного анализа стратегий поиска, реализуемых пользователем путем кластеризации всего фонда электронной информотеки. Идея использования различных поисковых образов связана с увеличением вероятности получения релевантных документов [3]. Учитывая множественность запросов пользователей, которые можно в некоторых случаях классифицировать по определенным группам, для решения поставленной задачи была проведена оценка эффективности различных алгоритмов поиска. С учетом трехуровневой классификационной системы рубрикатора, на основе которого пользователь может сформировать свой запрос поиска, вся база генерального каталога и полнотекстовые ЭД разбиваются на три уровня с формированием отдельных кластеров на каждом уровне, соответствующих элементу классификационного куба знаний (ККЗ).

В табл. 1. приведены значения выигрыша в получении пользователем релевантных информационных карт при различных объемах накопленных электронных документов в ЭИ и алгоритмах поиска, основанных на использовании классификационного куба знаний и ДК Дьюи с равномерным распределением электронных документов по рубрикатору.

Из полученных результатов следует, что применение классификационного куба знаний обеспечивает больший выигрыш при получении релевантных документов на запрос. Однако на начальном этапе формирования ЭИ этот теоретический выигрыш оказывается нереализованным при робастном режиме работы программно-аппаратного комплекса ЭИ, так как в большинстве классификационных единиц рубрикатора второго и третьего уровней будут отсутствовать электронные документы. Поэтому для реализации эффективных алгоритмов поиска на всех этапах развития ЭИ необходимо использовать адаптивный режим работы программно-аппаратного комплекса в саморазвивающихся электронных информотеках, при котором сохраняется полнота использования всего рубрикатора для авторов, создающих свои информационные ресурсы во всех областях человеческой деятельности, и частичное использование рубрикатора при формировании запроса пользователем. При этом достигается теоретический выигрыш при кластеризации всего массива документов с исключением вариантов поиска в тех классификационных единицах, где отсутствуют на момент поиска электронные документы.

Таблица 1. Зависимость выигрыша от стратегии поиска и объема накопленных полнотекстовых электронных документов

Объем электронного каталога ЭД, хранимых в ЭИ, шт.	Доля электронного каталога ЭД, доступного для поиска, Дьюи/ККЗ	Объем электронного каталога ЭД, доступного для поиска (рубрикатор Дьюи)	Объем электронного каталога ЭД, доступного для поиска (ККЗ),	Поисковый выигрыш по рубрикатору Дьюи	Поисковый выигрыш по ККЗ
1000 5000 10000 100000 1000000	$0,1/10^{-2}$ $0,1/10^{-2}$ $0,1/10^{-2}$ $0,1/10^{-2}$ $0,1/10^{-2}$	100 500 1000 10000 100000	10 50 100 1000 10000	3,3	5,6
1000 5000 10000 100000 1000000	$0,01/10^{-4}$ $0,01/10^{-4}$ $0,01/10^{-4}$ $0,01/10^{-4}$ $0,01/10^{-4}$	10 50 100 1000 10000	0,1 0,5 1 10 100	5,6	10,2
1000 5000 10000 100000 1000000	$10^{-3}/10^{-6}$ $10^{-3}/10^{-6}$ $10^{-3}/10^{-6}$ $10^{-3}/10^{-6}$ $10^{-3}/10^{-6}$	1 5 10 100 1000	0,001 0,005 0,01 0,1 1	10,2	14,8

Использование классификационного куба знаний при систематизации интеллектуальных информационных ресурсов по всем областям человеческих знаний позволяет реализовать оптимальные алгоритмы работы информационно-поисковых систем как для авторов, так для пользователей, при использовании адаптивного режима. Создаваемые электронные информотеки являются динамическими информационными системами, и для повышения эффективности работы их информационно-поисковых систем необходимо использовать рубрикатор с алфавитно-вербальным принципом описания классификационных единиц, а сами электронные информотеки должны работать в адаптивных режимах.

Список литературы

1. Котиков В. И. Электронная библиотека: от концепции до реализации //Научный вестник МГТУ ГА, сер. «Информатика». – М.: МГТУ ГА, вып. 45, 2003, - С. 53-64
2. Котиков В. И. Электронная информотека – новый класс информационных систем //Труды Международного форума по проблемам, науки, техники и образования. Том 1. /Под ред. В. П. Савиных, В. В. Вишневого. – М.: Академия наук о Земле, 2004 – С. 131-133
3. Дж. Солтон. Динамические библиотечно-информационные системы //Пер. с англ. под ред. В. Р. Хисамутдинова. – М.: Мир, 1979, - 558 с.
4. Государственный рубрикатор научно-технической информации. - 5-е изд. – М.: ВИНТИ, 2001. – 391 с.
5. Универсальная десятичная классификация. – 3-е изд. – М.: Издательство стандартов, 1987
6. Котиков В. И., Денисова Е. М. Разработка классификационной системы и рубрикатора для информационно-поисковых систем электронных информотек. - //Тезисы докладов 8-й Международной конференции EVA 2005 Москва. – Москва, декабрь

Нейросетевые агенты в аппаратно-программной системе контроля роторного оборудования

Крехов Е. В.

*Инженер, программист, Московский государственный университет приборостроения и информатики,
Серпуховской факультет
E-mail: krehov@yandex.ru*

Нурматова Е. В.

*Кандидат технических наук, доцент Московского государственного университета приборостроения и информатики, Серпуховской факультет
E-mail: elenur@yandex.ru*

142200, г. Серпухов, Московская обл., ул. 2-я Московская, д. 44/10

В данной работе рассматривается решение задачи прогнозирования технического состояния магистрально-насосных агрегатов применением искусственных нейронных сетей (ИНС). Применение ИНС позволяет снизить расходы на программное обеспечение и повысить эффективность прогнозирования, тем самым увеличивая эксплуатационную надежность оборудования.

Ключевые слова: техническая диагностика; вибродиагностика; эмулятор, обучение, качество работы нейронной сети; MLP сеть.

В процессе эксплуатации роторного оборудования возникают задачи определения и прогнозирования технического состояния каждого отдельного агрегата или его узлов. Определение и прогнозирование технического состояния агрегата требуется производить без его разборки. Эту задачу решает техническая диагностика в рамках технологии технического обслуживания (ТО) по состоянию (ТОС) [1]. В основу ТОС положены анализ вибрационного сигнала и методы анализа трендов общих уровней вибрации, чтобы определить текущее состояние узлов машины и предсказать, когда потребуется ТО.

Сущность вибродиагностики заключается в следующем: во время работы агрегата движение деталей сопровождается соударением, в результате которого по механизмам распространяются колебания. При износе или возникновении дефектов в механизме нарушаются кинематические связи между деталями, вследствие чего меняется характер вибрации. Это свойство используется для оценки технического состояния агрегатов по параметрам вибрации. Сложный многокомпонентный колебательный процесс описывается выражением:

$$x_k(t) = \sum_{k=\frac{1}{p}}^q A_k \cdot \cos(k\omega t - \varphi_k) , \quad (1)$$

где A_k, φ_k - амплитуда и фазовый угол k -ой гармоники; k - порядок гармоники; p, q - целые положительные числа; ω - угловая частота вращения ротора [2].

В технической литературе [1,2] описана целая гамма диагностических признаков неисправностей роторного оборудования, которые нуждаются в уточнении для каждого конкретного механизма и условий эксплуатации (климат, режимы работы и др.).

Для роторного оборудования (на примере магистральных насосных агрегатов) характерны следующие виды неисправностей [3]:

- 1) неуравновешенность (дисбаланс) ротора;
- 2) несоосность (расцентровка) валов агрегата;
- 3) дефекты подшипников скольжения;
- 4) биеение шнека;
- 5) электромагнитные дефекты двигателя.

В состав системы выполняющей ТОС входят программные компоненты, предоставляющие пользовательский интерфейс экспертам, для определения и прогнозирования технического состояния роторного оборудования. В основу работы экспертов положено выделение экстремальных векторов виброскорости, характерных для каждого вида неисправности. Экстремальные вектора выделяются из диагностических спектров, пример одного из которых изображён на рисунке 1. Для автоматизации процесса предусматривается внедрение нейросетевых агентов в программные компоненты системы. Нейросетевые агенты выполняют определение и прогнозирование технического состояния по выделенной неисправности.

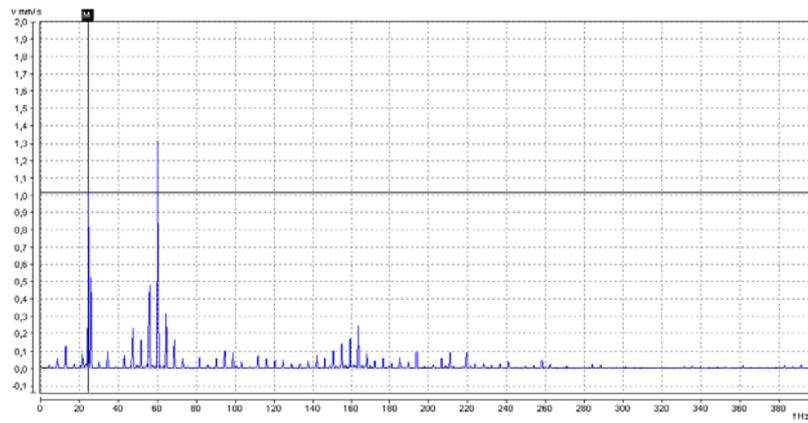


Рисунок 1 - Диагностический спектр при неисправности «расцентровка»

Для определения технического состояния используются нейронные сети Кохонена по одной на каждую неисправность. Таким образом, задача каждой сети сводится к отнесению входного вектора к одному из двух классов – «неисправность присутствует» и «неисправность отсутствует». Поэтому выходной слой сети содержит нейроны с функцией активации SOFTMAX, необходимой для выделения вероятности возникновения неисправности. Задача обучения такой нейронной сети – научить выходной слой активировать один и тот же нейрон для похожих входных векторов.

Алгоритм обучения слоя Кохонена [4]:

- 1) присвоение начальных значений весовым коэффициентам;
- 2) подача на вход одного из векторов, обучающего множества;
- 3) расчет выхода слоя Кохонена и определение номера выигравшего нейрона m_0 , выход которого максимален;
- 4) коррекция весов только выигравшего нейрона m_0 :

$$w_{m_0} = w_{m_0} + \alpha(x - w_{m_0}), \quad (2)$$

где w_{m_0} - вектор весов нейрона m_0 , x - входной вектор. α - скорость обучения.

Выход нейронов слоя Кохонена рассчитывается по формуле:

$$D = \sum_{i=5} x_i c_i, \quad (3)$$

где D - выход нейрона, x_i - компонент вектора x , c_i - весовой коэффициент.

Выигравший нейрон определяется по следующей формуле:

$$m_0 = \max_m (D^m), \quad (4)$$

где D^m - выход нейрона m , m_0 - выигравший нейрон.

После обучения сети, выделяются реакции сети на значения входных векторов, характеризующих исправные и неисправные состояния узла. В ходе тестовых испытаний были получены тестовые результаты, графическая интерпретация на примере неисправности «расцентровка» приведена на рисунке 2. Для контроля работы классификатора используется ошибка обобщения, с учётом баланса относительных частот встречаемости классов[5]:

$$BER = \frac{1}{2} \left(\frac{b}{a+b} + \frac{c}{c+d} \right), \quad (5)$$

где a и b - число правильных и ошибочных классификаций примеров класса «неисправность присутствует», d и c - число правильных и ошибочных классификаций примеров класса «неисправность отсутствует».

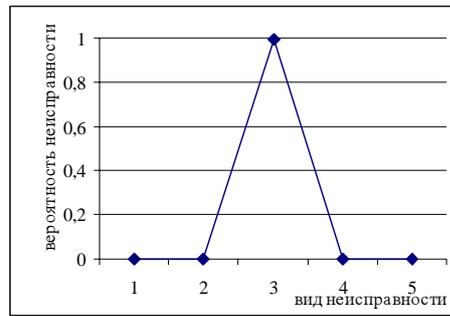


Рисунок 2 - Реакция нейронной сети, при определении неисправности «расцентровка»

Нейронные сети являются популярной методикой для моделирования нелинейной многомерной динамики систем, вследствие их уникальных аппроксимационных свойств [6]. Поэтому в качестве программного компонента, выполняющего прогнозирование технического состояния роторного оборудования, используется эмулятор нейронной сети.

В ходе моделирования анализируются различные топологии нейронных сетей: многослойный перцептрон (MLP - Multilayer Perceptron), сеть типа рациональной базисной функции (RBF - Radial Basis Function), обобщённо-регрессионная нейронная сеть (GRNN - Generalized Regression Natural Network). Выбор нейронной сети производится на основании размера поступающей обучающей выборки и качества получаемых результатов по Парето. Размер обучающей выборки (300 многомерных векторов), влияет на скорость обучения и позволяет выбрать архитектуру нейросети (топологию и алгоритм обучения).

Качество работы нейронных сетей определяется из соотношения [7,8]:

$$K_p = \frac{\sigma^2}{\sigma_o} \quad (6)$$

где K_p - коэффициент, определяющий качество работы нейронной сети, σ^2 - дисперсия, определяющая ошибку прогноза, σ_o - стандартное отклонение обучающих данных.

Экспериментальные испытания показали, что наилучшим решением данной задачи является MLP сеть. Результаты применения нейросетевого агента в графической интерпретации изображены на рисунке 3. Для контроля качества работы нейросетевой модели, выполняющей прогнозирование неисправности, используется отношение (6).

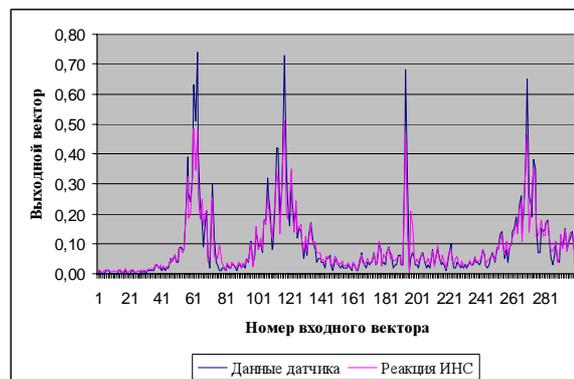


Рисунок 3 - Прогноз искусственной нейронной сети и реальные данные

Эмулирование нейронных сетей производится по средствам разработанных нами программных модулей, динамически подключаемых к прикладным программам и работающим под управлением операционной системой Windows, с использованием программного интерфейса приложение Win32 [9].

Результаты экспериментов показывают, что использованием нейросетевых агентов в системе аппаратно-программного контроля технического состояния роторного оборудования позволяет корректно определять неисправности и повторять динамику работы оборудования при прогнозировании. Таким образом, использование нейросетевых агентов позволяет автоматизировать работу аппаратно-программной системы по контролю технического состояния роторного оборудования, исключая привлечение группы экспертов в технологические процессы ТОС.

Список литературы.

1. Изерманн Р. Перспективные методы контроля, обнаружения и диагностики неисправностей и их применение. Приборы и системы управления. 1998 №4.
2. Александров А. А., Барков А. В., Баркова Н. А., Шафранский В. А. Вибрация и вибродиагностика судового электрооборудования.- Л.: Судостроение, 1986.
3. Текин А. Д. Разработка и исследование экспертных систем диагностики магистральных насосных агрегатов на базе портативных виброанализаторов: диссертация канд. техн. наук: 05.11.13 - М.2000 - 152 с.
4. Крехов Е. В. Методика определения технического состояния магистральных насосных агрегатов с использованием нейронных сетей // "Нейроинформатика - 2006", Сборник научных трудов – М.: МИФИ, 2006 – с 88 - 96.
5. Терехов С. А. Технологические аспекты обучения нейросетевых машин // "Нейроинформатика - 2006", Лекции по нейроинформатике. - М.: МИФИ, 2006–с 7-72.
6. Горбань А. Н. Обобщённая аппроксимационная теорема и свойства нейронных сетей. Соросовский образовательный журнал, 1998.
7. Крехов Е. В., Нурматова Е. В. Анализ алгоритмов обучения нейронных сетей при прогнозировании технического состояния магистральных насосных агрегатов // Информационные технологии – 2006.
8. Крехов Е. В. О нейросетевом прогнозировании технического состояния роторного оборудования, сборник тезисов – М.: МГУ, 2006 – с. 27 - 28.
9. Крехов Е. В. Методика программно-модульного эмулирования нейронных сетей // Программное и информационное обеспечение систем различного назначения на базе персональных ЭВМ, Сборник научных трудов – М.: МГУПИ, 2006.

Исследование характеристик периодического потока сигналов в сложных вычислительных сетях

Кудрявцев А. А., н.с.

*Институт проблем управления им. В. А.Трапезникова РАН
Россия, 117997, Москва, Профсоюзная ул., 65
E-mail: gvozd@skypoint.ru*

Промыслов В. Г., с.н.с.

*Институт проблем управления им.В.А.Трапезникова РАН
Россия, 117997, Москва, Профсоюзная ул., 65*

Поскребышев И. А., н.с.

*Институт проблем управления им. В. А.Трапезникова РАН
Россия, 117997, Москва, Профсоюзная ул., 65*

В данной статье были сформулированы основные причины нелинейности передачи сигнала в распределенной сети. Показано, что для некоторых случаев (особенно для систем с «реальным временем») требуется учет нелинейности характеристик ЛВС при разработке алгоритма программы. Для примера рассмотрена распределенная система диагностирования элементов АСУ ТП с встроенными SNMP агентами

In the paper there were formulated main reasons of nonlinearity of signal transmission in a distributed network. It was shown that in some cases (especially for real-time systems) within development of the program algorithm, accounting of LAN parameter nonlinearity is required. For example, it was considered the distributed diagnosing system of the process control system elements with inbuilt SNMP-agents.

В настоящее время вырос масштаб и сложность компьютерных сетей для построения современных АСУ ТП. И как следствие, увеличилась доля автоматизации управления распределенными вычислительными ресурсами, взаимодействующими между собой.

На Рис.1 представлена модель передачи сигнала в сети АСУ ТП. В качестве источника сигнала выступают встроенные в диагностируемые устройства агенты.

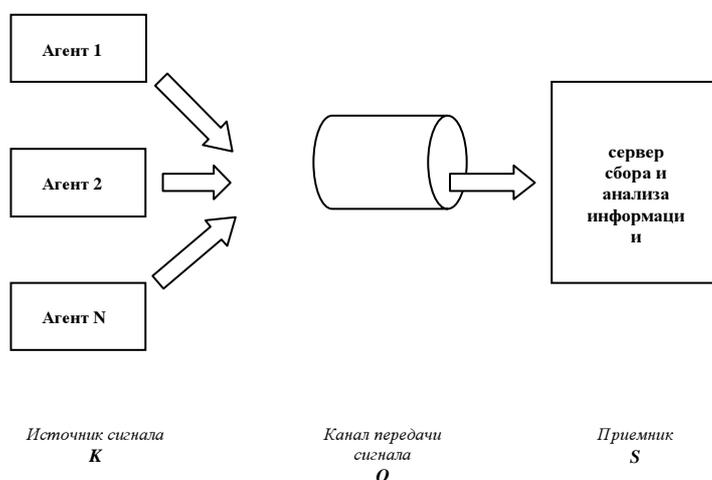


Рис. 1 Модель передачи сигнала в АСУ ТП

Распространение сигнала от агента к серверу может носить нелинейный характер вследствие задержек при прохождении через устройства компьютерной сети (ОС, сетевое устройство компьютера, коммутатор). По этому даже в случае регулярного (периодического) потока сигналов на выходе каждого из агентов входной поток сигналов на сервер будет иметь некоторое стохастическое распределение.

Определение, каким будет спектр сигнала на входе в сервер в зависимости от канала связи и от спектра сигнала от каждого из агентов – важная задача в распределении ресурсов в сети АСУ ТП.

Программа Колмогорова и секвенциальные интеллектуальные системы

Кузичев А. С., МГУ

Под секвенциальными интеллектуальными системами понимаются двухъярусные исчисления с принципом комбинаторной полноты, вводимые секвенциально по Генцену на основе бестиповой теории алгоритмов в форме неразрешимых исчислений Шейнфинкеля–Карри–Чёрча с принципом комбинаторной полноты (или λ -полноты), без известных ограничений теоремами Гёделя о неполноте: первый ярус исчислений, обеспечивающий принцип интеллектуальности, задаёт неограниченное теоретико-множественное свёртывание Кантора в алгоритмической (вычислительной) форме, например, бестипового исчисления λ -конверсии Чёрча с λ -секвенциями (имеющими вид $a \rightarrow v$); второй ярус исчислений задаёт классическую логику (предикатов 1-го порядка) в секвенциальной (без постулируемого правила сечения) форме Генцена с дедуктивными секвенциями вида $\Gamma \Rightarrow \Delta$; связь между ярусами обеспечивают правила λ -сечения $*\lambda$ и $\lambda*$ (пишем: $*\lambda*$), введенные автором и названные канторовскими, с целью синтаксического перенесения принципа интеллектуальности на 2-ой ярус исчислений. Причем все логические постулаты распространяются на объекты исчислений Шейнфинкеля–Карри–Чёрча. Такие двухъярусные (интеллектуальные) исчисления строятся, следуя идее ступенчатых конструкций А.Н. Колмогорова и А.А. Маркова, исследуются автором на механико-математическом факультете МГУ с 1968 г. и публикуются с 1970 г.

Формулируется программа Колмогорова по основаниям КМ (классической теоретико-множественной математики в её целостности). Обсуждаются историко-методологические вопросы её

становления, связь с Центральной проблемой Гильберта (построения доказуемо полных и доказуемо непротиворечивых оснований КМ) и проблематикой искусственного интеллекта.

Одновременно решаются две хорошо известные проблемы оснований наук: проблема введения логических операторов в алгоритмические комбинаторно полные (λ -полные) неразрешимые исчисления Шейнфинкеля–Карри–Чёрча и Центральная проблема Гильберта построения доказуемо полных и доказуемо непротиворечивых оснований КМ.

В силу теорем Гёделя о неполноте некоторых аксиоматических теорий 1-го порядка и в силу парадоксов типа парадокса Рассела, доказуемо полные и доказуемо непротиворечивые интеллектуальные системы (такие, например, как интеллектуальные системы компьютерных логик), в частности, реализующие программу Колмогорова по КМ, естественно строить в их основной логико-предикатной части не как всем известные аксиоматические формульные теории 1-го порядка с собственными нелогическими аксиомами, а как вторые, логические, ярусы секвенциальных двухъярусных теорий, базирующихся на знаменитых результатах Кантора, Гильберта, Шейнфинкеля, Карри, Чёрча и Генцена в основаниях наук и введённых автором по публикациям с 1970 года. Основное внимание сосредоточим на КМ, естественно предполагая распространение результатов на любые интеллектуальные конструкции в их целостности.

Программа Колмогорова по основаниям КМ состоит в том, что при построении исчисления, решающего Центральную проблему Гильберта, надо учитывать две компоненты КМ (классической теоретико-множественной математики) – вычислительную (алгоритмическую) и дедуктивную (логическую) и одновременно отражать без ограничений два канторовских принципа теории множеств – неограниченное свёртывание и неограниченную логику, применением которых строятся все выводы КМ.

В работе предлагается вариант реализации программы А.Н. Колмогорова по основаниям математики. Показывается, что КМ (классическая теоретико-множественная математика) доказуемо полностью и доказуемо непротиворечиво (как абсолютно, так и относительно отрицания \neg) представляется одним интеллектуальным исчислением (теорией).

Тем самым Центральная проблема Гильберта построения доказуемо полных и доказуемо непротиворечивых (как абсолютно, так и относительно отрицания \neg) оснований классической теоретико-множественной математики КМ в виде одного (хотя и двухъярусного) интеллектуального исчисления решается автором по Колмогорову.

На основании сказанного интеллектуальные системы строятся как ДИС (двухъярусное секвенциальное исчисление) \mathcal{M} (без постулируемого правила сечения) из [1, 2]. Результат получает завершение ниже формулируемой Теоремой Cut (о допустимости в ДИС правила сечения), из которой в дополнение к абсолютной непротиворечивости [1, 2] вытекает непротиворечивость двухъярусной интеллектуальной системы относительно отрицания.

Формулировка Теоремы Cut (с использованием понятий и терминов, вводимых аналогично логико-предикатным генценовским)

Если в ДИС выводимы секвенции $(\Delta \Rightarrow A, A^r)$ и $(A^r, \Gamma \Rightarrow \Theta)$, то в ДИС существует вывод секвенции $(\Delta, \Gamma \Rightarrow A, \Theta)$, где A есть \mathcal{M} -формула с индикатором r ; $\Delta, \Gamma, A, \Theta$ суть наборы оснащенных \mathcal{M} -формул.

Доказательство Теоремы непосредственно следует из следующей леммы о смешении в ДИС, по формулировке и доказательству являющейся обобщением генценовской леммы о смешении в логике предикатов.

Лемма о смешении в ДИС

Если \aleph и \aleph – выводы секвенций $E = (\Delta \Rightarrow A)$ и $G = (\Gamma \Rightarrow \Theta)$ в ДИС, A^r – оснащенная \mathcal{M} -формула, r – индикатор, то в ДИС можно построить вывод \aleph секвенции $H = (\Delta, \Gamma_A \Rightarrow A_A, \Theta)$, где выражение вида Φ_A обозначает результат вычеркивания из набора Φ всех оснащенных \mathcal{M} -формул B^r таких, что B есть \mathcal{M} -формула и $B \leftrightarrow A$ (B конвертируется в A , то есть в исчислении λ -конверсии [1, IV, п. б] выводимы λ -секвенции $A \rightarrow B$ и $B \rightarrow A$), здесь $=$ выступает как символ равенства по определению, Δ, A, Γ и Θ – наборы оснащенных \mathcal{M} -формул.

Итак, два яруса построенной двухъярусной интеллектуальной системы (ДИС), как аналоги двух составляющих КМ, позволяют в ДИС применением двух принципов канторовской теории множеств, представленных в ДИС логическими и алгоритмическими постулатами, получить в соответствии с программой А.Н. Колмогорова доказуемо полным и непротиворечивым образом все выводы КМ без ограничений естественно на основе колмогоровского пакета законов рассуждений, заданного указываемыми по постулатам ДИС всеми свойствами выводимости при замене символа дедуктивной секвенции \Rightarrow на знак выводимости \vdash , или секвенциально в ДИС.

Первый, алгоритмический ярус ДИС с его неразрешимостью используется не только при построении 2-го, логического яруса ДИС, но и в других случаях, например, при задании исходных элементов (М-термов и М-формул) исчисления ДИС.

Негёделевость ДИС, как и невозможность формализации ДИС в теориях первого порядка, построенных на пути Фреге, обеспечивается неразрешимостью исчислений Шейнфинкеля–Карри–Чёрча, используемых как в теории ДИС, так и в её метатеории (см. также [3–7]).

Список литературы

1. Кузичев А. С. Вариант формализации канторовской теории множеств. // Доклады Академии наук, 1999, т. 369, № 6, с. 740–742.

2. Кузичев А. С. Решение проблемы Гильберта по Колмогорову. // Доклады Академии наук, 2000, т. 371, № 3, с. 303–306.

3. Кузичев А. С. О негёделевской перестройке арифметики и других аксиоматических теорий первого порядка по Колмогорову. Доказательство их непротиворечивости. М., Изд-во мех-мат. ф-та МГУ, 2004, 36 с.

4. Кузичев А. С. О роли теорем Гёделя о неполноте в основаниях наук. // “Философия и будущее цивилизации. Тезисы докладов и выступлений IV Российского философского конгресса”, в 5 тт., Том 1, М., Современные тетради, 2005, с. 726–727.

5. Кузичев А. С. Колмогоровские основания математики и образование. // В сборнике научных трудов “Математика. Компьютер. Образование”. Выпуск 12, том 1. Москва–Ижевск, НИЦ “РХД”, 2005, с. 100–112.

6. Кузичев А. С. Секвенциальное построение интеллектуальных систем с принципом комбинаторной полноты. В сб. научных трудов “Машины. Люди. Ценности”, Курган, Изд-во Курганского гос. ун-та, 2006, с. 31–32.

7. Кузичев А. С. Программа Колмогорова, интеллектуальные системы и теоремы Гёделя о неполноте. В кн. “Искусственный интеллект: междисциплинарный подход”, М., ИИнтелЛЛ, 2006, с. 330–346.

Основы технологии по мониторингу и анализу состояния дорог, характеристик автотранспортных потоков (АТП) и правил дорожного движения (ПДД), дисциплины водителей

Докладчик Кузьмин Д. М.,

e-mail: hugodi@mail.ru

Гл. консультант: д.ф.-м.н. профессор Буслаев А. П.,

e-mail: busl@math.madi.ru

г. Москва, МАДИ, Ленинградский проспект, 64., каф. Высшей математики

В работе описана методология стационарного и мобильного видео- (аудио-) мониторинга и анализа состояния автомобильных дорог, характеристик транспортных потоков и факторов, определяющих безопасность движения. Использование этой технологии позволяет вести оперативный контроль условий дорожного движения с целью управления, создает предпосылки для существенного сокращения числа ДТП и неотвратимости наказания за нарушение ПДД.

Автоматическая обработка видео и аудиоинформации о характеристиках автотранспортных потоков разрабатывается в МАДИ (ГТУ) с начала 2000-х годов. Технологии, выполняющие компьютерную обработку в лабораторных условиях видеозаписи многополосного автотранспортного потока, обрабатывались на примыкающих к Москве участках автомагистралей «Центральная Россия». Впоследствии задачи оценки интенсивности, плотности, состава автотранспортных потоков были

реализованы в реальном режиме времени без промежуточной лабораторной обработки видеоматериала. Проведена также работа распознавания автотранспортного потока по аудиосигналам, что имеет практический смысл в условиях недостаточной видимости. В процессе работы выяснилось, что спектр практических задач, которые можно решать посредством обработки сигнала видеисточника, чрезвычайно широк. Большой пакет алгоритмов по автоматическому мониторингу на автодороге, которые могут работать и параллельно, получил название MonStr (Street Monitoring). Данный пакет программ основан на анализе видео- или фотопотока, получаемого с видео- и фотокамер, которые установлены на мобильной лаборатории и позволяют проводить исследования и многоцелевой мониторинг. Среди целей можно отметить: мониторинг состояния дороги, оценка количественных и качественных характеристик состава АТП, оценка средств управления дорожным движением, оценка соблюдения правил дорожного движения. Видео- и фотоматериал, получаемый с источника, обрабатывается на компьютере для каждой из поставленной задачи. Работа программы основана на анализе матрицы цветности пикселей каждого из кадров потока. Пиксел – неделимая точка в графическом изображении, т.н. наименьший адресуемый элемент растрового изображения (кадра, фотографии).

Источниками информации, с помощью которых осуществляется мониторинг, являются устройства, классифицируемые по типовому признаку получения данных: видеокамера, фотоаппарат, прибор навигации (GPS), микрофон, различные датчики (света, колебаний) и т.д. В зависимости от поставленной задачи и разработанной схемы алгоритма устройство может находиться как в салоне мобильной лаборатории (МЛ), так и закреплено снаружи.

Так, крепёж видеокамеры осуществляется на внешнем штативе, прикрепленном на корпусе мобильной лаборатории (например, на багажнике), однако в зимнее время года, при низких температурах, крепёж может быть осуществлен внутри МЛ на специально оборудованном штативе. Работу всех устройств координирует компьютер с установленным пакетом программ (для каждой подзадачи своя подпрограмма). Компьютер установлен в салоне МЛ и осуществляет двунаправленную работу и сбор данных со всех подключенных устройств, управляет режимом записи виде- и фотоаппаратуры, осуществляет сбор данных о текущих координатах, получаемых с прибора навигации.

После получения центральным компьютером данных с видеокамеры, матрица кадра видеопотока сканируется в реальном режиме времени по маске заданной области и выдает команду в зависимости от реализуемой задачи либо видеокамере на переход в режим записи, либо фотоаппарату на режим фотографирования. Независимо от выданной команды, полученная информация сохраняется в компьютере, что позволяет в дальнейшем осуществить анализ и обработку данных в лаборатории. Также компьютер запоминает в памяти координаты навигации (широта и долгота) и сохраняет их в базе данных, что позволяет осуществить привязку с точностью до 1м.

Разрабатываемая технология предназначена для автоматической обработки информации в реальном времени, получаемой с видеискателя (камеры, фотоаппарата и др.) о состоянии инфраструктуры дороги и автотранспортного потока. Следует отметить, что видеокамера «создает» поток видеок кадров со скоростью равной 25 кадров/сек., и все кадры видеопотока могут быть переданы в компьютер для дальнейшей обработки подзадачами. Однако ограниченная производительность компьютера в реальном времени ставит ряд ограничений. Так, загружая компьютер многозадачностью, количество кадров, обрабатываемых компьютером, снижается. Следовательно, в условиях многозадачности необходимо распараллелить выполняемые процессы подзадач и распределить их на несколько процессоров. Таким образом, каждая из подзадач может работать параллельно и независимо друг от друга и при этом получать доступ к одним и тем же исходным данным, таким как снимки фотоаппарата, карта навигации GPS и др. Следовательно, все записи, создаваемые пакетом программ, могут быть обработаны разным классом подпрограмм. Данная концептуальная особенность позволяет вести комплексный мониторинг и обрабатывать информацию не только в режиме реального времени, но и в лабораторных условиях.

Контрольный автомобиль – мобильная лаборатория (МЛ) движется по одной из полос многополосной дороги. На борту МЛ имеется комплект «Видеокамера – компьютер». Камера работает в режиме наблюдения, причем избранные кадры в соответствии с текущим заданием обрабатываются. Если, кроме того, камера закреплена на мобильном управляемом штативе, то компьютер может управлять ракурсом съемки и оптическим приближением. Это моделирует работу человеческого глаза. При этом пара камера – компьютер, в отличие от связки глаз-мозг, может выполнять параллельно большое количество заданий. Опишем основные.

1. Задача определения собственной скорости МЛ как функции времени может быть решена посредством GPS технологий.

2. Аналогично 1 решается задача определения скорости АТС, движущихся по многополосной дороге, если известна скорость МЛ. Практический смысл состоит в классификации автотранспортных средств по скорости перемещения (быстрые, правильные, медленные (препятствия)) и соблюдении ПДД.

3. Контроль за девственностью сплошных линий разметки реализован в нескольких модификациях. Во-первых, при стационарном наблюдении с высокой точки за большим фрагментом сплошной линии (двойной сплошной линии). При нарушении происходит идентификация АТС и запись в базу. Во-

вторых, проверка качества разметки и локализация участков, где необходимо ее обновление. В-третьих, анализ дорожного полотна на наличие посторонних компонентов (отдельных предметов, грязи, снега и т.д.).

4. Охрана динамического габарита (ДГ) – части полосы перед МЛ, необходимой для безопасности движения динамического габарита зависит от скорости МЛ и определяется автоматически. В случае вторжения в область ДГ какого-либо АТС происходит видеозапись и идентификация нарушителя. Динамический габарит необходим для анализа дорожного полотна по описываемой технологии в условиях насыщенных потоков.

5. Интегральная оценка гладкости дороги. Путем построения функции изменения положения фиксированной точки на экране компьютера МЛ, движущейся с фиксированной скоростью, определяется совокупная гладкость движения. Затем на участке ровной дороги при той же скорости определяются собственные колебания МЛ. По результатам этих двух функций дается оценка гладкости.

6. Проверка однородности дорожного покрытия. Цветовой анализ неприкосновенного динамического габарита МЛ позволяет выявить на дорожном полотне такие ее компоненты, как заплаты, трещины, ямы, посторонние предметы, грязь, лужи. Каждый из компонентов имеет физическую природу и, следовательно, может быть представлен в виде суммарных компонентов цвета.

7. Определение площади заплат дорожного покрытия. Метод основан на проверке однородности дорожного покрытия на предмет поиска компонентов – заплат (8) и использует формулы преобразования перспективы изображения в единицы площади. Используются параметры контролируемой области (ДГ), такие как: ширина, длина, расстояние от камеры до области и высота камеры над дорожным полотном.

8. Оценка объема дефектов дорожного покрытия. Подход основан на оценке объемов ям и колдобин на дороге при аппроксимировании цилиндрическими телами, причем боковая сторона отделяется от основания по разнице освещенности.

9. Оценка интенсивности и состава АТП. В одном случае осуществляется «изнутри» в режиме движения по одной из полос автомагистрали. Пересчет интенсивности с учетом известной скорости движения МЛ. В другом случае осуществляется с доступных возвышенностей (столбы) для мониторинга всего АТП по полосам движения.

10. Построение матрицы перемешивания и оценка пропускной способности сложного перекрестка.

11. Построение карты освещенности дороги. Выявление мест с недостаточной или неудачной освещенностью.

12. Портрет дороги и технология узнавания. Создается фото- (видео) карта дороги, которая хранится в памяти компьютера и служит эталоном. В процессе наблюдения в реальном времени происходит распознавание участка и сравнение с эталоном. Цель – определение изменений.

Таким образом, предложена технология постоянного и дискретного мониторинга и анализа состояния автомобильных дорог, характеристик транспортных потоков и факторов, определяющих безопасность движения. Использование описанной технологии позволит вести оперативный контроль условий дорожного движения с целью его управления, создаст предпосылки для существенного сокращения числа ДТП и неотвратимости наказания за нарушение ПДД.

Проектирование и разработка интеллектуальной системы управления финансами предприятия с применением инструментов нечеткой логики и искусственного интеллекта

Куливец С. Г.,

*механико-математический факультет
Пермского государственного университета,
614990 г. Пермь, ул.Букирева, 15,
skulivec@yandex.ru*

В статье представлены результаты работы по созданию интеллектуальной информационной системы для решения задачи диагностики финансового состояния реального предприятия и выработки рекомендаций по корректировке его финансовой политики при возникновении угрозы банкротства. Для получения правдоподобной оценки финансового состояния предприятия интеллектуальная система

опирается на использование знаний экспертов и алгоритмы нечеткого логического вывода [1-5,7]. Исходя из нечеткой информации о допустимых ресурсах предприятия, заданной экспертами в форме нечетких подмножеств, а также результатов качественного анализа внешних факторов рыночной среды, осуществляется поиск рекомендуемых действий по улучшению финансовой ситуации. Поиск решения осуществляется с использованием идей генетических алгоритмов [4,7] для решения оптимизационной задачи по достижению желаемого уровня финансового благосостояния с максимальным соответствием нечетким ограничениям на имеющиеся ресурсы.

Рассматриваемое предприятие является обществом с ограниченной ответственностью, основная форма деятельности которого состоит в производстве и реализации продукта питания. Оно относится к числу средних (в осуществлении хозяйственной деятельности занято около 80 человек). Требуется изучить проблему финансового менеджмента о минимизации риска банкротства предприятия и оценке эффективности процедур по его предотвращению. Подобный анализ актуален как для собственников компании, так и для ее кредиторов и инвесторов. Ведь решение об инвестировании в активы предприятия принимается, главным образом, на основе данных о его финансовом здоровье.

Согласно российскому законодательству, должник (предприятие или гражданин РФ) признается несостоятельным (банкротом), если он не способен удовлетворить требования кредиторов (в течение трех месяцев со дня наступления сроков их исполнения) по оплате товаров, работ, услуг, а также по обязательным платежам в бюджет и внебюджетные фонды. Банкротство может иметь место в результате судебного решения или внесудебной процедуры при официальном объявлении должника о своей несостоятельности.

В этой связи задача менеджмента заключается в установлении граничных значений показателей состояния компании, при которых финансовая ситуация становится критической, и все попытки ее исправить оказываются тщетными. По существу, степень риска банкротства – это комбинированный критерий, характеризующий как финансовое положение предприятия, так и качество управления им, которое, в конечном счете, получает свое выражение в денежном эквиваленте, но не исчерпывается одним лишь финансовым эффектом.

В проектируемой системе базу для комплексного описания экономического состояния предприятия предоставляет диагностика, т.е. исследование, позволяющее установить причины сложившегося положения дел.

В практике финансового анализа используются такие характеристики текущего финансового положения компании, как показатели ликвидности, рентабельности, устойчивости, оборачиваемости капитала и т.д. По ряду признаков известны нормативы, позволяющие оценить их уровень положительно или отрицательно. Например, когда собственные средства некоторого предприятия превышают половину его пассивов, значение коэффициента автономии считается “хорошим”, а в противном случае “плохим”. Но многие показатели однозначно охарактеризовать невозможно. Это связано со спецификой отраслей экономики, с особенностями действующих предприятий, с состоянием экономической среды, в которой они функционируют. Кроме того, любое заинтересованное лицо (руководитель, кредитор, инвестор) не довольствуется лишь численными значениями финансовых показателей. Для него важно знать истинное состояние дел: свидетельствуют ли полученные значения о финансовом благополучии компании или они указывают на недостаточную эффективность экономической деятельности, что может привести к банкротству. Задача осложняется большим количеством показателей, их возможным разнонаправленным влиянием на результат. Рациональным представляется введение одного или нескольких агрегированных критериев, сформированных из указанных финансовых показателей, которые характеризовали бы возможность получения предприятием прибыли и возврата инвестированных средств.

Наряду с определением причин финансовых проблем, актуальной задачей, для решения которой предназначена разрабатываемая интеллектуальная информационная система, является выработка рекомендуемой оптимальной программы действий по оздоровлению хозяйственной деятельности предприятия, повышению уровня его инвестиционной привлекательности, предотвращению банкротства, исходя из нечетких ограничений на ресурсы. Такие рекомендации зависят от степени финансовых осложнений, являющихся результатом возможных стратегических и тактических ошибок в осуществлении производственного процесса либо в понимании рыночной конъюнктуры. Меры, которые могут исправить ситуацию, и тем самым предотвратить банкротство или введение конкурсного управления, представляют собой результат моделирования финансового состояния компании и исследования его устойчивости на основе углубленного анализа. Выбор варианта-рекомендации осуществляется из требования максимального приближения текущего финансового благосостояния к желаемому, при условии максимального удовлетворения системе нечетких ограничений на ресурсы.

Выполним структуризацию изучаемой сферы деятельности для построения базовых моделей. В данной работе для суждения о “здоровье” хромосомы-решения в процессе поиска оптимального набора мероприятий с помощью генетического алгоритма использована модель комплексной оценки финансового состояния предприятия. Эта оценка получена на базе выполненной нами модификации модели финансового состояния предприятия, полученной средствами теории нечетких множеств [6].

В предлагаемой модели состояние предприятия описывается лингвистической переменной E [1-5]. Она обладает терм-множеством из пяти значений, которым соответствуют нечеткие подмножества “предельного неблагоприятия (E_1), неблагоприятия (E_2), среднего качества (E_3), относительного благополучия (E_4), предельного благополучия (E_5)”.

Для характеристики компании введем набор из следующих показателей:

1. X_1 – коэффициент обновления (КО), который рассчитывается как отношение балансовой стоимости основных средств, поступивших за период, к балансовой стоимости основных средств на конец периода;
2. X_2 – коэффициент текущей ликвидности (КТЛ), который определяется как отношение текущих (оборотных) активов к текущим обязательствам компании;
3. X_3 – коэффициент абсолютной ликвидности (КАЛ), который вычисляется как отношение денежных средств и краткосрочных финансовых вложений к сумме наиболее срочных и краткосрочных обязательств;
4. X_4 – коэффициент финансовой автономии (КА), который показывает долю активов должника, обеспечиваемых собственными средствами, и определяется как отношение собственных средств к совокупным активам;
5. X_5 – коэффициент оборачиваемости средств в активах (КОС), который находится как отношение выручки от реализации к средней за период стоимости активов;
6. X_6 – рентабельность совокупного капитала (РК), которая равняется отношению чистой прибыли к средней за период стоимости активов.

Мы выбрали представителей из классов показателей имущественного положения, финансовой устойчивости, деловой активности, рентабельности и ликвидности. Для оценки степеней значимости показателей будем использовать набор весов Фишберна.

Для каждого показателя построим одноименную лингвистическую переменную и сформируем набор определителей для текущих значений показателей X_i , которые позволят выполнить разбиение полного множества их значений на пять нечетких подмножества. Так как рост значений всех выбранных показателей свидетельствует об улучшении состояния предприятия, то мы предполагаем, что каждое из пяти возможных значений лингвистической оценки переменной X_i интерпретируется в пользу соответствующего значения переменной E , характеризующей состояние предприятия. Этот результат обещает нам монотонность комплексной оценки, т.е. возрастание значений одного из шести показателей, при неизменных остальных, не приведет к снижению итогового уровня. Реализация описанной методики позволяет построить агрегированную оценку состояния предприятия на базе шести показателей в виде нечеткой модели Мамдани [3,4].

Расчет влияния каждого правила осуществляется, исходя из вклада соответствующего показателя, который пропорционален величине его весового коэффициента, найденной по методу Фишберна. Сумма всех весовых коэффициентов равна единице. В качестве способа вывода заключений (импликации) используется метод алгебраического произведения, а в качестве способа агрегирования – метод граничной суммы.

Чтобы дать содержательную оценку финансового состояния предприятия, построим дополнительную модель типа Мамдани для нечеткой классификации полученного результата, что позволит охарактеризовать уровень благосостояния компании, или степень угрозы банкротства.

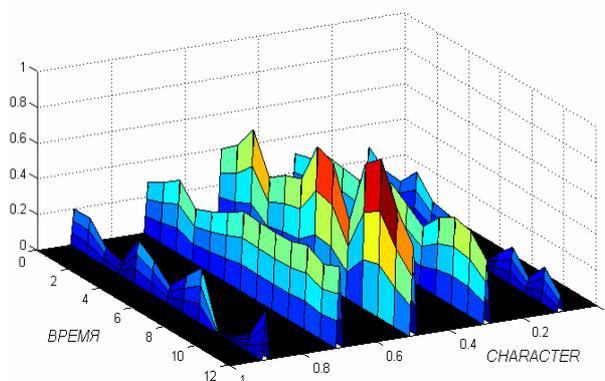


Рис. 1. Динамика результатов нечеткой классификации комплексной оценки

Вспомогательная модель Мамдани использует базу знаний, которая устанавливает соответствие между уровнем значения комплексного показателя в форме нечеткого подмножества и классом состояния, принадлежащего универсуму “*предельное неблагоприятие, неблагоприятие, среднее качество, относительное благоприятие, предельное благоприятие*”. В результате оценка положения дел на предприятии получается в виде нечеткого подмножества универсума (см. пример динамики изменения состояния предприятия в течение года на рис. 1).

В данной работе построена модель, выдвигающая предположения о причинах невысокого уровня благосостояния. Эта проблема рассматривается как задача идентификации [4], которая, в соответствии с выбранным подходом, отличается следующими свойствами: выходная переменная ассоциируется с объектом идентификации, т.е. с видом принимаемого решения; роль входных переменных выполняют параметры объекта идентификации; выходная и входные переменные могут иметь как количественные, так и качественные оценки; структура взаимосвязи между выходной и входными переменными описывается правилами ЕСЛИ <входы>, ТО <выход>, использующими лингвистические оценки переменных и представляющими собой нечеткие базы знаний.

Описанная система хранения знаний и логического вывода реализована с максимальным использованием существующих средств нечеткого моделирования для проектирования и тестирования нечетких моделей. Таким средством является расширение **Fuzzy Logic Toolbox** вычислительной системы **MATLAB**. Однако пакет устроен так, что реализация взаимодействия отдельных моделей может осуществляться только через дефазифицированные результаты. Дефазификация промежуточных результатов крайне нежелательна, так как эта процедура всегда сопровождается потерей информации. Кроме того, практическое применение описанной системы потребовало создания независимого приложения с использованием Microsoft Visual Studio .NET 2003. В рамках разработанного программного продукта реализованы все указанные процедуры по нечеткому логическому выводу и решению оптимизационной задачи с использованием идей генетических алгоритмов. Таким образом, в рамках данного исследования спроектирована и создана интеллектуальная информационная система для решения задачи комплексной оценки положения предприятия и выработки рекомендаций по корректировке его финансовой политики при возникновении угрозы банкротства. В процессе конструирования применены иерархическая система нечеткого вывода для комплексной оценки финансового состояния и идеи генетических алгоритмов для решения задачи поиска рекомендуемых оптимальных действий по улучшению ситуации при наличии нечетких ограничений на имеющиеся ресурсы.

Список литературы.

1. Заде Л. Понятие лингвистической переменной и ее применение к принятию приближенных решений / Пер. с англ. М.: Мир, 1976. 167 с.
2. Романов В. П. Интеллектуальные информационные системы в экономике. М.: Экзамен, 2003. 496 с.
3. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2003. 736 с.
4. Ротштейн А. П. Интеллектуальные технологии идентификации. <http://matlab.exponenta.ru/fuzzylogic/book5/index.php>.
5. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику. <http://matlab.exponenta.ru/fuzzylogic/book1/index.php>.
6. Недосекин А. О. Комплексная оценка риска банкротства корпорации на основе нечетких описаний. <http://www.sedok.narod.ru>.
7. Тим Джонс М. Программирование искусственного интеллекта в приложениях. М.: ДМК Пресс, 2004. 312 с.

Информационные процессы в системах управления обучением

Куликова Е. Н., к.т.н., e-mail: Ekaterina.Kulikova@ritzio.com
Руководитель отдела дистанционного обучения Ritzio Entertainment Group
129337 Москва Ярославское ш., 26, корп.2-204

Деятельность современного образовательного учреждения не может ограничиваться только классическими формами обучения. Необходимость внедрения дистанционных технологий обучения

обусловлена как территориально распределенным спросом на образовательные программы, так и необходимостью в обучении без отрыва от работы.

Надо отметить, что, дистанционная форма обучения не создается с «нуля». По одной простой причине – нужны учебные материалы и люди, умеющие их «вкладывать» в головы обучаемых. Создание таких материалов и коллективов – задача многолетняя и сложная. Поэтому, дистанционное обучение предлагается уже существующими, работающими учебными заведениями, которые имеют продукт, т.е. учебные материалы и имеют мастеров по внедрению данного продукта – преподавателей.

Но, зачастую, понимание, каким именно должен быть это продукт, весьма различно. Причем различия характерны не только для разных учебных организаций, но и с точки зрения обучаемых и преподавателей. Если при очной форме обучения студент точно знает, что он должен делать (посещать лекции, писать курсовые проекты, выполнять лабораторные работы), то в дистанционной форме тот или иной электронный курс может оценивать действия пользователя по-разному. У каждого преподавателя также может существовать своя точка зрения на подачу материала. Поэтому, для того, чтобы достигнуть понимания сторон, необходима единая методика как представления учебного материала в виде электронного курса, так и оценки действий обучаемого при прохождении курса.

В любом учебном процессе можно выявить следующие основные информационные процессы: передача информации (теоретической и практической) и мониторинг деятельности обучаемого.

Следовательно, учитывая особенности дистанционной формы обучения, состоящие в используемых инструментах представления и передачи учебной информации, для формализации процесса передачи информации необходимо сформулировать требования к форме электронных курсов, а для формализации процесса мониторинга учебной деятельности – выработать регламенты организации информационных процессов.

Разработанная информационная модель предполагает двухуровневую декомпозицию учебного материала электронного курса, позволяющую реализовать передачу знаний, выработку навыков, контроль и самоконтроль. Каждый элемент структуры обладает детерминированными и динамическими параметрами, позволяющими контролировать учебную деятельность с различной степенью детализации.

**МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ
"ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И КОМПЬЮТЕРНЫЕ
НАУКИ" (23-27 октября 2006 г.), том 2, часть 1. Под общей редакцией
академика Садовниченко В. А., проф. Кудрявцева В. Б., проф.
Михалева А. В., 2006 г., 176 с.**

Подписано в печать 29.09.2006.

Формат 60 × 90 1/16. Объем 11.5 п.л.

Заказ 18. Тираж 200 экз.

Издательство ЦПИ при механико–математическом факультете МГУ
г. Москва, Воробьевы горы.

Лицензия на издательскую деятельность ИД № 04059 от 20.02.2001 г.

Отпечатано на типографском оборудовании механико-математического факуль-
тета

