

## **Использование нейросетевых моделей для извлечения и анализа информации баз данных**

**Солодовников В.И., МГИЭМ**

В настоящее время нейросетевые модели являются одним из наиболее эффективных средств анализа и синтеза сложных объектов, таких как динамические объекты, экономические системы, базы данных и т.д. Это обусловлено способностью нейронных сетей к моделированию нелинейных процессов, работе с зашумленными данными, адаптивностью к изменяющимся условиям функционирования (обучение и самообучение), возможность работы с малым объемом априорной информации.

Первостепенной задачей, стоящей перед разработчиком, является проблема выбора топологии сети, которая бы наилучшим образом удовлетворяла тем или иным условиям, предъявляемым системе. Что в свою очередь зависит от задач ставящихся перед нейросетью, какими, могут являться задачи классификации (отнесение предъявляемого образца к одному из уже известных классов) и кластеризации (разбиение поступающих данных на кластеры, причем количество классов может быть изначально неизвестно). Так, для решения задач классификации хорошо подходят сети прямого распространения, типа многослойного персептрона и RBF-сети и их модификации, обучаемые по алгоритму обратного распространения ошибки. Для задач кластеризации, часто используются самоорганизующиеся сети Кохонена, обучение которых проходит без учителя, и сеть сама решает, к какому из классов лучше отнести предъявляемый образец.

В работе ставится задача рассмотрения вопроса построения приложения, позволяющего:

1. Осуществлять генерацию нейронных сетей, имеющих различную структуру, в зависимости от решаемых ими задач.
2. Хранить, модифицировать и обучать полученные сети.

3. Использовать в качестве источника данных сети приложения уже существующих баз данных (БД).

4. Анализировать результаты работы системы.

Для решения первой задачи предлагается использование БД, в которой хранится описание нейронной сети. Это позволяет упростить процесс его создания, избавив разработчика от написания кода (формирование описания осуществляется в ходе диалога), и дает возможность осуществлять анализ сети и результатов ее работы средствами, предоставляемыми БД. Таким образом, создание, модификация и анализ нейронной сети осуществляется в режиме взаимодействия с базой данных, а обучение и испытания сети осуществляются в оперативной памяти. Это потребовало разработки информационной модели нейронной сети для ее хранения в БД и объектно-ориентированной модели для ее программной реализации.

С точки зрения объектно-ориентированного подхода нейросеть представляет собой иерархию классов: сеть, слой, нейрон и т.д.. Фактически, именно, описание этой иерархии и хранится в базе данных.

Рассмотрим более подробно эту иерархию. Отметим, что здесь приводится упрощенный вариант, ориентированный на описание наиболее характерных сетей, для которых фиксировано число слоев, заданы функции активации и связи [1].

Класс `NeuroNet` представляет собой общее описание сети, включающее ее идентификатор, вид сети, назначение и область применения, а также другие наиболее общие характеристики.

Класс `NeuroStruct` задает архитектуру сети: количество входов, слоев и выходов, определяет связи (в общем случае выделяется в отдельный класс) и т.д.

Класс `NeuroEducate` предназначен для описания характеристик обучения и содержит такие атрибуты как: вид алгоритма обучения, объем обучающей и тестовой последовательности, источник данных.

Класс NeuroStratum является описание слоев нейронной сети и содержит информацию о количестве нейронов в слое, наличии задержек и некоторых других свойствах.

Класс NeuroResult используется для описания получаемых результатов и определяет вид получаемых результатов, критерии их обработки и т.д.

Класс Neuron задает основные характеристики нейрона, позволяющие выделить его среди других нейронов в слое.

Класс Activation определяет вид функции активации.

Класс InAct задает правила вычисления входа функции активации.

Класс Weights задает весовые коэффициенты.

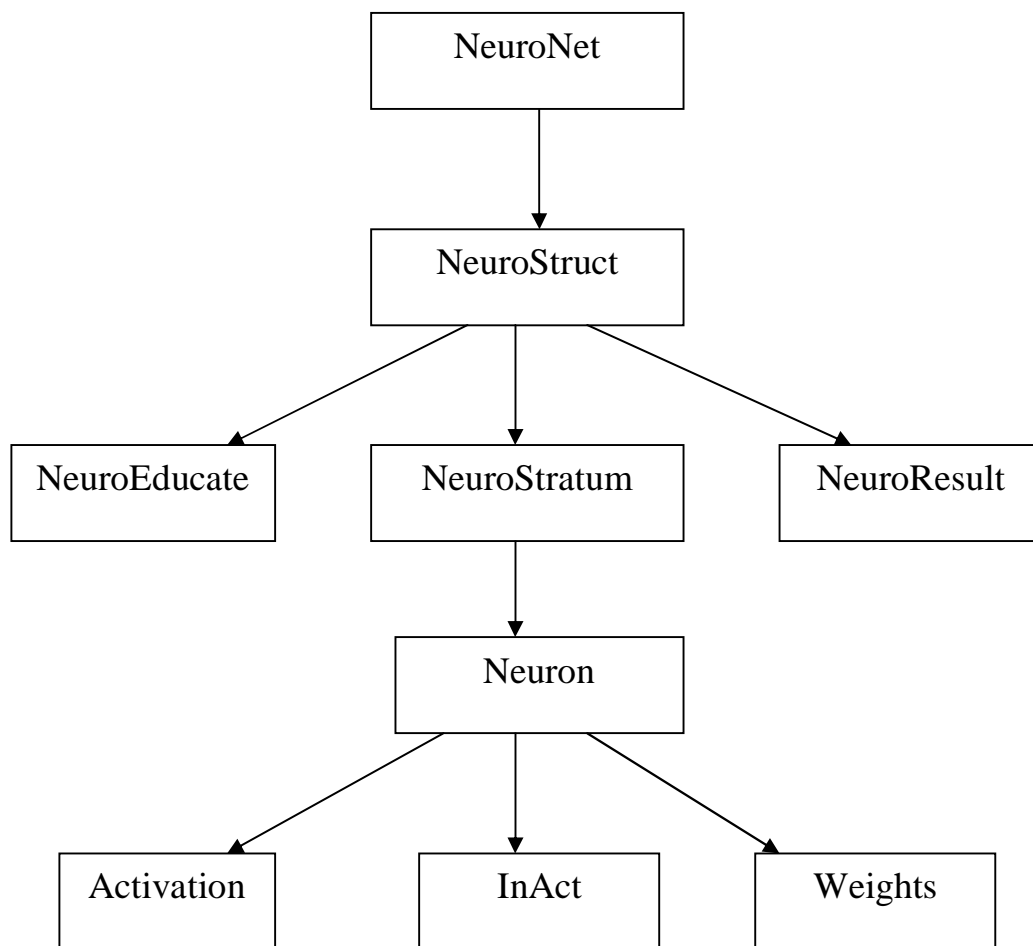


Рис.1. Иерархия классов

Фактически данная иерархия классов соответствует информационной модели и таблицам БД с учетом того, что для хранения некоторых классов,

используются несколько таблиц, возникших на этапе проектирования БД с использованием CASE – технологии, носящих в основном справочный характер и устраняющих избыточность и аномалии обновления[3]. Кроме того, в БД используется и ряд дополнительных отношений, к которым можно отнести системную таблицу, в которой фиксируются статистические данные, таблицы обучающей и тестовой последовательности и результатов испытаний.

В силу того, что данная модель ориентирована на модели нейронных сетей с частью характеристик, известных заранее, некоторые атрибуты классов могут формироваться автоматически, другие вводятся в процессе взаимодействия, причем это взаимодействие зависит от вида сети. Поэтому для того чтобы формализовать этот процесс, получить возможность включать в класс хорошо определенных сетей новые варианты, а также для упрощения для пользователя задачи, что хочет получить пользователь от своей базы данных, используется разновидность деревьев решений.

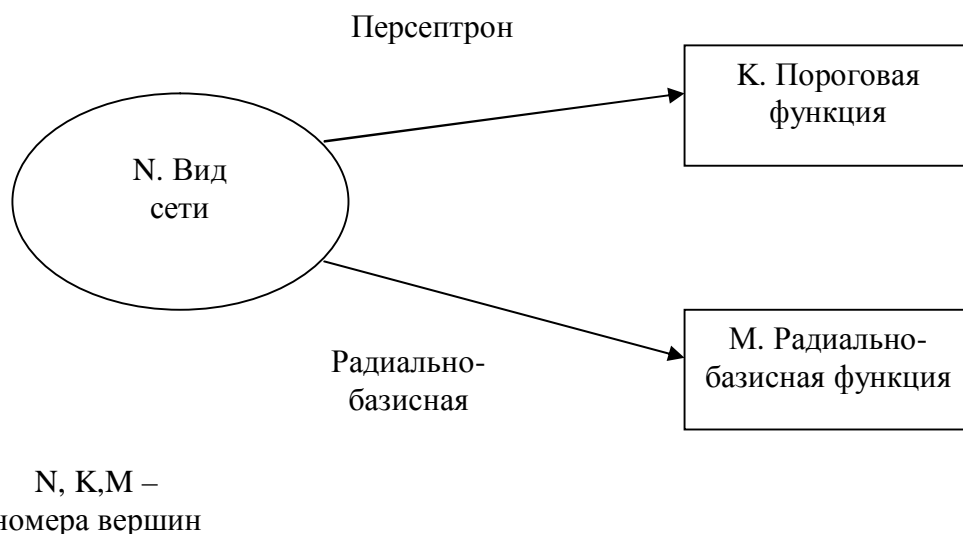


Рис.2. Пример фрагмента дерева решений

Дерево решений состоит из вершин двух типов. Вершины решений, содержащие вопросы, обозначаются окружностью. Цели или логические выводы обозначаются прямоугольниками. Условия задаются на дугах. Каждая вершина нумеруется [2].

Например, если в ходе диалога возникает вопрос о выборе функции активации в сети, решение этого вопроса зависит от вида сети. Это можно представить следующим образом: если при ответе на вопрос о выборе сети дан ответ, что выбирается персептрон, то функция активации имеет вид пороговой функции, а при выборе радиально-базисной сети выбирается радиально-базисная функция, что в дереве решений представляется в виде, показанном на рис.2.

Для хранения деревьев решений в БД приложения формируется таблица, имеющая вид

<b>№ вершины</b>	<b>Переменная</b>	<b>Значение</b>	<b>Исходная вершина</b>	<b>Дуга</b>	<b>Тип вершины</b>
N	Вид сети	-	-	-	решение
K	Функция активации	Пороговая	N	Персептрон	вывод
M	Функция активации	Радиально-базисная	N	Радиально-базисная	вывод

Кроме того, источник данных для обучающей последовательности и испытания сети может быть задан с помощью SQL – запроса.

Данный подход позволяет производить анализ широкого класса задач в различных предметных областях на основе накопленной информации с использованием разных видов нейронных сетей. При этом появляется возможность вводить новые виды нейронных сетей и осуществлять их анализ.

#### Литература

1. В. А. Головки. Нейронные сети: обучение, организация и применение. – М.: Радиотехника, 2001.
2. Р. Левин, Д. Дранг, Б.Эделсон. Практическое введение в технологию искусственного интеллекта и экспертных систем. – М.: Финансы и статистика, 1991.
3. Т. Конелли, К.Бегг. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – М.: Изд. «Вильямс», 2000

## Оценка влияния условий труда на параметры крови с помощью нейросетей

Решетникова Н.В., Петровская В.А., Товбис Е.М., Привалко Е.Ю., Жуков Л.А.  
Сибирский Государственный Технологический Университет  
Красноярская городская детская больница 2  
Красноярская краевая станция переливания крови  
E-mail: [zhukov@wallst.ru](mailto:zhukov@wallst.ru)  
<http://zhukov.wallst.ru>

В настоящей работе производилась обработка гематологических, биохимических и иммунологических исследований доноров электролизных цехов алюминиевого завода, подвергающихся иммунному плазмаферезу на основе нейросетей с учителем [1].

Использовались следующие параметры: величина титра антител, число плазмаферезов и некоторые изучаемые гематологические, биохимические, иммунологические характеристики, указывающие на функциональные сдвиги в системе гомеостаза в ответ на введение стафилококкового антигена и под влиянием многолетних плазмаферезов [2].

Данные параметры были сведены в файлы данных формата DBASE, отдельные для каждой категории рабочих. Файл с анодчиками содержит 266 записей, электролизники – 186 записей, крановщики 168, слесари – 198, сварщики – 130, студенты – 472.

При обработке использовался нейроимитатор NeuroPro версии 0.25. Были созданы и обучены несколько нейронных сетей с различными параметрами [1]. В качестве поля ответа использовалось поле – общее количество лейкоцитов [3].

Таблица 1. Описание входных параметров

№	Поле	Xmax	Описание
Гематологические показатели			
1	X2	450,0	Гемоглобин (г/л)
2	X6	11,4	Лейкоциты ( $\times 10^{**9}/л$ )
3	X7	13,0	Эозинофилы (%)
4	X8	10,0	Палочкоядерные (%)
5	X9	73,0	Сегментоядерные (%)
6	X10	57,0	Лимфоциты (%)

7	X11	73,0	Моноциты (%)
8	X12	23,0	СОЭ (мм/ч)
Печеночные пробы			
9	X13	27,4	Билирубин общий (мкмоль/л)
10	X14	12,0	Ас Ат (ед.)
11	X15	11,0	Ал Ат (ед.)
12	X16	5,3	Тимоловая проба (ед.)
Биохимические показатели			
13	X17	91,4	Белок общий (г/л)
14	X18	72,5	Альбумин (г/л)
15	X19	7,5	Альфа 1%
16	X20	12,7	Альфа 2%
17	X21	11,7	Бета %
18	X22	23,3	Гамма %
Плазмаферез (ПФ)			
19	X25	450,0	Доза взятой крови (мл)
20	X28	270,0	Криопреципитат (мл)
21	X29	9815,0	Суммарная доза извлеченной плазмы (мл)
22	X30	310,0	Интервал между последующими ПФ
23	X31	450,0	Разность между x25, не нулевым x26,x27,x28

Xmax – максимальное значение в поле

Для определения средней и максимальной ошибок было обучено и протестировано несколько нейросетей с 3 слоями нейронов с точностью 1,3 и характеристикой нейронов 0,3.

Таблица 2. Средняя и максимальная ошибки по категориям рабочих

Параметры	Анод-ники	Электролизники	Крановщики	Слесари	Сварщики	Студенты
Средняя ошибка	1,82	1,85	2,12	2,05	1,93	2,35
Максимальная ошибка	7,68	8,15	11,10	29,86	7,13	11,86

\* В данной таблице анодчики обучались по данным электролизников, все остальные – по данным анодчиков.

Как видно из таблицы 2 наибольшая средняя ошибка наблюдается при тестировании данных студентов, что можно объяснить более благоприятными условиями их “работы”. Так как изначально данные студентов предполагалось использовать в качестве тестовой выборки.

Было проведено сравнение тестовых выборок разных профессий по количеству правильно определенных примеров для чего использовалась нейросеть с точностью 1,5 (таблицы 3,4).

Таблица 3. Количество правильно определенных примеров для тестовых выборок

ТВ	Anod	Elec	Kran	Sles	Stu2	Svar
OB						
Anod	191	81	71	91	182	41
Elec	86	182	96	121	279	70
Kran	83	112	166	126	271	69
Sles	57	80	79	198	222	65
Stu2	92	141	133	142	472	91
Svar	72	77	88	85	229	130
Сумма	191	182	166	198	472	130

Прим.1. Сумма представляет собой общее количество использованных примеров.

Прим.2. По диагонали все ответы правильные, т.к. обучающая и тестовая выборки совпадают.

Таблица 4. Процент правильно определенных примеров для тестовых выборок

ТВ	Anod	Elec	Kran	Sles	Stu2	Svar	Среднее
OB							
Anod	100	44	42	45	38	31	41
Elec	45	100	57	61	59	54	56
Kran	43	61	100	63	57	53	56
Sles	29	43	48	100	47	50	44
Stu2	48	77	80	71	100	70	69
Svar	37	42	53	42	48	100	46
Среднее	40	53	56	56	50	52	

\* Средний процент вычисляется без учета диагональных значений.

Из таблиц 3 и 4 видно, что данные анализов крови разных профессий сильно различаются по количеству правильно решенных примеров. Причем, при обучении по данным анализов сварщиков самый большой процент верно определенных примеров (53%) имеет тестовая выборка крановщиков, что может быть объяснено схожими условиями труда данных категорий рабочих. В целом, правильность тестирования сети, обученной по анодчикам, наихудшая, что может объясняться существенно разными условиями труда. К сожалению,



не все результаты тестирования можно легко объяснить. Например, при обучении по электролизникам, крановщикам и анодчикам наибольший процент верных прогнозов, соответственно 61%, 63% и 45%, давало тестирование по файлу слесарей. Наибольший процент правильных ответов (80%) дало тестирование данных крановщиков при обучении по файлу студентов. Исходно данные по студентам собирались для контроля, однако полученные результаты не сильно выделяют студентов из большинства групп других профессий.

В среднем наибольший процент верных ответов наблюдается при тестировании выборок крановщиков и слесарей. Для обучающих выборок максимален процент по студентам (69%) и чуть меньше – по электролизникам и крановщикам.

Было проведено также сравнение тестовых выборок разных профессий по количеству правильно определенных примеров для чего использовалась нейросеть с точностью 1,5 при обучении по половине выборки и тестировании по другой половине (таблицы 5,6).

Объемы обучающих выборок составили: анодчики – 105, электролизники – 91, крановщики – 81, слесари – 99, студенты – 236, сварщики – 65.

Таблица 5. Количество правильно определенных примеров для тестовых выборок

ТВ	Anod	Elec	Kran	Sles	Stu2	Svar
OB						
Anod	46	48	47	62	128	21
Elec	51	39	45	57	106	26
Kran	42	40	44	50	121	27
Sles	28	41	38	40	90	34
Stu2	34	40	35	65	153	27
Svar	46	39	33	57	131	27
Сумма	86	91	85	99	236	65

Прим.1. Сумма представляет собой общее количество использованных примеров.

Прим.2. По диагонали не все ответы правильные, т.к. обучающая и тестовая выборки не совпадают.

Таблица 6. Процент правильно определенных примеров для тестовых выборок

TB	Anod	Elec	Kran	Sles	Stu2	Svar	Среднее
OB							
Anod	53	53	55	63	54	32	47
Elec	59	43	53	58	45	40	45
Kran	49	44	52	51	51	42	44
Sles	33	45	45	40	38	52	38
Stu2	39	44	41	66	65	42	45
Svar	53	43	39	58	56	42	44
Среднее	48	45	48	56	52	42	

\* Средний процент вычисляется с учетом диагональных значений.

При обучении и тестировании по половинным выборкам, как видно из таблиц 5,6, результаты немного отличаются от тех, которые были получены при обучении и тестировании по полным выборкам. Например, при обучении по данным электролизников наибольший процент правильных прогнозов давало тестирование по выборке анодчиков, что может объясняться схожими условиями труда данных категорий рабочих.

Были определены наиболее значимые входные параметры тестовых выборок разных профессий для чего использовалась одна нейросеть с точностью 1,5 (таблица 7).

Таблица 7. Наиболее значимые входные параметры для тестовых выборок (ОВ/ТВ)

Поле	Anod/ Anod	Anod/ Elec	Anod/ Kran	Anod/ Sles	Anod/ Stu2	Anod/ Svar	Elec/ Anod	Elec/ Elec	Elec/ Kran	Elec/ Sles	Elec/ Stu2	Elec/ Svar
X2	0,47	0,49	0,43	0,46	0,46	0,56	0,41	0,44	0,48	0,45	0,45	0,42
X7	0,29	0,21	0,32	0,29	0,29	0,20	0,34	0,38	0,31	0,29	0,35	0,31
X8	0,62	0,41	0,37	0,29	0,41	0,40	0,44	0,49	0,40	0,43	0,48	0,35
X9	0,50	0,50	0,52	0,62	0,63	0,62	0,53	0,58	0,60	0,61	0,71	0,53
X10	0,36	0,19	0,16	0,26	0,39	0,19	0,44	0,22	0,21	0,24	0,41	0,15
X11	0,64	0,74	0,56	0,68	0,64	0,72	0,65	<b>0,80</b>	<b>0,80</b>	0,61	0,74	<b>0,70</b>
X12	<b>1,00</b>	<b>0,81</b>	<b>0,67</b>	<b>0,75</b>	<b>0,89</b>	<b>1,00</b>	<b>1,00</b>	0,79	<b>1,00</b>	<b>0,94</b>	<b>1,00</b>	<b>1,00</b>
X13	0,51	0,45	0,45	0,35	0,53	0,79	0,53	0,48	0,55	0,54	0,54	0,49
X14	0,22	0,23	0,18	0,18	0,35	0,19	0,14	0,16	0,18	0,16	0,39	0,25
X15	0,26	0,21	0,17	0,22	0,35	0,25	0,41	0,24	0,21	0,26	0,39	0,35
X16	0,37	0,30	0,22	0,33	0,42	0,40	0,29	0,33	0,33	0,33	0,41	0,40
X17	0,46	0,33	0,35	0,40	0,50	0,37	0,27	0,32	0,26	0,37	0,39	0,31
X25	0,51	0,43	0,41	0,42	0,39	0,45	0,30	0,50	0,42	0,36	0,50	0,41
X28	<b>0,98</b>	<b>1,00</b>	<b>0,66</b>	<b>1,00</b>	<b>1,00</b>	<b>0,91</b>	<b>0,75</b>	<b>1,00</b>	0,74	<b>1,00</b>	<b>0,83</b>	0,70
X29	<b>0,90</b>	<b>0,86</b>	<b>1,00</b>	<b>0,72</b>	<b>0,71</b>	<b>0,82</b>	<b>0,87</b>	<b>0,96</b>	<b>0,92</b>	<b>0,75</b>	<b>0,90</b>	<b>0,79</b>
X30	0,60	0,65	0,62	0,54	0,50	0,65	0,47	0,52	0,54	0,43	0,59	0,55
X31	0,30	0,13	0,18	0,17	0,19	0,16	0,33	0,20	0,24	0,14	0,24	0,21

Поле	Kran/ Anod	Kran/ Elec	Kran/ Kran	Kran/ Sles	Kran/ Stu2	Kran/ Svar	Sles/ Anod	Sles/ Elec	Sles/ Kran	Sles/ Sles	Sles/ Stu2	Sles/ Svar
X2	0,46	0,40	0,37	0,57	0,42	0,46	0,48	0,38	0,34	0,38	0,47	0,30
X7	0,36	0,27	0,26	0,34	0,37	0,27	0,29	0,29	0,25	0,33	0,34	0,18
X8	0,52	0,40	0,34	0,36	0,38	0,37	0,58	0,38	0,21	0,24	0,39	0,20
X9	0,58	0,56	0,58	0,55	0,55	0,48	0,55	0,71	0,55	0,59	0,66	<b>0,56</b>
X10	0,53	0,29	0,28	0,36	0,58	0,17	0,45	0,34	0,29	0,32	0,62	0,21
X11	0,65	0,64	0,55	0,67	0,54	0,42	<b>0,83</b>	0,69	0,58	0,55	0,62	0,44
X12	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	0,45	0,75	<b>0,74</b>	<b>0,81</b>	<b>0,83</b>	<b>0,60</b>

X13	0,56	0,56	0,52	0,52	0,53	0,39	0,72	0,36	0,29	0,32	0,40	0,24
X14	0,18	0,19	0,17	0,15	0,23	0,11	0,47	0,29	0,21	0,19	0,42	0,10
X15	0,45	0,18	0,19	0,36	0,35	0,27	<b>0,81</b>	0,27	0,26	0,26	0,47	0,24
X16	0,30	0,22	0,24	0,35	0,38	0,25	0,37	0,43	0,33	0,37	0,43	0,36
X17	0,32	0,26	0,21	0,36	0,27	0,24	0,26	0,53	0,46	0,37	0,49	0,30
X25	0,43	0,46	0,41	0,55	0,49	0,41	0,41	0,73	0,63	0,70	0,64	0,49
X28	<b>0,94</b>	<b>0,84</b>	<b>0,64</b>	<b>0,99</b>	<b>0,64</b>	<b>0,69</b>	0,62	<b>0,81</b>	0,68	<b>0,76</b>	0,71	0,46
X29	<b>0,91</b>	<b>0,92</b>	<b>0,79</b>	0,66	0,64	0,62	0,68	<b>0,90</b>	<b>0,88</b>	0,63	<b>0,78</b>	0,56
X30	0,73	0,72	0,72	<b>0,84</b>	<b>0,67</b>	<b>0,75</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>
X31	0,31	0,17	0,23	0,22	0,16	0,22	0,31	0,24	0,22	0,22	0,24	0,18

Поле	Stu2/ Anod	Stu2/ Elec	Stu2/ Kran	Stu2/ Sles	Stu2/ Stu2	Stu2/ Svar	Svar/ Anod	Svar/ Elec	Svar/ Kran	Svar/ Sles	Svar/ Stu2	Svar/ Svar
X2	0,63	0,51	0,55	0,54	0,44	0,45	0,55	0,28	0,25	0,39	0,41	0,41
X7	0,33	0,17	0,23	0,23	0,16	0,19	0,55	0,30	0,37	0,44	0,40	0,29
X8	0,39	0,24	0,29	0,20	0,20	0,20	0,38	0,17	0,12	0,17	0,23	0,13
X9	0,74	0,57	0,65	0,67	0,59	0,57	<b>0,95</b>	0,45	0,49	0,55	0,48	0,43
X10	0,59	0,33	0,37	0,25	0,57	0,28	0,67	0,35	0,27	0,34	0,54	0,24
X11	0,69	0,50	0,59	0,62	<b>0,67</b>	0,49	<b>0,93</b>	0,47	0,50	0,57	0,58	0,50
X12	<b>0,78</b>	0,51	0,54	0,55	<b>0,67</b>	0,53	0,83	<b>0,67</b>	<b>0,62</b>	0,59	0,75	<b>0,72</b>
X13	0,42	0,20	0,29	0,33	0,42	0,29	0,42	0,23	0,31	0,37	0,48	0,26
X14	0,42	0,19	0,19	0,31	0,34	0,18	0,56	0,28	0,18	0,36	0,68	0,35
X15	0,44	0,14	0,23	0,41	0,20	0,15	0,40	0,22	0,29	0,29	0,39	0,24
X16	0,57	0,36	0,41	0,50	0,60	0,35	0,82	0,51	0,48	0,57	0,74	0,49
X17	0,44	0,45	0,47	0,47	0,37	0,38	0,30	0,28	0,27	0,31	0,35	0,29
X25	<b>1,00</b>	<b>0,92</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	0,90	<b>0,95</b>	<b>0,94</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>
X28	0,51	0,50	0,43	0,57	0,45	0,30	0,46	0,45	0,29	0,49	0,44	0,39
X29	<b>0,91</b>	<b>1,00</b>	<b>0,73</b>	<b>0,77</b>	0,58	<b>0,72</b>	0,91	0,60	<b>0,67</b>	<b>0,71</b>	<b>0,80</b>	<b>0,74</b>
X30	0,78	<b>0,82</b>	<b>0,75</b>	<b>0,87</b>	<b>0,73</b>	<b>0,65</b>	<b>1,00</b>	<b>1,00</b>	<b>1,00</b>	<b>0,95</b>	<b>0,99</b>	<b>0,94</b>
X31	0,45	0,16	0,20	0,17	0,24	0,15	0,66	0,23	0,30	0,34	0,36	0,24

Как видно из таблицы 7 наиболее значимыми входными параметрами являются:

гематологические показатели:

X11 – Моноциты (%)

X12 – СОЭ (мм/ч)

плазмаферез:

X25 – Доза взятой крови (мл)

X28 – Криопреципитат (мл)

X29 – Суммарная доза извлеченной плазмы (мл)

X30 – Интервал между последующими плазмаферезами.

Следовательно, среди наиболее значимых параметров выделяются суммарная доза извлеченной плазмы, СОЭ и интервал между плазмаферезами, что указывает на большую значимость показателей плазмафереза по сравнению с другими.

### **Список литературы**

1. Горбань А.Н., Россиев Д.А. Нейронные сети на персональных компьютерах. Н.: Наука, 1996
2. Петровская В.А., Жуков Л.А., Оводова А.А. Нейросетевая обработка данных по гематологии // 2-й семинар “Новые информационные технологии” Москва, 1999. С.30
3. Петровская В.А., Жуков Л.А., Доценко О.Н., Гулько Т.И. Изучение влияния плазмафереза на функциональное состояние организма доноров с использованием нейросетей // “Информационные технологии”: межвуз. конференция. Красноярск, 1999. С.46-49

## Реализация механизмов вывода на основе деревьев решений и семантических сетей

Солодовников И.В., МГИЭМ

Важным классом задач, решаемых в информационных системах, являются задачи принятия решения. Одним из подходов для их представления является подход, основанный на использовании деревьев решений. Значительно реже в последнее время используется подход с применением семантических сетей. Однако, сочетание семантических сетей и деревьев решений дает дополнительные возможности для этого класса задач и позволяет повысить качество принимаемых решений.

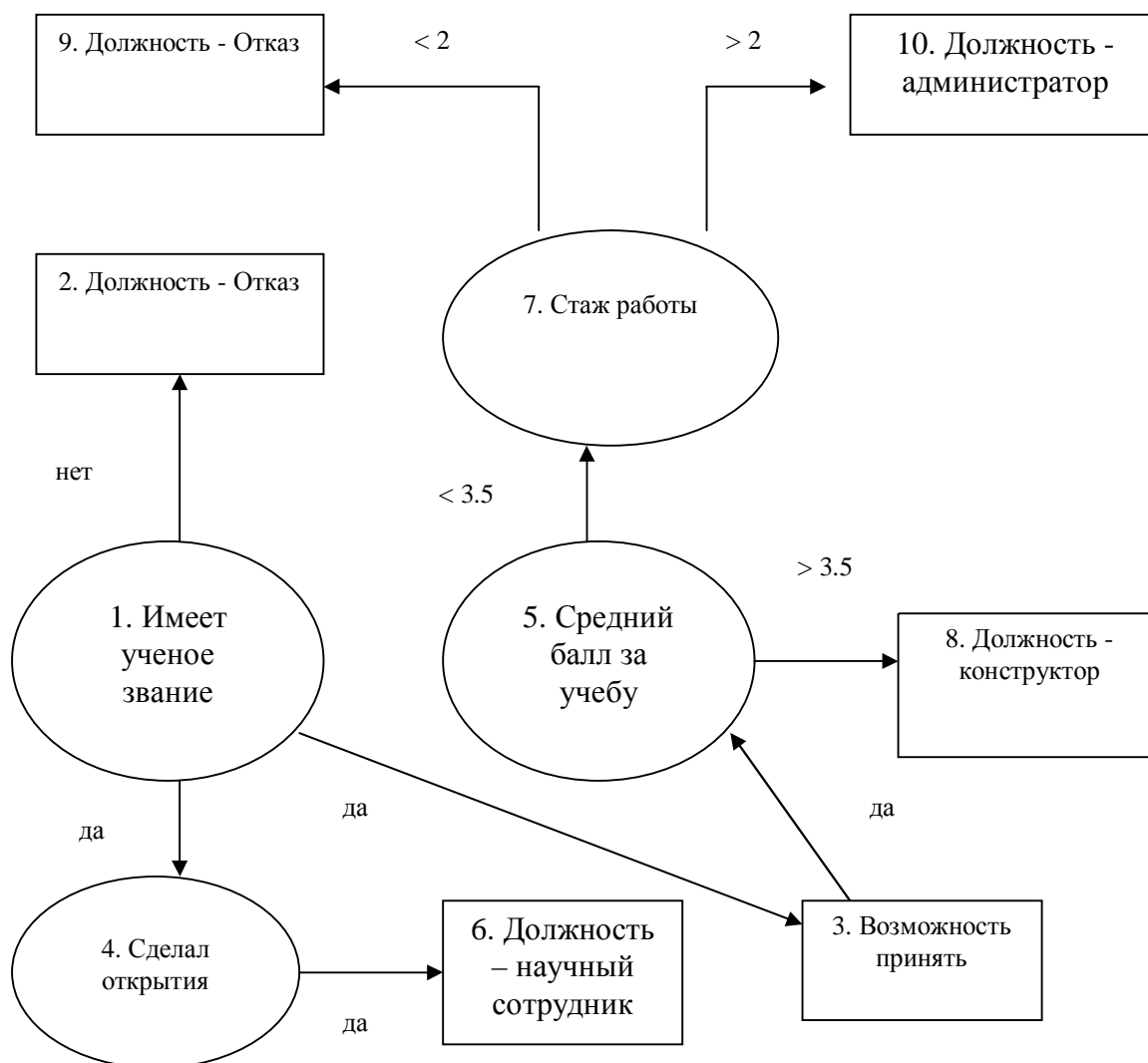


Рис.1. Пример дерева решений

Дерево решений состоит из вершин двух типов. Вершины решений, содержащие вопросы, обозначаются окружностями. Цели или логические выводы обозначаются прямоугольниками. Вершины нумеруются. Каждая вершина может иметь не более одного входа. Для сравнения подходов, основанных на деревьях решений и простейших семантических сетях, рассмотрим пример с приемом на работу, часто используемый в литературе [1].

*Таблица дерева решений*

<b>№ вершины</b>	<b>Переменная</b>	<b>Значение</b>	<b>Исходная вершина</b>	<b>Дуга</b>	<b>Тип вершины</b>
1	Звание	-	-	-	решение
2	Должность	Отказ	1	нет	вывод
3	Возможность	да	1	да	вывод
4	Открытия	-	1	да	решение
5	Средний балл	-	3	да	решение
6	Должность	Научный сотрудник	4	да	вывод
7	стаж	-	5	$< 3.5$	решение
8	должность	конструктор	5	$> 3.5$	вывод
9	должность	Отказ	7	$< 2$	вывод
10	должность	администратор	7	$> 2$	вывод

Для сохранения результатов будем использовать таблицу вывода (в начальный момент таблица пуста).

**Таблица вывода**

<b>№ варианта</b>	<b>Переменная</b>	<b>Значение</b>

Алгоритмы вывода для такого представления дерева решений достаточно просты [2]. Теперь сравним это представление с представлением для этой же задачи, получаемом в простой семантической сети, называемой иногда вычислительной семантической сетью и представляющей собой фактически двудольный граф [3].

Пусть задано конечное множество символов  $A = \{A_1, \dots, A_r\}$ , называемых атрибутами, и конечное множество  $R = \{R_1, \dots, R_n\}$  отношений. Схемой или интенционалом отношения  $R_i$  называют набор пар:

$$\text{INT}(R_i) = \{ \dots, [A_j, \text{DOM}(A_j)], \dots \},$$

где  $R_i$  – имя отношения,  $\text{DOM}(A_j)$  – домен  $A_j$ , т.е. множество значений атрибута  $A_j$  отношения  $R_i$ .

Объединение всех доменов называется базовым множеством модели или множеством объектов, на которых задаются отношения  $R$ .

Экстенционалом отношения  $R_i$  называют множество

$$\text{EXT}(R_i) = \{F_1, \dots, F_p\},$$

где  $F_k$  ( $k = 1 \div p$ ) – факт отношения  $R_i$ . Факт задается совокупностью пар атрибут – значение, называемых атрибутивными парами. Под фактом понимается конкретизация определенного отношения между указанными объектами. В графической интерпретации факт – это подграф семантической сети, имеющий звездообразную структуру. Корень подграфа – вершина предикатного типа, помеченная уникальной меткой, включающей имя соответствующего отношения. Из вершины факта выходят ребра, помеченные именами атрибутов данного факта, ведущие в вершины базового множества, которые являются значениями этих атрибутов.

Для примера с приемом на работу получим:

$\text{INT}(\text{Должность}) = \{ [ \text{Звание}(\text{да, нет}) ], [ \text{Возможность принять}(\text{да, нет}) ], [ \text{Открытия}(\text{да, нет}) ], [ \text{Средний балл}(<3.5, >3.5) ], [ \text{Стаж работы}(<2, >2) ], [ \text{Должность}(\text{отказ, научный сотрудник, конструктор, администратор}) ] \}$

$\text{INT}(\text{Возможность принять}) = \{ [ \text{Звание}(\text{да, нет}) ], [ \text{Возможность принять}(\text{да, нет}) ] \}$



Отметим, что каждой переменной вывода соответствует свое отношение. Переменные вывода отметим жирным шрифтом. Кроме того, введем еще одно значение Nil, которое может принимать любая переменная, и которое будет обозначать, что данная переменная не учитывается при формировании факта. Дополнительный смысл значения Nil будет пояснен ниже.

Экстенционалы этих отношений запишем в виде фактов, которые удобно представлять в виде таблиц.

**Таблица. Экстенционал «Возможность принять»**

Метка факта	Звание	Возможность принять
F1	да	да

**Таблица. Экстенционал «Должность»**

Метка факта	Звание	Возможность принять	Открытия	Средний балл	Стаж работы	Должность
F2	нет	Nil	Nil	Nil	Nil	отказ
F3	да	Nil	да	Nil	Nil	Научный сотрудник
F4	да	да	нет	>3.5	Nil	Конструктор
F5	да	да	нет	<3.5	>2	администратор
F6	да	да	нет	<3.5	<2	отказ

Получим следующий граф для рассматриваемого примера (рис.2).

Алгоритм преобразования дерева решений в семантическую сеть достаточно прост. Пусть заданы список  $V$  переменных вывода и список интенционалов  $I$ . В начальный момент эти списки пусты. Просматривается таблица хранения дерева решений. Для каждой вершины вывода реализуется обратная цепочка рассуждений. Описание семантической сети строится по следующим правилам:

1. Если имя переменной отсутствует в списке  $V$ , то оно добавляется в список и делается текущим. Если переменная есть в списке, она делается текущей.

2. Факту соответствует цепочка вывода с одним номером варианта из таблицы вывода и соответствующим именем переменной вывода. Атрибуты и значения факта соответствуют переменной и ее значению из каждой строки таблицы вывода с соответствующим номером варианта и именем переменной вывода. Метка факта может быть определена как текущее имя переменной вывода  $\cap$  номер варианта, где  $\cap$  операция конкатенации.
3. Одновременно формируется интенционал. Если в списке I отсутствует элемент с текущим именем переменной вывода, переменной и значением переменной из таблицы вывода, соответствующий элемент добавляется к этому списку.
4. После полного просмотра таблицы дерева решений окончательно формируется экстенционал. Для этого проверяется все ли переменные из списка I для каждого полученного отношения, упомянуты в факте, соответствующем отношению. Если какая-либо переменная отсутствует, она включается со значением Nil.

Проведем сравнение этих двух представлений. Дерево решений более просто с точки зрения пользователя, поскольку может быть реализовано, как последовательность ответов на вопросы, причем значения ответов выбираются из заранее определенного перечня. Таким образом, процесс принятия решения достаточно нагляден, так как отображается в виде естественной цепочки рассуждений. Однако, это справедливо лишь для небольших по объему деревьев решений. Действительно, чем больше дерево решений, чем сложнее его структура, тем сложнее отследить логику проводимых рассуждений. Поэтому возникает задача выделения поддеревьев (частичных деревьев решений), которые отображают отдельные аспекты вывода решений. При этом возникает ряд проблем:

- как разбить исходное дерево на поддеревья;
- какое из поддеревьев наиболее информативно для пользователя с точки зрения обоснования принимаемого решения и т.д.

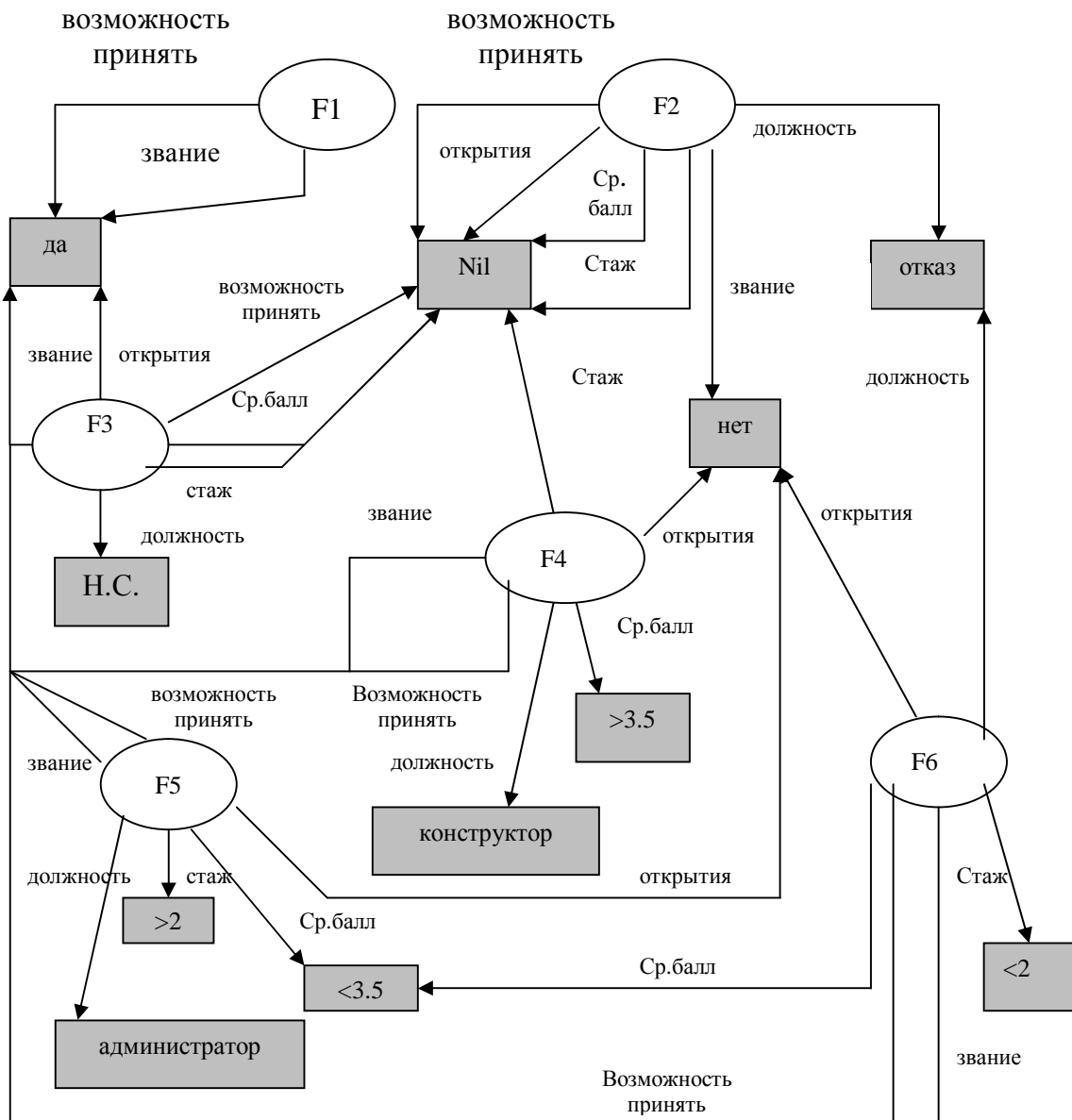


Рис.2. Пример семантической сети

Теперь рассмотрим приведенный здесь вариант семантической сети. Сразу обратим внимание на то, что поскольку существует отображение одного представления в другое, эти представления с формальной точки зрения тождественны. Однако, представление в виде семантической сети позволяет упростить решение целого ряда задач.

Как видно из алгоритма преобразования дерева решений в семантическую сеть, каждый факт представляет уже готовый вариант вывода. Такая сеть содержит все возможные варианты выводов в виде фактов. Это дает дополнительные возможности для анализа. Например, обычными средствами

баз данных можно выделить факты, относящиеся даже к разным отношениям, обладающие одинаковыми атрибутами и значениями их характеризующими.

В семантической сети можно осуществить выводы, которые далеко не очевидны для дерева решений. Возьмем такой простейший пример. Пусть в дереве решений определен вывод установления родительских отношений в первом поколении. Необходимо определить родительские отношения во втором поколении. Решение этой задачи для семантической сети очевидно. Необходимо выделить факты родительских отношений. Удалить из интенционала объекты, не соответствующие родителям и повторить процесс выделения фактов родительских отношений.

Это означает, что в семантической сети возможно введение средств построения функциональных зависимостей подобно тому, как это сделано в языках функционального программирования таких как, например, ЛИСП [4]. Вернемся к примеру с приемом на работу. Рассмотрим экстенционал «Должность». В описание фактов этого экстенционала входит атрибут «Возможность принять», который соответствует вершине вывода, а, следовательно, дереву решений (частичному дереву решений). Можно рассматривать факт, как список атомов, каждому из которых ставится в соответствие значение либо непосредственно, либо в процессе вывода. Если ввести операции манипулирования над такими списками, появляется возможность создавать и модифицировать деревья решений формальными методами.

Таким образом, представляется целесообразным сочетать в единой системе представление в виде деревьев решений и класса семантических сетей, что дает возможность наглядного отображения процесса принятия решений и дополнительные возможности в конструировании механизмов вывода.

## Литература

1. Р.Левин, Д.Дранг, Б.Эделсон. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрацией на Бейсике. – М., Финансы и статистика, 1991.
2. Робототехника и гибкие автоматизированные производства. В 9-ти книгах. Кн. 6. Техническая имитация интеллекта / В.М. Назаретов, Д.П. Ким. Под ред. И.М. Макарова. – М., Высшая школа, 1986.
3. Солодовников В.И., Солодовников И.В. Разработка ВС с использованием графовых моделей принятия решений – Новые информационные технологии. Материалы пятого научно-практического семинара. М., 2002.
4. Э.Хювенен, И.Сеппянен. Мир ЛИСПА. Методы и системы программирования. – М., Мир, 1990.

## **Общие принципы преобразования изображений в целях улучшения их визуального качества**

**С. М. Кулясов**

**Кафедра информатики и математики МПГУ (г. Москва)**

При разработке математических методов улучшения визуального качества изображений используем модель цифрового изображения, в которой исходное изображение (ИИ) представлено в виде двух аддитивных составляющих:

- исходная локально усредненная (преимущественно по малоконтрастным объектам) составляющая, называемая также исходной локальной освещенностью (ИЛО);
- исходная фоно-объектовая составляющая (ИФО).

**Таким образом, получаем следующую модель  
 $ИИ=ИЛО+ИФО$ .**

Общая схема преобразования состоит в том, что сначала формируется ИЛО путем «сильного» локального усреднения ИИ, а затем, в силу аддитивности, ИФО определяется как разность ИИ и ИЛО. Далее ИЛО и ИФО обрабатываются независимо, в результате чего образуются две новые скорректированные аддитивные составляющие:

- скорректированная локальная освещенность (СЛО);
- скорректированная фоно-объектовая составляющая (СФО).

Путем сложения СЛО и СФО образуется скорректированное изображение (СИ) с улучшенными качествами, которые позволяют более комфортно воспринимать изображение человеком.

ИЛО должно представлять собой сильно локально усредненное несмещенным способом ИИ, причем при явном весовом преимуществе слабоконтрастных (мало различимых между собой) объектов и элементов фона. Именно ими должно определяться среднее локальное значение. Таким образом, ИЛО – это усреднение ИИ по большинству пикселей соседей, имеющих сходные значения и относящиеся либо к малоразличимой фоновой специфике,

либо к малоразмерным объектам, или к отдельным деталям объектов больших размеров некоторого уровня яркости.

ИЛО используется для двух целей:

- 1). «Снятие» с исходного изображения ИФО.
- 2). Формирование скорректированной локальной освещенности.

Цель корректировки ИЛО заключается в уменьшении диапазона значений для выведения малоразличимых объектов в область видимости. Т.е. область определения ИЛО, которой является отрезок  $[0; 255]$ , сужается до диапазона  $[8; 247]$  или  $[16; 239]$ .

Смысл преобразования ИФО в СФО состоит в том, чтобы, сохраняя хорошо видимые объекты (даже допуская снижение их амплитуды), усилить плохо различимые или совсем зрительно неразличимые объекты и их детали, путем «поднятия» их границ, до уровня зрительной видимости человеком. Само преобразование включает две процедуры. Цель первой из них – повышение контрастности границ объектов. Вторая процедура направлена на обеспечение нелинейного уменьшения значений, сводящаяся преимущественно к подавлению больших всплесков.

Обе операции должны быть сбалансированы так, чтобы обеспечивался плавный рост малых значений (усиление слаборазличимых и неразличимых объектов) и сохранение или слабое подавление «очень ярких» и «очень темных» участков изображения – мягкое приближение их к умеренно ярким и хорошо различимым.

Таким образом, после проведенных преобразований и формирования СЛО и СФО, получаем скорректированное изображение с необходимым, для комфортного восприятия человеком, качеством.

## Алгоритм сравнения статей на основе семантической близости понятий\*

Д.Е. Кондратьев, О.В. Тихонова  
Центр «Проблемы исторического познания»  
Институт всеобщей истории РАН  
Тел: (095) 938-58-69

Среди многообразия задач анализа текстов на естественном языке [1, 2] выделим задачу сравнения текстовых фрагментов – статей. Под статьей будем понимать последовательность предложений в обычной орфографической записи. Вообще говоря, две статьи можно сравнивать по различным критериям [2]. Рассмотрим метод сравнения статей на основе семантической близости понятий, выраженных словами статей.

Перед рассмотрением статей необходимо создать категориальную структуру понятий. Эта структура представляет собой определенную иерархию понятий. На нижнем уровне иерархии находятся так называемые понятия первого уровня. Каждое такое понятие выражает лексическое значение некоторого слова. Понятиям первого уровня в соответствие поставлены лексемы. Лексемой мы называем совокупность всех грамматических форм, которые свойственны одному и тому же слову при всех его употреблениях. Например: *вол, вола, волу, волом, волы* и т.д. На следующем уровне иерархии расположены обобщающие понятия. Каждое такое понятие включает в себя группу понятий первого уровня. Например: *животноводство = {животное, вол, баран, скотина, бык и т.д.}*. Понятие первого уровня может соответствовать нескольким обобщающим понятиям. В свою очередь обобщающие понятия могут группироваться в категории более высокого уровня. Полученная структура понятий может быть использована для анализа (сравнения) различных статей, оперирующих с заданными понятиями.

Перейдем непосредственно к процедуре сравнения двух статей.

---

\* Исследование выполнено при финансовой поддержке РГНФ. Проекты 01-01-00261а, 02-05-12023в.



Будем называть ключевыми словами обобщающие понятия в структуре понятий. Общее количество ключевых слов в структуре понятий обозначим как  $N$ . Их порядок строго определен нумерацией от 1 до  $N$ .

Для определения сходства или различия двух статей следует проделать следующие действия. В первую очередь необходимо определить список лексем, характеризующих данную статью. Для этого требуется провести лексико-морфологический анализ текста статьи. Входная цепочка символов текста разбивается на подцепочки, представляющие отдельные словоформы и затем каждой словоформе сопоставляется ее морфологическая структура, т.е. имя соответствующей лексемы с указанием морфологических характеристик. Существует два основных метода морфологического анализа: декларативный и процедурный [3, 4]. Для декларативного метода характерно наличие полного словаря всех возможных словоформ для каждого слова. При этом каждая словоформа снабжается полной и однозначной морфологической информацией. Задача морфологического анализа в этом случае сводится к поиску нужной словоформы в словаре. В процедурных методах каждое слово разбивается на сегменты (морфы) соответствующие определенным лингвистически содержательным позициям, и с этими сегментами связываются некоторые морфологические характеристики [5]. Набор характеристик всей словоформы образуется объединением характеристик составляющих ее морф. Морфологический анализ в этом случае сводится к определению набора морф, составляющих заданную словоформу. На практике обычно применяется комбинированный вариант анализа, включающий в себя как элементы декларативного, так и элементы процедурного морфологического анализа.

Следующий шаг – определить набор ключевых слов каждой статьи по полученным лексемам. Для этого по структуре понятий определяем, каким понятиям первого уровня соответствуют полученные лексеммы, и, в свою очередь, каким обобщающим понятиям (т.е. ключевым словам) соответствуют полученные понятия первого уровня.

Далее определяем последовательности  $\{x_i\}$  и  $\{y_i\}$  частот вхождения ключевых слов в первую и вторую статьи соответственно. Эти последовательности определяем по следующей процедуре. Рассматриваем список ключевых слов от 1 до  $N$ . Если в наборе ключевых слов первой статьи не встречается ключевое слово с индексом  $i$ , то соответствующее значение  $x_i$  приравнивается нулю. В противном случае значение  $x_i$  равно числу повторений ключевого слова с индексом  $i$  в составе ключевых слов первой статьи. Значения  $y_i$  определяются аналогично.

Наконец, для двух последовательностей  $\{x_i\}$  и  $\{y_i\}$ , характеризующих две статьи, определяется функционал сходства

$$L = \frac{\sum_1^N |x_i - y_i| \cdot m_i}{\sum_1^N (x_i + y_i) \cdot m_i} \quad (1)$$

где  $m_i$  – весовой коэффициент значимости ключевого слова, определяемый методом экспертных оценок.

Если значение функционала  $L$  равно 0, то это означает абсолютную похожесть статей, то есть все ключевые слова, соответствующие одной статье, совпадают с ключевыми словами второй статьи. При полном различии статей функционал  $L$  равен 1, то есть ключевые слова одной статьи полностью отличны от ключевых слов второй статьи. Если некоторые ключевые слова одной статьи совпадают с ключевыми словами второй статьи, а некоторые не совпадают, то значения функционала  $L$  лежат в промежутке между 0 и 1, что означает частичную схожесть статей, причем, чем меньше это значение, тем более сходны статьи. Например, если все ключевые слова одной статьи совпадают с ключевыми словами второй, отличие лишь в том, что во второй статье дополнительно есть некоторое ключевое слово, то значение функционала будет близко к 0. Если все ключевые слова одной статьи отличаются от ключевых слов второй статьи за исключением лишь одного некоторого ключевого слова, то значение функционала будет близко к 1.

Наряду с выбранным функционалом  $L$  следует рассмотреть функционал  $L_2$

$$L_2 = \frac{\sum_1^N |x_i^2 - y_i^2| \cdot m_i}{\sum_1^N (x_i^2 + y_i^2) \cdot m_i} \quad (2)$$

Этот функционал отличается от предыдущего тем, что в нем численное значение сходства (лежащее в пределах от 0 до 1) более чувствительно к отличиям, состоящим в том, что одно и то же ключевое слово встречается в двух статьях разное количество раз. Например, если некоторое ключевое слово встречается в одной статье один раз, а в другой, скажем, пять, то численные значения второго критерия покажут более значительные отличия. Применение того или иного критерия сходства будет зависеть от рассматриваемых задач.

Нельзя недооценивать важность правильного выбора значений коэффициентов  $m_i$ . При рассмотрении статей на предмет сходства лишь по некоторой совокупности ключевых слов (понятий) целесообразно для этих ключевых слов задать значение  $m_i$  равное 1 (или, например, 10 для особо определяющих ключевых слов), тогда как для ключевых слов, не характеризующих рассматриваемую данную совокупность, значения  $m_i$  принять равными 0.

Рассмотрим применение данного метода на примере сравнения четырех статей “Византийского земледельческого закона” – памятника Византийского права VIII в.:

1. *Если кто-либо возьмет вола для работы и вол подойдет, пусть обследуют судьи, и если он подойдет на той работе, для которой его выпрашивал взявший вола, да будет неповинен; если же на другой работе подойдет, пусть отдаст вполне здорового вола.*
2. *Если кто-либо рубящий в дубраве дрова не обратит внимания и дерево упадет и убьет вола, или осла, или какую-либо другую скотину, пусть отдаст голову за голову.*
3. *Если грызутся две собаки и хозяин одной ударит другую мечом, или палкой, или камнем и если от этого удара собака ослепнет, или*

*подохнет, или что-либо иное сопряженное с опасностью претерпит, то пусть убивший возместит ущерб ее хозяину.*

4. *Если кто-либо имеющий сильного пса, разъяренного против себе подобных, направит сильного пса на более слабых и случится, что какой-либо пес будет изувечен или подохнет, пусть возместит ущерб хозяину его и получит 12 ударов плетью.*

Для данных статей была определена категориальная структура понятий.

Вычисление значений функционалов  $L$  и  $L_2$  осуществлялось при всех  $m_i = 1$ .

Полученные результаты представлены в таблице 1.

Таблица 1.

Номера сравниваемых статей	1+2	1+3	1+4	2+3	2+4	3+4
Значение функционала сходства $L$	0,5	0,83	0,8	0,9	0,89	0,26
Значение функционала сходства $L_2$	0,55	0,93	0,92	0,96	0,95	0,33
Интерпретации	Сходны	Различны	Различны	Различны	Различны	Похожи

Интерпретация результатов строится по следующему правилу: при значении функционала  $L$  от 0 до 0.4 статьи похожи, от 0.4 до 0.65 – схожи, более 0.65 - различны. В общем случае выбор интерпретации численных значений – это задача, решаемая методом экспертных оценок. Полученные результаты позволяют сделать вывод о применимости разработанного алгоритма для анализа фрагментов исторических контекстов.

## Список литературы

1. Шемакин Ю.И. Начала компьютерной лингвистики: Учебное пособие. – М.: Изд-во МГОУ, АО Росвузнаука, 1992.
2. Виноград Т. Работа с естественными языками. // Современный компьютер. Сб. научно-популярных статей. Пер. с англ. / Под ред. В.М. Курочкина. – М.: Мир, 1986.
3. Белоногов Г.Г., Новоселов А.П. Автоматизация процессов накопления, поиска и обобщения информации. – М.: Наука, 1979.
4. Попов Э.В. Общение с ЭВМ на естественном языке. – М.: Наука, 1982.
5. Апресян Ю.Д. Лингвистический процессор для сложных информационных систем. – М.: Наука, 1992.

## К проблеме математического описания многоязыковой машинной транскрипции

Клышинский Э.С., МГИЭМ [klyshinsky@mail.ru](mailto:klyshinsky@mail.ru)

Слезкниа О.Ю., РГГУ [alienta@yandex.ru](mailto:alienta@yandex.ru)

Необходимость передачи с одного языка на другой имен и фамилий людей, географических названий, а также названий различных организаций, пароходов, кинотеатров и мн. др. появилась уже довольно давно. И с увеличением количества информации в современном мире потребность в подобной передаче все больше увеличивается, что и делает актуальным создание машинной системы передачи имен с одного языка на другой.

Самый простой способ передачи имен собственных с одного языка на другой – это просто заимствование слова без его видоизменения. Так сейчас и происходит во многих странах Европы. То есть, французское имя *Francois* – Франсуа и в английском и в турецком языках будет иметь вид *Francois*, хотя читаться уже будет в первом случае – Френкойс, а во втором - Франджойс. Правильное прочтение такого заимствованного имени требует от читающего знания соответствующего иностранного языка, если же он им не владеет, то подобным образом заимствованное имя будет читаться по правилам уже другого языка, что неизбежно приведет к искажениям его звукового облика и, как следствие, потери идентификации объекта по его имени, то есть нарушению основной функции имени. Но непосредственно заимствование возможно лишь когда и исходный язык и язык «перевода» используют одну и ту же графическую систему (например, латиницу). Если же нужно передать слово с латиницы на кириллицу, необходимо уже ставить в соответствие буквы одной графической системы<sup>1</sup> буквам другой (например, **a** → **а**, **b** → **б** и т. д.). Такая побуквенная передача слов, записанных с помощью одной графической

---

<sup>1</sup> Здесь имеется в виду именно графическая система (например, латиница или кириллица), но не конкретный язык.

системы, средствами другой графической системы, называется транслитерацией. При помощи этого метода передается тоже лишь графический образ слова, но не его звучание. Например, имя *Francois* будет транслитерировано на кириллицу как *Франкоис*. Недостатки транслитерации очевидны – как и в случае с заимствованием нарушается функция имени. Достаточно того, что если Франсуа назвать Франкоисом, то человек даже не догадается, что обращаются именно к нему.

Второй теоретически возможный метод передачи имен собственных с одного языка на другой – это перевод, при котором фактически используется «словарь имен», состоящий из словарных статей вида: Arthur - Артур, Shakespear – Шекспир и др. (с английского на русский); Hilda – Ильда, Rodriguez – Родригес и др. (с испанского на русский) и тому подобные базы «перевода» имен с одного языка на другой. Однако количество имен собственных в языке не ограничено – каждый день возникают все новые названия, поэтому для их *перевода* невозможно составить конечную базу имен собственных (или базы, хотя бы охватывающей около 90% имен собственных).

И последний, на наш взгляд наиболее приемлемый метод передачи имен собственных с одного языка на другой – это метод транскрипции (точнее говоря, практической транскрипции),

когда имени собственному одного языка ставится в соответствие слово другого языка, наиболее точно отражающее его звучание в родном языке. Этот метод (как и метод перевода) ориентируется на конкретные языки, но опирается не на «словарь имен», а на правила передачи отдельных букв и буквосочетаний с одного языка на другой. Так, например, в базе правил транскрипции с английского языка на русский должны быть правила вида: **c** перед гласными заднего ряда → **к**, **o** → **о**, **p** → **п**, **r** → **р**, **ation** на конце слова переходит в **ейшен**. Вышеперечисленные правила позволяют передать английское слово *corporation* на русский язык как «корпорейшен» (а не «корпоратион», как получилось бы, если бы использовался метод транслитерации).

Однако, например, английское сочетание **ation** (на конце слова) должно обозначать «эйшен» не только в русском, но и других языках (так как основным при передаче является сохранение именно звукового образа слова). Поэтому составлять подобные базы правил передачи с каждого исходного языка на каждый язык транскрипции представляется нерациональным. Гораздо эффективнее, как нам кажется, составить общую фонетическую таблицу, которая будет лежать в основе промежуточного фонетического представления. То есть, практическая транскрипция будет осуществляться по следующей схеме:

#### СЛОВО X ИСХОДНОГО ЯЗЫКА L1

1. каждому элементу слова исходного языка подбирается соответствующий элемент общей фонетической таблицы ( $\cong$  звук);

#### ПРОМЕЖУТОЧНОЕ ФОНЕТИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ (=ЯЗЫК-ПОСРЕДНИК) СЛОВА X

2. каждому элементу промежуточного фонетического представления подбирается соответствующая буква или буквосочетание, выражающие этот звук в языке L2;

#### СЛОВО X ЯЗЫКА ТРАНСКРИПЦИИ L2

Плюсом такого подхода является сокращение количества наборов правил транскрипции в случае работы со многими языками. При отсутствии языка-посредника приходилось бы создавать базы для транскрипции с каждого языка на все остальные, что составило бы  $N_L * (N_L - 1)$  баз, где  $N_L$  – количество языков, с которыми производится работа. При транскрипции через язык-посредник это количество составит лишь  $2 * N_L$ , так как потребуются базы лишь для транскрипции на язык-посредник и с него.

С лингвистической точки зрения основные сложности при создании системы машинной транскрипции, использующей промежуточное фонетическое представление, заключается:



1. в определении того, как же слово читается в языке перевода (с которого осуществляется транскрипция);
2. в составлении такой базы правил (для каждого языка в случае многоязычной системы транскрипции), которые позволяли бы для каждого звука или звукосочетания языка L1 находить максимально похожий звук (звукосочетание) языка L2.

Успешное решение задачи практической транскрипции на современном этапе требует создания программы или как минимум алгоритма машинной транскрипции. Однако создание подобного алгоритма не возможно без формального математического описания процесса транскрипции. В данной статье авторы ставили своей целью восполнение данного пробела.

В отличие от общепринятого в лингвистике соглашения, в данной статье мы принимаем, что сама буква, а не только обозначаемый ею звук, обладает некоторыми параметрами (например, гласность/согласность, ряд и так далее). Это необходимо для того, чтобы выяснить, какой звук обозначает данный символ в определенной позиции в слове и какой набор параметров будет соответствовать данному звуку. Без такого предположения подобная операция представляется затруднительной или трудоемкой.

Определим параметр как пару  $P = \langle N, V \rangle$ , где  $N$  – имя параметра, а  $V$  – его значение. Параметр будет отображать некоторые характеристики буквы, важные для транскрипции, или позволяющие классифицировать буквы по группам. Например: <"ряд", "передний">, <"тип", "гласная">, <"ударение", "безударная">. Будем считать, что два параметра равны, если совпадают их имена и значения.

Букву определим как пару, состоящую из графемы, однозначно идентифицирующей данную букву, и набора параметров, либо изначально присущих данной букве, либо отражающих положение буквы в слове.  $S = \langle C, \{P\} \rangle$ , где  $C$  – фиксированный символ (графема), обозначающий данную букву, а  $P$  – набор ее параметров. При этом будем считать, что различные написания одной и той же буквы (например, строчное и прописное или

начальное, срединное, конечное и изолированное) имеют одно и то же обозначение, однако могут обладать (в зависимости от применения) различными значениями определенных параметров. Набор параметров определяется критичностью различения таких написаний при транскрипции и особенностями языка.

Примером буквы может служить пара  $\langle 'A', \{ \langle \text{“тип”}, \text{“гласн”} \rangle, \langle \text{“написание”}, \text{“прописн”} \rangle, \langle \text{“ряд”}, \text{“задний”} \rangle \} \rangle$ , где 'A' – графема, идентифицирующая данную букву, а множество, заключенное в фигурные скобки – множество параметров данной буквы. Здесь и в дальнейшем выделим с помощью апострофов графемы, относящиеся к символам некоторого языка. Служебные графемы, предназначенные для обеспечения процесса транскрипции, будут обозначаться несколькими символами и не будут заключаться в апострофы.

Введем следующие операторы сравнения букв.

Оператор  $=$  производит сравнение как графем букв, так и их наборов параметров. Две буквы  $S_1$  и  $S_2$  равны в смысле оператора  $=$  ( $S_1=S_2$ ), если равны их графемы и множество параметров  $S_2$  является подмножеством параметров  $S_1$ .

Оператор  $\approx$  производит сравнение только наборов параметров букв. Две буквы  $S_1$  и  $S_2$  равны в смысле оператора  $\approx$  ( $S_1 \approx S_2$ ), если множество параметров  $S_2$  является подмножеством параметров  $S_1$ .

Как это уже было замечено выше, при многоязыковой машинной транскрипции удобнее производить транскрипцию через промежуточное фонетическое представление (язык-посредник). В связи с этим транскрипция в нашем случае будет состоять из двух частей – перевода с языка оригинала на язык-посредник и перевода с языка-посредника на язык транскрипции. Плюсом такого подхода является сокращение количества наборов правил транскрипции в случае работы со многими языками. При отсутствии языка-посредника приходилось бы создавать базы для транскрипции с каждого языка на все остальные, что составило бы  $N_L \cdot (N_L - 1)$  баз, где  $N_L$  – количество языков, с

которыми производится работа. При транскрипции через язык-посредник это количество составит лишь  $2 * N_L$ , так как потребуются базы лишь для транскрипции на язык-посредник и с него.

Однако подобный подход налагает дополнительные требования на язык-посредник. Алфавит языка-посредника должен содержать звуки всех языков, с которых производится транскрипция. Кроме алфавита для языка-посредника должен определяться набор параметров, которыми могут обладать буквы этого языка. Для того, чтобы корректно произвести транскрипцию, правила транскрипции с языка-посредника должны охватывать все буквы алфавита этого языка, что несколько увеличивает объем правил. Одновременно с этим за счет проведения дополнительных работ скорость транскрипции падает.

Также имеется необходимость определить алфавит каждого языка с тем, чтобы сопоставить любому символу, встречающемуся в данном языке, букву из этого алфавита (графему и набор параметров).

В целом, процесс транскрипции разобьем на пять этапов:

1. преобразование написания слова на языке оригинала во внутреннее представление;
2. выделение слогов, расстановка переносов и ударений;
3. перевод внутреннего представления слова в промежуточное фонетическое написание;
4. перевод промежуточного фонетического написания слова во внутреннее представление слова на языке транскрипции;
5. преобразование внутреннего представления слова на языке транскрипции в написание слова на языке транскрипции.

Опишем каждый из этих этапов подробнее

**1. Преобразование написания слова на языке оригинала во внутреннее представление** состоит в преобразовании слова языка, записанного как множество символов  $W = \langle G \rangle$ , во множество букв  $W' = \langle S \rangle$ . Здесь  $G$  – символ (знак), а в случае машинной транскрипции - информационный код знака в одной из компьютерных кодировок (ASCII, ANSI или иной другой). Для такого

преобразования вводится множество правил, называемых правилами алфавита, сопоставляющих символу (информационному коду знака)  $G$  букву  $S$ .  $\mathfrak{R}_a = \{R_a\}$ , где  $\mathfrak{R}_a$  – база правил алфавита, а  $R_a = \langle G, S \rangle$  – правило.

Примерами правил алфавита могут служить следующие множества.

$\langle 'A', \langle 'A', \{ \langle \text{“тип”}, \text{“гласн”} \rangle, \langle \text{“написание”}, \text{“прописн”} \rangle, \langle \text{“ряд”}, \text{“задний”} \rangle \} \rangle \rangle$

$\langle 'a', \langle 'A', \{ \langle \text{“тип”}, \text{“гласн”} \rangle, \langle \text{“написание”}, \text{“строчн”} \rangle, \langle \text{“ряд”}, \text{“задний”} \rangle \} \rangle \rangle$

$\langle 'B', \langle 'B', \{ \langle \text{“тип”}, \text{“согласн”} \rangle, \langle \text{“написание”}, \text{“прописн”} \rangle, \langle \text{“звонкость”}, \text{“звонкая”} \rangle \} \rangle \rangle$

$\langle 'b', \langle 'B', \{ \langle \text{“тип”}, \text{“согласн”} \rangle, \langle \text{“написание”}, \text{“строчн”} \rangle, \langle \text{“звонкость”}, \text{“звонкая”} \rangle \} \rangle \rangle$

Курсивом здесь выделена часть, относящаяся к букве ( $S$ ), а полужирным шрифтом – параметры буквы.

Для всех графем входного слова последовательно находятся такие правила, что графема входного слова совпадает с графемой из найденного правила. Внутреннее представление слова получается путем последовательной конкатенации букв, входящих в полученные правила. Кроме того, в начало и конец слова добавляются специальные буквы, обозначающие начало и конец слова. Все графемы, для которых не было найдено соответствия в правилах алфавита, считаются знаками препинания и передаются дальше без изменений с соответствующей пометкой. Перед началом группы знаков препинаний ставится буква конца слова, после нее – начала слова. Подобный подход позволяет вычленить не только знаки препинания, но и символы из других алфавитов, которые не должны транскрибироваться в рамках данного языка.

Таким образом  $W \Rightarrow W' = \bigcup_{m=1}^N S_m$ , причем

a)  $S_1 = \langle BEG, \{ \} \rangle$ ,

b)  $S_N = \langle END, \{ \} \rangle$ , здесь  $BEG$  и  $END$  – графемы, обозначающие начало и конец слова,

c)  $S_m = S$ , если  $\exists R_a = \langle G, S \rangle \in \mathfrak{R}_a : G = G_j$ , здесь  $j = 1..M$ , где  $M$  – общее количество графем во входном слове,

d)  $S_m = \langle G_j, \{ \} \rangle$ , если не  $\exists R_a = \langle G, S \rangle \in \mathfrak{R}_a : G = G_j$ ,

е)  $S_m = \langle \text{BEG}, \{ \} \rangle$ , если  $S_{m-1}$  получено по правилу d), а  $S_{m+1}$  получено по правилу с),

ф)  $S_m = \langle \text{END}, \{ \} \rangle$ , если  $S_{m-1}$  получено по правилу с), а  $S_{m+1}$  получено по правилу d),

Здесь  $m \in (1, N)$ , где  $N$  – общее количество букв в выходном слове (во внутреннем формате).

**2. Выделение слогов и расстановка переносов** производятся для того, чтобы определить закрытые/открытые слоги и ударные/безударные буквы. Любая буква, находящаяся в конце слога, приобретает дополнительный параметр «буква в слоге» со значением «открытая». Для остальных букв значение этого параметра – «закрытая».

Выделение слогов производится по следующему алгоритму. Для алфавита каждого языка может быть задан набор слогообразующих букв. В качестве части слога, присоединяемой к слогообразующей букве, берется половина букв между двумя слогообразующими. При нечетном количестве букв, средняя передается следующему слогу. Исключение делается для приставок, суффиксов и окончаний, разделение на слоги которых фиксировано. Они присоединяются к остальной части слова как отдельный слог или несколько самостоятельно выделенных слогов. Написание и деление на слоги таких приставок, суффиксов и окончаний задается отдельной базой правил.

Расстановка ударений, как и выделение слогов, не является обязательной. Их необходимо производить для языков, в которых буквы читаются различным образом в зависимости от того, в какой позиции находится данная буква – в ударной или безударной, в конце слога или нет.

**3. Задачей перевода внутреннего представления слова в промежуточное фонетическое написание** является приведение слов различных языков к единой записи в рамках алфавита языка-посредника. На вход данного этапа поступает последовательность букв языка. Выходом этапа является набор фонем, входящих в состав языка-посредника.

Под строкой (словом) здесь будем понимать упорядоченное множество букв. Подстрокой слова будет являться подмножество последовательно идущих букв данного слова. Обозначим через  $W_l^i$  подстроку слова  $W$  длиной  $l$ , начинающуюся с буквы в позиции  $i$ . В дальнейшем верхний индекс подстроки будет обозначать позицию, с которой начинается данная подстрока в слове, а нижний индекс будет обозначать длину подстроки. Символом  $*$  будем обозначать произвольное значение позиции.

Под правилом перевода будем понимать пару  $R_t = \langle W_{l_1}^*, \bar{W}_{l_2} \rangle$ , где  $W_{l_1}^*$  - строка-образец, а  $\bar{W}_{l_2}$  - строка-результат. Правило  $R$  применимо к подстроке  $W_{l_1}^i$ , если строка-образец сравнима с  $W_{l_1}^i$ . Под сравнимостью понимается нахождение равенства букв из  $W_{l_1}^*$  и  $W_{l_1}^i$  в одних и тех же позициях подстрок. При этом здесь две буквы  $S_1$  и  $S_2$  равны, если  $S_1 = S_2$  или  $S_1 \approx S_2$ . Подробный алгоритм определения применимости правила к строке приведен ниже.

Под переводом подстроки  $W_{l_1}^i$  будем понимать функцию  $\bar{W}_{l_2} = F^t(W_{l_1}^i)$ , такую, что  $\exists R_t = \langle W_{l_1}^*, \bar{W}_{l_2} \rangle \in \mathfrak{R}_t$  применимое к  $W_{l_1}^i$ . Здесь  $\mathfrak{R}_t = \{R_t\}$  - база правил перевода.

Задача перевода в промежуточное фонетическое написание в этом случае может быть представлена следующим образом.

Пусть имеем на входе на данный этап некоторое слово  $W = \langle S_1, S_2, \dots, S_a \rangle$  и набор правил перевода  $\mathfrak{R}_t$ . Перевод внутреннего представления в промежуточное фонетическое написание в этом случае будет заключаться в нахождении и применении упорядоченного подмножества правил  $\mathfrak{R} = \langle W_{l_1}^*, \bar{W}_{l_2} \rangle$ , таких что:

- 1)  $i = \langle i_1, i_2, \dots, i_n \rangle$ , где  $n$  – число правил в подмножестве  $\mathfrak{R}$ ;
- 2)  $l = \langle l_1, l_2, \dots, l_n \rangle$ ;
- 3)  $\sum_{j=1}^n l_j = a$ ;
- 4)  $i_j = 1$ ;

5)  $i_{k+l} = i_k + l_k$  для  $k < n$  и  $i_n + l_n = a + 1$ ;

6)  $\forall i, l \exists R_i = \langle W_i^*, \bar{W}_{l2} \rangle : \exists \bar{W}_{l2} = F^l(W_i^i)$ .

Здесь множество  $i$  – это множество позиций с которых применимы правила, а множество  $l$  – множество длин подстрок.

Результатом перевода будет являться конкатенация результатов последовательного применения правил перевода.

$$\bar{W} = \bigcup_{i,l} F^l(W_i^i)$$

Проверка применимости правила к строке производится следующим образом. Правила могут содержать в себе буквы со специально определенной графемой ЕМРТУ. Сравнение буквы правила и буквы строки производится при помощи оператора =, если графема буквы правила не равна ЕМРТУ, и при помощи оператора  $\approx$  в противном случае.

В начале перевода внутреннего представления слова в промежуточное фонетическое написание текущая позиция во входной строке устанавливается в 1. Далее, до тех пор, пока не будет достигнут конец слова, последовательно применяется следующий алгоритм.

Сохраняем текущую позицию. Далее пытаемся найти все правила, применимые для строки, начинающейся с текущей позиции. Если первые несколько последовательно идущих букв в правиле имеют графему, равную ЕМРТУ, то уменьшаем текущую позицию на количество таких букв. Если текущая позиция меньше 1, то считаем, что правило не применимо, восстанавливаем текущую позицию и переходим к следующему правилу.

Начиная с полученной текущей позиции последовательно сравниваем буквы строки и правила. Если хотя бы одна буква строки не равна соответствующей букве правила, то считаем, что правило не применимо, восстанавливаем текущую позицию и переходим к следующему правилу. Если сравнение всех букв прошло успешно, то считаем, что правило применимо. В этом случае помещаем сохраненную текущую позицию в множество  $i$ . Во множество  $l$  помещаем количество букв в правиле за вычетом последовательно

идущих букв в начале и в конце правила, имеющих графему равную ЕМРТУ. Далее восстанавливается сохраненная текущая позиция и алгоритм переходит к следующему правилу.

По окончании перебора всех правил текущая позиция увеличивается на величину, сохраненную в множестве  $l$ . В случае если к одной и той же позиции в слове применимо несколько правил, то для каждого правила заводятся свои множества  $i$  и  $l$ .

**4. Этап перевода промежуточного фонетического написания слова во внутреннее представление слова на языке транскрипции** аналогичен этапу 3, но имеет противоположные задачи. Он служит для того, чтобы сформировать последовательность букв, отражающих полученное звучание слова в языке транскрипции. Работа этапа осуществляется по тем же принципам, что и этапа 3. Здесь правила являются не столь многозначными, как на этапе 3, так как имеется возможность задать одно определенное правило для передачи данного набора звуков при наличии альтернативы.

**5. Преобразование внутреннего представления слова на языке транскрипции в написание слова на языке транскрипции** является обратным относительно этапа 1. Здесь могут использоваться те же самые правила, что и на этапе 1, так как в большинстве случаев должно существовать взаимнооднозначное соответствие между графемой и буквой с данным набором параметров. Буквы с графемами BEG и END удаляются, знаки препинания передаются соответствующими символами.

Предложенный метод позволяет формально подойти к проблеме машинной транскрипции в многоязыковых системах. Это позволит строго сформулировать требования к языку-посреднику и языкам, участвующим в транскрипции, исследовать их особенности и свойства. Формализация процесса транскрипции позволяет проще перейти к решению задачи машинной транскрипции.



## **Язык имитационного моделирования дискретных систем ВИСМО 2: математическая модель, алгоритмы управления, структура языка**

А.Б. Корякин

г. Москва, Московский государственный технический университет им. Н.Э. Баумана

Имитационное моделирование является одним из методов экспериментального исследования реальных или проектируемых систем. Имитационные модели позволяют просто учитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов системы, многочисленные случайные воздействия и другое. Имитационное моделирование – один из методов получения информации о поведении системы, особенно на этапе ее проектирования.

В настоящее время известно большое количество различных программных комплексов для имитации дискретных систем. Процесс разработки модели можно разбить на два этапа: построение блочной модели, детальное описание алгоритмов поведения модели. Блочная модель – это совокупность графических блоков и связей между ними (визуальное представление модели). Описание алгоритмов производится с использованием дополнительного языка. Таким образом, происходит разделение между визуальным и текстовым представлением модели, аналогичным соответствующим методикам программирования: визуальным и классическим - текстовым. В большинстве систем имитационного моделирования описания модели осуществляется только в текстовом виде с помощью специального языка моделирования, что еще на ранних стадиях моделирования заставляет пользователя задумываться о ненужных деталях модели.

При построении модели объекта необходимо учитывать его сложность, и отсюда возникает вопрос декомпозиции системы на более мелкие подсистемы и использования библиотечных модулей с уже отлаженными простыми моделями. Большинство языков имитационного моделирования не имеют таких возможностей, что существенно усложняет построение модели, и заставляют

рассматривать сложную систему, как нечто целое, неделимое, не позволяя взглянуть на нее как на объект, состоящий из простых частей.

В рассматриваемой системе имитационного моделирования сделана попытка, решить эти и другие проблемы, возникающие перед экспериментатором при программировании модели сложной системы.

Математическая модель сложной дискретной системы строится на базовых понятиях теории массового обслуживания и некоторых дополнительных понятиях.

Основным понятием рассматриваемой сложной дискретной системы является понятие блока. Блок представляется в виде пятерки: множество атрибутов (параметров) блока, характеризующих его свойства; множество выходных связей, которые соединяют соседние блоки и являются маршрутами движения выходящих из блоков транзактов; множество событий, обрабатываемых блоком; множество обработчиков событий, элементами данного множества являются функции преобразования и обслуживания транзактов; отношение соответствия между элементами двух последних множеств.

Любое изменение состояния моделируемой системы связано с возникновением событий в системе, и дальнейшей обработки с помощью специальных функций. События делятся на три категории: события инициализации; события, привязанные к определенному времени срабатывания; условные события.

Управляющие алгоритмы рассматривают систему с точки зрения возникающих в системе событий, что говорит об использовании событийного подхода. При этом наличие условных событий, т.е. активных элементов, характеризуют систему моделирования, как ориентированную на сканирование активностей. Алгоритмы проверки условий срабатывания событий оптимизируются путем использования специальных индексных структур и массивов.

Описание модели производится в текстовом виде с помощью объектно-ориентированного языка. Он позволяет описывать модель, как в текстовом виде, так и в графическом, с помощью визуального представления сети массового обслуживания (см. рис. 1). Синхронизация представлений осуществляется автоматически.

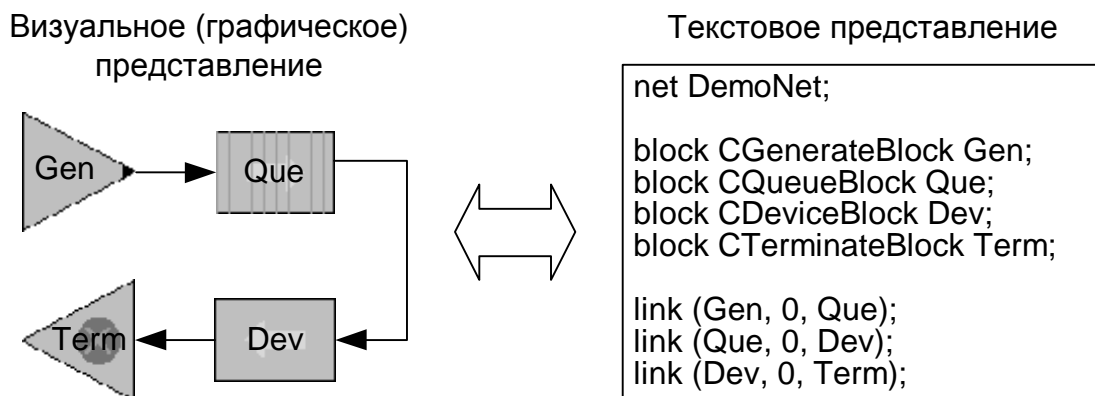


Рис. 1. Двойное представление модели сложной дискретной системы на языке ВИСМО 2

Язык позволяет организовать модульную структуру модели. Каждый из модулей может содержать описание произвольных элементов модели. Организация модульной структуры позволяет группировать часть элементов по какому-либо принципу и так же организовать реализацию библиотечных модулей, в которых будут описываться стандартные функции, классы и другие элементы, которые могут использоваться не только в текущем проекте, но и в дальнейших исследованиях.

## **К вопросу оценки эффективности алгоритмов**

**А.Ю. Гриднева, К.А. Майков**  
**Программное обеспечение ЭВМ и информационные технологии**  
**МГТУ им. Н.Э. Баумана**

Развивая изложенную в [2] концепцию векторной оценки устойчивости, сформулируем некоторые положения метода комплексной оценки эффективности вычислительных алгоритмов.

В основу данного метода может быть положена концепция многомерного пространства качества. Для ряда вычислительных алгоритмов, например для алгоритмов решения интегральных уравнений Фредгольма II рода, пространство качества может быть образовано критериями устойчивости, сходимости, сложности. При этом для некоторых показателей качества алгоритма (устойчивость и сходимость), можно рассматривать вектор качества, каждая компонента которого характеризует влияние определенной группы исходных данных или параметров алгоритма на его эффективность. Полученная совокупность векторов будет описывать некоторое пространство качества алгоритма, причем размерность этого пространства должна определяться не только количеством рассматриваемых характеристик алгоритма, но и особенностями прикладной задачи, для решения которой предполагается использовать исследуемый алгоритм.

Для вычисления значения отдельного показателя качества можно использовать несколько различных методов. Так в [2] предложено для оценки устойчивости использовать вероятностный метод [4], зональный метод и метод определения степени неустойчивости [3]. Можно предположить, что такой подход к анализу качества алгоритма позволит получить взаимодополняющие оценки эффективности в частично пересекающихся областях пространства параметров.

В построенном многомерном пространстве качества эффективность отдельного алгоритма будет рассматриваться как тело качества. При этом

отдельные характеристики, например устойчивость к ошибкам определенной группы исходных данных, получаются в результате сечения тела качества некоторой плоскостью. В этом же пространстве качества можно построить тело требуемой эффективности алгоритма, которое будет образовано системой требований к эффективности алгоритма, обусловленных спецификой решаемой прикладной задачи. Пересечение тела качества алгоритма и тела требуемой эффективности позволит оценить применимость исследуемого алгоритма для решения данной прикладной задачи. Проецируя пересечение этих тел на пространство исходных данных и параметров алгоритма, можно определить области, в которых данный алгоритм удовлетворяет требованиям прикладной задачи.

Рассмотрим основные сферы применения метода комплексной оценки эффективности алгоритма.

1. Определение оптимальных областей пространства исходных данных. В [1, 3] показано, что многие характеристики вычислительного алгоритма, такие как точность, устойчивость, зависят от конкретных исходных данных, то есть при различных данных различные алгоритмы могут иметь лучшие характеристики. При этом важное практическое значение имеет задача определения областей исходных данных, для которых исследуемый алгоритм является наилучшим среди множества рассматриваемых алгоритмов. Для решения этой задачи можно применять различные методы оптимизации, причем для построения целевой функции будет использоваться метод комплексной оценки эффективности алгоритма.

2. Построение комбинированного алгоритма [3]. Результаты, полученные при решении первой задачи, могут быть использованы при построении алгоритма оптимального во всей области параметров задачи. Примером такого алгоритма может служить комбинированный алгоритм [4], который совпадает с различными алгоритмами из числа альтернативных в зависимости от конкретных исходных данных и имеет во всей области параметров лучшие характеристики, чем любой из алгоритмов, из которых он составлен.

3. Оптимизация существующего алгоритма. Метод комплексной оценки эффективности позволяет выделить группу исходных данных, оказывающих наиболее сильное влияние на значение показателя качества. Используя этот подход при анализе эффективности блоков исследуемого алгоритма, можно для каждого из них вычислить значение показателя качества. Тогда появится способ определить, какие блоки алгоритма следует модифицировать при его оптимизации.

Таким образом, разработка программного и алгоритмического обеспечения метода комплексной оценки эффективности вычислительных алгоритмов позволит провести комплексное исследование алгоритма, определить зависимости его характеристик от исходных данных и параметров, выделить неэффективные блоки. Причем результаты такого исследования могут быть использованы при автоматизации процедуры построения оптимального алгоритма, что, безусловно, имеет важное практическое значение.

#### **Список литературы.**

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М. Физматлит, 2002.
2. Гриднева А.Ю. К вопросу количественной оценки устойчивости алгоритмов решения интегральных уравнений Фредгольма II рода. (в печати).
3. Евневич Е.Л. Автоматизация выбора и синтеза комбинированных алгоритмов. Диссертация. - СПб., 1996.
4. Тоценко В.Г., Александров А.В., Парамонов Н.Б. Корректность, устойчивость, точность программного обеспечения. - Киев, Наукова думка, 1990.

# Некоторые аспекты построения программно-алгоритмического комплекса для количественной оценки устойчивости решения линейного интегрального уравнения Фредгольма II рода

А.Ю. Гриднева, К.А. Майков  
Программное обеспечение ЭВМ и информационные технологии  
МГТУ им. Н.Э. Баумана

В основу разработанного программно-алгоритмического комплекса положена концепция векторной оценки [2] устойчивости алгоритмов решения интегральных уравнений Фредгольма II рода, предлагающая описывать устойчивость алгоритма трехкомпонентным вектором, определяющим чувствительность решения к ошибкам правой части, ядра и параметра уравнения. Рассмотрим особенности построенных алгоритмов экспериментальных методов [4] количественной оценки устойчивости, основанных на анализе зависимости погрешности результата от погрешности исходных данных, причем предполагается, что случайная величина  $\Delta x$ , определяющая отклонение исходных данных, равномерно распределена на отрезке  $[0, \Delta_{\max}]$ , где  $\Delta_{\max}$  - максимальное допустимое искажение.

Вероятностный метод [5] оценки устойчивости алгоритмов предлагает для оценки устойчивости использовать вероятность  $P_y$  того, что отклонение выходных величин находится в заранее заданных пределах. При построении алгоритма данного метода использовалось ограничение вида  $\Delta y \leq \Delta_{\text{дон}}$ , где  $\Delta y$  - отклонение выходных величин,  $\Delta_{\text{дон}}$  - величина допустимого отклонения результатов. В этом случае для определения  $P_y$  достаточно найти такое значение отклонения исходных данных  $\Delta_{\text{крит}}$ , начиная с которого отклонение выходной величины превышает  $\Delta_{\text{дон}}$ . Тогда показатель устойчивости алгоритма вычисляется по формуле

$$P_y = \frac{\Delta_{\text{крит}}}{\Delta_{\max}}. \quad (1)$$

Для нахождения  $\Delta_{крит}$  строится зависимость  $\Delta y = f(\Delta x)$  отклонения результата вычислений  $\Delta y$  от отклонения исходных данных  $\Delta x$ . Так как функция  $f(\Delta x)$  монотонно возрастает [1, 4], то задача отыскания  $\Delta_{крит}$  сводится к решению уравнения

$$f(\Delta x) - \Delta_{дон} = 0. \quad (2)$$

Развитием рассмотренного вероятностного метода оценки устойчивости является зональный метод [2], предлагающий разбить отрезок  $[0, \Delta_{max}]$  на четыре части, выделив зону абсолютной устойчивости, зону квазиустойчивости, зоны полиномиальной и экспоненциальной неустойчивости.

При разработке алгоритма зонального метода предполагалось, что зависимость погрешности результата от погрешности исходных данных имеет вид

$$f(x) = \sum_{k=0}^m a_k x^k. \quad (3)$$

В этом случае зона абсолютной устойчивости вычисляется по формуле

$$Z_{abc} = \frac{x^{abc}}{\Delta_{max}}, \quad (4)$$

где  $x^{abc}$  будет являться корнем уравнения  $f'(x) = \sum_{k=1}^m k a_k x^{k-1} = 1$ .

Аналогично определяется зона квазиустойчивости исследуемого алгоритма

$$Z_{ycm} = \frac{x^{ycm} - x^{abc}}{\Delta_{max}}, \quad (5)$$

где  $x^{ycm}$  - решение уравнения  $f'(x) = C$ ,  $C > 1$  - заранее заданная константа.

Так как в зоне полиномиальной неустойчивости погрешность результата растет не быстрее, чем полином заданной степени  $n$ ,  $n < m$ , то можно утверждать, что  $(n+1)$ -ая производная функции  $f(x)$  на отрезке  $[x^{ycm}, x^{нол}]$  отличается от нуля не более, чем на  $\epsilon$ , то есть

$$|f^{(n+1)}(x)| < \epsilon, \quad \forall x \in [x^{ycm}, x^{нол}], \quad (6)$$



где  $\epsilon$  - требуемая точность вычислений. Таким образом, найдя точку  $x^{пол}$ , для которой  $|f^{(n+1)}(x^{пол})| \geq \epsilon$ , вычислим зону полиномиальной неустойчивости

$$Z_{пол} = \frac{x^{пол} - x^{уст}}{\Delta_{max}}. \quad (7)$$

Для определения зоны экспоненциальной неустойчивости применяется следующее условие нормировки:

$$Z_{abc} + Z_{уст} + Z_{пол} + Z_{эксн} = 1. \quad (8)$$

Заметим, что при оценке устойчивости алгоритма могут использоваться не только величины отдельных зон устойчивости, но и их суммы. В этом случае в каждую зону устойчивости включаются все предшествующие ей зоны [2], и значение показателя устойчивости соответствует вероятности того, что при случайном искажении исходных данных погрешность результата будет изменяться по закону, определяемому типом исследуемой зоны устойчивости.

Альтернативный способ анализа эффективности алгоритма предлагает метод определения степени неустойчивости [3]. Для оценки устойчивости алгоритма используется специальный показатель качества – степень неустойчивости, характеризующий чувствительность исследуемого алгоритма к ошибкам исходных данных. При этом различают устойчивость алгоритма в заданной точке (локальная степень неустойчивости) и в области (глобальная степень неустойчивости) пространства исходных данных.

В реализованном алгоритме оценки эффективности для вычисления степени неустойчивости строится функция  $U_f(\Delta x)$ , описывающая локальную неустойчивость алгоритма. При этом для определения локальной неустойчивости алгоритма используется полученное в [3] соотношение

$$U_f(\Delta x) = \frac{\Delta f(\Delta x)}{\Delta x}, \quad (9)$$

где  $\Delta f$  - среднее отклонение результата, соответствующее  $\Delta x$  - заданному отклонению исходных данных. С использованием функции  $U_f(\Delta x)$  локальная

степень неустойчивости  $U_f^{лок}$  определяется как значение этой функции при  $\Delta x = \Delta x^{лок} = 0.01$ , если  $\Delta x^{лок} \in (0, \Delta_{max}]$ , или как  $U_f(\Delta_{max})$  в противном случае

$$U_f^{лок} = \begin{cases} U_f(\Delta x^{лок} = 0.01), & \text{если } \Delta x^{лок} \in (0, \Delta_{max}] \\ U_f(\Delta_{max}), & \text{если } \Delta x^{лок} \notin (0, \Delta_{max}] \end{cases} \quad (10)$$

В зависимости от особенностей решаемой задачи для оценки эффективности алгоритма в заданной области можно использовать среднюю или максимальную степень неустойчивости. Для определения глобальной средней и глобальной максимальной степеней неустойчивости достаточно найти соответствующее значение функции  $U_f(\Delta x)$  в интервале  $(0, \Delta_{max}]$

$$U_f^{сред} = \frac{\sum_{i=1}^N U_f(\Delta x^i)}{N}, \quad (11)$$

$$U_f^{max} = \max U_f(\Delta x_i), \quad i = \overline{1, N}, \quad (12)$$

где  $N$  - количество проведенных вычислительных экспериментов.

Описанные алгоритмы оценки эффективности реализованы в составе программно-алгоритмического комплекса для количественной оценки устойчивости метода Положего при решении интегральных уравнений Фредгольма II рода с вырожденным ядром. Проанализируем представленные в таблице 1 результаты вычислительного эксперимента, проведенного с использованием построенного комплекса, при следующих исходных данных:

- исследуемое уравнение

$$y(x) - \int_0^{2p} |p - s| \sin x y(s) ds = x; \quad (13)$$

- максимальное отклонение исходных данных – 10%;
- количество вычислений на каждом шаге – 10;
- используемая степень аппроксимации – 5;
- максимальное допустимое отклонение результата (для вероятностного метода) – 5%.

Таблица 1.

Компоненты вектора устойчивости	Метод оценки устойчивости			
	Вероятностный (5%)	Зональный		Степень неустойчивости (глобальная средняя)
		Зона абсолютной устойчивости	Суммарная зона квазиустойчивости	
	$V_1$	$V_2$	$V_3$	$V_4$
Ошибки правой части	0,967	0,926	1,000	0,546
Ошибки ядра уравнения	0,186	0,017	0,096	2,661
Ошибки параметра	0,924	0,980	1,000	0,626
Норма вектора	1,350	1,348	1,417	2,788
Нормированный показатель устойчивости	78,0%	77,9%	81,8%	(отсутствует)

Сопоставляя векторы устойчивости, полученные с помощью вероятностного и зонального методов, заметим, что для исследуемого уравнения зона абсолютной устойчивости соответствует вероятности того, что при случайном искажении исходных данных погрешность результата не превысит 5%. Метод определения степени неустойчивости позволяет выделить группу исходных данных, погрешность которых наиболее сильно влияет на погрешность решения уравнения. В рассматриваемом случае погрешность решения в основном определяется погрешностью ядра уравнения.

С целью проведения комплексного исследования эффективности алгоритма, полученные векторы  $V_1$ ,  $V_2$  и  $V_3$  можно объединить в матрицу устойчивости, элементы которой характеризуют устойчивость решения интегрального уравнения к ошибкам исходных данных, оцененную с использованием реализованных алгоритмов. Добавление вектора  $V_4$  в рассматриваемую матрицу возможно только после модификации метода определения степени неустойчивости, так как показатель качества, используемый этим методом, является ненормированным и определяет не устойчивость решения, а его чувствительность к ошибкам исходных данных.

Таким образом, использование нескольких взаимодополняющих методов анализа устойчивости при реализации концепции векторной оценки

эффективности алгоритмов делает возможным проведение комплексного исследования устойчивости алгоритма. Полученная в результате исследования совокупность векторов устойчивости позволяет определить влияние погрешности определенной группы исходных данных на погрешность решения интегрального уравнения. Эти результаты могут быть использованы для поиска неустойчивых блоков алгоритма решения, что имеет важное практическое значение при оптимизации алгоритма.

### **Список литературы.**

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Физматлит, 2002.
2. Гриднева А.Ю. К вопросу количественной оценки устойчивости алгоритмов решения интегральных уравнений Фредгольма II рода. (в печати).
3. Евневич Е.Л. Автоматизация выбора и синтеза комбинированных алгоритмов. Диссертация. – СПб., 1996.
4. Иванов В.В. Методы вычислений на ЭВМ. – Киев: Наукова думка, 1986.
5. Тоценко В.Г., Александров А.В., Парамонов Н.Б. Корректность, устойчивость, точность программного обеспечения. – Киев: Наукова думка, 1990.

## **Иерархическое исследование дискретных систем с очередями**

**И.В. Рудаков**

(г. Москва, Московский государственный технический университет им. Н.Э. Баумана)

При анализе и проектировании структур сложных систем с очередями используется блочно-иерархический метод, который предусматривает расчленение процесса проектирования на ряд последовательных уровней и сведения задачи большей размерности к совокупности задач значительно меньшей размерности.

Одной из задач при разработке структур сложных систем является задача анализа и контроля правильности функционирования системы на ранних этапах ее разработки, в результате чего определяется состав функциональных блоков и их взаимодействие. На последующих этапах осуществляется анализ отдельных блоков. Предлагаемый подход позволяет осуществить анализ функционирования сложной структуры с использованием методов макро моделирования и декомпозиции на базе языка моделирования ориентированного на транзакты.

Для реализации макро моделирования в качестве элементов моделирования предлагается использовать макро модели функциональных блоков сложной системы на основе вероятностных автоматов и систем массового обслуживания.

Метод анализа сложных структур только на базе макро моделей функциональных блоков недостаточен, так как не позволяет учитывать такие характеристики системы как, например, временные параметры элементов, входящих в макро модель, а, следовательно, выполнять надежную верификацию проекта. С целью уточнения характеристик сложной системы используется принцип “увеличительного стекла”, когда различные участки структуры системы анализируются с разной степенью детализации. В отличие от традиционного подхода, при котором переход с одного иерархического уровня на другой выполняется разработчиком системы, предлагаемый метод декомпозиции, реализованный в виде программной подсистемы, позволяет

выполнить автоматический переход от макромоделей функционального блока к моделям элементов структуры при выполнении смешанного моделирования сложных систем.

При формализации процесса функционирования структуры сложной системы как иерархической, она должна отвечать следующим требованиям:

- последовательное вертикальное расположение моделей, составляющих данную систему;
- приоритетное воздействие моделей верхнего уровня;
- зависимость функционирования моделей верхнего уровня от функционирования моделей нижнего уровня.

При описании макромоделей функционального блока как иерархической системы необходимо учитывать:

- множество входных и выходных воздействий, поступающих на блок в соответствии с алгоритмом функционирования  $PS(P,T)$  и множество выходных воздействий, вызываемых блоком после реализации алгоритма  $QS(P,T)$  с учетом вероятностного ( $P$ ) и временного характера параметров блока;
- множество связей между макромоделью функционального блока и моделями элементов  $VS(P,T)$  с учетом множества обратных связей  $WS(P,T)$ ;
- множество состояний макромоделей функционального блока  $\Phi(P,T)$ .

Тогда функционирование макромоделей функционального блока можно описать следующим выражением:

$$PS(P,T) \times WS(P,T) \times \Phi(P,T) \rightarrow QS(P,T)$$

- Практическая реализация иерархического исследования структуры сложных систем с очередями опирается на технологию объектно-ориентированного программирования (ОПП). В качестве инструментального средства анализа и верификации структуры разработан язык моделирования на базе Visual C++, ориентированной на транзакты .

Программное обеспечение иерархического проектирования структур с очередями функционирует в рамках ОС Windows95 и выше.

## **Применение шаблонов проектирования при построении информационных систем предприятий машиностроения**

**И.В. Илларионов**  
Воронежский государственный университет,  
кафедра Программирования и информационных технологий, [illar@cs.vsu.ru](mailto:illar@cs.vsu.ru)  
**В.Н. Старов**  
Воронежский государственный технический университет,  
кафедра Автоматизации оборудования  
**В. А. Муранов**  
Воронежский государственный технический университет,  
кафедра Автоматизации оборудования, [tertium@mailru.com](mailto:tertium@mailru.com)

Новая редакция международных стандартов в области качества ISO 9000 2000 года предполагает использование процессоориентированного подхода, как идеологической основы построения систем управления качеством продукции. Различные методики такого подхода на настоящий момент известны и активно используются как элемент информационных технологий, например, при проектировании программных средств и информационных систем управления предприятиями. Каждая из них включает в себя в той или иной степени стандартизованную графическую нотацию, реализующие ее программные средства и методические материалы по использованию. Особо отметим важность графической нотации, которая наряду с предоставлением исходной информации для программных средств, обеспечивающих генерацию кодов, выполняет методически не менее важную задачу – служит своеобразным «интерфейсом» между специалистом в предметной области (менеджером, технологом, финансистом) и специалистами в области информационных технологий. Именно на этапе их взаимодействия во многом закладывается степень адекватности, с которой проектируемая система сможет описывать функционирование предприятия или организации и, в конечном итоге, качество программного продукта.

Предприятия материального производства или сферы услуг, рассматриваемые во всем многообразии их внешних связей и взаимодействий составляющих их частей, очевидно, требуется рассматривать как сложную

систему /1,2/, для описания различных сторон функционирования которых целесообразно использование различных способов. Например, известная группа методик IDEF /3/ позволяет описать функциональную структуру (IDEF0), информационную структуру (IDEF1X), динамику функционирования (IDEF3) объекта. При использовании каждой из них приходится учитывать возможности соответствующих программных средств. Так, если для IDEF0 и IDEF1X существует значительное количество активно используемых программных пакетов, то для описания динамического поведения вместо IDEF3 гораздо удобнее и экономически целесообразнее использовать иные нотации (UML), лучше поддерживаемые программными средствами.

На начальном этапе проектирования требуется определить системы предприятия и входящие в них процессы и объекты с целью последующего управления ими. Такое описание начинается с наиболее общих задач с последующим уточнением до элементарных составляющих.

В ГОСТ Р ИСО 9001:2001 к обязательным бизнес-процессам относятся:

- реализация ответственности высшего руководства в рамках системы качества;
- менеджмент ресурсов (кадры, инфраструктура, рабочая среда);
- менеджмент производственных процессов (процессов жизненного цикла продукции приравненных к ним по степени важности вспомогательных процессов обеспечения);
- процессы измерения, контроля и улучшения системы качества.

Кроме того, как обязательные элементы процессов, составляющих бизнес-процесс, рассматриваются:

- документы, содержащие политику, цели организации в сфере менеджмента качества, руководство по качеству;
- документированные процедуры, в том числе документы, содержащие ответственность сотрудников организации;
- документация на процессы, необходимая для обеспечения их эффективного планирования, управления и улучшения;
- записи качества, и т.д.



Соответственно, функциональная модель должна содержать все обязательные процессы и элементы в соответствии с требованиями ГОСТ Р ИСО 9000:2001.

В пользу применения методологии IDEF0 для описания и классификации процессов говорит то, что она является официальным стандартом функционального моделирования в ряде стран, включая США и Россию. Последнее обстоятельство делает возможным использовать методологию IDEF0 в качестве единого языка для обмена информацией между организациями.

Для промышленного предприятия действующего в современной конкурентной среде наряду с выполнением задачи построения его информационной модели не менее важной является способность построенной модели обеспечивать эффективную поддержку задачи постоянной реструктуризации производства (в частности технологических процессов материального производства и соответствующих им информационных технологий). Наиболее эффективно эти задачи решаются при поддержке программных средств. Так, развитие продукции, направленное на повышение её характеристик, эффективно поддерживается CAD/CAM/CAE – системами, обеспечивающими конструирование, проектные расчеты, кинематическое моделирование и др.

Цена продукции определяется, в свою первую очередь, её себестоимостью, на которую оказывает влияние значительное количество факторов. Оптимизация здесь может быть осуществлена при поддержке систем управления предприятием - ERP – систем, обеспечивающих оптимальное планирование, повышение загрузки оборудования, сокращение складских запасов, сокращение длительности производственных циклов, и т.д.

С внедрением в практику стандартов CALS место идеологической основы проектирования информационных систем прочно занял объектный подход, для которого, в свою очередь, стандартом проектирования стала нотация языка UML /4,5/. Далее рассматривается формальное описание процессов

машиностроительного производства с учетом динамики поведения и взаимодействия отдельных объектов.

Наибольший эффект от внедрения автоматизированного оборудования может наблюдаться лишь при полной интеграции всех систем производства. Дело в том, что в контексте крупного предприятия очень трудно заставить работать все системы слаженно и синхронно, обеспечить выполнение и взаимодействие множества параллельных процессов в реальном времени. Много предприятий, вышедших на рынок, показали полную свою несостоятельность и неспособность конкурировать с вновь созданными фирмами. Однако, повышение их эффективности всё же возможно. Одно из направлений – построение единой информационной системы предприятия.

С развитием компьютерных технологий можно было наблюдать революцию в интеграции различных этапов проектирования и производства изделий; пришли новые технологии, таких производителей CAE/CAD/CAM-систем как Unigraphics Solutions, Dassault Systemes, Autodesk, Structural Dynamics Research Corporation и других. Но у отечественной промышленности зачастую просто не хватает средств на покупку этого, вне всякого сомнения, мощного современного программного обеспечения и его поддержку.

Итак, перед отечественной промышленностью стоит задача построения информационной системы, интегрирующей в своих рамках различные подсистемы производства. В данной работе предлагается подход к построению такой системы, основанный на шаблонах проектирования.

На предприятиях работают специалисты предметной области. Для осуществления проекта построения информационной системы необходим контроль над его реализацией со стороны экспертов в предметной области, при этом надо чётко разделить труд программиста и эксперта. К сожалению, такая практика у нас используется не всегда.

Разработка крупномасштабного программного обеспечения – достаточно сложная задача, требующая использования CASE-технологий (Computer-Aided Software Engineering – автоматизированная разработка программного

обеспечения) и UML (Universal Modeling Language – универсальный язык моделирования). Перечисленные средства позволяют жестко формализовать процесс создания ПО и разбить его на приемлемые по размеру части – для параллельной разработки несколькими группами /6,7/ .

При выполнении работы авторы исходили из следующих задач:

1) Изучение возможностей CASE-технологий и UML применительно к разработке программного обеспечения для материального производства.

2) Построение общей концептуальной модели техпроцесса (на примере сверления) как потомка абстрактного техпроцесса и словаря начального уровня информационной системы машиностроительного производства с использованием UML и шаблонов проектирования.

3) Подetailная проработка системы управления предприятием на уровне шаблонов.

4) Проектирование:

- услуг, предоставляемых системой, на диаграммах прецедентов или случаев использования (use case diagrams);

- их реализаций в виде ассоциаций классов на диаграммах коопераций (collaboration diagrams);

- классов на диаграммах классов (class diagrams);

- процессов и объектов во временных рамках на диаграммах последовательностей и состояний (sequence diagrams, state diagrams);

- физического развёртывания системы в пространстве на диаграммах развёртывания (deployment diagrams) для лучшего понимания масштаба системы и физической связи её отдельных звеньев.

Чтобы потребителем данной системы могло стать любое предприятие машиностроения, проектирование её ведётся с использованием шаблонов.

5) Проектирование целевой части проекта (САМ-системы) до самого логически низкого уровня. Этим демонстрируется реальный пример разработки ПО с использованием CASE-технологии.

б) Разработка с использованием технологии CASE универсального САМ – программного обеспечения (АСТПП) для упрощения обслуживания и повышения эффективности работы оборудования с ЧПУ в рамках информационной системы предприятия. Разрабатываемая САМ-система – среда визуального проектирования формализованных техпроцессов сверления для операционной системы Windows с интуитивно понятным графическим интерфейсом, требующим минимум времени для освоения. САМ-система имеет клиентскую и серверную составляющие: собственно среда визуального проектирования с локальной базой данных (БД) и ПО оснастки сервера БД техпроцессов. Так реализуется принцип сотрудничества разработчиков техпроцессов: доступ к серверу базы данных возможен с любого компьютера, подключённого к корпоративной сети (рассматриваются сети, работающие по протоколу TCP/IP), на котором имеется клиентская часть САМ-системы. Права на удаление техпроцессов и управление базой данных пользователей и паролей для простых разработчиков отсутствуют. Для этих возможностей реализован вход для администратора через веб-интерфейс с любого компьютера, с которого опять же есть доступ к корпоративной сети, и установлен любой обозреватель сети (Internet Explorer, Netscape, Opera). Таким образом, администрирование главной БД техпроцессов может производиться с компьютера с любой операционной системой. Так частично реализован принцип кроссплатформенности. Для полной его реализации следует переписать клиентскую часть САМ-системы на Java, что, к сожалению, приведёт к снижению производительности и появлению ограничений, накладываемых на графический интерфейс, поэтому в рамках этого проекта мы не используем Java. Для конкретизации клиентская часть САМ-системы ориентирована на создание техпроцессов сверления.

Написание САМ-системы не имело целью создать систему прямого программирования оборудования с ЧПУ. На выходе системы предполагалось получать именно формализованный техпроцесс, который на следующем этапе должен обрабатываться парсером (синтаксическим анализатором), написанным

для конкретного станка. Так реализован принцип универсальности (аналогия с виртуальной Java машиной).

Проектирование велось в три этапа, результатом чего стали диаграммы верхнего, нижнего и среднего уровня.

Самый верхний уровень абстракции представлен девятью диаграммами прецедентов, с разных позиций и с разным уровнем детализации отражающие процессы, происходящие в АСУП.

- Диаграмма общей системы. На ней присутствуют актёры, представляющие людей или системы, внешних относительно АСУП, и прецеденты, представляющие в рамках АСУП те услуги, которые она предлагает потребителям. Обобщённо говоря, прецеденты показывают процессы, происходящие в системе, а актёры – их клиентов.
- Диаграмма системы поддержки жизненного цикла продукта. Как говорилось выше, предприятие, применительно к которому рассматривается разработка АСУП, работает в рамках стандартов CALS или ISO9000:2000, то есть обслуживает весь жизненный цикл своих продуктов – маркетинговое исследование, проектирование, разработка, испытание, техническая поддержка
- Диаграмма системы управления кадрами. АСУП в частности обеспечивает автоматизированную работу с кадрами, включая набор кадров через биржи труда, их обучение и т.д.
- Диаграмма системы работы с репозиторием (базой знаний). Репозиторий – это база знаний, в которой хранятся все наработки данного предприятия и других, схожих по профилю, а также нормативная документация (ГОСТ, ОСТ и пр.), каталоги моделей стандартизированных деталей и материалов и подобная информация. Репозиторий не однороден по своей природе и удачная организация его может служить залогом успешной работы в дальнейшем. Нами избрана объектно-релятивистская модель базы данных для организации базы знаний. Как сказано выше, такие базы данных основаны на следующих принципах:

Каждая сущность, информация о которой хранится в БД, — это объект.

1. Каждый объект уникален в пределах БД и имеет уникальный идентификатор.

2. Объект имеет свойства (строковые, числовые, временные, перечислимые), которые описывают атрибуты сущности.

3. Объекты могут быть связаны между собой произвольным образом. Связь характеризуется связанными объектами и типом связи. Например, сотрудник фирмы может быть связан с отделом, в котором он работает, связью типа «сотрудник в отделе» и т.п. Связь в определенном смысле аналогична понятию ссылки на таблицу-справочник в традиционной модели БД.

4. Объект может быть хранилищем. В этом случае допускается хранение в нем других объектов (например, товара на складе).

5. Такая БД не привязана к конкретной бизнес-модели и позволяет реализовать «над собой» практически любую бизнес-логику. Логика выделяется в отдельный программный слой и, как правило, реализуется на сервере приложений, где по запросу клиента создаются объекты, загружающие информацию о себе из БД и реализующие «поведение» объектов реального мира. В то же время, в силу однообразности модели хранения, эти объекты довольно легко создаются на основе базовых классов, инкапсулирующих функциональность по загрузке и сохранению свойств и связей в БД.

Таким образом, для организации репозитория требуется пополняемая библиотека классов. Задача создания и поддержки такой библиотеки, а также введения отдельного программного слоя бизнес-логики достаточно трудоёмка и с точки зрения программирования, и с точки зрения проектирования, тем не менее, усилия, затраченные на это, в дальнейшем принесут немало пользы.

- Диаграмма системы управления ресурсами. Система управления ресурсами является неотъемлемой частью АСУП. Она включает в себя системы прогнозирования ресурсоёмкости, снабжения ресурсами и энергией и т.д.
- Диаграмма процессов, связанных с оборудованием.

- Диаграмма системы контроля. Конечной целью система контроля является обеспечение высокого и стабильного уровня качества продукции и повышение эффективности производства за счёт слаженной работы всех его подсистем.
- Диаграмма процессов, связанных с реализацией продукта и его поддержкой. Рассматриваемое предприятие является звеном в сети аналогичных и, одновременно, самодостаточным образованием. Следовательно, у него должна быть развитая система реализации продукции и поддержки продукции.
- Диаграмма подготовки производства. Наиболее важная для настоящего проекта диаграмма высокого уровня – это диаграмма прецедентов системы подготовки производства. Это последняя диаграмма уровня прецедентов. Она состоит собственно из диаграммы АСТПП, диаграммы базы знаний (репозитория) и актёров.

Для диаграмм среднего уровня выделяются

- Диаграмма концептуальной модели техпроцесса - служит для иллюстрации концептуальных связей реальных понятий предметной области. На диаграмме концептуальной модели присутствуют два основных понятия: «Техпроцесс» и производное от него «Техпроцесс сверления». В верхней ветви диаграмма содержит отношение абстрактного техпроцесса и его составляющих частей – этапов техпроцесса, состоящих из «Операций»; «Операции» – из «Позиций», «Позиции» – из «Технологических переходов». Для описания конкретного техпроцесса (сверление) его структура несколько упрощена - «Техпроцесс сверления» имеет в своём составе три основных операции: «Позиционирование», «Смена инструмента» и «Режим резания». Также он соотносится с понятиями «Станок» и «Инструмент» для получения информации об имеющемся оборудовании для подбора верных параметров операций (здесь техпроцесс принимается как некая сущность, способная к самоформированию).

- Диаграммы состояний рабочего и оборудования - иллюстрируют поведение объекта в системе. Согласно методологии объектно-ориентированного программирования, всякий объект на протяжении времени его жизни находится в различных состояниях. Простейший пример – окно Windows может находиться в состоянии «активно» или «неактивно». Так и рабочий в контексте системы имеет ряд состояний.
- Диаграмма последовательностей «Заготовка-техпроцесс-рабочий» - служит для демонстрации взаимодействий объектов во времени. Взаимодействие осуществляется с помощью событий, которые могут быть как синхронными, так и асинхронными. При проектировании данной диаграммы мы ограничились использованием синхронных процедурных событий, суть которых состоит в вызове методов одного объекта другим. В диаграмме использована также разновидность синхронных событий, при посылке которых объект сам запускает свой метод.
- Диаграмма прецедентов САМ-системы.  
 Диаграммы нижнего уровня -
  - Диаграмма физических данных БД техпроцессов
  - Диаграмма классов САМ-системы
  - Диаграмма развёртывания САМ-системы
  - Диаграмма компонентов системы визуального проектирования техпроцессов

Они строятся непосредственно перед разработкой программной системы, описывают все необходимые аспекты разработки и служат для генерации скриптов базы данных, а также «каркаса» программы – заголовочных и исходных программных файлов (\*.h и \*.cpp), в которых прописаны все заголовки и описания классов и проставлены все связи. Скрипты серверного приложения администрирования базы данных достаточно просты в написании, поэтому на этапе создания диаграмм нижнего уровня упоминаются лишь в диаграмме развёртывания системы.



- Диаграмма физических данных БД техпроцессов - моделирует структуру базы данных техпроцессов. Использование нотации UML позволяет на основе этой диаграммы сгенерировать специальную подпрограмму, так называемый «скрипт базы данных», которая позволяет создать базу данных с определёнными нами атрибутами. При разработке сложных баз данных очень важно, чтобы и программисты, и специалисты по базам данных работали в рамках одной нотации. Поэтому и в рамках настоящего проекта очень полезно будет использовать UML, чтобы проиллюстрировать многогранность этой нотации.
- Диаграмма классов САМ-системы - описывает классы, используемые в системе. Ограничимся двумя главными классами: класс техпроцесса (CTechProcess) и класс элемента техпроцесса (CTPItem). Класс техпроцесса является листовым, то есть на данном этапе не имеет потомков, а класс элемента техпроцесса – абстрактным базовым, то есть экземпляр (объект) этого класса создать невозможно, но есть классы-потомки этого класса, создание экземпляров (объектов) которых допускается.
- Диаграмма развёртывания САМ-системы. Диаграммы развёртывания очень полезны при проектировании систем с уровнем сложности выше малого. Они позволяют спланировать, как будут размещены в пространстве элементы проектируемой системы, на каких устройствах будут установлены, как будут связаны между собой (вплоть до физических параметров коммуникаций). Строго говоря, для эффективного функционирования проектируемого элемента САМ-системы необходимо пять процессоров. Процессорами в методологии UML называются любые устройства, имеющие некоторую вычислительную мощность. В данном случае это четыре компьютера и один станок с ЧПУ. Файлы, относящиеся к модулю

администрирования базы данных, помещаются на любой сервер сети, имеющий доступ к серверу базы данных и открытый для доступа с рабочих станций сети. Фактически производительность у скрипт-языков на несколько порядков ниже производительности приложений на C++, однако, в отличие от них, скрипты платформеннонезависимы, то есть работают в любой операционной системе. К тому же доступ к базе данных занимает какое-то время, что при определённых условиях почти сравнивает по производительности приложения на C++ и на PHP, так что использование скрипт-языка вполне допустимо в проектируемой нами системе. Серверу HTTP необходимо иметь двухстороннюю цифровую связь с сервером БД по протоколу TCP/IP.

- Диаграмма компонентов системы визуального проектирования техпроцессов. Диаграмма компонентов позволяет правильно расставить связи между компонентами разрабатываемой системы. Компоненты – это файлы, из которых состоит проект. На диаграммах компонентов часто изображают динамически подгружаемые библиотеки (\*.dll), дополнительные компоненты Windows (\*.ocx) и исполнимые файлы (\*.exe), то есть те элементы программных систем, которые имеются в наличии только после компиляции их исходных кодов.

Разработка системы визуального проектирования техпроцессов велась с использованием методологии RAD, то есть присутствовало несколько циклов разработки с распределением задач между группами разработчиков. Это позволяет снизить фактический уровень сложности разработки и повысить оперативность обмена информацией с предметной областью. На каждом этапе разработки ведётся планирование следующего этапа, а также исследования в предметной и программной области с целью оптимизации работы системы и максимального удовлетворения требований предметной области.

Разработанная программная система состоит из двух частей: система визуального проектирования техпроцессов сверления и модуль администрирования базы данных. Система визуального проектирования позволяет создавать техпроцессы сверления, состоящие из операций трёх типов, у которых можно задавать параметры. На выходе системы – формализованный технологический процесс, который может быть использован для генерации технологической, экономической и пр. документации. Программа поддерживает сохранение формализованных техпроцессов в базу и на диск и открытие их из базы и с диска. Модуль администрирования – позволяет просматривать базу и удалять техпроцессы из неё.

Всё это обеспечение включено в инсталляционный комплект, который также производит установку на компьютер HTTP-сервера Apache, языка PHP /8/ , СУБД MySQL /9/ и драйвера MyODBC последних (на момент завершения проекта) версий.

### **Список литературы**

- 1 Зиндер Е.З. Бизнес-реинжиниринг и технологии системного проектирования. Учебное пособие. - М., Центр Информационных технологий, 1996
- 2 Международные стандарты, поддерживающие жизненный цикл программных средств. - М., МП «Экономика», 1996
- 3 Дэвид А. Марка. Методология структурного анализа и проектирования. /Дэвид А. Марка, Клемент МакГоуэн/ - Электронная версия.
- 4 Крег Ларман. Применение UML и шаблонов проектирования. – СПб., Питер-пресс. 2001
- 5 Inside the Unified Modeling Language. Rational Corp., © 2000
- 6 Илларионов И.В. «Проектирование иерархической структуры классов для систем механообработки при проектировании системы управления качеством продукции» / Новые информационные технологии: материалы

четвёртого научно-практического семинара. /Муранов В.А., Илларионов И.В./ - Моск. гос. ин-т электроники и математики. М., 2000. стр. 220-230

7 Илларионов И.В. «Информационное моделирование процессов механической обработки» / Новые информационные технологии: материалы пятого научно-практического семинара. /Муранов В.А., Илларионов И.В., Старов В.Н./ - Моск. гос. ин-т электроники и математики. М., 2001. стр. 180-191

8 Хьюгс Стерлинг. PHP: руководство разработчика. / Хьюгс Стерлинг, Андрей Змиевски/ - К.: «ДиаСофт». 2001ю – 384 с.

9 David Axmark. MySQL reference manual. David Axmark, Michael (Monty) Widenius, Paul DuBois, Kim Aldale

## **Кодировщик математических моделей надежности ЭРИ подсистемы АСОНИКА-К**

**А. Ю. Фридер**

**Кафедра Радиоэлектронных и телекоммуникационных устройств и систем  
Московский государственный институт электроники и математики**

**E-mail: [Asonika-k@mail.ru](mailto:Asonika-k@mail.ru), [http: www.asonika-k.ru](http://www.asonika-k.ru)**

Осуществляющийся в настоящее время переход от радиоэлектронной аппаратуры (РЭА) пятого поколения к РЭА шестого поколения невозможен без создания новых классов электрорадиоизделий (ЭРИ). Это приводит к постоянным изменениям научно-технической документации (НТД), и, в частности, справочника «Надежность ЭРИ», который является основным документом для расчета надежности.

Несмотря на то, что в состав интерфейса администратора подсистемы АСОНИКА-К [1] входит модуль кодирования формул предназначенный для создания кода формул по ее обычному математическому представлению и занесении этого кода в СЧБД, при модификации справочной части базы данных (СЧБД) на основе данных справочника «Надежность ЭРИ» (ред. 2002 г.) был выявлен ряд его недостатков, в частности, любая ошибка при вводе любого компонента формулы требовала повторение ввода с самого начала.

Поэтому была поставлена задача доработать кодировщик математических моделей для подсистемы АСОНИКА-К. Требовалось улучшить интерфейс программы и облегчить ее взаимодействие с пользователем.

В результате был значительно упрощен процесс ввода формул пользователем за счет использования ввода формул с клавиатуры, введена удобная система оповещения пользователя об ошибках в введенной формуле. Для более наглядного представления структуры математических моделей в программу введено их представление в виде древовидной структуры (рис. 1), т. е. каждая формула представлена в виде дерева, которое включает в себя используемые в ней константы и другие формулы. В окне программы отображается древовидная структура классов, групп и подгрупп ИЭТ. (рис. 2)

Кодировщик позволяет описывать формулы, используя предоставленные в базе константы и формулы, а также числовые константы и стандартные математические операции и функции. Пользователь может вводить формулы с клавиатуры, используя стиль их описания, распространенный в большинстве используемых языков программирования и математических программ. В случае если при вводе формулы допущена ошибка, то при попытке ввести формулу в базу данных будет выведен список допущенных ошибок. Правильно введенные формулы сохраняются в базе данных в виде кода.

### **Список литературы**

1. Жаднов В.В., Жаднов И.В., Измайлов А.С., Сотников В.В., Марченков К.В. Подсистема АСОНИКА-К – расчет надежности аппаратуры и ЭРИ // EDA Express: Научно-технический журнал. № 5 2002. – с. 17-20.

2. Жаднов В.В., Измайлов А.С. Реализация математического ядра подсистемы «АСОНИКА-К» // Сборник научных трудов «Интернет и автоматизация проектирования» / Под ред. Тумковского С.Р. - М.: МГИЭМ, 2001. - с. 134-140.

## Метод анализа результатов расчета надежности РЭА и ЭРИ

Н.В. Смирнов

Кафедра «Радиотелекоммуникационных устройств и систем»  
Московский Государственный институт математики и электроники  
E-mail: [asonika-k@mail.ru](mailto:asonika-k@mail.ru)

Расчет надежности РЭА является необходимой проектной процедурой, предусмотренной в ПОН (Программа Обеспечение Надежности). ПОН регламентирует порядок проведения расчетов надежности на всех этапах процесса проектирования РЭА. На основе результатов расчетов принимаются решения о проведении тех или иных мероприятий, направленных на обеспечение требуемого уровня надежности.

Как правило, это происходит следующим образом: сравнивается требуемое и расчетное значения показателей надежности, и, если они существенно различаются, то предлагается некоторый перечень направлений по повышению надежности.

В качестве примера таких направлений можно привести следующие:

1. Снижение эксплуатационной интенсивности отказов ЭРИ. Это может быть достигнуто за счет:
  - применения ЭРИ с более низким значением базовой интенсивности отказов. Для этого необходимо рассмотреть следующие возможности:
    - а. Изменения типономиналов ЭРИ;
    - б. Применения зарубежных ЭРИ с более высоким уровнем качества на основе данных, приведенных в *MIL-HDBK-217F*;
    - в. Введения отбора для отечественных ЭРИ по критерию на надежности, например с использованием методов, разработанных в ЦНИИ ЦИКЛОН.
  - снижение величин рабочих температур ЭРИ, коэффициентов их электрической нагрузки и уровней внешних воздействующих факторов. Для этого необходимо рассмотреть следующие возможности:

- а. Применения системы охлаждения, изменения компоновочных схем изделия и его модулей, применения дополнительных мер для охлаждения ЭРИ и т.д.;
  - б. Изменения схем электрических принципиальных изделия с целью снижения величин рабочих токов, напряжений и мощностей на ЭРИ, а так же изменения типоминалов ЭРИ (применение ЭРИ, имеющих более высокие предельно-допустимые электрические параметры);
  - в. Применения дополнительных средств защиты от ВВФ(внешние воздействующие факторы) изделия в целом, модулей и ЭРИ с целью изменения класса аппаратуры по классификации ГОСТ РВ 20.39.304-98 (с использованием данных «Справочника НАДЕЖНОСТЬ ЭРИ» и *MIL-HDBK-217F*).
2. Рассмотреть возможность снижения количества комплектующих ЭРИ, например, за счет применения заказных БИС и СБИС.
  3. Рассмотреть возможность введения резервирования изделия в целом, его отдельных модулей и ЭРИ.
  4. Рассмотреть возможность изменения требований по надежности изделия в сторону их снижения.

При этом выбор конкретных направлений и оценка их эффективности перекладывается на плечи проектировщика РЭА. К сожалению, основным критерием выбора является минимум изменений, вносимых в схему и конструкцию. Повсеместная распространенность такого критерия обусловлена, с одной стороны, естественным желанием снизить объем изменений, вносимых в проект, а, с другой стороны, отсутствием достоверной информации об эффективности мероприятий. Получение таких оценок является весьма непростой задачей, а с точки зрения трудоемкости в несколько раз превышает трудоемкость самого расчета надежности.



Так, простое изменение базовой интенсивности отказов ЭРИ, может повлечь за собой лавину изменений:

- изменение типа ЭРИ (изменение его геометрических размеров);
- изменение конструкции РЭА (изменение компоновки, размещения, трассировки);
- изменение тепловых и механических характеристик конструкции РЭА и режимов работы ЭРИ;
- проведение расчета надежности РЭА.

При этом может оказаться, что все эти изменения приведут не к повышению, а даже к снижению общей надежности РЭА!

К сожалению, при создании программных средств (ПС) расчета надежности РЭА как отечественных (например, САПР АСРН), так и зарубежных (например, подсистема *RELIABILITY* системы *CADENCE*) проблеме анализа результатов расчетов не уделено достаточно внимания. Поэтому решение задачи обеспечения надежности РЭА с применением таких систем может быть получено только путем проведения многократных повторных расчетов.

Если принять во внимание сжатые сроки проектирования современной РЭА поколения и большие временные затраты на проведение расчета надежности (время расчета определяется количеством комплектующих ЭРИ), то применение этих систем для обеспечения надежности просто неэффективно.

Поэтому для новой версии подсистемы АСОНИКА-К был разработан метод автоматизированного анализа результатов расчетов надежности РЭА[51, 63, 74, 78, 102]. Метод позволяет проводить анализ результатов расчетов надежности РЭА, имеющей до 4-х уровней разукрупнения и различные виды резервирования. В результате анализа надежности компонентов всех уровней пользователем может быть получена количественная информация следующего характера:

- величины показателей надежности как РЭА в целом, так и его компонентов (в т.ч. и ЭРИ);

- виды и параметры резервирования как РЭА в целом, так и отдельных ее компонентов (в т.ч. и ЭРИ).

Кроме того, для ЭРИ определяются

- максимально-допустимые значения рабочей температуры и коэффициента электрической нагрузки или рабочих токов, напряжений, мощностей, нормированных на их предельные значения;
- вид приемки для отечественных ЭРИ и уровень качества для зарубежных ЭРИ;
- максимально-допустимые значения внешних воздействующих факторов (климатических и механических).

Использование "Системы анализа результатов" позволяет пользователю найти типономиналы ЭРИ, требуемые значения электрических и тепловых режимов работы ЭРИ, значение внешних воздействующих факторов (ВВФ) и тип и параметры резервирования, при которых обеспечиваются требование ТЗ по надежности.

Таким образом, полученные в результате анализа количественные данные позволяют наполнить конкретным содержанием рекомендации общего характера (например, вместо рекомендации «Снизить рабочую температуру резистора  $R1$ » дать рекомендацию «Обеспечить рабочую температуру резистора  $R1$  не выше  $40\text{ }^{\circ}\text{C}$ »).

Что же касается выбора конкретных схемных, конструкторских и технологических решений, позволяющих практически реализовать рекомендации, которые получены в результате анализа, то решение и этой задачи вполне возможно.

Однако для этого требуется применение экспертных систем, основанных на использовании аппарата нечетких множеств. Очевидно, что задача эта является не самой простой и ее решение является одним из перспективных направлений дальнейшего развития эксплуатационных возможностей подсистемы АСОНИКА-К.

## Список литературы

1. Сотников В.В. Справочная система подсистемы АСОНИКА-К // Научно-техническая конференция студентов, аспирантов и молодых специалистов, посвященная 40-летию МИЭМ. Тезисы докладов – М.: МИЭМ, 2002. – с. 71-73.
2. Жаднов В.В., Мазница Е.М. Подсистема анализа и обеспечения показателей надежности и качества на ЭВМ *IBM PC/AT* // Международная научно-техническая конференция «Методы и средства оценки и повышения надежности приборов, устройств и систем». Тезисы докладов Пенза: ППИ, 1992. - с. 80.
3. Жаднов В.В., Мазница Е.М., Гусев А.Ю. Пакет программ обработки результатов анализа стабильности для подсистемы АСОНИКА-К // Международная научно-техническая конференция «Методы и средства оценки и повышения надежности приборов, устройств и систем». Тезисы докладов – Пенза: ППИ, 1992. - с. 20.
4. Борковский А.В., Жаднов В.В., Коноплев Я.И. Автоматизация анализа отказов ИЭТ // Современные методы обеспечения качества и надежности электронных приборов. Материалы семинара - М.: МДНТП, 1990. - с. 123-125.
5. Жаднов В.В., Полегайко А.В., Полесский С.Н. Метод автоматизированного анализа результатов расчетов надежности РЭА // Сборник научных трудов «Электромагнитная совместимость и проектирование электронных средств». / Составление - Кечиев Л.Н. – М.: МИЭМ, 2002. – с. 73-77.

## Проблемы повышения достоверности надежностных расчетов ЭРИ

С.А. Пращикин

Кафедра «Радиоэлектронные и телекоммуникационные устройства и системы»  
Московский Государственный институт электроники и математики

E-mail: [asonika-k@mail.ru](mailto:asonika-k@mail.ru)

E-mail: [asonika-k@mail.vandex.ru](mailto:asonika-k@mail.vandex.ru)

Web site: <http://asonika-k.ru>

В настоящее время расчет надежности электрорадиоизделий (ЭРИ) обладает, на мой взгляд, двумя существенными недостатками – математическим, связанным с оценкой интенсивности отказов (вероятности безотказной работы), и технологическим, обусловленным большим количеством новых и обновленных ЭРИ.

Измерение надежности ЭРИ сводится к оценке интенсивности отказов (вероятности безотказной работы) в интервале времени, составляющем сравнительно малую долю полной наработки изделий. Для обработки этих оценок используется математический аппарат, в основе которого лежит экспоненциальная модель, предполагающая, что и в дальнейшем интенсивность отказов будет такой же. В результате, прогнозирование средней наработки до отказа оказывается явно грубым и завышенным. Однако это допущение позволяет рассчитывать надежность радиоэлектронной аппаратуры (РЭА), в состав которой входят новые ЭРИ, «персональные» характеристики надежности которых не удастся получить в короткие сроки.

Первый недостаток можно устранить путем использования более адекватных моделей надежности (более точных), например, двухпараметрического  $DN$ -распределения [1]. Однако вследствие малой статистики, а также отсутствия надежного аппарата расчета надежности систем на основе двухпараметрических распределений, вполне вероятно появление больших погрешностей, как в определении параметров распределения, так и в самом расчете.

При разрешении математического затруднения становятся ощутимыми проблемы технологического характера – при создании новых ЭРИ

производители лишь приблизительно приписывают им те или иные надежностные характеристики (групповые) [2]. Естественно предположение, что более детальная информация об изделиях повысит достоверность расчетов. Однако, даже имея результаты «персональных» испытаний ЭРИ, конечные пользователи с громадным опозданием (иногда превышающим «срок жизни» ЭРИ) получают эту информацию, в доступной для расчетов форме.

Ведущаяся в данное время работа [3], позволит решить указанные проблемы:

- отказ от аналитического решения вероятностно-физических уравнений и переход на машинный уровень (программирование) позволит более точно, быстрее и менее трудоемко рассчитывать двухпараметрические системы
- использование уже накопленной статистики и данных по надежности ЭРИ для экспоненциальных моделей разрешит проблему более точного определения параметров моделей на основе *DN*-распределения
- *Internet* и *intranet*-технологии позволят в корне изменить качество и скорость доступа пользователей к новой и обновленной информации о ЭРИ.

### Список литературы

1. Погребинский С.Б., Стрельников В.П. Проектирование и надежность многопроцессорных ЭВМ – М.: Радио и связь, 1988. - 168 с.: ил.
2. Надежность ЭРИ: Справочник. CD-версия // С.Ф. Прытков, В.М. Горбачева, А.А. Борисов и др. / Научный руководитель С.Ф. Прытков – М.: МО РФ, 2000. – 508 с.
3. Працикин С.А. Программно-методический комплекс для идентификации надежностных параметров ИЭТ // Научно-техническая конференция студентов, аспирантов и молодых специалистов. Тезисы докладов – М.: МГИЭМ, 1999. – с. 105.

## **Программа синтеза систем виброудароизоляции конструкций РЭС**

**Кожевников А.М., Жарков**

**(Московский государственный институт электроники и математики)**

При проектировании систем виброудароизоляции РЭС ( радиоэлектронных средств ) необходимо выполнить ряд условий, позволяющих обеспечить оптимальную по качеству ее работу. К таким условиям относятся [ 1 ]:

1. суммарная грузоподъемность всех виброудароизоляторов равна силе тяжести виброудароизолируемой аппаратуры и статические нагрузки на виброизоляторы должны быть в допустимых пределах;

2. координаты центра масс виброудароизолируемой аппаратуры должны совпадать с центром жесткостей системы виброудароизоляторов по всем возможным направлениям воздействий;

3. резонансные частоты системы виброудароизоляции должны быть сгруппированы в узкой полосе частот, которая может быть задана исходя из амплитудно-частотных характеристик виброудароизоляторов при номинальной статической нагрузке;

4. система виброудароизоляции должна обеспечивать заданное уменьшение уровня вибрационных и ударных перегрузок и амплитуд колебаний, передаваемых на аппаратуру от источника внешних механических воздействий.

В настоящее время синтез систем виброудароизоляции решается путем выбора ее варианта инженером-конструктором с последующей проверкой путем либо расчета на ЭВМ, либо экспериментально. Нахождение оптимального варианта при этом представляет трудоемкую задачу.

Предлагается алгоритм автоматического синтеза систем виброизоляции, основные этапы которого приведены ниже:

1. Задаются исходные данные-описание виброудароизолируемой конструкции, характеристики возможных к применению

виброудароизоляторов, координаты возможных мест установки виброудароизоляторов, параметры внешних механических воздействий;

2. В цикле по вариантам установки виброудароизоляторов и их типам задаются параметры виброудароизоляторов исходя из условий 1,3 и производится расчет деформаций виброудароизоляторов под действием силы тяжести или статической силы в центре масс по направлениям внешнего механического воздействия используя математическую модель, описанную в работе [ 2 ];

3. По результатам анализа величин деформаций виброудароизоляторов под воздействием статической силы определяются виброудароизоляторы, жесткость которых надо увеличить или уменьшить, чтобы добиться равенства деформаций всех виброудароизоляторов при воздействии вдоль соответствующей оси координат, что соответствует условию 2 рационального монтажа. При этом должно выполняться также условие 1.

4. Из проанализированных вариантов выбирается вариант с наименьшим количеством виброудароизоляторов, обеспечивающий наилучшим образом условия 1,2,3 рационального монтажа;

5. Производится расчет амплитудно-частотной характеристики системы виброизоляции согласно [ 2 ] и, если она удовлетворяет поставленным требованиям по ослаблению вибрации, то данный вариант системы виброизоляции принимается к практической реализации.

Синтез систем виброудароизоляции при ударном воздействии состоит из следующих этапов:

1. Выполняются п.п. 1,2,3 предыдущего алгоритма ( условие 3 может не выполняться, если нет требований по одновременной защите от вибраций и ударов);

2. Для каждого варианта рационального расположения и выбора виброудароизоляторов строятся графики суммарной силовой ударной характеристики и характеристики суммарной энергоемкости системы виброудароизоляции и производится приближенный расчет максимального

ускорения аппаратуры при ударном воздействии либо исходя из условия полного перехода кинетической энергии аппаратуры в потенциальную энергию виброудароизоляторов, либо методом эквивалентных прямоугольных импульсов;

3. Из проанализированных вариантов выбирается вариант с наименьшим количеством виброудароизоляторов, обеспечивающий наилучшим образом выполнение условий 1,2,3 рационального монтажа и условие ослабления ускорений на основе расчета в предыдущем пункте;

4. Производится расчет переходного процесса в системе виброудароизоляции согласно [ 2 ] и, если она удовлетворяет требованиям по ослаблению ударного воздействия, то данный вариант принимается к практической реализации.

На основе приведенных алгоритмов разработана программа для ПЭВМ.

### **Список литературы**

1. Ильинский В.С. Защита РЭА и прецизионного оборудования от динамических воздействий. – М.: Радио и связь, 1982.

2. Кожевников А.М. Оптимальный выбор эксплуатационных режимов и допусков на параметры электрорадиоизделий устройств телекоммуникаций // Информационные технологии в проектировании и производстве: Науч.-техн. журн.- ГУП "ВИМИ", 2001, N3, с. 72-77.



## Метод построения макромоделей процессов в РЭС в частотной области

А.М.Кожевников

Московский государственный институт электроники и математики, каф. ИТАС

Для проведения параметрической и структурной оптимизации частотных характеристик эквивалентных электронных схем радиоэлектронных средств (РЭС), в том числе моделирующих вибрационные и другие процессы в различных объектах на основе электроаналогий [3], был разработан, излагаемый ниже, метод построения макромоделей процессов в РЭС и анализа чувствительности выходных характеристик эквивалентных электронных схем к большим изменениям параметров и структуры в частотной области.

Модель электронной схемы в частотной области, построенная в расширенном однородном координатном базисе, имеет вид:

$$[C_0 s + G_0] \bar{X}_0 = \bar{Y},$$

(1)

где  $C_0, G_0$  - ( $n \times n$ ) - исходные вещественные матрицы коэффициентов;  $\bar{Y}(n)$  - вектор возмущающих воздействий;  $\bar{X}_0(n)$  - искомый вектор напряжений в узлах схемы и токов через индуктивности;  $n$  - количество неизвестных;  $s = j\omega$ ;  $j = \sqrt{-1}$ ;  $\omega$  - круговая частота воздействия.

При анализе чувствительности схем к большим изменениям параметров электрорадиоизделий (ЭРИ), от значений которых зависят коэффициенты матриц  $C_0$  и  $G_0$ , параметр может изменяться от нуля до бесконечности, что позволяет варьировать и структурой схемы подключая или отключая элементы.

Определим сначала чувствительность выходной характеристики к изменению параметров матриц  $C_0$  и  $G_0$ . Пусть варьируются  $m \ll n$  параметров и они изменяются на величину  $\delta_{c1}, \delta_{c2}, \dots, \delta_{cm}$  или  $\delta_{g1}, \delta_{g2}, \dots, \delta_{gm}$  в матрицах  $C_0$  и  $G_0$  соответственно, а также в различных возможных вариантах.

Проводимость ветви -  $u$  (или ее сопротивление), включенная между  $i$  - м и  $j$  - м узлами, появляется в матрице  $C_0$  и  $G_0$  в столбцах и строках с номерами  $i$  и  $j$ ,

при этом имеет знак плюс в диагональных элементах (i, i) и (j, j) и знак минус во внедиагональных элементах (i, j) и (j, i). Эту зависимость можно записать, например, для матрицы  $C_0$  в форме произведения векторов [ 1 ]:

$$y(e_i - e_j)(e_i - e_j)^t,$$

где  $t$ - индекс транспонирования;  $e_i$  - единичный вектор, все компоненты которого, кроме  $i$ -го, равны нулю, а  $i$  - й равен единице;  $e_j$  - единичный вектор, все компоненты которого, кроме  $j$ -го, равны нулю, а  $j$  - й равен единице.

Тогда матрица  $C_0 = \sum_k y_k (e_{i_k} - e_{j_k})(e_{i_k} - e_{j_k})^t$ . Для четырехполюсников, имеющих два входных узла  $i_1$  и  $i_2$  и два выходных узла  $j_1$  и  $j_2$  элемент матрицы  $C_0$ , равный  $y$  можно записать в виде:

$$y(e_{i_1} - e_{i_2})(e_{j_1} - e_{j_2})^t,$$

Для  $k$ -го четырехполюсника алгоритм занесения  $y_k$  в матрицу  $C_0$  (или  $G_0$ ) можно записать в виде  $y_k p_k q_k^t$ , где  $p_k = e_{k1} - e_{k2}$ ,  $q_k = e_{k1} - e_{k2}$ .

Выходную величину представим в виде элементов вектора  $\bar{X}_0$ , где  $d^t$  - транспонированный вектор, определяющий номер элемента вектора  $\bar{X}$ , являющегося выходной характеристикой  $F$  при измененных параметрах:

$$F_0 = d^t \bar{X}_0, F = d^t \bar{X}.$$

Обозначим  $[C_0 s + G_0] = A_0(s)$ , при этом  $A_{ij} = C_{ij} s + G_{ij}$ . Тогда при изменении параметров можно записать:

$$A_0 \bar{X} + P_c \delta_c s Q_c^t \bar{X} + P_g \delta_g Q_g^t \bar{X} = \bar{Y}, \quad (2)$$

где  $P_c = [p_{c1}, p_{c2}, \dots, p_{cm}]$ ,  $Q_c = [q_{c1}, q_{c2}, \dots, q_{cm}]$ ,  $P_g = [p_{g1}, p_{g2}, \dots, p_{gm}]$ ,  $Q_g = [q_{g1}, q_{g2}, \dots, q_{gm}]$ ,

$\bar{X}$  - вектор решения при измененных параметрах, а  $\delta_c, \delta_g$  - диагональные матрицы ( $m \times m$ ) изменений параметров.

Матрицы  $P_c, Q_c, P_g, Q_g$  имеют размерность  $n \times m$  и содержат нули и  $\pm 1$ . Обозначим  $d_c = Q_c^t \bar{X}$ ;  $z_c = \delta_c s d_c$ ;  $d_g = Q_g^t \bar{X}$ ;  $z_g = \delta_g d_g$ . Уравнение ( 2 ) представим в виде системы:

$$\begin{aligned}
& A_0 \bar{X} + P_c Z_c + P_g Z_g = \bar{Y}, \\
& \delta_c s d_c - z_c = 0, \\
& \delta_g d_g \quad \quad \quad - \quad \quad \quad z_g \quad \quad \quad = \quad \quad \quad 0, \\
(3) \quad & -Q_c^t \bar{X} + d_c = 0, \\
& -Q_g^t \bar{X} + d_g = 0.
\end{aligned}$$

Подставив  $\bar{X} = A_0^{-1} \bar{Y} - A_0^{-1} P_c Z_c - A_0^{-1} P_g Z_g$  из (2) в систему (3) и исключив из нее путем подстановки  $d_c$  и  $d_g$ , получим систему уравнений:

$$\begin{aligned}
& (sF_c + \delta_c^{-1}) Z_c + sF_{cg} Z_g = s\hat{Y}_c; \\
& F_{gc} Z_c \quad \quad \quad + \quad \quad \quad (F_g \quad \quad \quad + \quad \quad \quad \delta_g^{-1}) Z_g \quad \quad \quad = \hat{Y}_g; \\
(4) \quad &
\end{aligned}$$

$$F = F_0 - t_c Z_c - t_g Z_g;$$

где  $F_c = Q_c^t A_0^{-1} P_c$ ,  $F_g = Q_g^t A_0^{-1} P_g$ ,  $F_{cg} = Q_c^t A_0^{-1} P_g$ ,  $F_{gc} = Q_g^t A_0^{-1} P_c$ ,  $\hat{Y}_c = Q_c^t A_0^{-1} \bar{Y}$ ,  $\hat{Y}_g = Q_g^t A_0^{-1} \bar{Y}$ ,  
 $t_c = d^t A_0^{-1} P_c$ ,  $t_g = d^t A_0^{-1} P_g$ ,  $F_0 = d^t A_0^{-1} \bar{Y}$ .

В системе уравнений (4) матрицы  $F_c$ ,  $F_g$ ,  $F_{cg}$ ,  $F_{gc}$  имеют размерность  $m \times m$ , а векторы  $\hat{Y}_c$ ,  $\hat{Y}_g$ ,  $Z_c$ ,  $Z_g$  и строки  $t_c$ ,  $t_g$  размерность  $m$ ,  $F_0$  и  $F$  - скаляры.

Для того, чтобы в выражении для  $F$  сохранить его аналитическую зависимость от  $s$ , вычисление  $A_0^{-1}$  производится по методу [2] и основано на вычислении собственных значений  $\lambda_1, \dots, \lambda_n$  матрицы  $A_0 = C_0 \lambda + G_0$  и всех ее правых  $V = [\bar{V}_1, \dots, \bar{V}_n]$  и левых  $H^T = [\bar{H}_1, \dots, \bar{H}_n]^t$  собственных векторов, связанных условием нормировки  $H^T G V = E$ , где  $E$  -  $(n \times n)$  единичная матрица. В результате матрица  $A_0^{-1} = V [D s + E]^{-1} H^t$ , где  $D = \text{diag} (-\lambda_1^{-1}, \dots, -\lambda_n^{-1})$ .

Таким образом, вычислив один раз  $A_0^{-1}$  в виде аналитической зависимости от  $s = j\omega$ , и подставив в систему (4), имеющую размерность  $(m \times m)$ , можно вычислять выходную характеристику на различных  $\omega$  при больших изменениях  $m$  параметров.

При необходимости вычислять выходную характеристику в нескольких узлах схемы необходимо вместо вектора  $d^t$  использовать диагональную

матрицу, в диагонали которой находятся единицы в тех строках, номера которых равны номерам узлов выходных характеристик, а остальные элементы равны нулю. В этом случае  $F_0$  и  $F$  будут векторами и изменятся также размерности некоторых используемых массивов.

### **Список литературы**

1. Влах И., Сингхал К. Машинные методы анализа и проектирования электронных схем: Пер с англ. - М.: Радио и связь.- 1988.-560 с.

2. Борисов Н.И. Методы повышения эффективности многовариантного анализа устойчивости и частотных свойств эквивалентных схем//Сб. лекций IV Всесоюзной школы-семинара "Теория и математическое моделирование объемных интегральных схем (ОИС) СВЧ и КВЧ". - Алма-Ата.: КГУ.-1989.- с. 63-69.

3. Кожевников А.М., Кофанов Ю.Н., Чернушенко А.М. Машинное моделирование для расчета на виброустойчивость //Известия вузов СССР, Сер. Радиоэлектроника .- 1980.- №6.- с. 74-77.

**Обработка экспериментальных данных  
в автоматизированном аппаратно-программном комплексе  
измерения параметров различных видов помех ДКМ диапазона**

**А.О. Щирый  
МарГТУ, каф. Высшей математики  
42400, Россия, г. Йошкар-Ола, пл. Ленина, 3; e-mail: [saovu@mail.ru](mailto:saovu@mail.ru)**

Измерение помех ДКМ диапазона является актуальной задачей. Для радиосвязи оперативная информация о помеховой обстановке необходима для повышения надежности связи и снижения мощности связного сигнала путем выбора канала и адаптивной фильтрации помех. Для радио- и геофизики интерес представляет информация о суточных, сезонных и годовых вариациях уровней помех. Целесообразно использовать для регистрации данных ПЭВМ общего назначения, а обработку осуществлять программно.

Разработан комплекс, позволяющий производить измерения автоматически, без непосредственного участия человека. В состав комплекса входят: ПЭВМ с разработанным программным обеспечением (ПО), антенна, радиоприемное устройство (РПУ) и источник сигналов точного времени. ПО комплекса работает под управлением ОС *Windows 9x/NT/2000/XP* и состоит из двух частей: измерение (и обработка) в реальном времени и вторичная обработка.

В реальном времени ПЭВМ осуществляет не только обработку, но и управление РПУ: перестройка РПУ происходит последовательно с установленным шагом в заданном диапазоне. Для оцифровки сигнала используется звуковая карта ПЭВМ. Оцифровка и анализ сигнала осуществляется в многозадачном режиме: одновременно выполняются несколько потоков (thread) с различным приоритетом. Поток с высшим приоритетом осуществляет сохранение в файле отсчетов приемного буфера по приходу прерывания от звуковой карты, означающего заполнение буфера; сообщает об этом другим потокам, устанавливая семафор. Потоки обработки считывают и обрабатывают информацию из буфера: отбрасываются отсчеты,

приходящиеся на время перестройки РПУ; умножают каждый отсчет временного ряда на коэффициент, обратный коэффициенту усиления РПУ на текущей частоте (для коррекции АЧХ РПУ), далее на отсчеты накладывается весовая функция, выполняется БПФ, а полученные спектры усредняются по методу Уэлча, записываются в файл спектральной плотности помех, результаты отображаются также на мониторе.

Вторичная обработка предназначена для выделения определенных видов помех и сбора статистической информации о них. При оценке уровня фонового шума (атмосферный, галактический) для отделения стационарных помех в заданном интервале выбирается наименьшее значение помех; оно и считается фоновым шумом на данном интервале [1]. Для измерения статистических параметров сосредоточенных помех задается некоторый уровень от среднего значения на некотором интервале, выше которого помехи считаются стационарными (большинство сосредоточенных помех).

Проведены измерения помех в г. Йошкар-Ола. В нижней части ДКМ диапазона (2 – 20 МГц) максимум общего уровня помех наблюдается вечером (16-18 часов), минимум – ночью и ранним утром (2-6 часов). Днем преобладающими здесь являются стационарные и промышленные помехи, что препятствует исследованию атмосферных помех (в городской местности). В верхней части ДКМ диапазона (20 - 30 МГц) преобладают галактические помехи, характеризующиеся незначительными суточными вариациями. Полученные экспериментальные данные хорошо согласуются с результатами моделирования по модели помех, рекомендованной МККР [2] для городской местности.

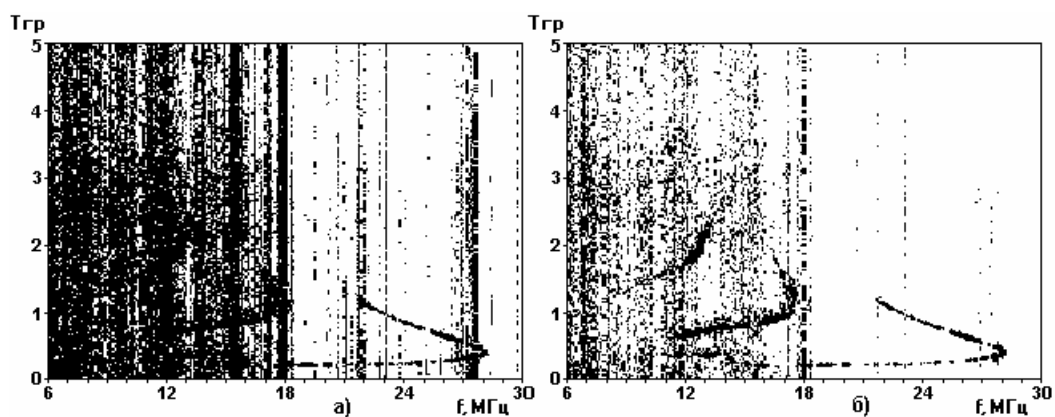


Рис.1. ИONOграмма радиолинии Шпицберген - Йошкар-Ола, 14.09.2001, Время: 11:26

В качестве примера адаптивной фильтрации помех на рис.1 показан результат фильтрации сосредоточенных помех при приеме непрерывного ЛЧМ-сигнала (рис.1.а - до фильтрации, рис.1.б - после): между сеансами наклонного ЛЧМ-зондирования производится оперативное измерение уровня помех, определяются параметры цифрового фильтра.

Работа выполнена при поддержке РФФИ, проекты 02-05-64950, 02-02-16318 и Минобразования РФ, проект Е00-8.0-41.

### Список литературы

1. Earl G.F. and Ward B.D. The frequency management system of the Jindalee over-the-horizon backscatter HF radar // Radio Science.-1987.-V.22.-№2.-P.275-291.
2. Отчет МККР N 322, Женева, 1963. - М.: Связь, 1965. - 80 с.

## **Программное обеспечение вычислительного процесса проектирования энергетических котлоагрегатов**

**В.С. Беднаржевский, E-mail: [bednar@s42.dcn-asu.ru](mailto:bednar@s42.dcn-asu.ru)**

*Разработан комплекс пакетов прикладных программ для моделирования и автоматизированного проектирования паровых котлов.*

Автоматизированное проектирование является одним из наиболее быстро развивающихся направлений научно-технического прогресса, современные темпы которого обуславливают необходимость повышения качества и сокращения продолжительности проектирования новых видов продукции.

Автоматизированное проектирование энергетических котлов относятся к классу сложных задач ввиду их единичного производства и, как следствие, при алгоритмизации такого вида задач возникают определенные трудности. Эта проблема может быть решена на основе широкого использования средств вычислительной техники и компьютерных технологий. Основными целями автоматизированного проектирования энергетических котлоагрегатов являются:

сокращение сроков разработки новых котлов и запуска их в производство вследствие более совершенной организации всего цикла проектирования и отладки проектных решений на моделях на ранних стадиях конструирования без изготовления дорогостоящего прототипа;

повышение качества и конкурентоспособности изделий путем внедрения компьютерных технологий, включая геометрическое моделирование, математические методы анализа и оптимизации будущей конструкции.

Разработано семь пакетов прикладных программ (ППП) для проектирования котлоагрегата в режиме диалога: «Тепловой расчет котлоагрегата»; «Расчет температуры металла стенки трубы»; «Расчет на прочность цельносварных газоплотных конструкций»; «Аэродинамический расчет котлоагрегата»; «Расчет теплофизических свойств теплоносителей»;



«Гидравлический расчет пароперегревателя»; «Расчет надежности работы котлоагрегата».

В ППП теплового расчета котлоагрегата входят расчеты: объемов газообразных продуктов сгорания, объемных долей трехатомных газов, концентрации золы; таблицы энтальпий теоретического расхода воздуха и газообразных продуктов сгорания; теплового баланса котлоагрегата; топки; ширмовых пароперегревателей; конвективных пароперегревателей; хвостовых поверхностей нагрева. При определении объемов газов, объемных долей трехатомных газов, концентрации золы на печать выводятся значения теоретических объемов воздуха, азота, трехатомных газов, водяных паров, номера поверхностей, коэффициенты избытка воздуха по поверхностям, средний объем водяных паров и дымовых газов, объемные доли трехатомных газов, водяных паров и суммарная концентрация золы в газоходе.

В результате расчета энтальпий продуктов сгорания выводятся на печать следующие данные: избыток воздуха по поверхностям нагрева, температура через каждые 100 °С, теплосодержание газов и воздуха по поверхностям нагрева.

При расчете теплового баланса выводятся на печать значения КПД котлоагрегата; коэффициента сохранения теплоты; потери теплоты (с физической теплотой шлаков, с уходящими газами); располагаемой теплоты; тепловосприятия в котлоагрегате; полного расхода топлива; расчетного расхода топлива; теплоты, внесенной воздухом.

Расчет топки предназначен для определения тепловых характеристик топочного устройства. Предусмотрен случай, когда ширмы находятся в активном топочном объеме. Расчетная температура газов на выходе из топки  $T_{вых}^{расч}$  сравнивается с заданной температурой газов  $T_{вых}^{зад}$ , пользователь с клавиатуры дисплея варьирует значение высоты топки  $H_T$  так, чтобы выполнялось неравенство  $(T_{вых}^{зад} - T_{вых}^{расч}) \leq 1^\circ\text{C}$ . При этом происходит изменение эффективной степени черноты факела, степени черноты факела в свободном объеме топки, толщины излучающего слоя, свободного объема и поверхности

стен топки. Промежуточные значения  $N_T$  и  $T_{вых}^{расч}$  выдаются на экран дисплея, что позволяет анализировать эти параметры.

При расчете ширмового пароперегревателя вначале определяется энтальпия газов за ширмами, расположенными в выходном окне топки, на основе решения уравнения теплового баланса.

Если тепловосприятие ширм, найденное по уравнению теплопередачи, больше (меньше) тепловосприятия, найденного по балансу, более чем на 0,1 %, то на экран дисплея выдается сообщение «Увеличить (уменьшить) среднюю температуру газов в ширмах». Пользователь соответственно изменяет значение температуры.

При расчете конвективных ступеней предварительно вводится значение температуры пара между ступенями, и если расхождение между тепловосприятием по уравнению теплопередачи и балансу будет больше 0,1 %, то пользователь с клавиатуры дисплея задает новое значение температуры пара.

При расчете водяного экономайзера проверяется расхождение тепловосприятия по тепловому балансу и условиям теплопередачи, если расхождение больше 0,1 %, то с клавиатуры дисплея изменяется температура газов за ступенью и операция повторяется.

При расчете регенеративного вращающегося воздухоподогревателя осуществляется проверка температуры стенки по температуре точки росы, и при нарушении граничных условий на экран дисплея выдается сообщение об изменении принятых значений температуры газов. В конце расчета печатаются таблицы тепловых характеристик поверхностей нагрева. Алгоритм соответствует нормам теплового расчета котлоагрегатов [1].

Тепловой расчет функционирует совместно с инвертированной базой данных (БД) «Котлоагрегат», позволяющей по известному значению ключевого признака определить адреса всех записей, которые его содержат.

Расчетную схему можно формировать в режиме диалога и результаты теплового расчета выводить на экран дисплея в виде двух схем: водопарового и газозвоздушного трактов котлоагрегата (Рис. 1). На схеме водопарового тракта

выводится значение температуры среды по поверхностям нагрева. На схеме газоздушного тракта выводится значение температуры продуктов сгорания до и после расчетного участка.

ППП «Расчет температуры металла стенки трубы» предназначен для определения средней температуры металла стенки неравномерно обогреваемой (развернутой) трубы ширмового и конвективного пароперегревателей. Алгоритм пакета состоит в следующем [1].

В расчетном сечении находится среднее значение удельного тепловосприятия наиболее нагруженной образующей трубы. Определяется максимальное удельное тепловосприятие. Рассчитываются разности температур газов и внутренней среды при входе в пакет и выходе из него, для прямотока и противотока. Вычисляется температурный напор участка. Находится поверхность нагрева. Определяется расчетная поверхность участка змеевика для одного из случаев: первых пяти рядов ширм, «видящих» топку; остальных рядов ширм и всех рядов конвективных пакетов, облучаемых из топки или из впереди лежащего объема; для рядов ширм и пакетов, облучаемых из сзади расположенного объема. Рассчитывается тепловосприятие участка змеевика (от начала до расчетного сечения). Эффективная толщина излучающего слоя вычисляется для ширм конвективного пароперегревателя. Определяется приращение энтальпии среды на участке змеевика от начала элемента до расчетного сечения. Находятся значения энтальпий среды в определенных точках, и по этим данным вычисляется температура пара. Определяются энтальпия и по ее значениям температура в расчетном сечении наиболее обогреваемой (развернутой) трубы элемента. Вычисляется величина превышения средней температуры среды в расчетном сечении наиболее нагруженной трубы. Найденная в расчетных точках температура, средняя по толщине металла стенки развернутой трубы, выдается на дисплей. Пользователь анализирует эти значения и при необходимости повторяет расчет, изменив исходные данные с клавиатуры дисплея.

ППП «Расчет на прочность цельносварных газоплотных конструкций»

предназначен для определения теплонапряжений в цельносварных газоплотных экранах котла в зоне максимального уровня тепловых потоков с учетом дополнительных нагрузок [2]. Пакет производит расчет напряжений от внутреннего давления, весовых нагрузок, хлопка в топке, неравномерного распределения тепловых потоков по ширине экранов, разности температур экрана и балки жесткости, находит суммарные напряжения и оценивает прочность.

При расчете напряжений определяют:

от внутреннего давления в трубе — средние окружные, осевые и радиальные напряжения;

от весовых нагрузок — осевые напряжения в оребренной трубе фронтального и бокового экранов;

от хлопка в топке — изгибные напряжения, максимальный изгибающий момент, момент сопротивления в балке жесткости, максимальный изгибающий момент оребренной трубы в середине сечения фронтального экрана под обвязочным поясом, изгибающий момент в оребренной трубе между поясами жесткости, осевые и окружные напряжения от изгибающих моментов. При расчете температурных напряжений, вызванных неравномерным распределением теплового потока по ширине экранов, определяют осевые напряжения в трубе в середине и углу бокового и фронтального экранов.

При расчете суммарных напряжений и оценке прочности определяют алгебраическую сумму окружных и осевых напряжений под поясом жесткости в середине фронтального экрана. Далее находят эквивалентные напряжения и проверяют условие прочности. Если условие не выполняется, то на дисплей выдают диагностическое сообщение и, изменив исходные данные, повторяют расчет. В конце расчета на дисплей и печать выдаются алгебраическая сумма окружных и осевых напряжений и эквивалентные напряжения.

ППП «Аэродинамический расчет котлоагрегата» предназначен для расчета аэродинамических сопротивлений газового и воздушного трактов котельного агрегата [3]. ППП позволяет для газового тракта котлоагрегата рассчитывать

следующие поверхности: пароперегреватель, фестон, водяной экономайзер, воздухоподогреватель. Поток газов может омывать шахматные, коридорные, продольные пучки. Предусмотрен расчет участков от воздухоподогревателя до дымовой трубы, дымовая труба. Для всех участков рассчитываются повороты в канале, изменение сечения, раздающие и собирающие короба, диффузоры. Для воздушного тракта котельной установки можно учесть следующие виды сопротивлений: измерительная шайба, поворот на  $N$  градусов, карман Рихтера, трение на участке, диффузор, изменение сечения, калорифер, воздухонагреватель, конфузор, сопротивление входа в трубу, шибер, сопла, горелки, собирающий тройник.

ППП «Расчет теплофизических свойств теплоносителей» предназначен для расчета термодинамических и теплофизических свойств воды, водяного пара и дымовых газов [4]. В состав пакета входят 17 программ. Программы расчета энтальпий газов по температуре, критерия Прандтля для дымовых газов, теплопроводности дымовых газов, кинематической вязкости дымовых газов применимы для температуры  $0...3000$  °С. Программы расчета температуры пара по энтальпии и давлению, энтальпии пара по давлению и температуре применяются при температуре  $330...560$  °С и давлении  $10...20$  МПа. Программы расчета энтальпии воды по температуре и давлению, температуры воды по энтальпии и давлению, удельного объема воды используются при температуре  $215...320$  °С и давлении  $10...20$  МПа. Расчет удельного объема воды применим для всей области докритического давления. Программы расчета удельного объема пара на кривой насыщения, энтальпии воды на линии насыщения, энтальпии пара на линии насыщения, температуры воды и водяного пара на линии насыщения, удельного объема воды на линии насыщения применяются при температуре насыщения. Программы расчета динамической вязкости воды и водяного пара, теплопроводности воды и водяного пара применяются для температуры  $0...800$  °С и давления до  $100$  МПа.

ППП «Гидравлический расчет пароперегревателя» предназначен для расчета перепада давления по паровому тракту пароперегревателя

котлоагрегата [5]. Основные программы пакета: расчет удельного объема пара, внутреннего сечения труб, перепада давления, скорости пара, гидравлических сопротивлений входа, выхода, поворота, трения по длине труб, вывода на печать результатов расчета.

На этапе конструирования необходимо кроме основных расчетов (теплового, гидравлического, прочностного и т. д.) проводить вероятностную оценку надежности каждого варианта конструкции котлоагрегата. ППП «Расчет надежности котлоагрегата» предназначен для расчета надежности работы поверхностей нагрева котлоагрегата [6, 7]. Параметр потока отказов по каждой поверхности рассчитывается как сумма произведений числа элементов на интенсивность отказов. Элемент складывается из прямых участков, бесштуцерной приварки змеевиков к камерам, гибов труб, контактных сварных стыков. Также определяется характеристика потока вынужденных остановов на  $10^5$  ч работы.

Во всех пакетах исходная информация в режиме диалога вводится с клавиатуры дисплея в последовательности, определяемой запросами ЭВМ. Режим диалога обеспечивают программы бесформатного ввода с дисплея числовых и символьных данных. В ходе расчетов на дисплей выводятся промежуточные результаты, на основании анализа которых пользователю предлагается выбрать стратегию поведения: продолжить расчет либо, изменив исходные данные, вернуться к началу программы,

## Литература

1. Тепловой расчет котлов: Нормативный метод. СПб: НПО ЦКТИ, 1998. 257 с.
2. РТМ 108.031.108—78. Котлы стационарные паровые и водогрейные: Расчеты на прочность цельносварных газоплотных конструкций. Л.: НПО ЦКТИ, 1985. 166 с.

3. Аэродинамический расчет котельных агрегатов: Нормативный метод. Л.: Энергия, 1977. 256 с.
4. Ривкин С. Л., Кременевская Е. А. Уравнения состояния воды и водяного пара для машинных расчетов процессов и оборудования электростанций // Теплоэнергетика. 1977. № 3. С. 69—73.
5. Гидравлический расчет котельных агрегатов: Нормативный метод. М.: Энергия, 1978. 256 с.
6. Щеглов А. Г., Логвинова Н. А. Характеристики надежности элементов поверхностей нагрева котлоагрегатов // Электрические станции. 1975. № 9. С. 8—11.
7. Щеглов А. Г. Расчет надежности поверхностей нагрева котельных агрегатов // Электрические станции. 1975. № 11. С. 10—12.

## Статистическая модель ошибки измерения псевдодальности навигационной системы GPS

Э.Л. Аким, Д.А. Тучин  
Институт прикладной математики им. М.В. Келдыша РАН

Широко распространенные в настоящее время GPS-приемники ориентированны в основном на информационное обеспечение потребителей, расположенных на поверхности Земли. Применение средств космической навигации в сложных технических системах, к которым относятся бортовые автономные системы управления космических аппаратов, предъявляет более высокие требования к надежности работы аппаратуры потребителя. В Институте прикладной математики проводятся исследования, направленные на изучение возможности применения средств космической навигации на борту космического аппарата (КА).

Малая измерительная база бортовых приемников заставила моделировать работу системы GPS на борту КА. Для этого был построен программный комплекс моделирования измерений. Комплекс основан на результатах статистического анализа ошибки измерения псевдодальности на C/A коде [1]. В этой работе приведены результаты построения статистической модели измерения псевдодальности.

Для проведения сравнительного анализа были использованы измерения трех зарубежных станций, измерения которых доступны в интернете.

Станции оснащены различными типами приемников и расположены на разных широтах.

станция	расположение	тип приемника	широта	долгота
bahr	Бахрейн	ASHTECH Z-X113	26.21 с.ш.	50.61
сена	Аляска	TRIMBLE 4000 SSI	65.50 с.ш.	215.32
usno	Вашингтон	AOA SNR-12 ACT	38.92 с.ш.	282.93

Таблица 1. Источники измерительной информации



Обработка измерений проводилась с помощью разработанных в ИПМ алгоритмов определения вектора состояния наблюдателя [2].

Источники ошибок измерений псевдодальности можно разделить на три группы:

- погрешности эфемеридного и частотно-временного обеспечения навигационных КА (НКА);
- погрешности измерений, вызываемые атмосферой Земли;
- погрешности, вносимые аппаратно-программным комплексом потребителя.

Отметим, что еще существует ошибка, связанная с многолучевостью принимающей антенны [3]. Данная ошибка в этой работе не рассмотрена.

Погрешности эфемеридного и частотно-временного обеспечения состоят из ошибок численно-аналитической модели движения, представленной в интерфейсном документе и ошибок прогноза этих эфемерид. Для оценки точности прогноза эфемерид и значения ухода часов, передаваемых с борта в составе эфемеридного сообщения, были использованы эталонные орбиты и значения ухода часов НКА, публикуемые в Internet на сервере NASA.

С использованием измерений псевдодальностей на двух частотах была определена ионосферная поправка измерений псевдодальности на C/A-коде для наземных приемников. Далее будут приведены статистические характеристики этой поправки.

При получении измерений псевдодальности и псевдоскорости возникают ошибки, связанные с нестабильностью значения частоты гетеродина приемника, с моментом регистрации измерений и уходом фазового сдвига псевдошумовой последовательности, который вызван различной синхронизацией часов приемника и GPS-системы. Изучена динамика ухода частоты гетеродина приемника, фазы генерации псевдошумовой последовательности и точность времени регистрации сигнала.

Для определения источников ошибок измерений наземных приемников была исключена ошибка, вызванная тропосферными эффектами. В таблице

представлены основные источники ошибок измерений псевдодальности и их величины.

источники ошибок измерений, СКО [м]	bahr ASHTECH Z-X113	cena TRIMBLE 4000 SSI	usno AOA SNR-12 АСТ
эфемериды НКА	1.06	0.98	1.55
частотно-временное обеспечение НКА	7.62	1.82	37.16
ионосфера	4.82	2.99	2.62
тропосфера	2.97	3.95	4.18
ошибка измерения	1.78	5.22	2.04
точность определения положения	15.11	36.49	16.65

Таблица 2. СКО источников ошибок измерений псевдодальности на С/А коде [м]

Передатчики НКА GPS излучают два непрерывных сигнала на частотах L1 и L2. GPS-приемник может принимать сигналы L1 на частоте  $f_{L1} = 1575,42 \text{ МГц}$  и модулировать псевдослучайный С/А код. Кодовое измерение псевдодальности на С/А коде есть результат сравнения генерируемой приемником псевдошумовой С/А последовательности и принимаемого кода.

НКА генерирует псевдошумовую С/А-последовательность синхронно с метками бортового времени. Уход бортового времени относительно системного передается в эфемеридных сообщениях. Поэтому временной сдвиг  $\Delta t$  между принятой и сформированной в приемнике последовательностями с точностью до начальной фазы сформированной последовательности определяет время прохождения сигнала от НКА. Один цикл передачи С/А-кода состоит из 1023 бит и повторяется 1000 раз в секунду, следовательно, зона однозначного измерения составляет  $D_{C/A} = 10^{-3} \cdot c = 299792.5 \text{ [м]}$ , где  $c$  – скорость света.

Под измерением псевдодальности на С/А коде понимают величину  $PR_{C/A}$ , удовлетворяющую следующему соотношению:

$$PR_{C/A} = n_i \cdot D_{C/A} + \Delta t \cdot c, \quad (1)$$

где  $n_i$  – целое число.

Приемник не может с высокой степенью точности определить момент регистрации измерения. Известна временная метка, в окрестности которой проведено измерение. Истинный момент регистрации измерения должен уточняться. Пусть  $(t - \Delta t)$  – истинный момент регистрации измерения псевдодальности  $PR^i_{C/A}$ , отличающийся от метки времени на величину  $\Delta t$ . Связь истинной дальности с псевдодальностью, измеренной от  $i$ -го НКА, описывается следующим соотношением:

$$PR^i_{C/A} = \left\| \vec{X}^i_{\text{НКА} \rightarrow \text{ПРИЕМНИК}}(t - \Delta t) \right\| + j \cdot c - \Delta t^i_{SV} \cdot c + \Delta D^i_{\text{ион}} + \Delta D^i_{\text{троп}} + x^i, \quad (2)$$

где  $\left\| \vec{X}^i_{\text{НКА} \rightarrow \text{ПРИЕМНИК}}(t - \Delta t) \right\|$  – истинная дальность до НКА или длина вектора, направленного от НКА в момент излучения к потребителю в момент приема сигнала;

$j$  – фазовый сдвиг псевдошумовой последовательности С/А-кода, вызванный различием в синхронизации часов НКА и приемника, один для всех измерений на момент времени  $t$ ;

$\Delta t^i_{SV}$  – уход часов НКА относительно эталонного времени GPS-системы;

$\Delta D^i_{\text{ион}}$  – ионосферная поправка измерения;

$\Delta D^i_{\text{троп}}$  – тропосферная поправка измерения;

$x^i$  – аппаратная ошибка измерения псевдодальности на С/А коде.

Невязки  $x^i$  будем интерпретировать как аппаратные ошибки измерения псевдодальности.

Для описания статистической модели ошибок измерений псевдодальности было взято семейство распределений с плотностью вида:

$$f(x, a_1, s_1, a_2, s_2, k) = k \cdot f_N(x, a_1, s_1) + (1 - k) \cdot f_N(x, a_2, s_2), \quad (3)$$

где  $f_N(x, a, s) = \frac{1}{\sqrt{2\pi}s} \cdot \exp\left(-\frac{(x-a)^2}{2s^2}\right)$  – плотность нормального распределения.

Поиск неизвестных параметров  $\{x, a_1, s_1, a_2, s_2, k\}$  распределения (3) проводился методом наименьших квадратов [4]. Проверка гипотезы о законе распределения проводилась с использованием критерия согласия хи-квадрат.

Найденные плотности распределения для каждой GPS-станции изображены на рис. 1, а их параметры представлены в таблице 3.

	$k$	$a_1$	$a_2$	$s_1$	$s_2$
bahr	0.335	-1.275	0.668	0.762	0.810
cena	0.295	-5.850	0.398	12.081	2.335
usno	0.110	-4.520	0.270	6.060	1.320

Таблица 3. Параметры распределения плотности ошибок измерений псевдодальности

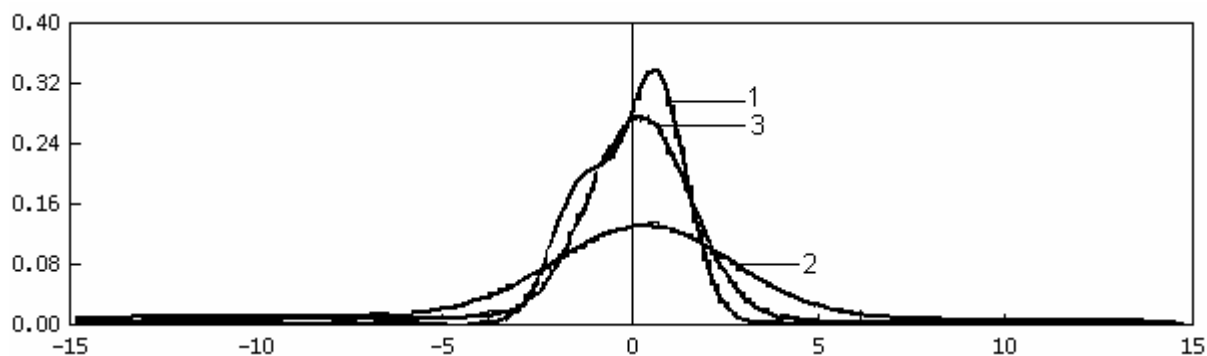


Рис. 1. Гистограмма и плотность распределения ошибок измерений псевдодальности 1) bahr, 2) cena, 3) usno

Из таблицы определения параметров распределения 3 видно, что процесс измерения псевдодальности различными типами приемников осуществляется по-разному. Это может объясняться наличием у приемников различных алгоритмов деления измеряемого бита C/A кода и различием в широте места расположения.

## Литература

1. Д.А. Тучин, Кодовые измерения псевдодальности системы GPS. Модель ошибок и априорная оценка точности определения вектора состояния, препринт Института прикладной математики им. М.В. Келдыша РАН, Редакционно-издательская группа, 2002, № 30.
2. Э.Л. Аким, Д.А. Тучин Апостериорная оценка точности определения вектора состояния земного наблюдателя по измерениям дальности и скорости системы космической навигации GPS, препринт Института прикладной математики им. М.В. Келдыша РАН, Редакционно-издательская группа, 2001, № 36.
3. Global Positioning System: Theory and Applications Bradford W. Parkinson, James J. Spiker Jr ISBN 1-56347-106-X
4. Ивченко Г.И., Медведев Ю.И. Математическая статистика: Учеб. пособие для вузов. — 2-е изд., доп. — М.: Высш. шк., 1992. — 304 с.: ил.

## **Подсистема имитационного моделирования локальных вычислительных сетей и инженерные основы ее реализации**

**Давыдов С.В., Солодовников В.И.  
Московский Государственный институт электроники и математики  
Кафедра ИТАС**

Подсистема имитационного моделирования для анализа работоспособности ЛВС предназначена для пользователей, не являющихся профессионалами в области сетевых технологий и имеющих небольшой опыт работы на персональном компьютере. Учитывая эти факторы, требования к подсистеме можно сформулировать следующим образом:

- простота и интуитивная доходчивость пользовательского интерфейса;
- возможность построения модели сети из заранее определенных блоков, назначение которых должно быть понятно из его схематического обозначения;
- возможность создания пользователем дополнительных элементов и/или модификации существующих, если требуемые элементы отсутствуют;
- возможность сохранения спроектированной модели и ее оперативной модификации, если среди хранящихся моделей существует подобная;
- возможность простого изменения входных параметров модели при проведении модельного эксперимента;
- наглядность в представлении результатов модельного эксперимента;
- возможность выдачи пользователю подсказок и рекомендаций в процессе проектирования компьютерной сети, если он нарушает основополагающие принципы.

Исходя из сформулированных требований в состав подсистемы анализа работоспособности ЛВС должны входить следующие модули (рис. 1.):

- Интерфейсный модуль;
- Модуль генерации модели;
- Моделирующая программа;
- База данных моделей объектов;

- База данных моделей ЛВС.

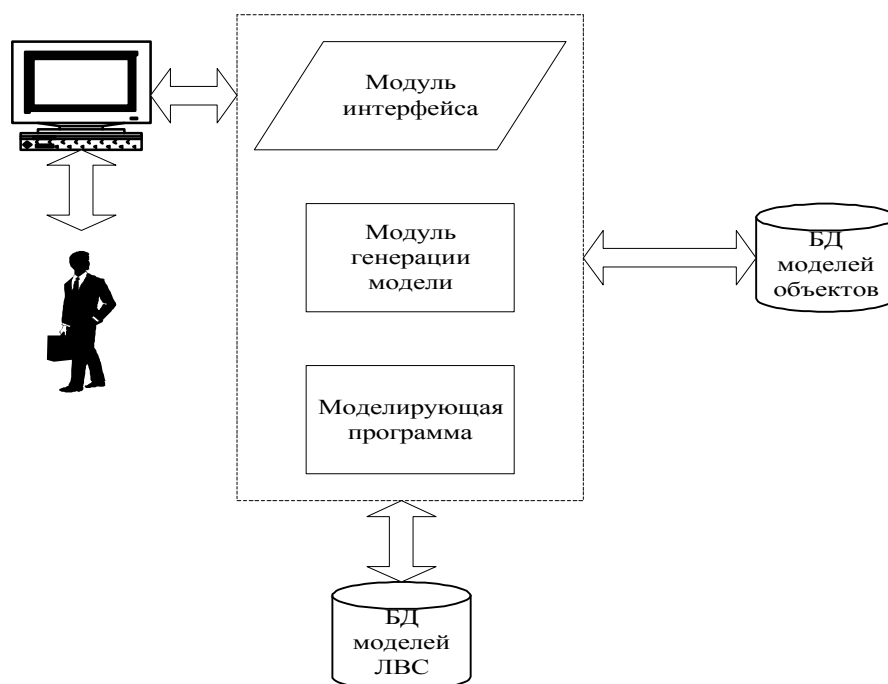


Рисунок 1. Функциональная схема подсистемы

Модуль интерфейса служит для обеспечения ввода информации в процессе построения модели и вывода результатов модельных экспериментов. На него также возлагается функция оповещения пользователя о несоответствии разрабатываемой модели принципам построения сети выбранной архитектуры.

Модуль генерации модели осуществляет преобразование данных, введенных пользователем с помощью интерфейсного модуля в вид, требуемый для моделирующей программы. Он подставляет конкретные значения атрибутов в модели объектов сети, хранящиеся в базе данных и по матрице связанности, полученной от модуля интерфейса, определяет связи между объектами модели.

Моделирующая программа предназначена для проведения модельного эксперимента и представляет собой реализацию управляющего моделирующего алгоритма. Она выполняет функции ведение календаря событий, активации блоков; контроля за блоками, готовыми к приему и передаче сообщений, и за наступлением момента окончания моделирования, анализа приоритета

сообщения на выходе блока и выбор наиболее приоритетного и прерывание обслуживания сообщения в блоке. Исходными данными для нее служат данные, полученные на этапе генерации модели.

База данных моделей объектов предназначена для хранения моделей объектов, которые могут быть использованы при построении модели ЛВС.

База данных моделей ЛВС служит для хранения уже созданных моделей. Ее использование связано с попыткой сокращения времени на моделирование, если моделирующий эксперимент на аналогичной сети уже когда-либо проводился, либо, если можно подобрать подобную модель и привести ее в соответствие требованиям заказчика путем незначительных изменений.

Выходными данными подсистемы являются результаты проведения модельного эксперимента, которые представляются в требуемой пользователем форме. Оценка результатов эксперимента возлагается на пользователя. При необходимости для оценки результатов может быть использована экспертная система, а сами результаты при положительной оценке их пользователем могут быть использованы для наполнения ее базы знаний.

Для генерации модели необходимо, чтобы была известна следующая информация:

- описание элементов сети;
- типы запросов (с частотами их проявления);
- описание структуры сети;
- требования пользователя по производительности сети.

В соответствии с этим база данных моделей сети должна содержать соответствующие отношения: таблица идентификаторов, таблица (матрица) связности, таблицы характеристик кабельных систем, сетевых адаптеров, концентраторов, коммутаторов и маршрутизаторов, таблица задач (запросов).

Компьютерную сеть определим как совокупность компьютеров и связей между ними. Для обеспечения локальных связей между компьютерами используются различные виды кабельных систем, сетевые адаптеры, концентраторы-повторители, мосты, коммутаторы и маршрутизаторы. Для



подключения локальных сетей к глобальным используются WAN-порты мостов и маршрутизаторов и аппаратура каналообразования для передачи данных по длинным линиям: модемы (для работы по аналоговым линиям) и устройства подключения к цифровым каналам (терминальные адаптеры сетей ISDN, устройства обслуживания цифровых выделенных каналов типа CSU/DSU и т.п.).

Исходя из этого выделим объекты, которые будут использованы как базовые элементы при построении модели сети и проведем их классификацию (рис. 2).

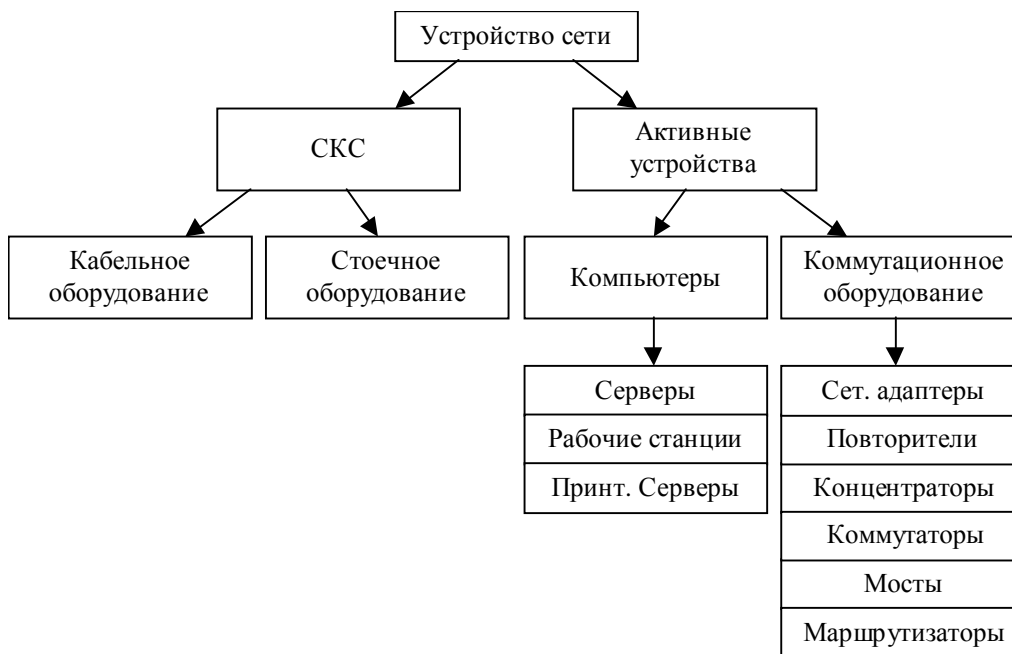


Рисунок 2. Классификация сетевых устройств

Каждый тип объектов имеет свое представление в базе данных. Сами же типы объектов в матрице связности представляются их идентификаторами (таблица 1).

Идентификатор	Наименование объекта	Имя таблицы
1	Кабельное оборудование	Cable
2	Сетевые адаптеры	NIC
3	Концентраторы	HUB
4	Коммутаторы	Switch
5	Маршрутизаторы	Router
6	Серверы	Server
7	Рабочие станции	Wstation
8	Принт-серверы	Pserv

Таблица 1 Идентификаторы объектов сети.

Из приведенной классификации (рис. 2) видно, что все сетевое оборудование можно разделить на активное и пассивное по следующему признаку: активные устройства сети для своей работы требуют подключения в сети электропитания. Они обеспечивают поддержание электрических параметров сигнала в требуемом заданном диапазоне.

Пассивные устройства не потребляют электроэнергию, а на электрические параметры сигнала оказывают влияние только с точки зрения их ухудшения. Пассивные устройства часто отождествляют со структурированной кабельной системой (СКС- Structured Cabling System, SCS), которая представляет собой набор коммутационных элементов (кабелей, разъемов, коннекторов, кроссовых панелей и шкафов), а также методику их совместного использования, позволяющую создавать легко расширяемые структуры связей в вычислительных сетях. СКС можно разделить на стоечное и кабельное оборудование.

К стоечному оборудованию относятся кроссовые панели, сетевые розетки, всевозможные коннекторы, коммутационные шкафы и стойки. В процесс функционирования компьютерной сети они практически не вносят никаких изменений. Их влияние на работу сети существенно только с точки зрения

вносимых ошибок, вызывающих нарушение работоспособности (плохой контакт, дребезг контактов и т.п.). Поэтому представлением этих объектов в модели сети является время отказа, определяемое как экспертная оценка пользователя.

К кабельному оборудованию относятся все типы кабелей, которые используются для построения локальных связей в вычислительных сетях в настоящее время. Исходя из номенклатуры кабельного оборудования и его параметров, влияющих на функционирование сети в целом можно использовать следующее информационное представление этих объектов в базе моделей:

Название	Тип
Cable_ID (идентификатор)	int
Cable_name (название кабеля)	Char[30]
Cable_type (тип кабеля)	Char[5]
Category (категория)	int
Speed (скорость передачи)	int
Material (материал изготовления)	Char[20]
Seg_len (максимальная длина сегмента)	int
Seg_point (максимальное количество компьютеров в сегменте)	int
Delay (задержка на единицу длины)	double
Cost (стоимость)	currency

Таблица 2. Таблица Cable.

В подкласс компьютеров входят всевозможные сервера (управляющие, баз данных и т.п.), рабочие станции и принт-серверы. Определим эти устройства.

Сервер представляет собой мощный компьютер, предназначенный для управления сетью, либо для хранения и обработки информации. Существует несколько типов серверов, ориентированных на разные применения: файл-сервер, сервер базы данных, принт-сервер, вычислительный сервер, сервер

приложений. Тип сервера определяется видом ресурса, которым он владеет (файловая система, база данных, принтеры, процессоры или прикладные пакеты программ).

Рабочая станция фактически является персональным рабочим местом пользователя сети (бухгалтер, программист, секретарь и т.п.). Чаще всего рабочая станция представляет собой персональный компьютер, подключенный к сети и имеющий доступ к каким-либо разделяемым ресурсам. Обычно в качестве программного обеспечения рабочие станции используют автоматизированные рабочие места (АРМ) различного назначения: АРМ бухгалтера, АРМ разработчика печатных плат, АРМ конструктора и т.п.

Принт-сервер является специальным устройством, к которому можно подключить принтер рабочей группы. С точки зрения сети он выполняет функции получения из сети данных, отправляемых на печать, обработки заданий печати, управление принтерами. По функциональному назначению он подобен рабочей станции сети, к которой подключен принтер.

Весь класс компьютеров с точки зрения анализа работоспособности сети интересен при построении ее модели, когда требуется произвести анализ функционирования сети на верхних уровнях модели OSI (сетезависимых). При этом, свойствами объектов класса компьютеров будут являться типы обрабатываемых и генерируемых запросов, частоты их появления и времена обработки. Все эти данные можно получить из предварительно накопленного опыта эксплуатации аналогичной сети, либо как экспертную оценку, задаваемую заказчиком.

Представление подкласса коммутационного оборудования рассмотрим на примере сетевых адаптеров. Сетевой адаптер (Network Interface Card, NIC) является периферийным устройством компьютера, непосредственно взаимодействующим со средой передачи данных, которое прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надежного обмена двоичными данными, представленными соответствующими электромагнитными

сигналами, по внешним линиям связи. Сетевой адаптер выполняет следующие функции:

- Оформление передаваемой информации в виде кадра определенного формата.
- Получение доступа к среде передачи данных.
- Кодирование последовательности бит кадра последовательностью электрических сигналов при передаче данных и декодирование при их приеме.
- Преобразование информации из параллельной формы в последовательную и обратно.
- Синхронизация битов, байтов и кадров.

Название	Тип
NIC_ID (идентификатор)	int
NIC_name (название)	Char[30]
NIC_speed (скорость передачи)	int
NIC_media (среда передачи)	Char[5]
Media_sense (автоопределение скорости передачи)	boolean
NIC_bus (компьютерная шина)	Char[5]
NIC_delay (задержка передачи)	double
MAC_addr (MAC- адрес)	Char[30]
Fault (вероятность отказа)	double
Team (возможность объединения)	boolean
Con_p (совместимость со стандартом IEEE802_1p)	boolean
Con_Q (совместимость со стандартом IEEE802_1Q)	boolean
Cost (цена)	currency

Таблица 3. Таблица NIC

Сетевые адаптеры различаются по типу и разрядности используемой в компьютере внутренней шины данных и по типу принятой в сети технологии (Ethernet, Token Ring и т.п.). Конкретная модель сетевого адаптера работает только по определенной сетевой технологии (например, Ethernet). Так как для каждой технологии имеется возможность использования различных сред передачи данных, то сетевой адаптер может поддерживать одну или несколько сред.

Так как сетевые адаптеры работают на физическом уровне и подуровне MAC канальном уровня модели OSI, то важнейшими характеристиками для их описания являются значения параметров, определяющие взаимодействие с устройствами, работающими на этих уровнях и сетевая технология по которой работает адаптер. Информационное представление сетевых адаптеров включает эти параметры (табл. 3).

Рассмотрим и определим типы информации, которые будут рассматриваться при проведении модельного эксперимента. Будем различать следующие типы запросов:

- Запрос типа «клиент-сервер», или запрос на обработку.
- Запрос типа «клиент-файл», или запрос к данным.
- Запрос на печать представляет собой запрос на передачу данных.

Название	Тип
ID (идентификатор)	
Req_name (название запроса)	
Req_type (тип запроса)	
Out_addr (адрес исходящий)	
In_addr (адрес входящий)	
Req_freq (частота появления запроса)	
Len_data (длина передаваемых данных)	
Req_time (время выполнения)	

Таблица 4. Таблица Request

Указанные типы запросов используются в модели сети, в случае, когда необходимо произвести оценку ее работоспособности с точки зрения работы пользователей с программными приложениями, т.е. если рассматриваемая компьютерная сеть моделируется на верхних уровнях модели OSI. Информационное представление запросов в базе данных подсистемы моделирования приведено в табл. 4. Временные должны быть определены как экспертная оценка заказчика.

В заключении хочется отметить что на основе анализа потребностей пользователя и предложенного подхода к построению имитационных моделей сети разработана функциональная схема подсистемы и на основе анализа типовых ЛВС определен круг объектов сети, которые принимаются как базовые при ее моделировании. Для этих объектов, по результатам рассмотрения принципов их функционирования и критическим параметрам, построено информационное представление в базе данных моделей объектов.

#### ЛИТЕРАТУРА

1. Блэк Ю. Сети ЭВМ: протоколы. Стандарты, интерфейсы/ Пер. с англ. – М.: Мир. 1990. 510 с.
2. Науманн Ш., Вер Х. Компьютерная сеть. Проектирование, создание, обслуживание. / Пер. с нем. – М.: ДМК 2000. 336 с.
3. Тиори Т., Фрай Дж. Проектирование структур баз данных: в 2-х кн. / Пер. с англ. –М.: Мир. 1985 кн.1 - 287с., кн.2 – 320 с.
4. Шрайбер Т. Дж. Моделирование на GPSS. / Пер. с англ. – М Машиностроение, 1980. 592 с.
5. Data Communications Networks, Open System Interconnection (OSI) System Description Techniques, Volume VIII, Fascicle VIII.5, CCITT Red Book, CCITT Plenary Assembly, October 8-19, 1984 (Malaga-Torremolinos).

## Управление деятельностью выпускающей кафедры

Г.М. Рудакова, Т.Н. Барина, Андриянова, С.Н. Клименок, О.В. Рожкова  
Сибирский государственный технологический университет  
Кафедра информационных технологий  
[gmr@fait.krs.ru](mailto:gmr@fait.krs.ru), [ck@fait.krs.ru](mailto:ck@fait.krs.ru)

Автоматизация деятельности кафедры является актуальной, так как основная задача ВУЗа - подготовка и выпуск специалистов, решается на кафедре. Разрабатываемая система обеспечит удобные и простые механизмы работы с данными, начиная от формирования исходной базы и заканчивая статистической обработкой и формированием отчетов и рекомендаций, проведение оперативного контроля [1]. Система позволит эффективно организовать деятельность кафедры, откроет возможность иметь полную информацию по обучению на кафедре, оценить перспективные направления деятельности специалистов и сотрудников, востребованность их на рынке труда, заведующему кафедрой оптимально принимать управленческие решения.

На кафедре информационных технологий СибГТУ возникла необходимость в интенсификации работ по освоению современных технологий разработки систем организационного управления, информационных систем и программных средств [3]. Эта необходимость обосновывается следующими факторами:

- существующие системы (Word, Excel), используемые для документационного обеспечения кафедры, не обеспечивают всех функций делопроизводства и не достаточно удобны для пользования;
- расширение спектра предоставляемых ВУЗом образовательных услуг требует от кафедры более быстрой, эффективной и точной обработки информации;
- рост объема анализа информации требует обеспечение удобных и простых механизмов работы с данными, начиная от формирования исходной базы, состоящей из множества справочников, и заканчивая статистической обработкой и формированием отчетов;



- существование необходимости организации более эффективного обмена данными с другими подразделениями ВУЗа;
- существование необходимости налаживания четких взаимосвязей документов с персоналом и между собой;
- существование необходимости приведения всех документов одного типа к единой форме, с учетом существующей нормативно-справочной документации ВУЗа;
- существование необходимости обеспечения быстрого и удобного поиска документов;
- обеспечение оперативности доведения распоряжений и усиление контроля за их исполнением;
- проводимые в СибГТУ работы по созданию системы качества образования требуют новых, современных подходов к разработке системы делопроизводства кафедры, которая должна обеспечивать информационную поддержку управления данным подразделением;
- необходимость развития научно-производственной базы СибГТУ и кафедры, как подразделения ВУЗа, в области информационных и управленческих технологий;
- обучение групп специалистов, способных выполнять заказы в области управленческого консалтинга и информационных технологий.

Обоснованием разработки служит необходимость создания программного продукта, обеспечивающего решение вышеперечисленных задач[3]. Не решая всех проблем кафедры, данный проект, тем не менее, открывает важнейшие перспективы, связанные с совершенствованием системы управления, информационной системы и информационных технологий.

Управление рутинной работой включает выполнение каждодневных задач – оперативных задач. Такие задачи являются более или менее регулярными и не сложными, но они составляют основу всей работы, обеспечивая правильное функционирование[2]. Исполнение оперативной задачи – это последовательность выполнения действий, которые продолжаются в течении

некоторого времени и могут выполняться разными сотрудниками. Система позволит не только структурировать деятельность кафедры, но и отслеживать планирование основных задач, контроль их исполнения, а также занятость сотрудников в выполнении цели, т.е. фиксировать изменения внесенные сотрудником, даты и времени его работы.

Применение реляционного подхода к проектированию баз данных, технологии "клиент-сервер", позволит обеспечить независимость хранимых данных от использующих их программ, выявить и устранить дублирование информационных объемов.

### **Библиографический список**

1. Барина Т.Н., Клименок С.Н., Рожкова О.В. и др. Система менеджмента качества выпускающей кафедры//Теоретические и прикладные вопросы современных информационных технологий, Материалы третьей конференции. - Улан-Удэ, 2002, стр.295-300;

2. Рудакова Г.М., Клименок С.Н. Построение и анализ моделей деятельности текущего и целевого состояния выпускающей кафедры // Проблемы информатизации региона ПИР-2001. - Красноярск: КГТУ, 2001;

3. Клименок С.Н. Системный анализ в построении модели управления деятельностью выпускающей кафедры // Труды второго Всесибирского конгресса женщин-математиков. - Красноярск, 2002, стр. 94-95.

## **Использование виртуальных приборов в учебном процессе и проектировании**

**Иванов С.Ю., Конопатченков А. (МГИЭМ)**

Изучение физических, производственных и других процессов требует наглядности и возможности манипулирования параметрами для их понимания. Для учебного процесса и научной работы необходимы лабораторные установки, материалы, демонстрационные стенды, контрольно-измерительные приборы. Будучи товарами высокотехнологичными, стоят все эти устройства дорого. Многие технические ВУЗы не могут себе позволить обновление лабораторного инструментария, лишь изредка приобретая жизненно необходимое и ремонтируя своими силами имеющееся в наличии.

С другой стороны, развитие современных вычислительных средств, привело к повсеместному использованию компьютеров практически во всех сферах человеческой деятельности. Моральное старение компонент вычислительных систем позволяет приобретать компьютеры по сравнительно недорогим ценам.

Руководствуясь этими двумя посылками, закономерно возникает идея о замене с помощью компьютера части дорогостоящего оборудования. Для этого нам нужно решить задачу представления информационных потоков аналоговых сигналов в цифровом виде. Компьютер не может измерить и обработать аналоговый сигнал без аналого-цифрового преобразователя (АЦП). Одной из самых удачных реализаций АЦП является плата сбора данных (ПСД), реализованная для архитектуры IBM PC. У ПСД имеются в наличии входы, соединяя которые с датчиками можно ввести аналоговый сигнал и преобразовать его в понятный компьютеру цифровой сигнал. Программное обеспечение, используемое для работы с ПСД, позволяет многократно упростить процедуру взаимодействия прибора и пользователя вследствие ряда преимуществ:

- настраиваемый пользовательский интерфейс, удобный оператору;
- богатейшие возможности по представлению информации;
- возможность адаптации к условиям измерения;
- учет статистики предыдущих измерений;

- работу в реальном времени для быстропротекающих процессов;
- размеры записи сигнала в реальном времени практически не ограничены;
- расширяемый специализированный анализ;
- язык интерфейса можно быстро изменить на нужный;
- быстрый отклик на изменения условий в процессе управления или измерения.

Программная часть виртуального прибора может эмулировать переднюю управляющую панель стационарного измерительного устройства. Сама панель, сформированная на экране дисплея, становится панелью управления виртуального прибора. В отличие от реальной панели управления стационарного прибора такая виртуальная панель может быть многократно реконфигурирована в процессе работы. Пользователь виртуального прибора активизирует объект графической панели с помощью мыши, клавиатуры или прикладной программы.

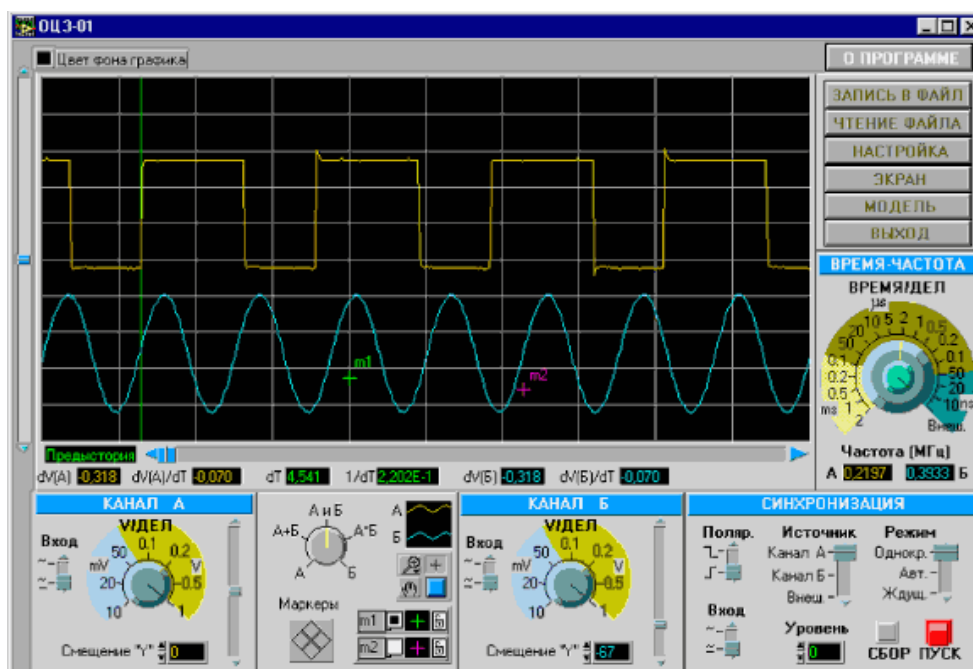
Несмотря на то, что стационарные приборы старались оснастить дисководами, улучшенными дисплеями и т.д., получить на них те же результаты, что и с использованием ПК, оказалось невозможным. Кроме того, появление нового стандарта сменных НГМД на 100 Мбайт и 1 Гбайт (применим и на портативных компьютерах) позволяет говорить о виртуальном эксперименте, для проведения которого необходимо только наличие компьютера, а все остальные программно-аппаратные средства подбираются, исходя из технических требований проводимого эксперимента.

Оснатив свой компьютер ПСД, получаем и осциллограф, и спектроанализатор, и вольтметр, и генераторы самых разных применений. Характеристики этих виртуальных приборов определяются только техническими параметрами выбранной ПСД.

Рассмотрим более подробно виртуальные приборы, реализованные на ПСД с АЦП: осциллограф, спектроанализатор и вольтметр.

**Виртуальный осциллограф** предназначен для электро- и радиоизмерений в различных областях науки и техники при проведении исследовательских

испытательных работ в лабораторных и производственных условиях. Лицевая панель осциллографа показана на рисунке



### Лицевая панель осциллографа

Сочетание виртуального осциллографа и персонального компьютера открывает новые возможности, недостижимые для стационарных аналогов. Работа с файлами позволяет документировать измеряемые процессы, сравнивать сигналы с образцовыми и отображать сигналы, созданные самим пользователем в его программах (чтение из файла), производить статистическую обработку сохраненных в файл экспериментальных данных.

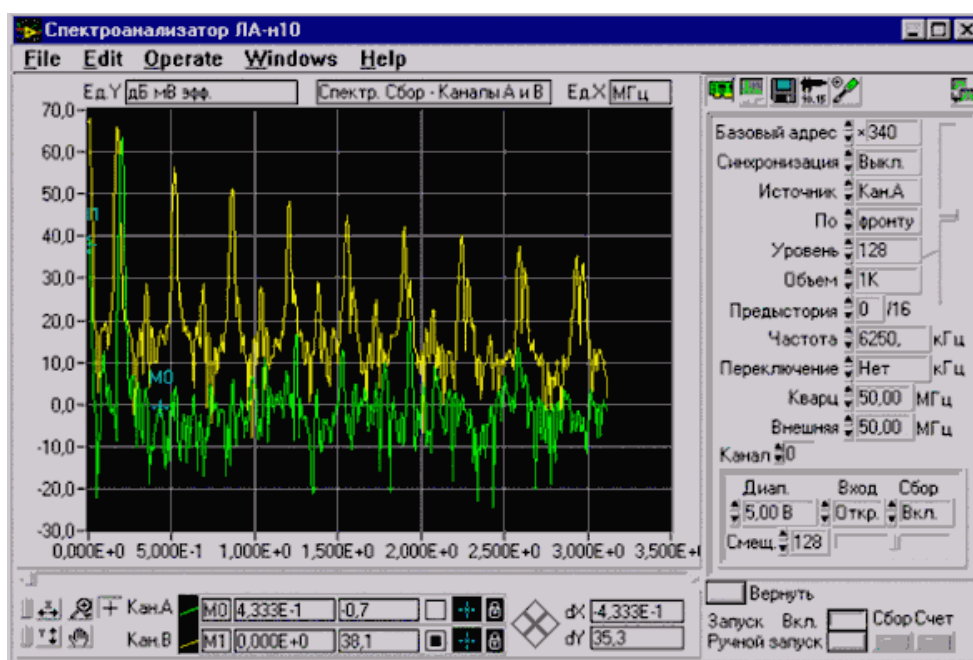
При работе с виртуальным осциллографом по сравнению с обычным электронно-лучевым осциллографом есть существенные преимущества:

- Высокая точность измерения временных и амплитудных интервалов;
- Широкая полоса пропускания за счет возможности работы виртуального осциллографа в режиме стробоскопа (в режиме стробоскопа достигается эквивалентная частота дискретизации свыше 1ГГц);
- Яркое, хорошо сфокусированное изображение на экране на любом времени развертки;
- Возможность отображения сигнала до момента запуска (предыстория - отрицательное время);

- Возможность запоминания данных на экране в течение произвольного времени;
- Возможность детектирования импульсных помех;
- Возможность работы в составе автоматизированных измерительных систем;
- Возможность статистической обработки сигнала;
- Возможность исследования переходных процессов;
- Считывание предварительно записанных данных;
- Возможность сравнения предварительно записанных данных с текущими.

**Спектроанализатор** представляет собой измерительную систему, которая без широкополосного приемника дает возможность получить автоспектры и частотные характеристики исследуемых сигналов в полосе частот до 25 МГц, синхронно по двум каналам, и автоспектр до 50 МГц по одному каналу при частотном разрешении 400 Гц и 800 Гц, соответственно. Время обзора полосы пропускания системы 12 мс.

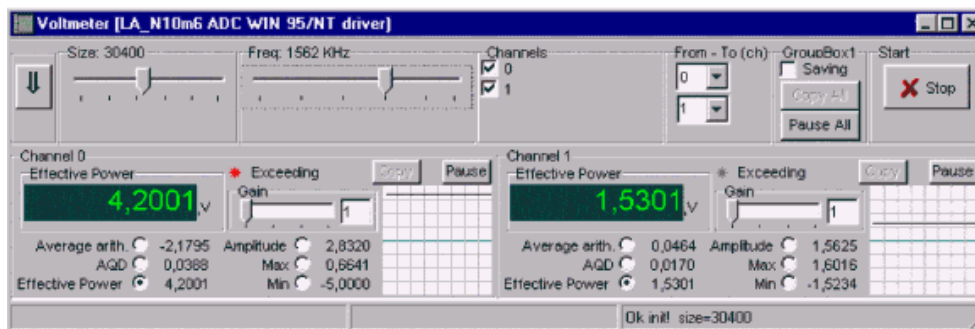
На рисунке изображена лицевая панель анализатора, где расположены управляющие элементы обеспечивающие задание режима измерения и визуализации результатов.



**Анализатор спектра**

Данная измерительная система может быть полезна для мониторинга каналов радиосвязи систем безопасности, телефонных систем сотовой связи, наличия электромагнитных излучений в системах экологического контроля.

**Виртуальный вольтметр** предназначен для измерений в различных областях науки и техники при проведении исследовательских испытательных работ в лабораторных и производственных условиях. Лицевая панель универсального вольтметра показаны на рисунке.



### Внешний вид универсального вольтметра

Сочетание виртуального вольтметра и персонального компьютера открывает новые возможности, недостижимые для аналогового прибора. Запись в файл позволяет документировать измеряемые процессы, производить статистическую обработку сохраненных в файл экспериментальных данных.

При работе с виртуальным вольтметром по сравнению с обычным аналоговым прибором есть существенные преимущества:

- *Высокая точность измерения;*
- *Широкая полоса пропускания;*
- *Визуализация измеряемого напряжения (истории измерения) на экране для каждого канала;*
- *Возможность запоминания данных в файле;*
- *Возможность работы в составе автоматизированных измерительных систем.*

Для удобства работы пользователя измеряется одновременно целая группа параметров сигнала:

- *Мгновенное значение измеряемого напряжения.*
- *Среднеквадратическое значение измеряемого напряжения.*

- *Среднеарифметическое значение измеряемого напряжения.*
- *Эффективное значение измеряемого напряжения.*
- *Амплитудное значение измеряемого напряжения.*
- *Максимальное значение измеряемого напряжения.*
- *Минимальное значение измеряемого напряжения.*

Из дополнительных возможностей этого прибора хочется отметить.

- *Индикацию превышения допустимого входного напряжения ( $\pm 10V$ ).*
- *Гальваническая развязка между измеряемым устройством и компьютером 400В.*
- *Возможность одновременно использовать до 8 каналов.*
- *Переключаемый коэффициент усиления.*
- *Возможность измерения периодического сигнала частотой до 99кГц.*

Кроме существующего ПО у пользователя есть возможность с помощью современных средств программирования создавать виртуальные приборы с необходимыми специфическими функциями. Эта возможность намного расширяет спектр применения ПСД. Таким образом, используя только одну ПСД и соответствующее ПО можно получить несколько приборов на одном компьютере, что бесспорно выгодно экономически.

Важной особенностью использования виртуальных приборов является сохранение полученных данных в виде наборов и размещение их в специализированной базе данных. Это позволяет использовать виртуальные приборы при разработке программно-аппаратных измерительных комплексов на ранних стадиях проектирования и производить предварительные оценки характеристик проекта, а так же вести разработку без наличия дорогостоящего оборудования. Интересным является соединение моделей комплексов с виртуальными приборами в качестве нагрузки или источника данных.



## **Обучающая система «Исследователь»**

**Ляпин Д. А.- учащийся 11 класса гимназии 1516;**

**Комалов С. С. – научный руководитель, доцент кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э.Баумана**

Программа «Исследователь» - это система программирования, выполненная в форме игры. «Кому сейчас это нужно?»- спросите Вы. Ведь есть масса мощных сред разработки, способных создать программу любой сложности. Но представьте школьника, лет 11-ти, который только что наигрался в компьютерные игры и решил, что теперь хочет писать программы сам. Освоить профессиональный язык программирования, например, Basic или Turbo Pascal, ему будет нелегко. И не потому, что мальчику или девочке 11-ти лет трудно понять, что такое «процедура», «функция», «переменная» и т.д., а потому, что голубой экран с мигающим курсором может показаться человеку, привыкшему играть в «настольные» игры, слишком скучным, и желание научиться программировать пропадёт.

«Исследователь» - это наполовину игра, наполовину язык программирования. От игры здесь исполнитель–машинка, которая движется по условному «полю»-карте и выполняет некоторый набор заданных допустимых действий. А от языка программирования то, чтобы управляя машинкой довести ее до финиша, алгоритмически описывая образ программы, с помощью которой эта задача может быть решена. Необходимость написания программы полностью отпадает, что должно, по моему мнению, активизировать интерес к освоению алгоритмического языка и написанию «собственных» программ решения стоящих задач. Игровой элемент является обязательным стимулом для младших школьников в освоении новой предметной области знаний.

Игровые элементы нужны для того, чтобы человеку было проще и интереснее общаться с системой. Пятикласснику будет гораздо понятнее задание, звучащее «возьмите ключ в правой части карты, затем откройте им ворота и поезжайте к финишу через болото», чем «составьте алгоритм вывода

на экран всех целых чисел в заданном интервале». В «Исследователе» задача, которая ставится перед пользователем, всегда ясна: например, «привезти продовольствие в голодающую деревню», либо «приехать к финишу за минимальное время или по кратчайшему расстоянию» и др. Естественно, мальчику или девочке младших классов понятнее, зачем надо привезти продукты в голодающую деревню, чем то, зачем надо вывести на экран числа.

Итак, задача, которую я поставил перед собой – это развитие логического мышления и попытка привить интерес к программированию у детей в возрасте от 9 до 12 лет. Реализована эта задача в совокупности игры, игрового интерфейса и непосредственно программирования. Все игровые элементы – это некая «приманка», т.е. то, что делает работу с системой понятнее и интереснее. А составление программы – это самое главное – то ядро, которое даст ребёнку возможность понять важнейшие элементы программирования, научиться логически мыслить.

Как работает эта система? В «Исследователе» есть набор уровней: начинаются они с самых простых, обучающих, и заканчиваются весьма сложными, требующими хорошо развитого логического мышления. Каждый уровень состоит из карты (игрового поля) и задания, которое машинка должна выполнить, передвигаясь по карте. Для того, чтобы пройти уровень, пользователю надо визуальнo изучить карту, предложенное задание и, составив алгоритм решения поставленной задачи, запрограммировать машинку. После запуска составленной программы машинка начинает двигаться по карте и последовательно выполнять заданные ей команды. После того, как она выполнит все запрограммированные пользователем действия, появится сообщение, выполнены ли все задания данного уровня и насколько рационально написана программа. Чем меньше действий сделала машинка для достижения цели и чем короче программа, тем лучше пользователь продумал алгоритм решения поставленной задачи.

Программа пишется на русском языке, способ программирования несколько похож на конструкции языка Паскаль. Пользователь может

оперировать набором из 7-ми команд для машинки (вперёд, назад, влево, вправо, проверить, нагрузить, выгрузить) и такими средствами оперирования, как циклы (детерминированный и итерационный), условный оператор, составной оператор, создание собственных процедур.

Карта, по которой движется машинка, состоит из различных элементов, таких, как стена, ворота, ключ, болото и др. К примеру, машинка может ехать по траве, но не может проехать сквозь стену, или, если Вы не подобрали ключ, то не откроете ворота. Если для прохождения какого-нибудь участка приходится выполнять несколько раз один и тот же набор команд, следует использовать цикл с заданным количеством повторений «повтори». Для реализации условного оператора на карте существуют псевдополя. Псевдополя – это такие элементы карты, которые имеют один вид, но разные свойства. Используется это следующим образом: например, машинке надо проехать через полосу болот к финишу, и известно, что в этой полосе есть одна ячейка псевдоболота, через которое машинка может проехать. На вид псевдоболота от болота отличить нельзя, так что для того, чтобы найти нужную ячейку, нужно использовать команду «проверить», условный оператор и цикл «пока». Или другой пример: нужно привезти ящик с продуктами на пустую площадку, а ящика на карте два, и известно, что в одном из них бомба. Для того, чтобы погрузить нужный ящик и не взорваться на бомбе, нужно использовать команду «проверить» и условный оператор. Также в системе существует переменная, но она всего одна – это память машинки. В ней может содержаться какой-нибудь элемент карты. Поместить в память этот элемент можно командой «проверить». Именно память машинки используется в условном операторе и итерационном цикле «пока». Если в разные моменты выполнения программы приходится выполнять одну последовательность команд, например, развернуть машинку (для этого надо выполнить две команды «вправо» и «влево», либо «влево» и «влево»), есть смысл создать процедуру (в данном примере вместо того, чтобы каждый раз писать «вправо» и «влево», достаточно будет написать «разворот»).

Несмотря на кажущуюся простоту, работа с системой является очень хорошим упражнением для развития логики мышления. Для прохождения некоторых уровней требуется сообразительность и острота ума: важно увидеть, как рациональнее и лучше составить алгоритм решения задачи; увидеть и правильно выбрать те элементы, которые удобно объединить в процедуру; найти повторяющиеся фрагменты; продумать такой алгоритм, чтобы машинка выполнила наименьшее количество действий (шагов). Так что поупражняться в прохождении сложных уровней, я считаю, было бы полезно не только ребёнку, но и взрослому.

В «Исследователе» открыт простор для фантазии. Пользователь может не только проходить те уровни, которые «жестко» заложены в программу, но и создавать свои собственные. Для этого в системе предусмотрен специальный редактор уровней. Очень полезным и увлекательным упражнением для мальчика или девочки было бы составление своей карты, задания к ней, а затем написания программы для прохождения созданного маршрута. А поскольку фантазия у детей развита очень хорошо, уровни, я думаю, получились бы очень интересные и сложные.

Как и где можно использовать систему? Прежде всего, её можно использовать в школе, на уроках информатики в младших классах. После того, как дети научатся работать с программой, учитель мог бы устроить, к примеру, такие соревнования: кто создаст наиболее сложный уровень и составит алгоритм прохождения этого уровня; кто напишет лучшую программу прохождения какого-нибудь очень сложного уровня. Принцип состязательности был бы весьма полезен и уместен, так как «Исследователь» - это частично игра.

Вне школы с этой системой может легко разобраться каждый, сидя дома за компьютером. В «Исследователе» есть очень подробный и несложный встроенный самоучитель. Он состоит из двенадцати глав или, точнее говоря, уроков. Первые уроки достаточно простые, в них описывается, как должна выглядеть простейшая программа и как заставить машинку передвигаться. В

следующих главах пользователь ознакомится с различными элементами программирования, такими, как циклы, процедуры и др. Кроме описания теории, в каждой главе есть пример прохождения уровня на рассматриваемую тему и задание: выполнить несколько уровней самостоятельно. Прочитав последовательно все главы и написав программы для всех обучающих уровней, пользователь будет полностью ознакомлен с системой и может приступать к реальному практическому программированию.

Надеюсь, что у школьника, разобравшегося с «Исследователем», возникнет интерес к программированию, и он начнёт писать программы на настоящих языках. Теперь сделать это будет намного проще, так как с наиболее важными принципами написания программ он уже знаком. Если пользователь запустит «Исследователь» для того, чтобы поиграть, а затем ему захочется разобраться с системой и он обнаружит свои способности и интерес к программированию, то можно считать, что цель будет достигнута.

Создание программы «Исследователь» - это моя первая попытка дать неподготовленному пользователю возможность проверить свои способности и логическое мышление. Возможно дальнейшее развитие этой идеи и создание более мощных обучающих программ, над чем я и работаю в настоящее время. Кроме того, я планирую провести пробное тестирование системы для групп школьников 4-6 классов нашей гимназии.

Программная часть системы написана на Delphi 6 и никаких дополнительных ресурсов, кроме WINDOWS/95/98/2000 при работе с программой не требуется.

# **Организация хранения, обработки и доступа к полнотекстовым документам в современных АБИС**

**Григорьев А.С.**

**МГТУ им. Баумана, г.Москва, 2-я Бауманская, дом 5, библиотека МГТУ**

## **Введение**

В настоящее время проблема создания электронного каталога как альтернативы бумажному (карточному) большей частью решена. Много делается для облегчения и ускорения работы сотрудников библиотек.

Так как библиотека существует для удовлетворения запросов третьего заинтересованного в результате лица - читателя, решению задачи предоставления именно этому пользователю системы новых возможностей, которые помогут ему удобнее и быстрее получить интересующую его информацию, уделяется внимание при проектировании современных библиотечных систем. Следующий этап в развитии библиотеки - исследование и организация новых форм хранения документов, а также способов предоставления их пользователям.

Создание фондов электронных документов и предоставление читателям доступа к этим ресурсам позволяет, в частности расширить круг читателей, получающих доступ к редким документам.

Хранение документов в электронном виде позволяет проектировать в библиотечной системе предоставление возможности полнотекстового поиска по содержимому электронных документов, независимо от двоичного формата документа.

## **Фонды электронных документов**

Основной объект хранения и предоставления услуг классической библиотеки - документ, как правило, бумажный носитель. Современная библиотека имеет дело не с документом, как таковым, а с информацией, которую он содержит. Документом, который хранит и обрабатывает библиотека, становится любое материальное представление информации.

Следовательно, для библиотеки становится актуальной возможность поддержки различных форм хранения документов. К привычной бумажной добавляется электронная форма хранения. Среди возможных форм наибольший интерес представляет именно она в связи с удобством компактного хранения двоичной информации и передачи её на большие расстояния.

Электронная форма хранения документа допускает различные форматы кодирования информации. Это могут быть текст, изображения, аудио или видео ролики.

Существует большое число определений электронного документа. Но все они пресекаются в одном: электронный документ - это закодированная в двоичном виде информация, предназначенная для обработки специально созданной программой для каждого конкретного формата хранения. Могут быть использованы произвольные пользовательские форматы, если они допускают возможность воспроизведения закодированной информации и предоставления её пользователю.

При решении поставленной задачи требуется охватить все многообразие документов, которые могут стать объектом обработки библиотечной системы.

Например, электронный документ может быть:

- а) зарегистрированным (электронная версия книги, которой был присвоен ISBN);
- б) незарегистрированным (автореферат диссертации, конспект лекций, аудиозапись конференции).

С точки зрения библиотековеда (как и с юридической) это принципиально различающиеся документы. Первый будет обработан по классическим библиотечным технологиям, а второй может даже и не оказаться в списке документов, предоставляемых библиотекой читателю, пока этот документ не будет зарегистрирован. Для определения статуса незарегистрированных документов было введено понятие (юридический термин) «выхода в свет».

Под выходом в свет (опубликованием) электронного объекта законом подразумевается [1] «предоставление экземпляров документа с согласия автора

неопределённому кругу лиц» (в том числе, записью на электронный носитель или выпуском печатного текста). То есть, документ считается вышедшим в свет, если к нему имеют доступ пользователи, которым он интересен.

Таким образом, будем считать документы, не зарегистрированные в книжной палате и не получившие индивидуальный книжный номер, вполне пригодными к размещению в фондах библиотеки и предоставлению пользователю (с разрешения автора). Это точно такие же документы, и обращаться с ними надо точно так же, как и с зарегистрированными. Значит, документы и того и другого типа будут включаться в создаваемые электронные библиотечные фонды. Видео ролик лекции наряду с бумажным конспектом займет место в фондах библиотеки с той лишь разницей, что типом документа будет "видеофильм" вместо "бумажный документ".

Велика вероятность, что со временем всё растущая доля электронных документов в потоке ресурсов, пополняющих фонды библиотеки, превысит поступления традиционных бумажных документов.

#### **Доступ читателей к редким документам**

Библиотека хранит большой объём традиционных документов. Случается, что количество некоторых из них не соответствует запросам читателей. Требуется большее число копий. Эту задачу легко решить, имея электронную копию бумажного документа. Современная библиотека может пополнять фонд электронных документов, квалифицированно создавая эти копии. Такой подход решит проблему, так как количество этих копий не ограничено. Переведенная в электронный вид информация, содержащаяся в традиционных документах, становится электронным документом. Работу с новым представлением информации должна продолжать обеспечивать современная библиотека.

Очевидно, что при создании электронного образа бумажного документа присутствует элемент творчества библиотекаря. При этом может возникнуть проблема расхождения форматов и, что намного серьезнее, структуры создаваемых документов (разбиение на отдельные файлы, оформление служебной информации, как то обложка, содержание или приложения). Во



избежание подобной неоднозначности необходимо создать некий документ, описывающий внутрибиблиотечный стандарт, в соответствии с которым будут оформляться создаваемые документы.

### **Полнотекстовый поиск по содержанию электронного документа**

Основная потребность читателя - это получение документов, носителей информации, функцию хранения которых выполняет библиотека. Предоставить читателю доступ к интересующим его документам - задача библиотеки. Для этого необходимо решить задачу поиска нужного документа.

Как правило, читатель не знает точно, какой именно документ ему требуется. Он может лишь описать интересующую его тему, перечислить набор специфичных для неё терминов.

После этого обычно производится поиск только по библиографическому описанию документа. При этом список предложенной литературы может оказаться не полным, или, наоборот, избыточным.

Максимальная релевантность поиска может быть достигнута только при поиске по содержанию документа. Теперь, когда появилась такая возможность, реализация данного механизма необходима в рамках современной библиотечной системы.

### **Организация хранения электронных документов**

Библиотечная система должна поддерживать полный жизненный цикл электронного документа. Она должна позволять получать, регистрировать, хранить, перерабатывать документ и предоставлять его пользователю. При этом необходимо минимизировать возможные исключительные ситуации (как то, недоступность запрашиваемого документа или ошибки при обращении к нему). Если же таковые происходят - предусмотреть корректную их обработку.

В общем случае внутренняя структура документа (расположение и значение полей) не известна. Библиотечная система должна позволять работать как с линейным, так и с ассоциативным или сетевым представлением информации в пределах документа, который может состоять из нескольких взаимосвязанных файлов. Это потребует допустить или реализовать

возможность организации ссылок между частями документа. Так же не следует исключать из рассмотрения обращение из одного обрабатываемого файла к другому, также находящемуся в хранилище системы.

Формальным эквивалентом электронного документа можно считать файл в случае линейного хранения информации, либо набор связанных файлов в общем случае. Таким образом, решение поставленной задачи можно свести к созданию системы, позволяющей производить требующиеся в библиотечном документообороте операции по манипулированию документами к операциям над файлами.

Сперва, необходимо определить способ хранения документов, так как остальные функции разрабатываемой системы будут ориентированы на использование функции хранения.

Эта функция допускает следующие способы реализации хранения документов:

а) в файловой системе (в обычном дереве каталогов, где за хранение и адресацию документа отвечает файловая система);

б) в таблице базы данных (как двоичные поля, содержащие все тело файла целиком).

Вне зависимости от того, какой подход будет использоваться, документ должен пройти дополнительную обработку перед размещением в хранилище. Поэтому потребуется создание отдельного модуля загрузки документа, будь это размещением в обычной файловой системе, или в базе данных. Ни в том, ни в другом случае документ не может быть просто «выложен» в хранилище – он должен быть, как минимум, зарегистрирован и привязан к записи своего описания.

Модуль выгрузки при использовании первого подхода к хранению документа должен реализовывать только «наблюдательную» функцию (как то, например, сбор статистики или контроль за санкционированностью доступа), так как загрузка будет осуществляться напрямую из доступной папки.

При использовании второго подхода модуль выгрузки кроме осуществления тех же функций, что и в первом случае, также должен «на лету» по запросу извлекать требуемый файл из базы и выводить его пользователю так, чтобы не было никакой разницы с загрузкой напрямую из файловой системы. Соответственно, на модуль выгрузки ложится ответственность и за обработку исключительных ситуаций при этих обращениях.

Обращение к требуемому документу напрямую, с указанием в точности его характеристик может составлять лишь малую долю всех запросов пользователей. Чаще всего при обращении в библиотеку читатель не знает, какие именно документы ему необходимы. Есть лишь интересующая пользователя тема и, иногда, набор специфичных терминов, которые могут быть введены в шаблоне для поиска. Именно этот набор может помочь пользователю получить искомый документ (или набор документов, релевантных запросу), если система предоставит ему возможность поиска по тексту документов. Для осуществления этого сервиса должен быть создан модуль поиска по текстовым образам документов, если таковые возможно создать.

Первый подход к хранению документа привлекателен тем, что в СУБД Microsoft® SQL Server 2000, которая используется при создании системы автоматизации библиотеки МГТУ имени Н.Э.Баумана, поддерживает функцию полнотекстового поиска.

В составе SQL Server 2000 имеется служба Full-Text Search, предоставляющая эффективные механизмы поиска. В данной системе возможно выполнять поиск не только в обычном тексте, но и в отформатированных материалах, сохраненных в полях для больших двоичных объектов (BLOB - Binary Large Object Block). В распоряжении разработчика имеются интерфейсы доступа к информации, обеспечивающие поиск нужных данных.

Индексирование отформатированных документов выполняется с помощью специальных фильтров спецификации [2] IFilter, которые в

соответствии с форматом документа производят выборку слов, отсеивая служебную и несущественную информацию. SQL Server 2000 имеет фильтры для HTML-файлов, текстовых файлов и документов Microsoft Office. Возможно создавать и использовать собственные фильтры.

В случае использования подхода к хранению документов в виде обычных файлов в дереве каталогов, задача обеспечения поиска по тексту документов намного сложнее.

Но можно существенно упростить поставленную задачу поиска, используя средство, предоставляемое операционной системой Microsoft® Windows 2000 Server (на основе которой создан сервер библиотечной системы). Это служба Microsoft Search, которая предоставляет эффективные механизмы поиска как по текстовым файлам, так и по документам Microsoft® Office. В таком случае результат будет схож с полученным предыдущим методом, так как механизмы полнотекстового индексирования используются одни и те же.

Если при разработке системы полнотекстового поиска предполагается создание собственной поисковой машины (с собственным механизмом построения поискового индекса), оба представленных подхода допустимы.

Рассмотрим подход к решению задачи хранения, предлагающий располагать документы в файловой системе. Он достаточно прост в реализации и напрашивается сам собой – файлы должны храниться в дереве каталогов. Но при этом ограничивается гибкость, управляемость и переносимость полученного множества данных. Например, исключается возможность обработки исключительных ситуаций при обращении к отсутствующему файлу.

Также, сложно будет дополнить систему своими обработчиками обращений к документам. Это доступно только через административные средства операционной системы, на которой расположится хранилище.

Очередная серьезная проблема возникает при создании резервных копий фонда электронных документов. Если база данных гарантирует единовременность создания «снимка» базы, сохранение каталога электронных

документов не дает таких гарантий. Таким образом, обеспечение целостности связей данных фонда должно обеспечиваться самой системой.

Применение второго подхода к хранению данных, предлагающего располагать документы в таблице базы данных, потребует создания механизма, эмулирующего все возможности и удобства работы со связанными документами. Чтобы второй вариант мог успешно конкурировать с первым, необходимо допустить полноценное функционирование перекрестных ссылок, используемых, например, в документах гипертекстовой разметки.

Использование такой системы хранения позволит здесь же осуществлять вызов функций, реализующих служебные операции (например, подсчет числа и частоты обращений).

### **Реализация механизма перекрёстных ссылок в файлах одного документа**

Библиотечная система должна поддерживать хранение документов даже с такой сложной организацией связей, как перекрёстные ссылки. Такие ссылки могут быть организованы внутри дерева хранимого документа.

Решая задачу создания хранилища фонда электронных документов с использованием возможностей файловой системы, для обеспечения работоспособности ссылок не потребуется ничего делать, так как изначально этот механизм разрабатывался под структуру каталогов дисков.

Если выбран подход, использующий собственную систему организации хранения документов (использующий базу данных), функция идентификации файлов, на которые указывают внутренние ссылки документа возлагается на модуль извлечения документа библиотечной системы.

Примером может служить то, как эта задача решается в файлах гипертекстовой разметки.

Пользователь обращается к конкретной части какого-то документа, указывая полный адрес. Этот адрес имеет вид:

```
address := document_address “/” file_name
```

document\_address := <описание местонахождения документа, по которому хранящая его система может его идентифицировать>

file\_name := <название отдельного файла – составной части документа>

Открытый пользователем документ содержит ссылки. Находящаяся в теле документа ссылка на объект, содержит относительный адрес местонахождения этого объекта. Таким образом, когда пользователь даёт команду перейти по этой ссылке, к строке полного адреса текущего документа (address) вместо названия самого документа (file\_name) добавляется относительный путь, указанный в ссылке.

Если спроецировать данную схему на разрабатываемую систему, то виртуальная адресация внутренних ссылок хранимых документов будет выглядеть следующим образом. Когда пользователь осуществляет переход на главную страницу документа, он должен указать полный адрес документа в хранилище. Далее, по описанному выше сценарию будет составляться строка, адресующая документ, на который указывает ссылка (относительная). Вид обращения при этом практически не изменится:

virtual\_address := virtual\_document\_address “/” file\_name

virtual\_document\_address := <описание местонахождения документа, по которому хранящая его система может его идентифицировать; постоянно для одного документа>

file\_name := <название отдельного файла – составной части документа; меняется в зависимости от выбранного фрагмента>

## **Выводы**

Очевидно, что необходимо развивать библиотеку в направлении расширения предоставляемых читателю сервисов. Требуется создать систему обработки электронных документов, которая должна поддерживать полный жизненный цикл такого документа. Библиотечная система должна уметь загружать, регистрировать, хранить, обрабатывать документ и предоставлять его пользователю.

Создание фондов электронных документов позволяет решить проблему недостаточного количества копий некоторых редких документов.

Электронная форма хранения предоставляет возможность доступа к содержимому документа, а не только к его описанию. Это расширяет возможности поиска материалов по запросу читателя.

В зависимости от потребностей разработчиков и сложности организации полнотекстового поиска выбирается способ хранения электронных документов фонда.

Простейший способ построения эффективной подсистемы полнотекстового поиска использует существующие СУБД и их средства построения полнотекстовых индексов. Он подразумевает помещение файлов документа в поля таблицы используемой СУБД.

Если же создается своя поисковая система – возможно использование обоих подходов к хранению файлов электронных документов. Полнотекстовый индекс должен быть построен поисковой системой по текстовым образам файлов, хранящихся либо в таблице базы данных, либо расположенных в каталогах на диске.

### **Литература**

1. Авторское право: Нормативные акты. Национальное законодательство и международные конвенции / Сост., авт. вступ. ст. И. Силонов; оформл. Г. Сыроватского.- М.: Элит-Клуб; Юридическая книга, 1998.- 429 с.

2. № 16, 2000. NetWeek. Web-функции и новые возможности SQL Server 2000 / Евгений Мамаев.- М.:СК Пресс, 2000.

## Дистанционное создание полнофункциональных сайтов

Мухамедвалеев Т.Р.

Сибирский государственный технологический университет, Красноярск

E\_mail: [tim1981@pochtamt.ru](mailto:tim1981@pochtamt.ru)

В проекте следует разработать информационную систему, позволяющую создавать автоматически полнофункциональные Web-сайты. Web-сайт с точки зрения такой системы выглядит как набор исполняемых страниц, sql-запросов на создание базы данных и прочих элементов, необходимых для формирования полноценного сайта. Разрабатываемый с помощью этой системы сайт автоматически выставляется в сети на сервере.

Для осуществления подобных операций необходимо создать мастера, которые по заранее описанным правилам должны обрабатывать переданные им шаблоны. С помощью них генерируются полнофункциональные html-документы (т.е. которые имеют необходимые объекты для действия полнофункциональной системы, при этом они (объекты) могут располагаться по желанию разработчика сайта и с дизайном, который был передан с шаблоном). Через мастера должны быть создаваемы, например, форум, гостевая книга, новости, файловый каталог и т.п. Для функционирования подобной системы необходимо создать средство передачи шаблонов на сервер, после чего они обрабатываются данными мастерами. В результате таких действий получаем html-документ, который способен обрабатывать данные на форме, вносить, редактировать, считывать из базы данных параметры и отображать их, т.е. полнофункциональный сайт. Обработка шаблонов будет зависеть от того, какой из модулей будет его интерпретировать.

Практически система функционирует подобным образом, на сервер отправляется некий шаблон, который имеет набор необходимых объектов. Описание объектов, как и привязка функций за которые этот объект будет отвечать, полностью определяются разработчиком системы. Это может быть непосредственное создание объектов в передаваемых шаблонах с необходимыми параметрами; или некий макрос который будет передавать



физическое название, логическое имя (например имя кнопки), размер, функциональную привязку и т.д и т.п.; либо какой ни будь другой метод. Предварительно пользователь подобной системы предложенными средствами выбирает мастера (модуль), которым посланный им шаблон будет интерпретироваться. После всех выше описанных процедур получается Web-сайт, он имеет дизайн, созданный пользователем и с объектами им указанные, которые функционируют согласно шаблону.

Подобная система способна автоматизировать создание полнофункциональных Web-сайтов, например таких как: форум, гостевая книга, новости, файловый каталог, internet магазин и т.п.

## Концептуально-графические модели механизмов регулирования темпов роста (решения проблем) социальных систем и процессов

А.В. Бондаренко, К.Х. Ипполитов, В.С. Нарыков, В.И. Пух  
Российская Академия естественных наук  
113105, Москва, Варшавское шоссе, 8  
[puh\\_victor@mail.ru](mailto:puh_victor@mail.ru)

I. Сутью (характеристикой) темпов развития любой социальной системы является быстрота или эффективность способа разрешения проблем (противоречий). Основные четыре глобальные проблемы, влияющие на всю иерархическую структуру эффективности механизмов и темпов развития социальных систем общества, в современных условиях следующие (см. Модель 1, “Объект”):

*первая* - крайне высокий уровень острейших социально-экологических противоречий в системе “природа-общество” и, как следствие, стремление к преимущественному использованию военно-силовых, а не интеллектуально - диалоговых мер, согласования интересов, при разрешении сложных глобальных, локальных, межгосударственных (межнациональных) конфликтов и проблем ;

*вторая* - непрерывно ускоряющиеся темпы развития информационно - телекоммуникационных систем и, как следствие, усложнение многообразия связей, зависимостей и взаимодействий элементов в социальных системах;

*третья* - преобладание стихийных механизмов над сознательными в решениях по развитию больших социальных систем и общества в целом;

*четвертая* - отсутствие адекватных современным условиям теоретических разработок и прикладных инструментов о сознательно проектируемых и корректируемых механизмах безопасного развития (решения проблем) социальных систем, более сложных чем корпорации.

Эти четыре глобальные проблемы тоже связаны причинно - следственными связями. Охарактеризуем подробнее содержание этих проблем

и связей между ними. Главной исходной причиной преобразования является очень высокий уровень противоречий в системе "природа-общество". Это вызвано тем, что, начиная с середины XX века, человеческое сообщество подошло к черте, когда процессы хозяйственной (технологической) деятельности человека по своему масштабу, массовости и сложности стали соизмеримы с процессами, происходящими в природе. Обострился (актуализировался) процесс смены основных принципов функционирования человека в системе "природа-общество". Если на предыдущем этапе развития в деятельности общества преобладало стремление к неограниченному и все большему потреблению природных ресурсов, то теперь перед человечеством стоит задача все большей экономии невозполнимых и трудновосполнимых природных ресурсов.

Основным в деятельности человека становится не увеличение потребления природных ресурсов, а нахождение новых способов их всемерной экономии. В результате происходит резкое снижение экзогенных (экстенсивных) факторов развития хозяйственной деятельности и возрастание эндогенных (интенсивных) факторов. Такими главными эндогенными факторами, прежде всего, становятся интеллектуальная (научная) деятельность человека и более продуктивная (эффективная) организация взаимодействия между членами общества, когда деятельность реализуется (вкладывается) и умножается не на уровне предметной деятельности (опредмечивания), а на уровне интеллектуальной деятельности (распредмечивания).

Перейдем ко второй проблеме - анализу непрерывно возрастающей сложности социально-экономических систем (СЭС).

С нашей точки зрения, возрастание сложности социально-экономических систем - это объективный процесс, который определяется тремя основными параметрами:

- возрастающей абсолютной численностью элементов системы;
- возрастающим многообразием родственных групп (классов) элементов системы с идентичными качественными характеристиками;

- возрастающим многообразием связей и взаимодействий между элементами системы.

Динамика процессов возрастания сложности социально-экономических систем изучена крайне слабо. Еще нет достаточно надежных методов формализации этих процессов. Однако, отдельные фрагменты идей, описывающие возрастание абсолютной численности элементов системы предложены Г.Н. Поваровым [1].

В соответствии с этой школой (шкалой Поварова), социально-экономические системы в зависимости от числа содержащихся в них элементов, можно представить в виде трех классов:

- большие (сложные системы) содержат порядка  $10^4 - 10^7$  - элементов;
- превращающиеся (ультрасложные, сверхбольшие) содержат  $10^8 - 10^{30}$  элементов;
- парадоксальные системы (сверхсистемы) содержат  $10^{30} - 10^{200}$  элементов.

В настоящее время главным предметом системных исследований являются большие системы. Поведение элементов этих систем описываются вероятностными, стохастическими методами. Особенность этих систем заключается в том, что нельзя предсказать поведение отдельного элемента, но можно предсказать суммарное (результатирующее) взаимодействие между элементами, т.е. механизмы поведения всей системы в целом. Для больших (сложных) систем число механизмов поведения на 3-4 порядка ниже, чем число элементов в них, т. е. в пределах  $10^1 - 10^3$ .

Вместе с изучением больших систем начинают изучаться сверхбольшие (превращающиеся системы). Примерами сверхбольших систем могут быть человеческий организм, социально-экономическая система государства и др. Отличительной особенностью этих систем является интенсивный обмен информацией с внешней средой, самоорганизация, саморазвитие. Главной особенностью систем этого класса являются гибкие перестраивающиеся механизмы. Ориентировочно число механизмов в таких

системах можно определить от  $10^4$ - до  $10^{10}$ . В связи с высокой сложностью превращающихся систем адекватный анализ их может проводиться только на основе вычислительных машин комплексов класса искусственный интеллект.

Перейдем к характеристике третьей проблемы - преодоление стихийности в развитии общества. К сожалению, мы должны признать, что наше общество до сих пор развивается преимущественно стихийными механизмами. Теоретическая разработка и реализация в общественной практике механизмов, сознательно управляющих развитием общества в русле объективных тенденций и закономерностей развития социально-экономических процессов, является одной из важных задач, стоящих перед нашим обществом.

Однако, вместе с признанием важности и актуальности этой проблемы, необходимо признать очень большую сложность разработки сознательных механизмов динамики социально-экономических систем. Эта сложность, прежде всего, объясняется высокой связанностью, зависимостью элементов социально-экономической системы. Такая связанность порождает необходимость конструировать иерархические, целостные модели механизмов динамики социально-экономических систем.

Четвертая проблема является ключевой по отношению к предыдущим трем проблемам. Эта проблема связана с отсутствием теоретических и прикладных методов и инструментов разрешения противоречий (устойчивого целенаправленного развития) в точках роста социальных систем и процессов. Инструментом практической реализации теоретических концепций может стать система информационно-аналитических порталов моделирования механизмов роста (решения проблем) социальных систем и процессов.

На верхнем уровне такой иерархической системы, с нашей точки зрения, должны находиться следующие подсистемы (инструменты и механизмы) :

- система принятия разрешений по разрешению противоречий в точках роста социальных систем;
- взаимодействие внешних и внутренних элементов и связей точек роста системы;

- взаимодействие объективных и субъективных факторов развития.

Таким образом, для разрешения возникших противоречий необходимо конструировать системы, функцией которых будет моделирование и инструменты реализации механизмов динамики (устойчивого развития) общественных структур. В связи с этим целью настоящей статьи является создать основы методов разработки языка концептуально - графического моделирования, и на базе этого языка инструментов формализации концептуальной информации о проблемах и механизмах разрешения социально - экономических противоречий, создание банков таких моделей и систем принятия решений, направленных на сознательное формирование механизмов динамики социально-экономических систем.

II. Сформулировав основные противоречия современного этапа развития и определив главные цели статьи, можно перейти к детальному описанию архитектуры системы, которая может обеспечить разрешение этих противоречий (см. Модель 1, “Субъект - портал” ).

В архитектуре такого портала, прежде всего, необходимо выделить семь блоков (подсистем):

- вербальное описание проблем механизмов социальных систем и процессов (ССП) (идентификация проблем);
- концептуально - графические модели (КГМ) механизмов развития ССП (принципиальные схемы, геометрическая “грубая” формализация );
- программное обеспечение механизмов развития ССП (“тонкая” формализация и получение системы несвязанных концептуально-графических моделей);
- информационно-аналитическое обеспечение механизмов развития ССП ( вербальные аналитические концепции и система статистических показателей ) ;

- принципиально новые, сознательно созданные механизмы развития ССП (результат, решенная проблема);
- интегрированные гипербазы концептуально - графических моделей ;
- математические и технические средства инфраструктуры функционирования информационно - аналитических порталов .

Необходимость развития процесса моделирования сверхбольших систем в такой последовательности одними из первых заметили М. Месорович и Я. Такахара. В своей книге "Общая теория систем: математические основы" они писали: "... вряд ли можно считать целесообразным начинать исследование сразу с подробной математической модели еще до того, как проверены основные гипотезы и достигнуто более глубокое понимание механизма работы системы. Гораздо эффективнее, особенно для больших систем, состоящих из большого числа взаимосвязанных подсистем, вначале наметить основные подсистемы, установить главные взаимосвязи между ними, а затем уже переходить к детальному моделированию механизмов функционирования различных подсистем" [2].

В результате такого общего анализа создается "принципиальная схема" [3] системы.

С нашей точки зрения "принципиальная схема системы" и "концептуальная модель системы" - понятия-синонимы.

В последние годы в связи с бурным развитием исследований в сфере системного анализа, искусственного интеллекта и экспертных систем резко возросла потребность как в конкретных концептуальных моделях систем, так и в новых методах их более строгой разработки.

Остановимся подробнее на более общих процедурах формирования содержательной (вербальной) информации о механизмах развития ССП и переходу этой информации к первичной (предварительной) формализации в виде концептуально-графических моделей, а затем уже к более глубокой

формализации в виде математических моделей и программному обеспечению развития ССП.

На этапе создания вербальной информации по отдельным фрагментам механизмов развития ССП выполняются следующие процедуры:

- осуществляется постановка целей. Предварительно определяется, как будут встраиваться полученные результаты в предметную область. Какие изменения эти результаты должны произвести в ней, какой дадут эффект;
- вычленяется и описывается проблемная область, строится иерархия проблем;
- определяется и описывается состав задач, необходимых для решения;
- определяется состав участников для решения задач, устанавливаются их роли (функции) в решении задач;
- определяются необходимые для решения задач ресурсы.

На следующем этапе создаются концептуальные модели. В ходе создания концептуальных моделей выполняются следующие процедуры:

- в предметной области выясняются основные понятия;
- между понятиями устанавливаются типы связей и отношений (причинно - следственные, пространственно - временные, родо - видовые, отношения часть - целое, корреляционные и др.);
- устанавливается характер необходимых для решения задачи информационных потоков;
- осуществляется более детальное разделение задач на подзадачи и устанавливается стратегия решения задач;
- определяются и устанавливаются ограничения для решения задач.

Как видно из перечисленных процедур главная задача КГМ - это приступить к “грубой” формализации предметной области в виде определения установления связей и отношений между выделенными основными понятиями. Недостатком таких моделей можно считать довольно высокий уровень абстрагирования от деталей предметной области. Однако, несомненно, есть и преимущества этих моделей: простота и возможность их агрегирования.



На этапе математического описания механизмов динамики ССП осуществляется более глубокая формализация основных (ключевых) понятий и отношений (связей). В результате этого этапа создается база знаний о фрагментах механизмов динамики ССП, которые в дальнейшем могут быть интегрированы в целостную модель механизмов ССП.

III. В предыдущих двух разделах были рассмотрены теоретические аспекты формирования концептуально-графических моделей механизмов регулирования темпов роста ССП. Этот третий раздел будет посвящен конкретно-прикладным аспектам формирования КГМ.

Любую социальную систему (объект) можно представить в виде множества карт точек роста. Применительно к Российской Федерации можно представить систему карт :

- карта точек роста (снятия проблем) Евразийской цивилизации;
- карта точек роста (снятия проблем) стран СНГ;
- карта точек роста (снятия проблем) России как целостного государства;
- карта точек роста (снятия проблем) регионов Российской Федерации;
- карта точек роста (снятия проблем) в силовых министерствах РФ;
- карта точек роста (снятия проблем) оборонной промышленности РФ;
- карта точек роста (снятия проблем) в отраслях экономики РФ;

Такая система карт точек роста в виде концептуально - графической модели опубликована в статье (4).

В каждой точке роста необходимо выделить семь базовых конструкторов языка концептуально-графического моделирования: I - объект; II - субъект; III- метасубъект; IV-связи, отношения, взаимодействия между объектом и субъектом; V- механизмы (отношения, взаимодействия) между связями объекта и субъекта; VI - связи, отношения, взаимодействия между субъектом и метасубъектом; VII - механизмы (отношения, взаимодействия)

между связями субъекта и метасубъекта, VIII - внешняя среда (см. Модель 2).

Каждая точка роста характеризуется (рассматривается) по двум основаниям : как целостная система и как структура элементов. Определяются цели, проблемы, ресурсы, системы статистических показателей отражающих ретроспективу темпов развития этой точки-индикаторы, и прогноз темпов развития - индексы. В итоге будет сформулирована система механизмов развития такой точки роста и выработана стратегия принятия решений по ее реализации.

Еще одним объектом применения инструментов концептуально - графического моделирования могут стать правовые проблемы безопасного развития социальных систем и процессов в России и механизмы вхождения ее в мировое сообщество. Современным, активным, системно-интегрированным субъектом решения этих проблем может стать Кодекс законов “О национальной безопасности Российской Федерации” (см. Модель 3). Структура механизмов формирования и реализации этого Кодекса законов может быть представлена в виде следующих этапов:

- формирование поля систем и точек роста (развития) объекта;
- определение проблем, целей, задач и ресурсов для обеспечения безопасного и эффективного развития правовых полей социальных систем и конкретных точек роста;
- формирование вербальных концепций механизмов безопасного развития правовых полей социальных систем и конкретных точек роста;
- формирование гипербаз концептуально-графических моделей механизмов развития правовых полей социальных систем и конкретных точек роста;
- формирование программ и проектов реализации механизмов развития правовых полей социальных систем и конкретных точек роста;
- принятие решений по реализации механизмов развития правовых полей социальных систем и конкретных точек роста;

- контроль процессов реализации механизмов развития правовых полей социальных систем и конкретных точек роста;
- оценка результатов реализации программ развития.

Рассмотрим еще один пример применения методов концептуально-графического моделирования для конструирования архитектуры и механизмов функционирования информационно-аналитического портала стратегического планирования развития Московского мегаполиса как центра (ядра) Евразийской цивилизации (см. Модель 4).

Прежде всего необходимо определить общую архитектуру портала и структуру подсистем портала: объекта, субъекта, метасубъекта, связей (взаимодействий) и механизмов, внешней среды. Метасубъектом развития Московского мегаполиса должен стать Институт проблем безопасности и устойчивого развития. Ядром, метасубъекта должна быть гипербаза концептуально-графических моделей. Рассмотрим детальнее структуру института:

- Центр проблем развития механизмов прогнозирования и системной интеграции;
- Центр проблем развития механизмов финансирования и инвестиций;
- Центр развития технологий и новшеств;
- Центр социально - демографических проблем;
- Центр природно-экологических проблем;
- Центр стандартов и нормативов;
- Центр социально- экономических проблем.

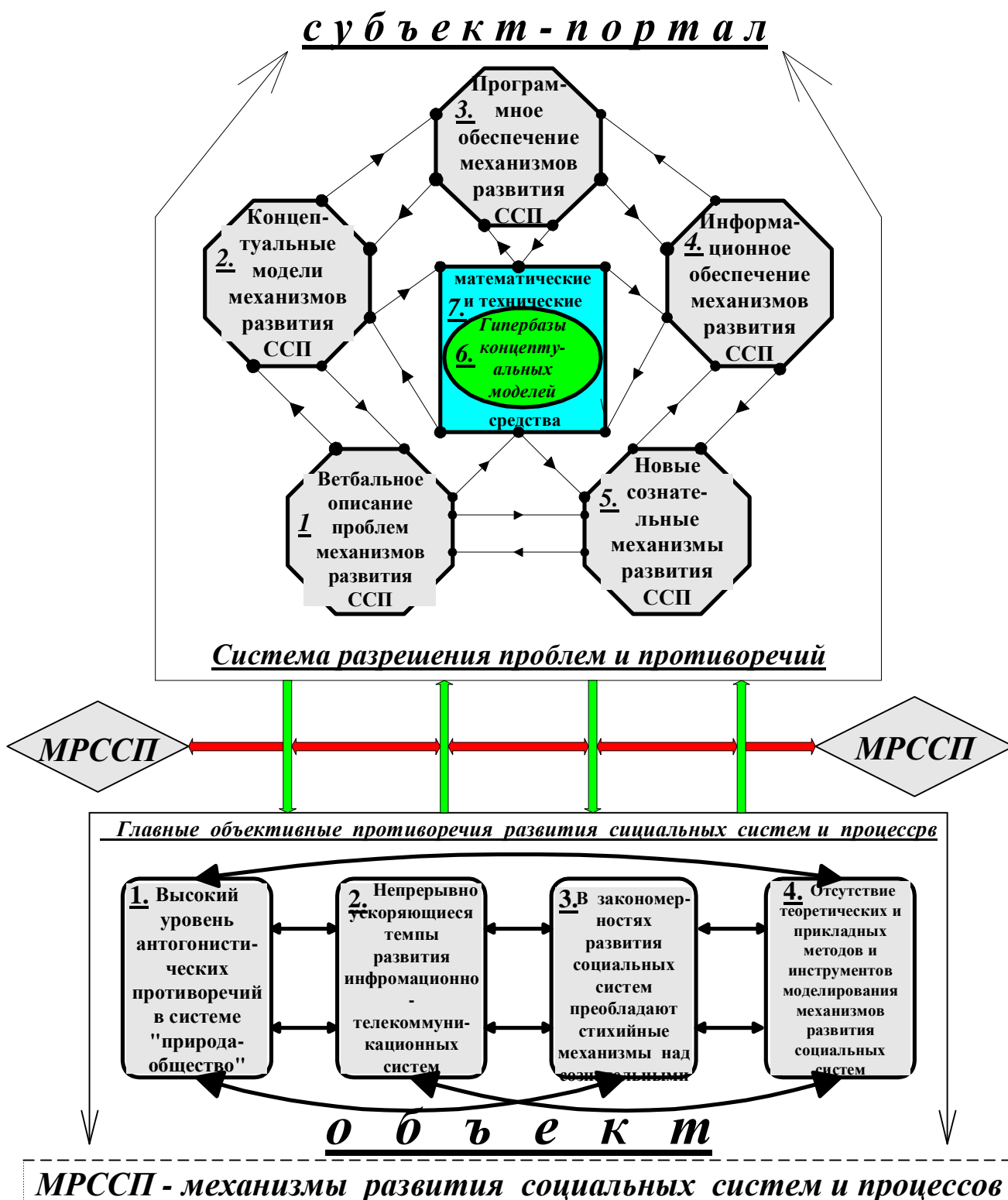
В заключении необходимо сделать вывод, что технологии концептуально-графического моделирования регулирования темпов роста социальных систем и процессов являются базой для создания автоматизированных рабочих мест аналитиков и конструирования самых современных информационно - аналитических порталов. А единое информационно-методологическое пространство аналитических порталов

(ситуационных центров) обеспечит управлением безопасным и эффективным развитием социальных систем и процессов.

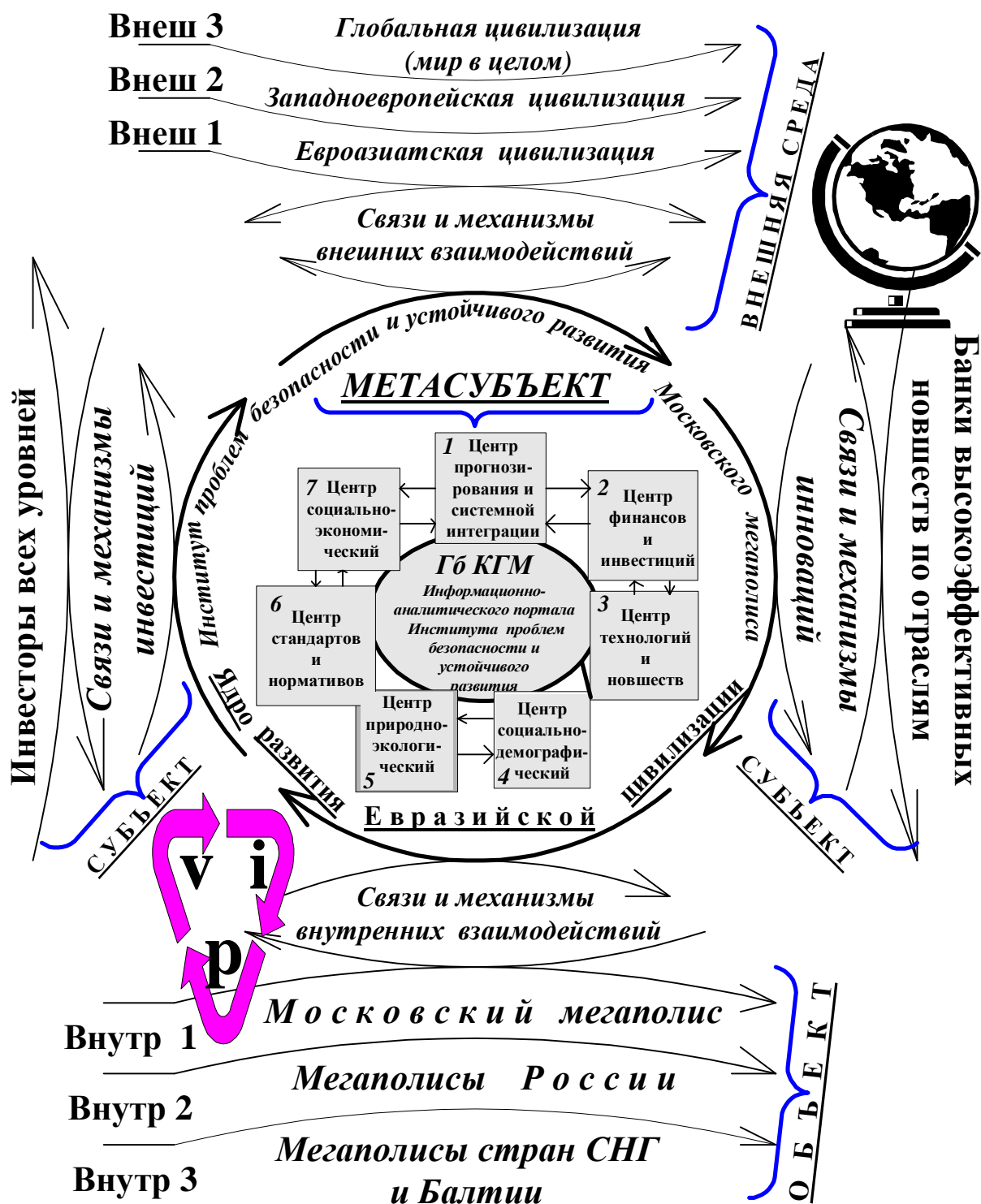
Список литературы:

1. Поваров Г.Н. Ступени сложности // Управление, информация, интеллект. М.: Мысль, 1976. с. 161-173.
2. Месарович М., Тахакара Я. Общая теория систем: математические основы. М. Мир, 1978. с.12
3. Месарович М., Тахакара Я. Общая теория систем: математические основы. М. Мир, 1978. с.13
4. Мехряков В.Д., Пух В.И. Творческое наследие А. А. Богданова: формирование языка концептуально-графического моделирования механизмов динамики глобальных и локальных социальных систем и процессов. // Международный институт А. Богданова. Труды международной научной конференции “Тектология в XXI веке”. М., 2000. С.200-216.

**Модель информационно - аналитического портала моделирования механизмов решения проблем социальных систем и процессов (ССП)**



**Концептуально - графическая модель информационно - аналитического портала стратегического планирования развития Московского мегаполиса как центра (ядра) Евразийской цивилизации.**

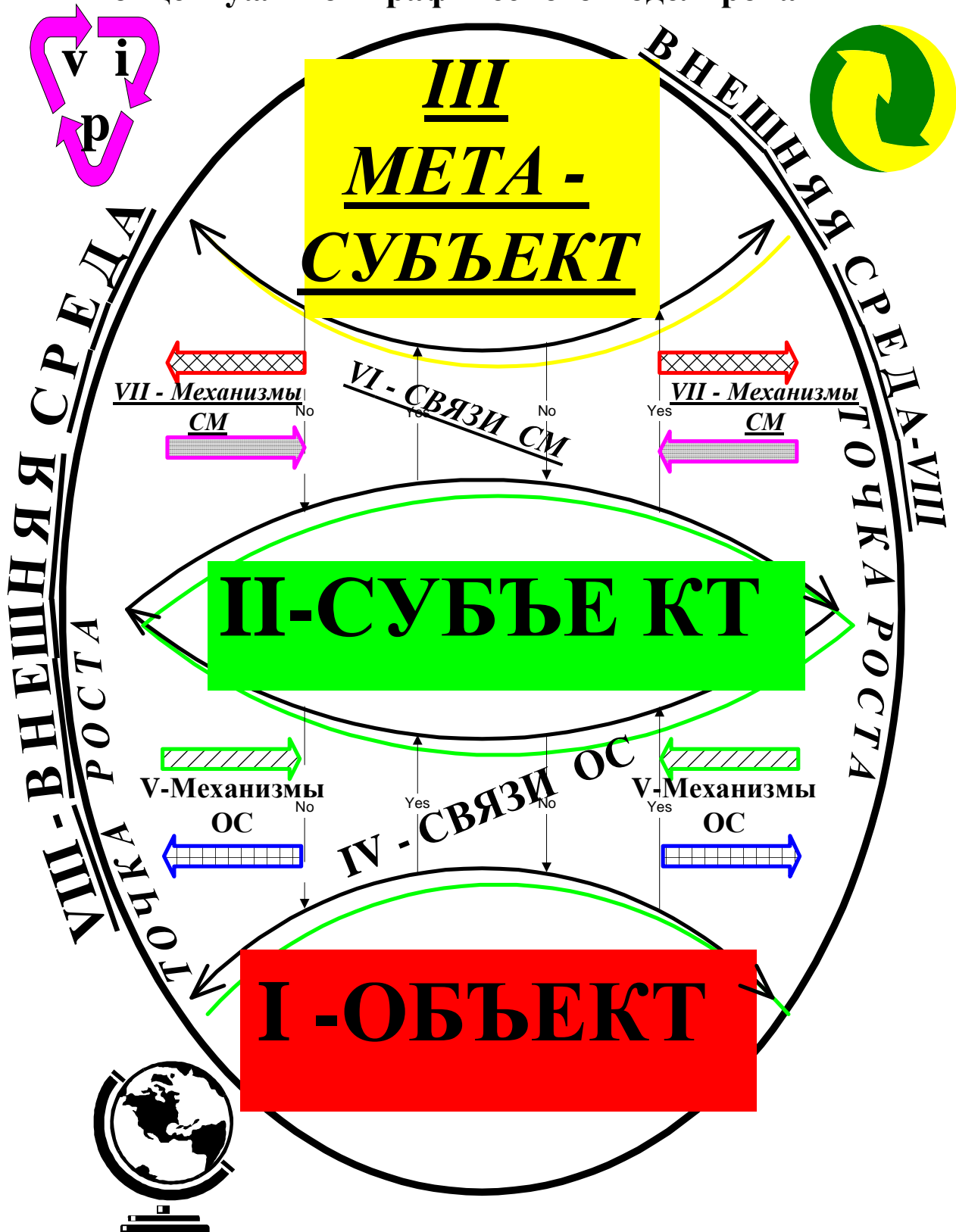


**Гб КГМ - Гипербаза концептуально-графических моделей**

# Концептуально-графическая модель механизмов формирования и реализации Кодекса законов "О национальной безопасности Российской Федерации"



Модель семи базовых конструктов (понятий)  
проблемных точек развивающейся системы и языка  
концептуально - графического моделирования





## **Использование имитационного моделирования при решении задачи анализа работоспособности ЛВС**

**Давыдов С.В.**  
**Московский Государственный институт электроники и математики**  
**Кафедра ИТАС**

Новейшие достижения в области микроэлектроники привели к новым концепциям в организации информационных служб. Благодаря быстрому развитию и широкому распространению персональных компьютеров (ПК), которые по стоимости эквивалентны терминалам больших и мини-ЭВМ, а по возможностям соответствуют ЭВМ третьего поколения, информационно-вычислительные ресурсы приближаются к рабочим местам менеджеров, бухгалтеров, плановиков, администраторов, инженеров и других категорий работников. На этой основе с середины 80-х годов наметилась тенденция развития информационно-вычислительной техники в виде создания локальных вычислительных сетей (ЛВС) различного назначения [1].

Компьютерные сети применяются при обработке текстов, работе с корпоративными базами данных, используются для выполнения числовых расчетов, являются информационными системами в управлении, планировании, учете, проектировании и др. [2]. Широта их использования определяется тем, что компьютерные сети позволяют организовать совместную работу пользователей с информацией, с обеспечением централизованного распределения ресурсов и разграничением прав доступа к ним.

Широкая и постоянно увеличивающаяся номенклатура ЛВС, сетевые программные продукты и технологии возлагают на потенциального пользователя сложную задачу выбора нужной системы из массы существующих [3]. Поэтому в настоящее время достаточно остро стоит вопрос о создании средства, которое может помочь им в этом выборе. Одним из возможных вариантов такого средства можно рассматривать специализированную систему имитационного моделирования ЛВС, проводя с помощью которой

моделирующие эксперименты можно оценить работоспособность проектируемой сети.

Для оценки работоспособности ЛВС при построении ее модели необходимо учитывать уровень детализации с которым необходимо представить моделируемую сеть. При этом в качестве уровней детализации сети можно использовать уровни эталонной модели ЛВС, в соответствии с требованиями Международной Организации по Стандартам (OSI) [5,6]. В этом случае исходными данными для процесса проектирования сети и ее модели будет являться множество объектов

$$C = \{C_i\}, i=1, N_k$$

Где  $C_i$  – объекты  $i$ -го уровня;

$N_k$  – количество уровней.

Каждый уровень представляется в виде:

$$s_i = T \cdot C_i$$

Где  $\sigma_i$  – структура  $i$ -го уровня;

$T$  – время существования.

В качестве средства формализованного описания данных на концептуальном уровне используется модель данных «объект-связь». Модель представляется в виде:

$$M = \langle C_a, C_p, \sigma \rangle$$

$$C_a = A \cup S \cup H \cup K$$

Где  $C_a$  – классы активного оборудования ЛВС:

$A$  – сетевые адаптеры;

$S$  – компьютеры сети

$$S = W_s \cup S_{er}:$$

$W_s$  – рабочие станции,

$S_{er}$  – серверы сети;

Так как сетевые адаптеры являются фактически неотъемлемой частью любого компьютера функционирующего в сети, то для компьютеров, в общем

случае, можно ввести обобщенный класс, который будет описывать объединение компьютера и установленного в него сетевого адаптера:

W – объединенный класс компьютер с сетевым адаптером.

H – концентраторы;

K – коммутаторы и репитеры;

B – маршрутизаторы.

$$C_p = S_p \cup C_{on}$$

Где  $C_p$  – классы пассивного оборудования ЛВС:

$S_p$  – среда передачи данных;

$C_{on}$  – коннекторы (все разъемные соединения).

В зависимости от уровня представления сети элементы  $C_a$  и  $C_p$  будут иметь различное модельное описание в соответствии с их представлением согласно уровням модели OSI для включения их в структуру модели.

При рассмотрении процесса функционирования сети можно выделить несколько направлений по которым можно оценивать качество ее работы, которые определяются количественными и качественными характеристиками используемого оборудования. При более детальном рассмотрении этих направлений можно отметить, что все они в конечном итоге сводятся к оценке времени получения ответа на запрос пользователя, которое в большей своей части определяются временем задержки на передачу данных [4]. Поэтому критерием качества спроектированной ЛВС можно считать задержку при получении отклика на запрос. Величина задержки при обработке запроса составляет:

$$t_{запр} = t_{обр} + t_{зо}, \quad \text{где:}$$

$t_{обр}$  - время обработки запроса на компьютере, к которому он адресован, зависит от типа запроса, и определяется нормальным распределением от величин, полученных экспериментально или определенных пользователем;

$t_{зо}$  - время задержки, вносимой оборудованием сети;

$$t_{зо} = t_{зпер} + 2 * t_{пп} + 2 * t_{за} + t_{зпр}, \quad \text{где:}$$

$t_{\text{зпер}}$  - время задержки передачи запроса: определяется временем прекращения коллизий или временем ожидания маркера;

$t_{\text{пп}}$  - время передачи/приема запроса по интегрированной кабельной системе сети:

$t_{\text{пп}} = L_{\text{кc}} * t_{\text{зс}}$ , где:

$L_{\text{кc}}$  – длина кабельной системы при прохождении запроса;

$t_{\text{зс}}$  - время задержки среды передачи сигнала на единицу длины.

$t_{\text{за}}$  - время задержки вносимое активным оборудованием сети, при прохождении запроса.

$$t_{\text{за}} = 2 * t_{\text{зса}} + \sum_{i=1}^n t_{\text{аoi}}, \quad (1)$$

где:  $n$  – общее количество активных элементов сети, через которые проходит запрос. Так как на пути от компьютера-источника к компьютеру-приемнику запрос проходит два сетевых адаптера (по одному в каждом компьютере), то времена задержек, вносимых ими можно задать явно ( $t_{\text{зса}}$ ).

Тогда формулу (1) можно представить в виде:

$$t_{\text{за}} = 2 * t_{\text{зса}} + \sum_{i=1}^n t_{\text{аoi}}$$

Критерий работоспособности представим в виде логического выражения

$$\forall x \in C_p ; \forall y \in C_a \exists \underline{K}_{\text{оп}} \sim K_{\text{оп}};$$

$$K_{\text{оп}} = \sum_{i \in I_s} a_i C_p + \sum_{j \in J_s} b_j C_a$$

Где:

$K_{\text{оп}}$  – критерий оптимальности;

$I_s$  – пассивное устройство заданного уровня;

$J_s$  – активное устройство заданного уровня;

$a_i, b_j$  – коэффициенты важности устройств с точки зрения оценки качества.

Теперь, определив критерий оценки работоспособности можно построить имитационную модель компьютерной сети и провести с ней модельные эксперименты. Работоспособной будем считать компьютерную сеть если

критерий, полученный в ходе эксперимента, попадает в заданный доверительный интервал с погрешностью, не превышающей заранее заданную. Оценку качества работы ЛВС будем определять путем анализа полученного значения задержки при выполнении запроса, если эта величина не превышает заранее заданную.

Таким образом, предложенный подход к построению моделей компьютерных сетей и анализу их работоспособности можно использовать для определения работоспособности сетей различной архитектуры на требуемом уровне детализации.

## ЛИТЕРАТУРА

6. Аппак М.А. Автоматизированные рабочие места на основе персональных ЭВМ. – М.: Радио и связь, 1989. 175 с.
7. Васильев Г.П., Егоров Г.А., Щербина Н.Н. Программное обеспечение сетей СМ ЭВМ. – М.: Финансы и статистика, 1983. 87 с.
8. Науманн Ш., Вер Х. Компьютерная сеть. Проектирование, создание, обслуживание. / Пер. с нем. – М.: ДМК 2000. 336 с.
9. Оглтри Т. Модернизация и ремонт сетей, 2-е изд. / пер. с англ. –М.: Изд. Дом «Вильямс», 2000. 928 с.
10. Information Processing Systems – Open System Interconnection – Basic Reference Model: International Standard 7498 – 1984. p. 1-40.
11. Data Communications Networks, Open System Interconnection (OSI) System Description Techniques, Volume VIII, Fascicle VIII.5, CCITT Red Book, CCITT Plenary Assembly, October 8-19, 1984 (Malaga-Torremolinos).