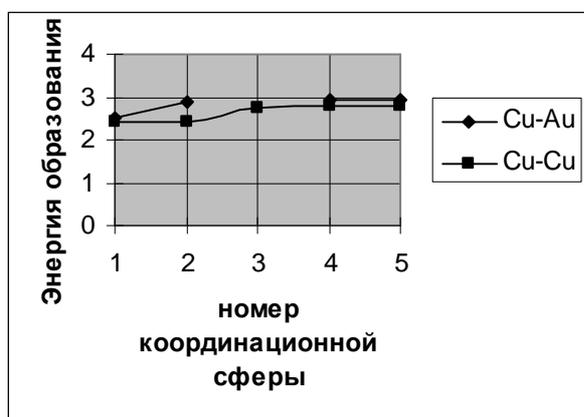
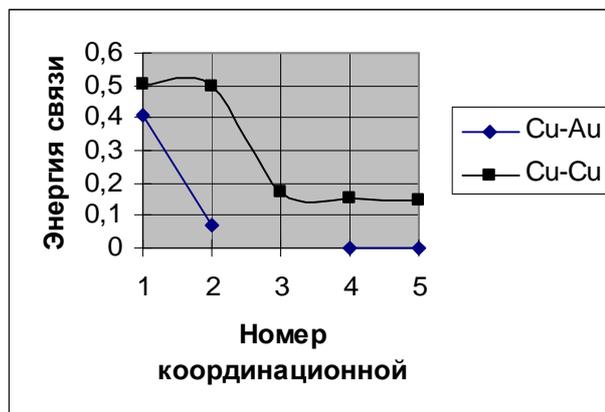


Таблица 1. Результаты эксперимента (Система Cu_3Au)

тип вакансий	число вакансий m	E_0 , эВ энергия идеального кристалла	E , эВ Энергия системы	E_1 , эВ энергия разорванных связей	E_2 эВ энергия образования	E_3 , эВ энергия связи	Расстояние между вакансиями p
Cu	0	-823,829					0
	1		-821,041	2,788	1,394		0
Au	1		-820,747	3,082	1,541		0
Au-Au	2		-817,694		3,053	-0,118	3
Cu-Au	2		-818,369		2,525	0,41	1
			-818,03		2,864	0,071	2
							3
			-817,962		2,932	0,003	4
			-817,961		2,933	0,002	5
Cu-Cu	2		-818,61		2,431	0,504	1
			-818,311		2,436	0,499	2
			-818,277		2,764	0,171	3
			-818,257		2,784	0,151	4
			-818,254		2,787	0,148	5



а)



б)

Рисунок 1. а) Энергия образования комплексов вакансий в зависимости от расстояния между вакансиями; б) Энергия связи в зависимости от расстояния между вакансиями.

СПИСОК ЛИТЕРАТУРЫ

1. Суппес В.Г., Полетаев Г.М. Компьютерный лабораторный практикум по молекулярной физике. Ж."Физическое образование в вузах" Издательский дом МФО 2003г., Т.9, №2, с.113-124.
2. Суппес В.Г., Полетаев Г.М. Компьютерный лабораторный практикум по молекулярной физике. Сб. науч. трудов "Проблемы учебного физического эксперимента" МММ., ИОСО РАО 2003, с.80-82.
3. Суппес В.Г. О компьютерном лабораторном практикуме. Межвузовский сборник научных статей под редакцией В.П.Горшенина, И.В.Резанович. Профессиональное мастерство: становление, формирование и развитие. Челябинск, Издательство ЮУрГУ, 2003, С.172-178.
4. Суппес В.Г., Старостенков М.Д., Дудник Е. А. О направлениях обучения с использованием ком-

пьютеров. Ж."Физическое образование в вузах" Издательский дом МФО 2004г., Т.10, №2, с.76-83.

ИНТУИТИВНЫЕ ОБЪЕКТЫ

Фельдман Я.А.

В программировании различают *физические* и *логические* структуры данных. Но пользователи мыслят структурами более высокого уровня, назовем их *интуитивными* (термин мой). Известны интуитивные структуры: объекты OLAP и объекты TreeLogu. Предлагаю свой вариант: *интуитивные объекты*.

Параметр или *типовой элемент данных* это пара (имя, формат). *Элемент данных* это пара (параметр, значение) У интуитивного объекта всегда есть набор элементов данных. Набор (упорядоченный список) типовых элементов это *тип*. Для таких типов

есть *наследование*. Например тип *сотрудник унаследован* от типа *человек*. При наследовании список *типа-отца дописывается* новыми типовыми элементами и так получается список *типа-ребенка*. Наследование объединяет все типы (как вершины) в *дерево наследования типов* (ДНТ).

У интуитивного объекта есть один и только один тип-центр (одна вершина ДНТ) и один и только один тип-множество (поддерево ДНТ, содержащее корень и тип-центр). Например *человек Иванов* может быть *сотрудником* и *студентом*. Тип-множество объекта *Иванов* это {объект, человек, сотрудник, студент}. Любой элемент этого множества может быть объявлен тип - центром. В течение жизни объекта его тип-множество и тип-центр могут меняться. При этом происходит приобретение и/или потеря данных.

Тип-вершина может получить *ярлык* (Синонимы: *тень, слабая копия*). Число ярлыков не ограничено, но у каждого ярлыка один и только один *оригинал*. *Ярлык от ярлыка* это *ярлык* от оригинала первого ярлыка. Ярлык можно добавить в ДНТ. *Под ярлык* ничего добавить нельзя. *Нельзя* добавить ярлык так, чтобы путь от ярлыка в *корень* ДНТ проходил *через* оригинал. Путь из оригинала в *корень* (*главный путь*) один и только один. Если разрешить *перескакивать из оригинала на ярлык*, то получится *слабый путь* в *корень*. Слабых путей может быть несколько.

Если у объекта А тип центр Т, то все типы-вершины на пути из Т в *корень* ДНТ *обязательно* входят в тип-множество объекта А. При демонстрации объекта А сперва демонстрируется список элементов *главного* пути. Затем демонстрируются элементы лежащие на *слабых* путях из Т в *корень*, если они не попали в *главный* путь и если эти тип-вершины попали в тип-множество объекта А. Тип-множество может включать больше вершин, но их элементы не демонстрируются.

Каждая тип-вершина имеет свой список функций (возможно пустой). Этот список дописывается по тем же правилам.

Права пользователей на создание, удаление, изменение объектов управляются еще двумя деревьями: *деревом объектов* (ДО) и *деревом вложенности типов* (ДВТ). ДВТ построено на тех же оригинал - вершинах, что и ДНТ и имеет тот же *корень*. ДО содержит все объекты. *Корень* ДО имеет тип - центром *корень* ДВТ. Пользователи – тоже объекты и тоже входят в ДО. Когда пользователь входит в систему, он видит поддерево ДО начиная с себя как с *корня*. Все это поддерево – его *зона ответственности*. В этой зоне он может создавать, изменять, удалять, перемещать объекты. Он может менять их тип, но так, чтобы объединение всех тип-множеств его объектов не увеличилось (это объединение называется *зоной компетенции*)

Можно создавать ярлыки в ДО и ДВТ (по тем же правилам – абзац третий).

Создать объект Б типа ТБ под объектом А типа ТА можно если и только если (А лежит в зоне ответственности и (в ДВТ вершина ТБ (или ее ярлык) - прямой потомок вершины ТА **или** ТБ = *папка или* ((ТА=ТБ и ТА - *рекурсивный тип*))

Если поместить ярлык объекта А в зону ответственности пользователя П, то это увеличит зону ответственности П на одну вершину, а *зону видимости* пользователя П на все поддерево с *корнем* в А. В зоне видимости можно видеть данные, но нельзя ничего менять.

Управление данными с помощью ДО, ДВТ и ДНТ позволяет строить и модифицировать систему без программистов. Только 4% работ требует знание SQL и 1% - знание JAVA. Типами управляет один человек – администратор = архитектор. Как пользователь он соответствует *корню* ДО.

Данная модель реализована на основе бесплатного ПО (http-сервер, СУБД) с применением JAVA/JSP и получила название FTS. Основное назначение – единая система управления предприятием. Экспериментально реализовано несколько задач.

В двух школах Петрозаводска уже год применяется Система Анкетирования и Тестирования (САТ) на основе FTS. Опыт показал *очень* высокую *гибкость* системы и ее *дружелюбное поведение*.

По затратам на создание и модификацию ИС технология FTS аналогов не имеет. (Я не рассматриваю область, где данные вводит автомат. Там объекты однотипны, структуры вечны, модификация исключена).

ИСПОЛЬЗОВАНИЕ ПРИНЦИПОВ ТЕОРИИ ПРИНЯТИЯ РЕШЕНИЯ В МЕТОДЕ МОРФОЛОГИЧЕСКОГО СИНТЕЗА

Чугунов Д.С., Бутенко Л.Н.

*Волгоградский государственный
технический университет,
Волгоград*

Как известно основу метода морфологического поиска составляют два этапа: анализ и синтез. На первом этапе производится изучение объекта, оптимальную модель которого мы должны получить. Известно несколько направлений анализа объекта, основными из них являются поэлементный и пофункциональный анализ, то есть когда объект рассматривается в виде составляющих его элементов либо выполняемых им функций.

Далее за этапом анализа, следует этап синтеза, на котором производится собственно расчет обобщенных показателей качества (если это необходимо) и в зависимости от метода, генерация готовых моделей, будь то полный перебор либо перебор после усечения морфологического множества альтернатив.

Теория принятия решения – это набор понятий и систематических методов, позволяющих всесторонне анализировать проблемы принятия решений в условиях неопределенности, возникающих при необходимости сравнений подобный технических систем – альтернатив. Под принятием решений понимается выбор одной альтернативы из полученного или представленного множества альтернатив.

Приведем реализацию метода морфологического синтеза опирающуюся на принципы теории принятия решений.