

УДК 517.977

Структуры компьютерных метафор

**В. Л. Авербух, А. Ю. Байдалин, М. О. Бахтерев, Д. Р. Исмагилов,
П. В. Трушенкова**

В работе рассматриваются некоторые положения теории метафоры. Вводятся понятия “работы метафоры” и “формулы метафоры”. На ряде примеров проводится структурный анализ конкретных компьютерных метафор, необходимый для понимания причин успехов одних и неудач других метафор визуализации и визуального интерфейса. Изучение структур объектов визуализации вместе с пониманием логики выбора и порождения метафор дают возможность формулировки критериев оценки когнитивной составляющей визуальных систем. Визуализация рассматривается как построение видимого отображения ментальных и/или компьютерных моделей, а интерпретация как двойственная к визуализации операция. Успешная визуализация, построенная на базе “хорошей” метафоры, должна опираться на уже имеющуюся у пользователя модель явлений и/или процессов и не противоречить его репрезентативным когнитивным структурам. Уменьшение сложности интерпретации визуализации можно рассматривать как условие “хорошей” метафоры.

Введение

Данная работа продолжает и обобщает наши исследования компьютерных метафор, результаты которых опубликованы в 2001 – 2006 гг. ([2–4, 16, 32]). Истоки этих исследований лежат не только в теоретических изысканиях, но и в практике проектирования специализированных систем визуализации [6–9, 13–15, 17]. Этап поиска или конструирования соответствующей метафоры является частью процесса разработки специализированной визуальной системы [33]. Здесь под метафорой понимается некоторая идея сближения понятий, используемая при создании специфической образности. Мы хотим научиться хорошо проектировать специализированные системы, а выбор хорошей (адекватной) метафоры — ключ к правильным решениям.

Существует обширная литература, посвященная проблематике компьютерной метафоры. В последние десятилетия метафоры рассматриваются не только с традиционных позиций филологии, но и философии, логики, когнитивистики, психологии, компьютерных наук. В этой связи выявлена роль метафоры в развитии научных дисциплин. (О научной метафоре см., в частности, [56, 57].) Весьма популярна среди специалистов работа [24], посвященная роли метафор в современном обществе и ее повсеместному проникновению в науку, политику, повседневную деятельность и мышление людей. Компьютерные науки, как и все новые научные дисциплины, развивались при активном использовании метафор, наиболее известные из которых, например, такие как “память” или “файл” почти потеряли отсылки к исходным понятиям. Осознание значения метафоры для человеко–компьютерного взаимодействия можно связать с успехом метафоры рабочего стола, являющейся господствующей в пользовательском интерфейсе персональных компьютеров в течение уже почти полутора десятков лет. (Вместе с тем, в литературе можно найти и обсуждение самой возможности использования метафор в компьютерном дискурсе [53].)

Значительная часть литературы по компьютерным метафорам посвящена именно проблематике метафоры человеко–компьютерного взаимодействия. Мы ранее достаточно жестко разделяли метафору интерфейса и метафору визуализации. Причем задачи создания и использования метафор визуального интерфейса нами практически не рассматривались. Однако в дальнейшем внимание исследователей было привлечено к материалам, посвященным опытным

разработкам специализированных визуальных сред, относящихся по нашему мнению к популярному некогда классу АРМ (автоматизированных рабочих мест) специалистов различного профиля, например, медиков, социологов, служащих муниципальных учреждений. Функции и операции, которые должны поддерживать эти системы, могут сильно отличаться от рутинных функций и операций, реализованных в традиционных системах автоматизации конторского труда (office automation systems). Для систем типа АРМ часто требуется глубокая специализация интерфейса. Причём отсутствие дополнительной локальной метафоры или её неудачный выбор приводят к сложностям при внедрении вполне функционально полноценных систем из-за возникающих у пользователей проблем с интерпретацией образов и операций. Всё это показывает, что необходим более широкий подход к исследованиям, когда в рамках одной теории рассматриваются различные случаи использования визуальных компьютерных метафор.

Нашей задачей является проектирование “хороших” метафор для специализированных систем визуализации и/или визуального интерфейса. Кроме того важными являются оценки качества метафор уже готовых или проектируемых систем.

Ниже разбираются некоторые положения теории метафоры в связи с задачами исследования. Вводятся понятия “работы метафоры” и “формулы метафоры”. На ряде примеров проводится структурный анализ конкретных компьютерных метафор, необходимый для понимания причин успехов одних и неудач других метафор визуализации и визуального интерфейса. Изучение структур объектов визуализации вместе с пониманием логики выбора и порождения метафор дают возможность формулировки критериев оценки когнитивной составляющей визуальных систем.

1. Теоретические аспекты изучения компьютерной метафоры

Наш интерес к теоретическим проблемам метафоры связан с основной задачей — научиться быстро и хорошо проектировать специализированные визуальные системы. В таких системах выделяются следующие составляющие:

- компьютерная графика и средства организации человеко-компьютерного интерфейса;
- инженерия программного обеспечения;
- когнитивные аспекты, связанные с проблематикой мышления и восприятия пользователя систем визуализации

В данной работе рассматриваются именно когнитивные аспекты разработки систем. Когнитивный аспект наиболее независим от техники и технологии. В тоже время существует достаточно примеров, когда неудачи с “когнитивными” составляющими проекта сводят на нет все успехи и достижения в области графики и программной инженерии. В этой связи нас интересуют языковые аспекты визуализации и человеко-компьютерного взаимодействия, то есть то из чего и как построены визуальные и диалоговые объекты, как они отображают необходимые понятия.

1.1. Знаковая природа человеко-компьютерного взаимодействия и визуализации

Рассмотрим (упрощенные) определения ряда понятий семиотики — науки о знаковых системах, включающей общие вопросы изучения языка. В наших работах [1], [3] приведена библиография переводных и оригинальных работ по семиотики, из которых эти определения частично заимствованы, поэтому здесь они пойдут без ссылок.

Знак есть нечто, обозначающее некоторый факт или объект, для некоторой интерпретирующей мысли. Ситуация, в которой употребляется знак или знаковое отношение, состоит из трёх элементов — *объект, знак, интерпретант*.

Согласно другому определению *знак* есть носитель, сообщающий уму что-то извне. То, что он обозначает, называется его объектом, то, что он сообщает — его *значением*, а идея, которую он вызывает — его *интерпретантом*.

Знаковой ситуацией называется пара из *знака* и *обозначаемого (денотата)*. В подавляющем числе знаковых ситуаций проявляются такие черты знака, как способность знака выступать в качестве заменителя обозначаемого; нетождественность знака и денотата (кроме случая автономного употребления знака); многозначность соответствия “*знак-денотат*”.

Знаковый процесс (семиозис) рассматривается на пятичленном отношении между *знаком*, его *значением*, его *интерпретантом*, *контекстом*, где знак встречается и, наконец, *интерпретатором* знака. Знак вызывает у интерпретатора определенную реакцию или предрасположенность к ней (интерпретант) на определенный вид объекта (на значение) при определенных условиях (в некотором контексте).

Выражаемые знаком свойства денотата, то понятие о денотате, которое несёт данный знак, называется *концептом*. Выбор денотата определяется конкретной знаковой ситуацией. Концепт показывает насколько отношение знака к денотату не случайно, а обусловлено стремлением определить денотат в том или ином аспекте. На уровне концепта можно отвлечься от обстоятельств конкретной знаковой ситуации и перейти к систематическим приёмам обозначения. Отношение знака к денотату через концепт даёт системность (сходные обозначения для сходных обозначаемых). Родство знаков часто соответствует родству предметов.

Знаковой системой называется некоторый набор знаков, в котором есть внутренние отношения между знаками, каким-то образом отображающими отношения между денотатами.

Языком называют некоторую форму существования знания в виде системы знаков, выражающих набор определенных понятий.

Употребление совокупности знаковых средств, составляющих *словарь* языка, определено *синтаксическими, семантическими и прагматическими* правилами сочетания знаков. При этом язык представляет собой определенную *иерархическую структуру*. (*Синтактика* - аспект знаковой системы, связанный с правильностью построения знака. Это система правил порождения одних знаковых систем из других. *Семантика* определяет отношения знака к значению и касается изучения знаковых систем как средства выражения смысла. *Прагматика* изучает тот аспект знака, который относится к восприятию этого знака адресатом.)

Таким образом, семиотика, оформившаяся как самостоятельная дисциплина в первой половине XX века, расширяет понятие языка по сравнению с его пониманием в традиционной лингвистике. В дальнейшем это понятие приобрело ещё больше значений. Говорят не только об языке пчёл или языке дорожных знаков (то есть о знаковых системах, которые, собственно говоря, и изучает классическая семиотика), но и об языке кино или языке математики. Появление математической лингвистики и формализация соответствующих понятий послужили основой и для создания теории формальных языков, и для практики широчайшего использования алгоритмических языков и языков программирования. Вместе с тем, освоение понятия и популяризация самого слова, привели к тому, что понятие языка стало весьма многозначным. Даже в рамках смежных дисциплин слово “язык” стало пониматься различным образом, что приводит к путанице и недоразумениям. Часто бывает не ясно, о чём же идёт речь — об языке программирования, на котором написана та или иная система, или об языке, как системе описания понятий, рассматриваемых в данном случае.

В случаях использования визуальных объектов в коммуникационной практике, информационной сфере, искусстве, также как и в компьютерных средах достаточно легко выделяются отношения между объектом визуального представления, то есть *означаемым (денотатом)*, и визуальным *знаком*. При задании некоторого *контекста* пользователь или наблюдатель (*интерпретатор*) выделяет вызванную визуализацией идею или интерпретирующую мысль (*интерпретант*). Налицо составляющие пятичленного отношения, описываемого семиозисом.

Как было сказано выше, для нас интерес представляют вопросы проектирования интерактивных визуальных систем, вопросы восприятия “визуальных текстов”, возможности мани-

пуляции визуальными объектами и взаимодействия с ними. (В качестве визуальных могут рассматриваться также такие объекты, как таблицы, схемы и пр.) Для изучения интересующих нас проблем необходимо осознать языковую природу человеко–компьютерного взаимодействия и компьютерной визуализации. Язык в данном случае понимается как систематическое описание способов представления рассматриваемых сущностей и методик их изменения, манипуляций и взаимодействия с ними. Семиотический анализ той или иной [интерактивной] визуальной системы (или [интерактивной] системы визуализации) требует выделения языка системы, то есть его словаря, синтаксиса и семантики, и, главное, этот анализ требует рассмотрения прагматики, то есть того, как воспринимаются и интерпретируются фразы языка адресатом. Этот анализ позволяет изучать существующие системы и систематизировать проектирование новых визуальных сред.

Очевидно, что здесь мы имеем дело с *визуальными языками*, то есть языками, систематически использующими визуальные значения для передачи понятий [72]. Однако в состав языка также включаются и возможности изменения положения и/или состояния объектов, интерактивные и манипулятивные возможности. Такой интерактивный язык, также как и любой другой, характеризуется своим словарем, синтаксисом, семантикой и прагматикой. В качестве элементов словаря в данном случае следует рассматривать возможные виды представления имеющихся сущностей вместе с возможностями манипуляций с ними. Именно на словарь обращается основное внимание при анализе и проектировании визуальных интерактивных систем.

В качестве примера можно рассмотреть языки систем визуального программирования, построенных на базе блок–схем [20, 54, 69]. Во всех случаях визуальный язык является диаграмматическим языком, основанным на потоке управления [1]. Набор образов для представления программных конструкций (словарь языка) диктуется стандартами на блок–схемы или же существующей традицией. Задача пользователя такой системы — выбрать и разместить элементы блок–схемы на экране. Методы соединения и размещения на экране (синтаксис) также определены самим понятием блок–схемы, как графа потока управления программы, что четко описывает и семантику языка.

Аналогично можно выполнить анализ некоторой условной интерактивной системы, где пользователь занят заполнением форм. В качестве словаря можно рассматривать возможные формы и возможные манипуляции с ними пользователей, а синтаксис задан порядком вывода форм.

В случаях сложных интерактивных сред подобное описание языка требует более тонких методов, однако в каждой интерактивной системе можно выделить языковые компоненты. При анализе видно, что языки интерактивных визуальных систем построены на некоторой основной идее уподобления сущностей моделируемой области приложения и визуальных объектов, которую мы считаем метафорой системы. Метафора, а не язык, с ней связанный, требует основного внимания при проектировании системы. Проектировщика и пользователя, как правило, больше занимают не абстракции языка системы, а конкретика видов отображения, также как и выбор (или построение) и оценка необходимых метафор. Методики семиотического анализа можно применить также и к метафорам (см., например, [36, 37]). Ниже будут рассмотрены наши подходы к пониманию оснований теории компьютерной метафоры.

1.2. Научная метафора

Компьютерная метафора рассматривалась нами, начиная с ранней работы [34], и далее в [2–4, 16, 32]. В статье [3] приведен ряд теоретических положений, связанных теорией научной метафоры и когнитивным подходом к метафоре, включая наши собственные результаты. Ниже эти положения будут частично повторены, а частично дополнены.

Исследования научной метафоры прежде всего связаны с именем Т. Куна [56, 57] (См. подробное изложение теории в [40], а также интересную отечественную работу [19]). Метафора

в этой связи понимается как использование информации и знаний из одной (исходной) области человеческого опыта для того, чтобы лучше понять и структурировать явления и понятия другой (целевой) области, которая как правило, является более абстрактной.

Научные метафоры приписывают объектам данной области ранее не выявленные у них свойства и направляют процесс поиска на обнаружение этих свойств. Тем самым абстракции как бы отождествляются с более привычными системами представления, обладающих утвердившимся набором ассоциаций, которые способствуют осмыслению новых идей [19]. Научные метафоры помогают создать язык для осознания и описания новых явлений. Метафоры такого типа особенно активно используются на ранних этапах развития той или иной научной дисциплины.

Примером научной метафоры в естественных науках служит знаменитая метафора электромагнитного поля. Хорошо известны “змеиная” метафора молекулы бензола, планетарная модель атома, изопренные метафоры физики элементарных частиц и т.п.

Научные метафоры могут играть объяснительную роль. Иногда они чрезвычайно просты и прямо отсылают к исходным понятиям, но могут также приводить к сложным выводам, далеко отстоящим от первоначальных идей. Конкретная метафора может выделять одни свойства исходной области и скрывать другие. Однако и эти, скрытые, элементы заимствования и переноса понятий, задают контекст, влияющий на восприятие метафоры, хотя он и не всегда осознается в полной мере.

Научная модель, которая создает основание для формирования теории, может рассматриваться как расширенная метафора. “Наследство” метафоры в модели связано с методами её создания и применения [19, 40]. Часто метафора и модель продолжают играть значительную роль в понимании явлений и развитии теории и тогда, когда теория, казалось бы, далеко ушла от первоначальной модели. Это вызвано сохранением её основной концептуальной структуры. Существует связь между языком метафоры и базирующейся на нее теорией. Поэтому никакая теория не является полностью абстрактной, так как должна ссылаться на нечто, описываемое в уже известных понятиях [57].

Когнитивный подход к метафоре активно разрабатывается в ставших уже классическими работах Дж. Лакоффа и его коллег [24, 60]. Этот подход рассматривает метафору как основную ментальную операцию, как способ познания, структурирования и объяснения мира. Считается, что человек не только выражает свои мысли при помощи метафор, но и мыслит метафорой, создает при помощи метафор тот мир, в котором он живет. Метафора пронизывает всю нашу повседневную жизнь и проявляется не только в языке, но и в мышлении и действии. Наша обыденная понятийная система, в рамках которой мы мыслим и действуем, метафорична по самой своей сути. Благодаря языку мы получили в свое распоряжение метафоры, структурирующие наше восприятие, наше мышление и поступки. Сущность метафоры состоит в осмыслении и переживании явлений одного рода в терминах явлений другого рода. Метафоризация основана на взаимодействии структур двух областей — исходной и целевой (source and target domains). По Дж. Лакоффу в процессе метафоризации некоторые объекты целевой области структурируются по образцу объектов исходной области и происходит метафорическое отображение (проекция) одной области на другую. То есть метафора может пониматься как отображение из исходной области на целевую область, причем это отображение сильно структурировано. (В качестве перевода понятия “domain” возможно использование также слов “пространство”, и “множество”.)

В примере классической метафоры “ЖИЗНЬ есть ПУТЕШЕСТВИЕ”, ЖИЗНЬ — целевая область, а ПУТЕШЕСТВИЕ — исходная область (область-источник). Некоторые структуры ПУТЕШЕСТВИЯ (начало, подъемы, спуски, завершение) рассматриваются в данной метафоре как основа для описания структуры жизни.

По нашему мнению необходимо сделать некоторые уточнения понимания метафоры, как отображения областей друг на друга. Ниже мы вернемся к изложению нашей позиции по данному вопросу, а пока проведем краткий анализ проблематики использования метафоры в

информационных технологиях, человеко–компьютерном взаимодействии, компьютерной графике и визуализации. С этим “метафорическим” направлением исследований смыкаются работы, посвященные использованию идей семиотики и формализации семиотических понятий при проектировании систем визуализации и человеко–компьютерного интерфейса [45, 46].

1.3. Метафора в компьютерных науках

Использование метафор в компьютерных науках в первый период их развития шло по обычным законам, описанным Т. Куном. Для создания языка новой науки, для описания новых понятий использовались привычные слова, становившиеся терминами. Эти слова приобретали новые значения и зачастую быстро теряли связь с исходной областью. Примеры таких метафор — “память компьютера”, в которой метафорическое наследование сохраняется, и “файл” данных, в которой современному, (и тем более, не англоязычному) пользователю сложно уловить следы исходного ящика с перфокартами. Интересно, что метафоры в компьютерных науках несут также “маркетинговую” функцию [43], обеспечивая броские и запоминающиеся названия для новых дисциплин и направлений исследований, что несомненно привлекает к ним внимание ученых и общественности. Всем известны, например, “искусственный интеллект”, “пятое поколение ЭВМ” или “data mining”. (Смысл последнего термина можно передать как “интеллектуальный поиск и анализ данных”).

В человеко–компьютерном взаимодействии метафоры на первых порах использовались с аналогичными целями. “Мышь”, “мышь” или джойстик (“joystick” — “палочка радости”) являются одновременно и метафорическими терминами и броскими “маркерами” новых устройств и возможностей.

Явное и активное использование понятия метафоры интерфейса началось после появления и огромного успеха метафоры рабочего стола (desktop metaphor), первоначально использованной в системах автоматизации конторского труда. (Ниже мы проведем анализ этой метафоры.) В дальнейшем и понятие и сами метафоры стали весьма популярными в теории и практике разработки человеко–компьютерных интерфейсов, а исследованиям в этой области посвящена обширная литература. Метафоры интерфейса актуализировали сами исследования в области метафоры. Из достаточно умозрительного рассмотрения одного из феноменов языка теория метафоры превратилась в инструмент проектирования конкретных программных систем. Следует отметить, что основы современной теории метафоры заложены в работах М. Блэка [38, 39], Т. Куна [56, 57], Дж. Лакоффа [24], [60] и ряда других специалистов в лингвистике и философии. Однако нас интересует специфика метафоры интерфейса, выделяющая её из других примеров использования метафоры, например, в художественной литературе, науке или политике.

1.4. Метафора интерфейса

В начале этого подраздела вспомним ситуацию, существовавшую в практике человеко–компьютерного интерфейса до появления визуальных метафор. В это время программисты (а именно они составляли основной контингент пользователей по всему миру) еще хорошо помнили работу за громоздкими пультами больших ЭВМ. В интерактивных системах господствовал командный интерфейс. Например, команды редактирования текста включали в себя указание операции, номер исходной строки (иногда и номер символа в строке) и (если необходимо) новый текст замены или вставки. Такой интерфейс, хотя и требовал от пользователя удерживать в голове постоянно изменяемое состояние текста, и, несмотря на случавшиеся казусы с сохранением этих изменений, был вполне приемлем для профессионалов. Однако внедрение массовых персональных ЭВМ просто не могло состояться без появления визуальных средств взаимодействия, новых устройств и принципиально новых концепций организации интерфейса с пользователем, например, таких, как концепция “непосредственного действия”

(direct manipulation). А визуальный интерфейс немислим без систематического использования элементов диалога, без некоторого языка, основанного на той или иной системе знаков. То есть, человеко–компьютерное взаимодействие (аналогично визуализации) может быть описано как знаковый процесс.

Роль метафоры интерфейса, понимаемой как основная идея уподобления интерактивных объектов и объектов прикладной области, заключается в том, что она способствует лучшему пониманию семантики взаимодействия, а также обеспечивает визуальное представление диалоговых объектов и определяет набор манипуляций пользователя с ними. Метафора интерфейса в этой связи представляется основой знаковой системы, которая в свою очередь лежит в основе диалогового языка. С помощью этого языка пользователь формулирует свою задачу и добивается от компьютера её решения.

Таким образом, функции метафоры интерфейса, значительно шире функций научной метафоры, так как она не только помогает описать абстракции, структурирует понимание новой прикладной области, но и задает объекты диалогового [визуального] языка. Отсюда следует, что метафоры интерфейса нужно рассматривать и с позиций теории научной метафоры, и с позиций когнитивного подхода к метафоре. Соответственно необходим семиотический анализ [визуальных] языков взаимодействия, получающихся на основе метафор интерфейса. То есть теоретический аппарат теории метафоры интерфейса включает в себя как различные аспекты теории метафоры, так и семиотику.

Отметим в этой связи наличие ещё одного класса компьютерных метафор — метафор системы (system metaphors), используемых в сравнительно новой методологии создания программных систем, известной как экстремальное программирование (Extreme Programming — XP). Метафора системы должна поддерживать взаимодействие проектировщика системы и её потенциального пользователя, который в принципе может не быть знакомым с соответствующей прикладной областью. Метафора системы структурирует ментальные модели участников проектирования и формирует логическую архитектуру будущей системы, что связано с необходимостью единства представления объектов проектирования для всех участников разработки и поддержания этого единства по ходу работы. В каком–то смысле метафора системы имеет визуальную природу, так как участники разработки активно используют графические представления в ходе коммуникаций. Описанный в [55] семиотический подход к рассмотрению этого класса метафор облегчает их разработку и использование, а также позволяет получить какие–то оценки качества.

1.5. Метафора визуализации

Визуализация (правда, с некоторой натяжкой) может рассматриваться, как особый случай человеко–компьютерного взаимодействия. Следовательно и метафора визуализации может изучаться, как частный случай метафоры интерфейса. Однако, хотя эти области близки, а по ряду пунктов и пересекаются, следует различать не только две самостоятельные дисциплины, но и два случая использования компьютерных метафор.

Компьютерная визуализация понимается здесь как методика перевода абстрактных представлений об объектах в геометрические образы, что дает возможность исследователю наблюдать результаты компьютерного моделирования явлений и процессов [79]. Алгоритмические и программные методики генерации изображения относятся непосредственно к машинной графике. Тогда как основная задача визуализации — обеспечение интерпретации и анализа результатов компьютерного моделирования.

Выделяются три подобласти компьютерной визуализации — научная визуализация, визуализация программного обеспечения и информационная визуализация. Это разделение проводится по различным направлениям и задачам приложений визуализации. Кроме того, следует рассмотреть три функции визуализации — иллюстративную, коммуникативную и когнитивную. Те же функции присущи и традиционной, “бескомпьютерной” (или “докомпьютерной”)

визуализации, которая понимается обычно, как процесс формирования зрительных образов для объектов, не имеющих зримых форм, или, как интерпретация явлений на визуальном языке.

Выше уже говорилось, что визуализация (и традиционная, и компьютерная) может быть описана, как знаковый процесс. Соответственно в качестве её оснований могут быть положены элементы понятийного аппарата семиотики.

В литературе было предложено несколько подходов к выявлению основ визуального языка, например, на базе элементарных форм, цветов, тонов и тому подобных признаков. (См. обсуждение проблем в [28].) Однако, представляется, что эти подходы не позволяют объяснить того, как эти языки воздействуют на наблюдателей (в нашем случае, пользователей системы). Необходимо выделение более крупных элементов восприятия визуализации пользователем.

В визуализации легко выделить составляющие знаковой ситуации, считая, например, конкретный графический вывод комплексным знаком, а визуализируемую сущность объектом, который обозначает знак. Однако, интерпретация единичной визуальной ситуации, находящейся вне некоторого контекста (как было сделано, например, в [71]), является проблематичной. Такое понимание явно недостаточно для получения интересных и полезных результатов ни при анализе визуальных систем, ни при их проектировании. Более продуктивным является рассмотрение конкретной визуализации как знаковой системы. Здесь в наличии есть набор сущностей визуализируемой модели и набор соответствующих им визуальных образов. Анализ отношений между модельными сущностями позволяет выстроить достаточно чёткие отношения между отдельными визуальными образами, в частности, при разработке конкретного примера визуализации. Однако, при проектировании полноценной специализированной системы визуализации мало описания знакового соответствия. Нас в этом случае интересует прагматика системы, а также правила представления визуальных образов. Необходимо сделать следующий шаг и выделить язык визуализации в полном семиотическом смысле этого термина.

Дадим определения ряда понятий компьютерной визуализации, частично используя заимствования из [59]:

Модельная сущность есть объект вычислительной модели, требующий изучения, чье состояние и поведение, свойства, атрибуты и особенности интересуют исследователя и, как следствие, подлежат визуализации. Абстракция модельной сущности может быть определена через выделение следующих категорий:

- спецификации характеристик моделируемого явления, которые должны быть получены из имеющихся данных;
- требуемый качественный анализ модельного объекта и его свойств;
- семантические атрибуты модельного объекта.

Вид отображения определим как абстракцию графического вывода, содержащую спецификацию визуальных объектов, их атрибутов, их взаиморасположения, возможной динамики и способов взаимодействия. При этом визуальные формы абстрактных данных не связаны ограничениями, накладываемыми определенными графическими системами.

Можно говорить о видах отображения, как о стандартных или придуманных для данного случая методиках визуального представления данных, своего рода, визуальных процедурах, которые при реализации в конкретных визуальных средах и при подстановке реальных данных выводятся на те или иные графические устройства. Кроме этого, в такой “процедуре” (то есть в виде отображения) могут предусматриваться возможные изменения изображений, включая анимацию, и допустимые способы взаимодействия с картинкой. Именно смена значимых и значащих картинок при возможном взаимодействии с изображением — вот внешняя сторона визуализации. Эти картинки (графические выводы) являются воплощением абстрактного понятия вида отображения.

Набор видов отображения данной системы можно считать словарем некоторого языка, тогда как в качестве грамматики можно рассматривать наборы правил образования конкретных

выводов и правила, задающие последовательность смены изображений.

Таким образом, достаточно легко осуществляется выделение языковых составляющих в визуализации. Теперь можно описать язык визуализации по всем правилам семиотики для каждого конкретного случая (так как в каждой системе имеет место свой конкретный язык).

Визуальное проектирование должно включать в себя учет знаний об абстракции модельных сущностей, обеспечивая тем самым основу для интерпретации различных аспектов модели. Основой визуального проектирования является проектирование видов отображения. Абстракция визуализации подразумевает связывание модельных сущностей с видом отображения так, чтобы суть, поведение, особенности и атрибуты модельных сущностей могли быть представлены в конкретном графическом выводе, точно идентифицирующем все визуальные свойства, в которые переходят атрибуты соответствующего вида отображения.

Понятие метафоры визуализации вводится для обобщения случаев использования метафоры (или близких к ней аналогий) во всех областях компьютерной визуализации и визуального человеко-компьютерного взаимодействия.

Определим метафору визуализации как отображение, ставящее в соответствие понятиям и объектам моделируемой прикладной области систему сближений и аналогий и порождающее некоторый изобразительный ряд (набор видов отображения) и набор методов взаимодействия с визуальными объектами.

Таким образом, предлагается (расширительный по сравнению с традиционным) подход к пониманию метафоры, как главной идеи при отображении прикладной области на визуальный мир.

Метафора, в противоположность стабильному знаку, может быть описана динамически, как акт или процесс обозначения одного понятия посредством знака, традиционно связанного с другим понятием. Визуализация есть знаковой процесс. Визуальное взаимодействие с ЭВМ, включая и визуальное программирование, тем более основано на знаковых системах. Визуальная знаковая система немыслима без использования аналогий, определяющих наше представление об исследуемом явлении. При этом, в отличие от литературной метафоры, мы должны взаимодействовать с нашим представлением, анализируя при этом полученные образы.

Выбор метафоры — это выбор знаковой системы, описание системы знаков, которая будет использоваться при визуализации. Другая функция метафоры состоит в задании контекста, помогающего правильной интерпретации элементов данного языка визуализации, выявлению значения визуального текста. Таким образом, метафора визуализации обеспечивает понимание отображаемых сущностей прикладной области, а также участвует в создании новых сущностей на базе внутренней логики самой метафоры [2].

Метафора в случае научной метафоры, визуальной метафоры и метафоры интерфейса является операционной, то есть помогающей принимать решение и действовать (в отличие от созерцательной художественной метафоры).

Интересно рассмотреть понятие “работа метафоры” (или “действие метафоры”). В данном случае мы имеем дело не с количественной характеристикой. Это скорее ответы на вопросы:

“как с помощью данной метафоры можно представлять информацию?”,

“какие свойства появляются у метафорических объектов?”,

“к каким результатам, то есть действиям (или мыслям и идеям в случае неинтерактивной визуализации) приводит пользователя взаимодействие (включая наблюдение, в случае неинтерактивной визуализации) с метафорическими объектами?”.

2. Анализ компьютерных метафор

В этом разделе анализируются несколько разноплановых метафор интерфейса и визуализации, среди которых повсеместно используемая метафора рабочего стола (*desk top metaphor*), метафора комнаты, а также метафоры, использованные в узко специализированных системах научной и информационной визуализации. Обращается внимание на генезис метафор и их

“действие”, то есть на возможности представления данных и манипуляции с ними при использовании конкретных метафор, потенциальные свойства метафорических объектов, а также потенциальные возможности пользовательской интерпретации этих объектов и манипуляций с ними. Цель анализа — выявить структуры успешных метафор и получить основу для сравнения и оценки метафор.

2.1. Метафора рабочего стола

Метафора рабочего стола уже давно используется на сотнях миллионах персональных компьютерах по всему миру. Благодаря ей понятие метафоры стало популярно среди специалистов в человеко–компьютерном взаимодействии. Формальный анализ этой метафоры был выполнен в работе [59], опубликованной ещё в 1991 году. Нас же интересует неформальное рассмотрение истории возникновения и возможностей по представлению сущностей интерфейса.

Впервые метафора рабочего стола была опробована в визуальном варианте системы Xerox Star, предназначенной для автоматизации конторского труда (office automation system). Автором этой метафоры является Д. Смит (D.C. Smith), который ещё в 1975 году успешно опробовал иконический интерфейс в системе программирования путем демонстраций Pegasus [80]. На рабочем столе Xerox Star были размещены иконы, представляющие документы, папки с бумагами, принтеры и т.п. Одновременно на экран выводилось несколько окон, содержащих как иконическую информацию, так и графику, дополненную текстами. Конечно, иконические возможности Xerox Star кажутся архаичными по сравнению с возможностями современных версий системы Windows, но начало было положено, и именно Xerox Star была первой коммерческой системой, активно использовавшей иконический интерфейс, и именно она оказала заметное влияние на развитие других систем. Важно, что первоначально метафора рабочего стола была применена в системе автоматизации конторского труда. Конторский стол, как область–источник, породил такие реалии, как произвольное размещение папок на поверхности, также, как и саму идею папки с документами (включая внешний вид американской папки, отличной от отечественной с завязочками).

Потенциальные возможности метафоры рабочего стола по организации иконического языка допускали различные решения. Вот идея, реализованная в одном из вариантов системы иконического программирования HI-VISUAL [51], также предназначенном для автоматизации конторского труда. (Авторами HI-VISUAL отмечается влияние Xerox Star при использовании элементов метафоры рабочего стола.) Здесь вообще нет специальных пиктограмм, представляющих функции. Использование пиктограмм ограничено представлением реальных объектов или понятиями уже введенными в заданной прикладной среде. Например, в одном из вариантов HI-VISUAL присутствовали пиктограммы “бухгалтерская книга” для представления данных о работе фирмы за год; “калькулятор”, для операции составления годового отчета; “готовальня” для графического вывода отчета и т.п. Функции представлялись путём комбинации двух различных пиктограмм, а точнее, совмещения в определенном порядке одной пиктограммы с другой.

Первые реализации метафоры рабочего стола для систем автоматизации конторского труда не получили по–настоящему широкого распространения. Кроме недостаточного технического уровня интерфейса свою роль вероятно сыграли и принципиальные особенности проектов. Думается, что здесь следовало большее внимание уделять представлению непосредственно самого документа, а не только папок с бумагами. (Ниже мы рассмотрим сочетание метафоры рабочего стола с метафорой картотеки в специализированной системе, предназначенной для автоматизации работы муниципальных служащих.) Известность метафоре рабочего стола принёс её перенос из первоначальной области применения — систем автоматизации конторского труда в другую область — пользовательский интерфейс визуальной оболочки операционной системы, где она претерпела некоторые изменения. Теперь на рабочем столе кроме папок размещаются иконы (пиктограммы), представляющие исполняемые программы. Вызов программ (также

как и открытие папок) осуществляется посредством двойного “клика”, двойного щелчка левой клавиши мыши. Свои значения имеют и щелчок правой клавишей, и простое наведение маркера на пиктограмму. Таким образом, в рамках данной метафоры получился очень простой иконический язык со словарем, состоящим из набора пиктограмм и манипуляций с этими пиктограммами при помощи мыши. Визуальный синтаксис практически отсутствует, так как месторасположение пиктограмм на поверхности стола не значимо.

На примере всем известной метафоры рабочего стола, про историю возникновения которой мы знаем достаточно много, можно построить некоторую “формулу” работы метафоры. Конечно, нам неизвестны реальные мысли авторов идей метафоры рабочего стола. Мы восстанавливаем возможный ход рассуждений, чтобы получить цельную (и, кстати, достаточно правдоподобную) картину работы по созданию метафоры. Отметим, что в случае анализа других метафор, у нас может не хватить данных для аналогичных построений. Поэтому то этот опыт должен пригодиться в дальнейших исследованиях.

В “формулу” метафоры очевидно следует включить упрощенное описание исходной и целевой областей, идею уподобления и результат действия метафоры. Тогда формулу метафоры рабочего стола можно записать следующим образом:

Исходная область:

офисный (конторский) стол, на котором лежат папки с документами (документы структурированы, сами папки — неупорядоченные);

Целевая область:

система автоматизации конторского труда с задачей обеспечить хранение и обработку документов.

Идея уподобления:

“папки с бумагами” = “структура данных, набор файлов”;

“открытие папки” = “показ структуры файлов и/или самих файлов”;

“обработка документов” = “исполнение функции, как выполнение команды некоего визуального языка”.

Результат:

непосредственный доступ к структурам данных за счет манипуляций с размещенными на экране иконами (пиктограммами) их представляющими; вызов некоторого набора предопределенных и задаваемых пользователем функций за счет использования визуального диалогового языка.

При переносе метафоры на случай организации интерфейса с оболочкой операционной системы рассмотрим случай ОС Windows. (Другие интерфейсы, например, для операционных систем компьютеров фирмы Apple рассматривать не будем. Хотя там иначе организована работа мыши, но принципиального различия в использовании метафоры нет.)

Дополнение исходной области:

Конторский стол совмещается с пультом, на котором размещены кнопки, запускающие процессы.

Кроме того добавляется некая “волшебная” идея:

Все действия в рамках системы производятся посредством двойного “клика” по пиктограмме.

Теперь данные доступны после двойного “клика” по пиктограмме, обозначающей папку — некую структуру файловой системы. Обработка документов производится через вызов программы, и этот вызов логично осуществлять таким же способом — через двойной “клик”.

Продолжим возможные логические рассуждения при построении метафорических операций.

Если обработка документов есть вызов программы, то вызов любой программы должен осуществляться через такое же двойное “клика” по пиктограмме, представляющей программу.

Результат:

Размещение на экране иконов (пиктограмм), представляющих как структуры данных, так и вызовы программ.

Структуры данных и программы обрабатываются одинаковым образом, как и положено в фон-неймановской машине.

Следует добавить к рассмотрению еще одну идею Windows — открытие новых окон с началом исполнения программ.

Можно говорить о появлении некоторого “метафорического” пространства интерфейса, построенного на основе реальностей конторского стола. Не все реалии офисного стола (области-источника метафоры), которые одновременно и богаче и беднее метафорических объектов, одинаково пригодились в новом метафорическом пространстве. Перемещение пиктограмм по экрану служит в основном для их группировки и соответственно для удобства работы конкретного пользователя. Папки с документами не играют решающей роли при работе пользователей операционной системы и зачастую постепенно вытесняются с “рабочего стола”. Зато важнейшее значение получил (не имеющий аналогов в исходной области) запуск программы по двойному “клику”, как правило, вместе с открытием нового окна. (Причем, в случае интернет-браузеров окно открывается почти в буквальном смысле.)

В результате получилась внутренне логичная система команд визуального (иконического) языка, основанная на базовой операции — двойном “кликанье” по пиктограмме. (В связи с этим вспомним существующие в визуальных языках проблемы с представлением функций и процессов и попытки их решения за счет либо использования определенных ухищрений визуального синтаксиса, либо использования различных меню и световых кнопок.)

Метафора рабочего стола стала основой удобного и понятного пользователям интерфейса. Успех метафоры несомненно связан не только с естественной образностью иконов, понятной пользователем ОС, но и с логичностью и системностью всего визуального диалогового языка. (Отметим, что успех метафоры рабочего стола связан также с наличием в сознании людей глобальной метафоры [компьютерного] мира как некоего сверхофиса.) Широчайшее распространение метафора получила благодаря распространению Windows, в свою очередь связанному не только с качествами самой ОС, но и со всей мощной инфраструктурой Microsoft. При более внимательном рассмотрении оказываются безосновательными претензии к данной метафоре, касающиеся навязывания пользователям-программистам чуждого для них понятийного ряда, заимствованного из конторской деятельности [2]. Задачи организации диалога с визуальной оболочкой современной ОС вообще не специфичны ни для какой профессии. Тем более, известно, что в случае крайнего раздражения иконическим интерфейсом пользователи-программисты просто переходят на использование командной строки. Интерес к метафоре рабочего стола сохраняется и до сих пор. Исследования по расширению набора операций в её рамках и самого содержания этой метафоры продолжаются и привели к интересным результатам [52], [63]. Успешное использование метафоры, связанной с работой в офисе, породило попытки эксплуатации данной области-источника для новых метафор. В частности была предложена метафора рабочей комнаты, о которой мы поговорим в следующем подразделе.

2.2. Метафора комнаты

Интерес к метафоре комнаты возник сразу после удачного внедрения метафоры рабочего стола. Разработчики интерфейсов попытались уже в первой половине 90-ых гг. развить успех и освоить средства трехмерного интерфейса на базе метафоры рабочей комнаты. Кроме того, в качестве источника метафоры комнаты можно рассматривать представление различных помещений в многочисленных компьютерных играх. (Отметим, кстати, что игры являются неисчерпаемым источником идей для построения метафор интерфейса и визуализации.) Несмотря на ряд интересных решений, чего-либо по настоящему сравнимого с интерфейсом на базе метафоры рабочего стола достичь все же не удалось. Применение средств виртуальной реальности в рамках этой метафоры не спасло положения, хотя разработчики пытались активно имитиро-

вать реальность офиса. (В одной из систем, например, можно было при помощи специальной перчатки открывать и закрывать ящики рабочего стола.) Наш анализ служит пониманию причин того, почему интересная идея, приложимая к различным предметным областям, осталась недостаточно востребованной в практике визуализации.

Есть целый ряд интересных решений пользовательского интерфейса с использованием метафоры комнаты в двумерном варианте (например, TeamWave WorkPlace [49], Clear [67]). Однако в этих реализациях нет сильных отличий от метафоры “рабочего стола” в плане визуального представления, а следовательно нет и выигрыша в выразительности.

Возможно основной характеристикой, которая побуждает использовать метафору комнаты в человеко-компьютерном интерфейсе, является ее естественность, наличие хорошо известного аналога в реальном мире. Примером здесь может служить трехмерное расширение метафоры рабочего стола, а точнее расположение нескольких “рабочих столов” на стенах комнаты, внутрь которой “смотрит” пользователь. Такое решение рассматривалось в проекте интерфейса новой ОС фирмы Microsoft — Windows Vista. Аналогично использует метафору и трехмерный браузер, точнее надстройка Cubic Eye от компании 2Ce над браузером Microsoft Internet Explorer, в некотором смысле “прародитель” этой идеи. В этом браузере на стенах “комнаты” отображаются загруженные из Интернета страницы.

Перспективным представляется использование различных модификаций метафоры комнаты в системах визуализации программного обеспечения.

К системам, неявно использующим метафору комнаты в трехмерном варианте, относится система Avatar [68], предназначенная для отладки производительности параллельных программ и функционировавшая на базе театра виртуальной реальности CAVE. Эта визуальная среда была предназначена для представления очень больших объемов данных о производительности параллельных систем, получаемых непосредственно в ходе их работы. Пользователь, оказывается внутри трехмерного помещения, где на стены проецируется видеоизображение. (В этой связи в системе используется понятие “scattercube”, производное от традиционного вида отображения “scatterplot”.) На внутренних гранях показаны оси, а на гранях куба — полу и стенах помещения выводятся кривые, описывающие метрики производительности параллельной программы. Реализован режим прозрачности потолка и пола. Рассматривается объединение отдельных элементов (scattercube-матрица), которое напоминает стеклянный небоскреб, каждая из комнат которого содержит трехмерный вывод, описывающий различные аспекты поведения параллельной программы. Путешествие (“виртуальный полет”) по небоскребу дает возможность полного исследования данных о производительности прикладной параллельной программы. Обратим внимание на то, что в системе Avatar данные о производительности параллельных программ отображаются в виде традиционных двумерных графиков.

Существует интересный пример расширения метафоры комнаты для визуализации функционирования параллельного вычислителя в системе INHOUSE [64, 75]. В этом случае вся мультипроцессорная система представляется как отель. Комнаты в нем представляют отдельные процессоры. Связь между процессорами представляется за счет внутригостиничной телефонной сети. В комнатах-процессорах находятся чемоданчики, представляющие процессы, которые размещены на этих процессорах и которые можно перемещать с процессора на процессор. На последнем уровне находится визуальное представление для самой программы. Для этого используются фрагменты картинок-головоломок, находящиеся в чемоданчиках-процессах. Картинки состоят из элементов структурированного набора форм, представляющих различные программные конструкции и размещаемых в определенном порядке.

Таким образом, в качестве целевого множества, представляемого метафорой комнаты при параллельном программировании (в рамках парадигмы обмена сообщениями), естественным образом выбирается процесс. Тогда упорядоченная (как правило, в здании) совокупность комнат представляет всю параллельную программу как совокупность параллельных процессов. Связи между комнатами в данном случае представляют передачу сообщений или данных. А

применение средств даже простой анимации (например, изменение толщины или цвета линий связи) может неплохо служить для представления динамики программы.

Другим примером применения метафоры является визуализация иерархии классов программы в [31]. На базе метафоры комнаты получен трехмерный аналог диаграммы классов языка UML. В качестве объекта предметной области, который представляет комната, рассмотрен класс. Содержимое класса (данные и методы) являются содержимым комнаты, а отношения наследования/агрегации — связями между комнатами. Иерархия классов очень важна на этапе проектирования программы. От удачного ее построения во многом зависит удобство написания и внесения изменения в программу, что в итоге сказывается на ее эффективности.

Нами предпринимался ряд попыток реализации метафоры комнаты в макетных системах визуального программирования и настройки производительности [11, 23].

В одной из таких систем метафора комнаты использовалась в рамках системы визуального программирования. Пиктограммы, размещенные на стенах, представляли управляющие конструкции и данные программной функции. Связи между конструкциями показывались в пространстве, а не на плоскости, как в традиционных двумерных диаграмматических и иконических системах визуального программирования. Вся программа в этом случае представлялась зданием (по примеру Avatar), а интересующая пользователя комната-функция вытягивалась из здания подобно блоку при блочном строительстве.

В другой нашей реализации метафора комнаты применялась при визуализации графа вызовов. Графы вызовов (call graphs) могут использоваться для анализа и настройки производительности сложных программных комплексов, в том числе распределенных и параллельных программ. Отображая различные характеристики работы программы в характеристиках комнаты, получим некое комбинированное отображение, по которому можно судить о “слабых” в смысле эффективности работы местах программы. Можно обнаружить такие важные часто вызываемые функции, функции, в которых программа “проводит” значительное время, функции, ответственные за передачу данных между процессами (это важно для синхронизации программы и минимизации простоя узлов многопроцессорного вычислителя) и так далее. Существуют и другие применения графа вызовов, например, при анализе потенциального параллелизма программы. Граф вызовов был представлен как набор связанных между собой помещений. Каждая комната — визуальное представление функции. Кроме того, за функцию “отвечает” и пиктограмма на стене комнаты функции-родителя в графе вызовов. Ребра графа естественным образом переносятся на наше изображение — они следуют от пиктограммы функции на стенке родителя к комнате, соответствующей данной функции. Наибольшую реалистичность изображения дает вид отображения “изнутри” комнаты в сочетании с возможностью “путешествия” между комнатами.

В случае метафоры комнаты нам не удаётся вывести формулу метафоры, так как нет единой для нее области приложения, нет успешного и убедительного опыта ее использования, а, значит, нет единства целевой области. Однако приведем обзор свойств метафоры и методов представления с ее помощью информации.

Метафора комнаты обладает следующими свойствами:

1. Способность содержать какие-либо объекты внутри себя. Комната не только представляет собой отдельный объект, но и является контейнером для других объектов.
2. Ограничение контекста восприятия. Объекты внутри комнаты можно рассматривать в отрыве от “внешнего мира”.
3. Замкнутость. Для работы с метафорой комнаты не требуются дополнительных элементов, кроме, возможно, объектов, расположенных внутри комнаты.
4. Включение в структуру. Из комнат можно “строить здания”, рассматривая совокупность комнат. Поэтому комната может являться элементом построения некоторой сложной конструкции.
5. Естественность метафоры. Комната является естественной метафорой, с наличием соответствующего объекта в реальном мире. Это свойство делает интуитивно понятными все

вышеописанные свойства. При использовании метафоры не возникает дополнительных аналогий и неестественных образов. Функциональность и характеристики реального объекта просто переносятся (хотя и в несколько расширенном понимании) в виртуальный мир.

Рассмотрим, какими способами можно передать информацию при использовании метафоры комнаты.

Так как комната есть контейнер, то естественно в качестве первичного способа представления использовать помещение в нее объектов. Причем можно рассматривать различные способы размещения объектов внутри комнаты. С одной стороны информация может быть представлена типом (видом) объекта, без учета его размещения, с другой стороны можно рассматривать однотипные объекты, а основную информационную нагрузку будет нести их местоположение в комнате. Так же можно рассматривать одновременное применение обоих способов. Располагать объекты внутри комнаты естественней всего на ее “стенах”. Однако расположение элементов во внутреннем пространстве, между стенками также может быть оправдано. Заметим, что расположение объектов на “стенах” комнаты является естественным. В повседневной жизни нас окружают комнаты, в которых на стенках расположены объекты, третье измерение которых часто не играет значительной роли в восприятии (в качестве примера — картина на стене). Информацию может нести и интерьер комнаты. Под интерьером можно понимать такие характеристики, как цвет и геометрическая форма. То есть, если рассматривать совокупность комнат, то по форме и/или цвету можно выделить различные их классы. Например, возможен обзор динамики выполнения программы “изнутри” посредством использования формы для передачи типа программной конструкции. Можно придумать соответствующую форму для каждой значимой языковой конструкции. Цвет комнаты вычисляться на основе содержания “комнаты”–функции. Цвет, например, может определяться количеством наследников или объемом обрабатываемых данных. Можно рассматривать совокупность разнотипных комнат. В этом случае связи между комнатами могут передавать структурные отношения в сложном объекте. При таком рассмотрении можно применить еще две характеристики — местоположение комнаты в пространстве и ее положение относительно других комнат. Также можно представить предопределенное, строго заданное расположение комнат в пространстве (стена, небоскреб и т.п.). Но такое расположение может нести меньший объем информации о комнатах, составляющих структуру.

Динамическое изменение характеристик комнат во времени может являться источником информации. Возможна анимация в рамках всей совокупности комнат. Кроме того, анимация может затрагивать не только изменение пространственного положения, но и других характеристик комнат — цвета, размера, формы и т.д.

В литературе встречается несколько попыток использования идеи контейнера (и не только в рамках метафоры комнаты — см. [61]). Но на наш взгляд в этих случаях даже интересное применение эффекта прозрачности не сильно повышает уровень интерпретации. Возможно, что частичные неудачи связаны с ограничениями в восприятии структур как объектов, вложенных друг в друга. Проводя анализ свойства ограничения контекста восприятия, оказывается, что это не всегда хорошо. Например, при работе с совокупностью взаимосвязанных комнат, может быть полезно посмотреть, что находится в комнатах, связанных с данной, причем, “не переключая” текущую комнату, то есть, не сменяя всего контекста. (Такая возможность реализована в системе Avatar.)

Можно рассмотреть и другие идеи использования метафоры комнаты при визуализации программного обеспечения.

Если существует какое-то разделение программной системы на отдельные модули со своими параметрами и механизмами обработки, то работу с системой можно построить с помощью разделения “рабочего пространства” на комнаты.

В качестве небольшого примера рассмотрим систему визуализации данных с возможной их фильтрацией [18]. Здесь можно выделить две “комнаты”. Первая — для получения, сохранения и предварительной обработки данных. Вторая — для создания и применения к ним фильтров.

В случае если работа с программой строится в виде последовательных шагов, то это может быть представлено как “путешествие” по комнатам. Из комнаты получения данных можно перейти в комнату применения фильтров, где на стенах комнаты можно изучать и сравнивать результаты.

Можно взглянуть на параллельную программу не с позиции процессов, а с позиции её данных. Кажется удобным с помощью метафоры комнаты описывать составные структуры данных программы (такие как массивы, структуры, объекты). Пользуясь свойством комнаты как контейнера, разместим внутри нее элементы составных типов данных. Кажется, что наибольший эффект от такого визуального представления данных можно ожидать в параллельных системах с общей памятью. Здесь также напрашивается рассмотрение динамики. Таким образом, будет возможность изучить как (и что также немаловажно — где) меняются данные. Это упрощает, например, поиск мест программы, где обрабатывается и/или пересылается большое количество данных.

Анализ метафоры комнаты показал важность вычленения основных свойств метафоры и способов представления с ее помощью различных объектов. Краткий (и естественно неполный) обзор показывает также, что трехмерный интерфейс (да еще и с использованием средств виртуальной реальности) возможно недостаточно удобен в массовых системах. Применение метафоры комнаты в визуализации программного обеспечения, конечно, перспективно. Однако в этом случае проблемы начинаются уже на этапе выбора тех программных сущностей, которые затем подлежат визуализации. Как правило, приходится работать с величинами, лишь косвенно отражающими реальное поведение процессов. Поэтому пользователь при использовании даже самых удачных метафор визуализации все равно сталкивается с множеством трудностей при анализе и интерпретации своих программ.

2.3. Метафора картотеки

Этот подраздел основан на опыте использования метафоры при проектировании пользовательского интерфейса для специализированной системы автоматизации конторского труда, предназначенной для муниципальных служащих — сотрудников паспортных столов. В подразделе описывается часть реального процесса принятия и обоснования решений в ходе разработки интерфейса клиентской части системы “Паспортный стол”, разрабатываемой в одной из крупных программистских организаций.

В начале работы была всесторонне изучена предметная область, то есть процесс трудовой деятельности сотрудников паспортных столов. Кроме того, особое внимание было уделено получению сведений о самих будущих пользователях системы, об их менталитете, сложившихся профессиональных привычках, уровне образования и способностях к обучению.

На основе полученных знаний была проведена работа по выбору и адаптации метафоры для приложения. В качестве главной метафоры естественно было использовать метафору рабочего стола. Локальной метафорой приложения была взята метафора картотеки.

Следующим этапом работы стало проектирование видов отображения. Оно основывалось на знаниях об операционной деятельности будущих пользователей. Конкретика реализации тех или иных задуманных элементов комплексного вида отображения здесь не рассматривалась. Эти задачи были решены на последнем этапе разработки, этапе конкретизации каждого вида отображения.

Далее мы рассмотрим как обоснование выбора метафор, так и анализ метафоры картотеки с позиций ее использования в конкретной среде автоматизации конторского труда.

Муниципальные служащие много работают с документами и различными формулярами, поэтому метафора рабочего стола как нельзя лучше отражает привычную рабочую обстановку и способна задать основу интерфейса.

В качестве второй, локальной метафоры приложения была выбрана естественная для сотрудников паспортных столов идея картотеки, представляющей собой набор ящичков с карточ-

ками. Рассмотрение карточек и картотечных ящиков (как совокупности карточек) в качестве структурных единиц картотеки, а также наличие правил составления картотеки из ящиков позволяет полностью определить локальную метафору.

Таким образом, получившуюся комплексную метафору можно охарактеризовать как “работу с картотекой, размещенной на рабочем столе”.

Выделим некоторые особенности метафоры картотеки, повлиявшие на сделанный выбор.

Во-первых, картотека является естественной метафорой. Существует общеизвестный объект реального мира, с которым имели дело все паспортисты. Знание объекта и опыт работы с ним очень важны для успешного освоения программы. Особенно это касается неопытных пользователей. Также в связи с предварительным анализом психологии восприятия и особенностей когнитивной деятельности потенциальных пользователей представляется нецелесообразным использование абстрактных понятий при выборе метафор. Поэтому лучшим решением является использование метафоры реального объекта. Трудовая деятельность должна быть отделена от бытовой, поэтому бытовые реалии также не подходят в этом случае.

Во-вторых, важную роль играет наличие структуры в рамках данной метафоры. Само понятие картотеки подразумевает четкое структурирование данных. Использование этой метафоры дает позитивный психологический эффект. Пользователь уверен, что всегда быстро найдет нужную информацию в структурированном хранилище. Он даже может составить ментальную модель такого хранилища данных программы на основе знаний о бумажных картотеках. Это даёт лучшее понимание работы с системой. Можно сказать, что эта метафора, как и реальный объект, выполняет организующую функцию. Информационная функция заключена в хранящихся карточках и непосредственно в строении картотеки. Данные о структуре картотеки позволяют более эффективно использовать всю программу.

Карточка, как отдельная структурная метафора, также очень важна для понимания локальной метафоры. Её можно считать единицей информации в картотеке, поскольку она отражает данные, связанные с одним объектом. Метафора карточки существенно проще метафоры картотеки, поэтому отметим лишь, что реальная бумажная карточка паспортиста имеет свою структуру и потому достаточно чётко определяет метафорическую карточку как визуально, так и функционально.

Существует еще один тонкий для понимания момент в работе паспортистов. В реальном мире диалог с людьми (например, просьба зарегистрировать человека по данному адресу) переносится на действия над карточками. Это еще одна неявная [некомпьютерная] метафора, к которой привыкли муниципальные служащие. Работа идет не с гражданами или семьями, а с их личными карточками или наборами карточек соответственно. Помня это, мы снова обращаемся к преимуществам метафоры картотеки, привычной и легко понимаемой.

При работе с картотекой пользователям интересны не сами ящики как таковые, а множество содержащихся в них карточек. Сама же концепция картотеки создает у людей определенное представление о хранении данных и вообще о работе системы. Многие сотрудники практически не имеют опыта работы на компьютере. Очень важно, чтобы они воспринимали новый порядок работы, как нечто им знакомое, похожее на то, чем они занимались раньше. Метафора картотеки как нельзя лучше выполняет эти задачи.

И структура электронной картотеки и возможности работы с ней, конечно, отличается от той, что используется в реальных паспортных столах в “бумажном” варианте. Имеет место (традиционный для систем офисной автоматизации) автоматический поиск карточек по первым буквам фамилии. После исправления карточка автоматически ставится на место. Работа происходит с конкретными, выбранными из картотек карточками, с целой “стопкой” документов. Причем в стопку попадают также те карточки, которые могут быть задействованы на следующем этапе данной операции, хотя паспортист еще не успел указать на требуемые имена (например, карточки родных человека при прописке). То есть метафора картотеки включает дополнительные операции, которые достаточно сложно реализовать в “бумажном” варианте работы, но которые очень полезны в специализированной системе автоматизации конторско-

го труда. Метафорические операции расширяют возможности операций исходного множества. Тесная связь интерфейса, построенного на базе естественной метафоры, с опытом “бумажной” работы вместе с добавлением давно желаемых “волшебных” операций (например, немедленное возвращение карточки на свое место после окончания ее обработки или появление в активной “стопке” карточек всех членов семьи сразу) позволяет пользователю быстро адаптироваться к компьютерной среде.

Можно отметить близость исходного и целевого множеств в данной метафоре. И в том и в другом случае пользователь имеет дело с документами определенного типа, только в одном случае используется бумажный, а в другом электронный вариант документов. Однако существенные дополнения пользовательских возможностей в случае системы автоматизации конторского труда создают новое метафорическое пространство, в котором и осуществляется теперь деятельность служащих–паспортистов.

2.4. Метафора в специализированных системах визуализации

Существуют две тенденции развития систем визуализации. С одной стороны — разработка универсальных средств визуализации, а с другой — специализация по всем направлениям, вплоть до создания специальных графических станций и реализации специально для данного случая графического, алгоритмического и программного обеспечения.

Таким образом, системы визуализации можно разделить на универсальные и специализированные.

Важной особенностью универсальных систем является наличие типового набора видов отображения для типовых моделируемых объектов. В принципе, при помощи универсальных систем можно отобразить любые объекты (конечно, в рамках применимости данной системы, например, информационных объектов для универсальной системы информационной визуализации). Другой вопрос, какие усилия требуются от пользователя, причем не только (и не столько) при самой визуализации, сколько при интерпретации результатов.

Специализированные системы поддерживают решение определенного класса задач, обслуживают определенный класс пользователей и содержат свои методики визуализации. Они облегчают работу пользователя, а, в случае исследования принципиально новых модельных объектов, только за счет их использования можно получить наглядное представление о природе и особенностях этих объектов. Несмотря на то, что зачастую пользователями таких специализированных систем оказывается несколько человек, (а иногда и один исследователь), их разработка оправдывается важностью получаемых результатов.

Изучение и опыт разработки специализированных систем визуализации различного назначения показывает, что в каждой из них присутствует основная идея уподобления сущностей прикладной области и визуальных объектов. Именно за счет этой идеи описывается соответствие между объектами вычислительной модели и визуальными объектами. Кроме того, этой идеей задаются методы взаимодействия с визуальными объектами (а через них и с модельными объектами). С полным основанием ее можно рассматривать как метафору визуализации.

В большинстве случаев специализированных систем компьютерной визуализации мы имеем дело с представлением математических, программных, информационных объектов, абстрактных по своей сути. Для части задач используется абстрактная, но ставшая в данной прикладной области привычной визуализация. Так, наборы представлений статистической графики успешно применяются для многих задач информационной визуализации, а также в системах настройки производительности параллельных систем. Однако бывают случаи, когда существующие приёмы визуализации не применимы. Тогда возникает необходимость поиска метафорических областей–источников, что зачастую является весьма серьезной проблемой.

Примером может служить визуализация четырехмерного множества, являющегося результатом моделирования сложных химических процессов на параллельном вычислителе. Параметрами этого моделирования служат скорости протекания четырех элементарных реакций.

(Отсюда и четырехмерность множества.) Размер результирующего множества составляет порядка 1 миллионов четырехмерных точек [22]. В этих условиях анализ и интерпретация результатов в численном виде трудно выполнимы. Вместе с тем изучение геометрического строения множества позволяет исследовать взаимосвязь элементарных реакций. Встает задача изучения геометрии информационных множеств, а вместе с ней проблема их графического представления. Требуется метод визуализации, с помощью которого можно было бы исследовать локальные и глобальные характеристики расположения множеств в четырехмерном пространстве. К ним относятся, например, локальное строение окрестностей, связность, выпуклость и ориентация в четырехмерном пространстве.

Очевидно, что не существует естественной образности для многомерных множеств. Был рассмотрен ряд приемов представления четырехмерных множеств, но все они оказались неприемлемы для данного случая. После длительного анализа структуры множеств, целей моделирования и опыта представления многомерных объектов был предложен ряд методик визуализации четырехмерных множеств. Однако на первых порах ни один из них не удовлетворял пользователя. После ряда таких попыток был реализован набор связанных между собой видов отображения, представляющих двумерные и трехмерные сечения основного множества, и включающих также набор манипуляций с визуальными объектами. Причём взаимодействие пользователя с визуальными объектами позволяет осуществить навигацию по четырехмерному множеству и обеспечивает извлечение его свойств. Важной идеей было введение специального объекта, не имеющего аналога ни в прикладной области, ни в модели. Был реализован так называемый росток точки, связанный с представлением двумерных сечений множества, где секущая гиперплоскость задается с помощью выбора точек из двумерной проекции. Пользователь в интерактивном режиме выбирает одну точку проекции и наблюдает строение ее ростка, тем самым получая возможность визуально оценить его конфигурацию. Перемещая выделенную точку, исследователь в динамике видит изменение строения ростка при изменении координат. Для преодоления ограничений двумерных представлений используется также трехмерные проекции. Вводится понятие трехмерного слоя, то есть набора ростков точек проекции, оказавшихся на данной прямой. Изобразив слой как трехмерное тело, можно получить визуальную оценку совокупности ростков. Представления свойств ростков на проекции осуществляется как за счет использования цвета, так и за счет передачи параметра ростка с помощью трехмерного графика над проекцией множества. Высота графика в данной точке равна значению изучаемого параметра. Разработанная на базе этих методов система визуализации позволяет генерировать проекции, используя цветопередачу и трехмерные графики, анализировать ростки и слои [17].

Таким образом, успех в очень сложном случае был связан с предложением комплексной метафоры визуализации, обеспечившей процесс интерпретации пользователем результатов компьютерного моделирования. Выделение всех компонент этой метафоры достаточно проблематично, однако чтобы приобрести опыт для разработки и/или адаптации новых метафор для новых специализированных систем нам необходимо провести (пусть даже поверхностный) анализ.

В первую очередь обратим внимание на абстрактность и модели прикладной области, и образности, использованной в визуализации (что, кстати, характерно для многих случаев компьютерной визуализации). Интерпретация пользователем в данном случае происходит не за счет узнаваемых и привычных картинок, а за счет возможности наблюдать за изменениями изображения на связанных между собой графических выводах. Причём пользовательские манипуляции со специальным объектом, не имеющим аналогов ни в целевой ни в исходных областях, и создают эту связь между элементами многомерного множества. Весьма абстрактная идея “ростка”, как дополнительного объекта, помогающего в интерпретации связанных между собой объектов, является основой данной метафоры.

Сама возможность представления и интерпретации четырехмерного множества в конкретном случае связана с спецификой моделируемого химического процесса, что позволило разде-

лить четверку скоростей, координат визуализируемого множества на две пары.

Визуализация строилась на основе знания особенностей математической структуры множества и анализа дополнительных объектов что позволило избежать трудностей при отображении на экран. Идеи привязаны к специфике конкретной модели и проверены на единичном пользователе, разработчике и модели, и программы и, главное, интерпретаторе самой визуализации.

Попытки использовать предложенные здесь приемы на других задачах визуализации многомерных множеств не привели к заметным успехам. Анализ обширного материала по теме визуализации многомерных множеств показывает, что успешный результат получается в каждом конкретном случае при учёте специфики прикладной области, модели и ментальных моделей пользователей–специалистов. Эти специалисты, как правило, достаточно хорошо “видят” структуру многомерного пространства и решительно отвергают неудачные с их точки зрения попытки представления (в принципе непредставимых) множеств. Перенос успешных идей в новую задачу проходит весьма редко.

По всем своим функциям использованные в таких случаях наборы идей являются научными метафорами, формирующими визуальные модели. Конечно, сам факт использования метафор в системах научной визуализации дискуссионен, так как эта область характеризуется абстрактностью визуализируемых сущностей. Но можно показать, что в основе даже самых привычных представлений научной визуализации, например, таких, как график в декартовых координатах, лежит (скрытая в настоящее время) метафора визуализации. Без использования метафоры ни визуализация, ни интерпретация невозможны. Однако, метафора здесь не отсылает пользователя к привычным понятиям, а формирует новые наборы образов и представлений.

Рассмотрим еще один пример метафоры визуализации, возникшей при проектировании информационной системы медицинского назначения.

Для занесения медицинской данных при организации запросов к разработанной нами информационной системе [13] использовалась традиционная интерфейсная методика, основанная на меню и последовательном уточнении области интереса. Это приводило к необходимости большого количества пользовательских манипуляций для любой информационной операции. В системе был предусмотрен удачный по нашему мнению набор видов отображения, показывающих по отдельности кровеносную систему, скелет, внутренние органы и пр. Пользователь должен был последовательно указать на соответствующем визуальном образе органы или части организма и ввести диагностирующие данные по ним в таблицы. Однако, несмотря на наглядное представление, трудоемкость работы с системой, обилие манипуляций с данными и переключений между окнами оказались неприемлемыми для конечного пользователя–медика.

Отметим, что такая ситуация нередко возникает при визуальном представлении информации и запросов к современным информационным системам в случае, когда интересующие пользователя объекты обладают сложной структурой и приходится выбирать их по зависимым друг от друга параметрам в рамках некоторой логики. (Причем эта логика хорошо понятна пользователю, но плохо формализуема.) Сложности возможно вызваны тем, что (естественно) абстрактные представления проектировщиков о структуре файловых систем и систем баз данных (как правило, это структуры типа “дерево”) предлагаются пользователям в качестве основы для организации запросов. Большинство запросов в визуальных информационных системах (или системах с элементами визуализации) основано на меню и дальнейшей детализации запроса за счет “спуска” по структуре дерева. Существуют примеры информационных визуальных систем, где запрос осуществляется за счет десятка (и более) манипуляций с объектами, переключений между окнами и выбора из меню. Сложная и абстрактная логическая модель данных представляется пользователю посредством не менее абстрактных визуальных видов отображения и методик взаимодействия.

Необходимо добиваться уменьшения сложности посредством визуализации и непосредственного указания на запрашиваемые сущности. Для этого необходим выбор метафоры ви-

зуализации, позволяющей описать данное информационное пространство и предусмотреть в него “семантическое погружение”. Под этим термином (который выбран по аналогии с семантическим зумингом [73] и расширяет это понятие) понимается методика выделения при запросе группы семантически связанных между собой объектов. При этом принципы задания связи зависят от задач конкретной специализированной системы. Предусматривается последовательный переход между структурными уровнями с возможностью углубленного изучения заданной группы объектов.

Поиск эффективного интерфейса для информационной “медицинской” системы привел к предложению метафоры для описания состояния человека — “фигура в стеклянном кубе” [12]. Метафора предполагает вывод трехмерной модели фигуры человека с возможностью сделать видимыми или невидимыми те или иные системы организма, включая кожный покров, скелет, нервную и кровеносную системы и т.п. При “погружении”, то есть запросе на вывод информации, могут быть показаны, в частности, кровеносные органы, отдельный желудочек сердца или несколько связанных в рамках данного заболевания органов. Получается нечто вроде набора семантических фильтров или специальных виртуальных (“волшебных”) очков, позволяющих увидеть и выделить, например, больной орган вместе с прилегающей пораженной тканью.

Образность данной метафоры — естественная с трехмерным отображением. Пользователь получает за счет семантической фильтрации дополнительные возможности при выводе данных, что резко уменьшает количество манипуляций, необходимых для запроса.

В принципе этот подход (при правильной метафоре) можно распространить на многие задачи информационной визуализации. В подобающей метафоре можно найти некоторые манипуляторы, используя которые, можно также реализовать запросы без увеличения уровня абстракции. Ими могут быть специальные визуальные объекты, не имеющих соответствия среди модельных (информационных) объектов, но обеспечивающих их анализ и интерпретацию (как в примере, рассмотренном выше).

Семантическое погружение в информационное пространство предлагает при проектировании пользовательского интерфейса вместо иерархии, заданной структурой файловой системы, использовать семантические связи между объектами. Подобный подход уже давно используется в гипертекстовых технологиях. Отметим, что полностью избавиться от меню, детализации запросов и т.п. нельзя, да наверное и не нужно.

В системах визуализации программного обеспечения, предназначенных для отладки производительности параллельных программ также применяются разнообразные метафоры. Одну из них — метафору комнаты мы рассматривали в предыдущем подразделе. Интересна метафора динамической механической системы, на которой основана среда настройки производительности Капоко [65].

В Капоко предлагается метод анимации, отображающий значения состояния параллельной программы, взятые из ее трассы, в набор состояний динамической модели системы. Затем воспроизводится работа параллельной программы (то есть моделируется динамическая система), а результаты моделирования визуализируются, а также сонифицируются. Анимация показывает изменения в балансе между вычислениями и обменов как обычные движения некоторой динамической системы. В частности определяются используемая динамическая система и отображение элементов и состояний параллельной ЭВМ на тела и силы динамической системы. Моделирование параллельной ЭВМ ведется на основе топологии ее (физической) сети. Так, для параллельного компьютера вычислительные модули и производимые на них вычисления, коммуникационная сеть и количество обменов могут отображаться на тела и их массу, пружины, связывающие тела, и силы притяжения между телами соответственно.

В чём-то аналогичная идея (метафора молекулы) была предложена для представления графа вызовов [11]. В этой метафоре структура молекулы отражает структуру исходного графа. Между атомами (отображающими функции — вершины графа) введено два типа взаимодействия — упругое между связанными вершинами и электростатическое между всеми

вершинами–атомами. Электростатическое взаимодействие отражает временные характеристики вызовов функций, тогда как упругое — количество вызовов. Анимация (вращение молекулы) позволяет изучить структуру графа. Анализ возможностей метафоры молекулы показал, однако, ее большую применимость не для отображения графа вызовов, а для представления структур больших систем, например, при визуализации баз данных.

Интересные метафоры были использованы в среде Chiron [47], которая позволяет исследовать влияние работы одного из процессоров на общую производительность параллельного вычислителя. Chiron разработан для решения таких проблем, как плохое использование кэша, неудачное размещение данных в памяти, разделения работы между процессорами, неэффективность синхронизации. Основой для проектирования трехмерных видов отображения являются “экономические” и “физические” метафоры, в частности, стоимость обращения к памяти, стоимость синхронизации, температура объекта. Несмотря на простоту самих идей, положенных в основу метафор, интерпретация достаточно сложных образов не элементарна, так как требует анализа и расшифровки кода форм и цветов, с помощью которых передается информация.

В этих случаях использования физических и экономических моделей, имеет место классический вариант метафоризации. Понятиям и объектам параллельного программирования ставятся в соответствие понятия и объекты модели, которые, в свою очередь, интерпретируются пользователем. Однако, необходимость двойной интерпретации как визуальных образов, так и стоящей за ними сложной физической модели может затруднить использование таких метафор в системах визуализации программного обеспечения.

3. Метафорическое пространство

Выше мы уже упоминали классические работы Дж. Лакоффа по теории метафоры и положения этой теории об исходной и целевой областях (source and target domains). В рамках когнитивного подхода, развитого Дж. Лакоффом, считается, что метафора формирует структуру отображения из исходной области на целевую. Причем целевая область в общем случае более абстрактна, а метафора позволяет уяснить ее понятия в терминах более конкретной исходной области. На этом и строятся основные положения теории метафоры интерфейса. Однако, если у Дж. Лакоффа говорится о сложном взаимодействии структур областей, принимающих участие в метафоризации, то в ряде работ по метафорам интерфейса, использующих его идеи, метафоры рассматриваются лишь как отображения исходной области на целевую (См., например, [58]). В литературе по теории интерфейса, как правило, концепция метафоры основывается на представлении новых или достаточно необычных для пользователя явлений посредством других явлений, хорошо ему известных из повседневной жизни, причем эти явления должны иметь те же основные свойства, что и явления, которые они объясняют [64, 75]. Выдвигаются требования привычности и полноты метафоры [43]. Конечно, апелляция к обыденному человеческому опыту и активизация интереса при использовании привычных аналогий облегчает понимание и усвоение принципиальных моментов исходного явления или процесса. Практика применения визуальных метафор интерфейса дает примеры привычных, полных метафор, в которых проектировщик добился скрупулезного соответствия между сущностями исходной и целевой областей и отличной узнаваемости почти всех метафорических объектов. (См. [75], [64] и анализ в [3].) Однако, эти метафоры могут оказаться практически бесполезными из-за своей громоздкости или появления дополнительных и нежелательных аналогий, связанных с обыденными вещами. С другой стороны существует множество примеров заведомо неполных, но плодотворных метафор. Так в Norton Commander'e (и его наследниках таких, как FAR) перепись файлов построена на метафоре бухгалтерской книги с её переносом соответствующих сумм из графы “дебет” в графу “кредит” и обратно. В этой чрезвычайно удачной визуальной метафоре отражается лишь один аспект исходного множества, а все остальные аспекты бухгалтерской книги не востребованы. Отметим, также, что эта метафора недостаточно чётко

отражает структуру файловой системы, а правила работы с Norton Commander'ом и его визуальный образ иногда порождали у пользователей-новичков нежелательные представления о логике его функционирования. Вместе с тем, несмотря на все видимые недостатки, успех метафоры в данном случае неоспорим. Да и в наиболее успешном примере применения метафоры, метафоре рабочего стола, используется минимальное число аспектов исходного множества. Наконец, можно вспомнить еще одну классическую, часто упоминаемую в литературе метафору “РИЧАРД — ЛЕВ”, где лишь некоторые качества льва (например, отвага, но отнюдь не наличие хвоста или клыков) переносятся на человека, который теперь находится в рамках метафорического пространства.

Возникает вопрос, когда хорошо работает метафора?

Примеры показывают, что не только (и не столько) в том случае, когда используются знакомые понятия и образы. Тут, как мы уже говорили, возможно возникновение дополнительных, “паразитных” смыслов, связанных у пользователя с реальными понятиями и образами, вредными для интерпретации. Не всегда пригодным оказывается и требование полноты отображения из исходной на целевую область. Метафоры успешны, когда использованные метафоры уменьшают абстрактность компьютерной модели, в том числе абстрактность интерфейса пользователя с системой. Интерпретация визуализации и интерактивных манипуляций, построенных на базе данной метафоры, восстанавливает (или создает заново) у пользователя некоторые ментальные структуры, в которых представляется картина явлений.

По сути метафора конструирует для пользователя некоторую реальность (часто при помощи “волшебных” объектов, понятий или операций.) Логика этой новой реальности с одной стороны отражает идеи пользователя об интерфейсе и объектах моделируемого мира, а с другой — должна совпадать (или быть близкой) с логикой развития процессов и изменения объектов в первоначальной прикладной области, включая логику деятельности самого пользователя.

Для наших задач важна идея взаимодействия исходного и целевого множеств в метафоре. Известно, что в классической теории метафоры рассматриваются два подхода к её функционированию — сравнительный и интеракционистский.

Сравнительный подход предусматривает, что метафора является актом сравнения двух объектов, служащим для того, чтобы выявить и подчеркнуть подобия между ними. Метафоры служат прежде всего средством отбора свойств в объектах целевой области, значения которых существенно независимы от сравнения.

В рамках интеракционистского подхода метафора рассматривается как сложное взаимодействие между системами отношений в целевом и исходном множествах, которое может вести к фундаментальным изменениям в нашем понимании обеих областей. При этом значение метафорических понятий не установлено заранее, но постепенно развивается посредством их участие в последовательности метафор. Метафора может фактически заставить значение метафорического понятия переместиться путем изменения окружающей ее системы отношений. Метафора не только обнаруживает подобия, но и может также создавать их между объектами, бывшими ранее совершенно несходными. Предусматривается, что метафора вовлекает полные системы отношений как в целевом, так и в исходном множествах. Структура отношений исходного множества действует как фильтр, преобразующий наше понимание целевого множества. Считается, что существует влияние и переход информации не только из целевого множества на исходное множество, но и обратно из исходного множества целевое. Причем изменения происходят в обоих компонентах метафоры [74].

Анализ реальных ситуаций использования метафоры приводит к рассмотрению все более сложных процессов. Так в [76] выделяются кроме традиционных исходного и целевого множеств (пространств) еще два — прототипное (generic), содержащее скелетные структуры, приложимые к обоим (исходному и целевому) множествам и смешанное (blended), объединяющее структуры из этих множеств.

Наш подход [3] расширяет классическую модель метафоризации Дж. Лакоффа и отличается тем, что в его рамках метафора порождает некоторое самостоятельное метафорическое

пространство.

Цель метафоризации состоит в увеличении выразительности изучаемых объектов. При метафоризации выбираются объекты целевого множества с набором характеристик, структур, свойств, которые мы и хотим рассмотреть с повышенной выразительностью. Причем выбираются не все объекты (и даже не все их свойства или элементы структуры), а лишь те, которые нас интересуют больше всего. Этим объектам ищутся аналоги (в плане структур, качественных свойств и пр.) в исходном множестве.

Далее имеет место следующая операция.

Объект целевого множества вместе с объектом из исходного множества помещается в метафорическое пространство, точнее тем самым порождается метафорическое пространство (или область метафоризации). В этом пространстве теперь начинает функционировать изучаемый объект. Можно считать, что это уже новый объект нового пространства. Метафорическое пространство приобретает автономность от породивших его областей. Многие свойства его объектов лишь опосредовано связаны (если вообще связаны) со свойствами объектов исходного множества. За счет проекции некоторых характеристик целевого множества на исходное множество появляется своя логика развития метафорического пространства. (Так, например, при использовании научной метафоры электромагнитного поля может изучаться его напряженность, явно отсутствующая у поля пшеницы.)

В частном случае метафоры визуализации имеет место отображением на некоторый мир визуализации, где безобразные объекты получают свое визуальное представление.

Возникают вопросы — какова же природа и структура метафорического пространства; как идёт его порождение? Естественный ответ на них связан с пониманием того, что рассмотрение метафоры как знака или как пары знаков не является плодотворным. Прежде всего метафора генерирует некоторую знаковую систему, то есть цельный набор знаков, в котором существующие внутренние отношения между знаками каким-то образом отображают отношения между обозначаемыми. Наше метафорическое пространство по сути и есть знаковая система. Такой подход, кстати, применим ко всем рассмотренным ранее пониманиям метафоры и к аристотелевскому украшению речи, и к научной метафоре, и к метафоре визуализации. В первом случае на базе новой знаковой системы генерируется некоторый вспомогательный язык, например, язык поэзии, во втором — основной язык новой дисциплины, в третьем — язык визуализации. Понимание метафоры как знаковой системы даёт и основу для оценки предлагаемой для конкретного случая метафоры. Если использованное сближение (сравнение или набор сравнений) отвечает требованиям системности, то тогда можно говорить о существовании полезной метафоры. Если нет, если изменения состояния объектов исходного множества слабо связано с изменениями целевого, то использование таких сравнений не поможет в понимании изучаемой ситуации (или не даст ничего нового в плане выразительности литературного текста).

Порождение основы знаковой системы в случае метафоры можно представить себе за счет применения двух операторов метафоры,

основного:

“пусть A подобно B ”

и оператора:

“следующие атрибуты/элементы/свойства A выбираются для уподобления следующим атрибутам/элементам/свойствам B ”,

где A — исходное, а B — целевое множества.

Подробнее процесс выбора/порождения метафор будет рассмотрен в разделе 8. А пока продолжим изучение знаковой природы метафоры.

Сделаем предположение о знаковой природе компьютерного моделирования. Можно утверждать, что процесс компьютерного моделирования есть процесс создания знаковых систем. Действительно, в той или иной модели можно легко вычленив в качестве набора означаемых объекты моделируемого явления или процесса, а в качестве знаков рассматривать объекты модели. Правда, для целей самого моделирования это рассмотрение не нужно, однако нам оно

позволяет делать дальнейшие построения.

Далее можно говорить об иерархии моделей, используя это понятие несколько не в том смысле, как было введено в [27], а скорее в смысле иерархии по уровням моделирования, то есть физическому, математическому, алгоритмическому, программному уровням [26].

Метафору (и, в частности, метафору визуализации) можно рассматривать как научную модель. Причем метафора может быть не только моделью явления, но и моделью восприятия этого явления автором (проектировщиком) и наблюдателем (пользователем).

Визуализация при моделировании обеспечивает интерпретацию и анализ полученных данных, а также поддержку работы на всех этапах цикла компьютерного моделирования. При этом, метафора, а ещё точнее, порожденная ей визуализация, является последней в иерархии моделей знаковой системой. В этом наложении моделей визуализация должна помочь в снятии хотя бы одного слоя, а не добавит новые. Необходимо обеспечить конкретику, присущую визуальным (то есть реально воспринимаемым при помощи зрения) образам. В общем случае визуализация должна уменьшать, а не увеличивать общий уровень сложности всей иерархии моделей.

Наиболее удачный и всем известный пример (докомпьютерной) универсальной методики визуализации — рисование двумерного графика функции в декартовых координатах. Если разобраться, то в основе самого понятия функции одного переменного лежит целая серия очень абстрактных идей, которые могут быть описаны как иерархия физических и математических моделей. Еще более абстрактными являются идеи декартовой плоскости и представления на ней значений аргумента и функции, по сути являющимися методами геометрического моделирования явлений. Однако рисование графика функции, позволяющего ясно показать все ее особенности, делает результирующее (визуальное) моделирование намного более конкретным и доступным для четкой интерпретации результатов всей серии моделирования. Отметим, что в основе этого рисования лежит полноценная, хотя и почти незаметная для привыкших к ней с детства людей, метафора визуализации.

В терминах семиотики можно говорить о “разозначивании”, то есть о снятии при визуализации нескольких уровней означивания, полученных за счет серии знаковых систем, присущей данной иерархией моделей. Особенно четко видно это “разозначивание” при непосредственной визуализации исходного объекта моделирования. Визуализация должна поддерживать непосредственную связь между картинкой, которая в данном случае является иконическим знаком (в смысле Ч. Пирса), и обозначаемыми объектами. То есть она должна поддерживать показ сущностей, а не рассказ о них посредством некоторых графических нотаций. (В необходимых случаях требуется также возможность непосредственного манипулирования объектами.) Это требование вовсе не означает необходимости использования только лишь естественной образности. Абстрактная образность часто требуется для показа абстрактных объектов или поддержки процесса создания этих абстракций.

Можно ввести оценку качества метафоры в системах визуализации за счет учета уровня сложности, задаваемую по числу слоёв (уровней иерархии), отделяющих рассматриваемую модель от первой моделируемой сущности. Тогда для оценки качества метафоры визуализации нужно оценить на сколько слоёв визуализация понизили общую сложность всей компьютерной модели. (Отметим, что график функции понижает уровень сложности сразу на несколько слоёв.) Однако, так можно оценивать только лишь метафоры, а не сами системы визуализации, так как в конкретных системах есть много привходящих деталей, связанных с реализацией.

Аналогичные оценки можно в принципе провести и для метафор в некоторых специализированных системах, использующих визуальные формы интерфейса.

4. Структуры объектов визуализации

Известно, что термин “визуализация” подразумевает как методики построения изображений или даже процесс построения конкретного визуального образа, так и название области

компьютерных наук (визуализация научная, информационная, программного обеспечения).

Несмотря на различия в приложениях визуализации, имеет место глубокое единство всех ее подразделов. Одним из следствий этой общности является такое явление, как дрейф методик визуализации, проявляющийся в переходе метафор визуализации и соответствующих им видов отображения из одной области визуализации в другую, в частности, в визуализацию программного обеспечения из информационной и/или научной визуализации [9]. Например, и в информационных системах и в системах настройки производительности параллельных программ часто необходимо отобразить изменяемые в течении времени состояния того или иного процесса (то есть по сути статистические данные). Для этого используются известные визуальные методики, имеющиеся в статистической графике — диаграммы различного типа (круговые, столбчатые, Гантта, Кивиата и пр.), графики и т.п. Аналогичные примеры можно привести в случаях переноса методик научной визуализации в системы отладки и настройки производительности параллельных программ, рассматривая в частности, “поверхности ошибок” в системе Guard [30] и поверхности, представляющие, в частности, “температуру объекта”, в Chiron [47].

Дрейф метафор и методик отображения может быть объяснён тем, что в рассматриваемых системах визуализируются схожие в каком-то смысле сущности. Однако, появляется целый ряд вопросов — за счёт чего выявляется это сходство, как выбираются в подобных случаях схожие способы отображения, существуют ли закономерности, определяющие успех или неудачу визуализации и независимые от природы визуализируемых данных. Если в простых примерах ответы достаточно очевидны, то в других случаях требуется изучение связей между внутренней природой и структурой модельных объектов (и соответствующих им данных) и внутренними ментальными структурами пользователя, в которых представлена сложившаяся у него картина явления.

При моделировании некоторого явления, процесса или объекта кроме абстрагирования используется декомпозиция изучаемой сущности, когда она представляется как совокупность некоторых составляющих. Сложные объекты и системы имеют иерархическое строение и состоят из более простых компонент, могущих также находиться в определенных отношениях между собой. Компоненты при необходимости можно подвергнуть дальнейшей декомпозиции. Это строение не всегда очевидно, и для выделения этой структуры необходимы специальные методики. Основными выделяемыми отношениями являются агрегация, или “быть частью”, а также наследование, или “ведёт себя как, является”. В частности, данные отношения активно используются при проектировании программных продуктов.

Компоненты системы могут быть связаны и другими отношениями. Например, существуют некоторое количество типовых структур — вектор (одномерная “цепочка”), матрица (двумерная регулярная структура) и др. Более сложные структуры в этом случае можно получить путем композиции простых. Такие структуры, не являющиеся иерархическими (по принципу наследования или агрегирования), можно называть организационными, или структурами устройства. Связи-отношения устанавливаются между компонентами одного ранга.

Можно сделать достаточно общее и “мягкое” описание структуры объекта и модели объекта.

Будем в качестве структуры объекта рассматривать выделенные в нём иерархические и организационные отношения.

Декомпонировать можно как сам объект, так и его модель-представление. Так, если сначала строится модель некоторого объекта, а потом представляется как она устроена, то можно получить декомпозицию модели (модельного объекта). В качестве структуры модели будем рассматривать набор отношений (агрегирования, последовательности и др.), которыми связаны компоненты модели. Если же имеется какая-либо информация о реальном строении объекта, то можно декомпонировать исходный объект и построить модели каждой из его составляющих. Потом возможно объединение модели фрагментов теми же отношениями, что и компоненты исходного объекта. Возникающий при этом набор отношений также можно рас-

смаатривать как структуру.

Успех визуализации обеспечивается наличием связи между внутренней природой и структурой модельных объектов (и соответствующих им данных) и внутренними ментальными структурами пользователя, в которых представлена сложившаяся у него картина явления (так называемые “репрезентативные когнитивные структуры”) [29]. При этом связи между теми и другими структурами не являются произвольными.

Человек выделяет в картинке какие-то общие закономерности, разбивая единую картинку на совокупность (быть может вложенных) фрагментов, абстрагируясь от второстепенных деталей [29].

Таким образом, можно говорить о когнитивных структурах, структурах анализируемых сущностей и структурах визуальных объектов и образов. При анализе визуального образа интерпретация происходит в двух фазах. Выделяются элементы, интерпретируемые в соответствии со знанием, полученным из предметной области. Из общего визуального образа выделяется (восстанавливается) информация о взаимосвязях между этими элементами и, тем самым, обеспечивается связное адекватное представление о визуализируемом объекте. То есть интерпретация частично происходит на основе самого визуального образа, а частично на основе анализа его частей и их взаиморасположения.

Информация о взаимосвязи элементов указывает на структуру визуализируемого объекта (точнее, нашего представления о нем).

Процесс визуализации можно рассматривать как построение визуального (геометрического) образа на основании абстрактных представлений об объекте. Эти абстрактные представления есть модель (исследуемого объекта, явления, или процесса), как-то связываемая с имеющимся у пользователя когнитивными структурами, описывающими данную сущность. Визуальные образы, представляющие моделируемую сущность, служат для того, чтобы создать или восстановить по нему когнитивные структуры. Цель исследований в области визуализации состоит в создании таких методик и принципов, которые обеспечат восстановление по визуальным образам адекватных когнитивных структур. Порождение когнитивных структур по визуальным образам и есть процесс интерпретации. Этот процесс является обратным или, точнее, двойственным визуализации. Аналогично принципам визуализации, существуют принципы, на которых базируется интерпретация.

Рассмотрение структуры модельного (исходного) объекта может снизить сложность разработки представляющих его видов отображения. В этом случае сложный вид отображения может разрабатываться как комплексный вид отображения. Для представления компонент исходного объекта можно использовать некоторый набор визуальных объектов. Также предполагается использование набора приемов визуализации отношений между элементами.

Виды отображения структуры предполагают не зависящие от конкретики способы изобразить элементы, находящиеся в данном отношении. Вид отображения структуры есть способ задания методик ее интерпретации. Конкретизация вида отображения с учетом специфики исходной сущности определяют сами объекты интерпретации.

Задача визуализации — получить визуальный образ, по которому будет правильно восстановлен ментальный образ (представление) об исходном объекте.

Таким образом, разработка конкретного вида отображения сложной исходной сущности может состоять из выбора способа отображения соответствующей структуры и способов представления ее элементов. Расположение визуальных образов элементов структуры необходимо проводить в соответствии с правилами, предписывающими способы визуализации отношений между этими элементами.

Вид отображения структуры включает в себя способ задания интерпретации этой структуры.

Ясно, что, если выбрать неправильные способы визуального представления исходных сущностей, то интерпретация будет затруднена или даже невозможна из-за “плохой” образности. Также может случиться, что выбран неправильный способ передачи отношений между эле-

ментами. В этом случае будет воссоздана ошибочная, хотя и схожая по составу с исходной, когнитивная структура. Поэтому необходим какой-то критерий правильности выбора (или конструирования) видов отображения. Построение видов отображения может основываться на том правиле, что структура построенного визуального образа не должна противоречить структуре исходного объекта. При интерпретации визуального представления объекта не должно возникать отношений, отсутствовавших в прообразе. В тоже время редукция размерности, также как и редукция всей структуры возможна при условии, что пользователь (интерпретатор) проинформирован о её осуществлении. Естественно, когда элемент не отображается, то отношения, в которых он находился с другими элементами, также не отображаются.

В связи с критерием сохранения структур интересно рассмотреть отображение в трехмерном виде сущностей, имеющих заведомо двумерную структуру. Известно, что повышение размерности изображения оказывается полезным для повышения наглядности взаиморасположения объектов, увеличения информационной плотности, а также для того, чтобы выделять более важные элементы и обеспечить лучшую интерпретацию изображений. Так, конические деревья [73], располагающие графы-деревья в трехмерном пространстве, обеспечивают фокусировку внимания пользователя на конкретном пути от корня к листу за счет выноса соответствующих рёбер на передний план. Это связано с особенностями человеческого восприятия визуальной информации.

Однако, в ряде случаев подобное отображение оказывается избыточным (и возможно приводит к неправильной интерпретации). Достаточно противоречива, в частности, ситуация с трёхмерными видами отображения графов вызовов, используемых в системах настройки эффективности распределенных и параллельных программ. Для адекватного отображения структуры этих графов в принципе достаточно двумерности. Трёхмерность в этих случаях, как правило, не даёт существенно нового качества при визуализации. (См., в частности, описание нашего опыта в [11].) Однако при рассмотрении каких-то сложных моментов возможен положительный эффект трехмерного отображения этого графа. Например, в случае распараллеливания в рамках парадигмы общей памяти рассматривается такая сущность, как “выполнение операций на матрице процессоров”. Здесь возникает граф вызовов, включающий некоторый цикл, распараллеленный по двумерному массиву процессоров. Трёхмерность здесь может быть полезна, например, для обеспечения отображения с сохранением отношений.

Полученный критерий выбора/построения видов отображения (критерий сохранения структур) по сути является и одним из критериев при выборе метафор, лежащих в основе видов отображения. Анализ функционирования метафор показывает, что кроме этого должны существовать “логические” принципы выбора метафор интерфейса и визуализации. Им и посвящен следующий раздел.

5. Структурный анализ порождения метафор

Мы уже задавались вопросом, почему одни метафоры (не только компьютерные, но и литературные или риторические) оказываются удачными, несмотря на заведомую неточность, а другие — точные и подробные — никуда не годятся?

На этот раз ответ будем искать, рассматривая как же происходит порождение метафор. Наша цель — определить отношение между метафорой и целевой системой значений, для которой метафора, собственно говоря, и создается. Это отношение может быть описано как понятийная метка.

Начнём с уже упоминавшегося выше примера известной метафоры “РИЧАРД — ЛЕВ”. В данном случае понятие “РИЧАРД” подвергается метафоризации. Известны его некоторые качества, в частности, храбрость, благородство, сила, etc. В нашем сознании существует некоторый образ [короля] Ричарда, вместе с представлениями о храбрости, благородстве, силе и пр. Имеют место связи между образом Ричарда и этими понятиями такие, что можно записать:

Ричард — храбрый;

Ричард — благородный;

Ричард — сильный.

Отметим, что Ричард — это простое имя собственное, которое вполне может быть оторвано от реального мира, и фактически, является свободной меткой. Метка "храбрый" в свою очередь подразумевает возможность навешивания на предмет и других меток, или наоборот, обязательное их отсутствие. То есть, "храбрый" — это настоящая понятийная метка, имеющая целое дерево следствий, также как и целое дерево предпосылок. С точки зрения теории речь идет о целевой области метафоры. Причём с формальной точки зрения данная понятийная система не является множеством, но системой, выстроенной в рамках естественной дедукции [66]. Чтобы не усложнять понимание, в дальнейшем будем применять упрощённые рассуждения на базе следования понятий и использовать отношение импликации (отношение "если-то"). Импликация важна для нас, так как является основой любого рассуждения, а метафора помогает этому процессу. Для обозначения этого типа следования будем использовать символ стрелочки (\rightarrow). Этот символ (в данном, отличном от формальной логики, случае) обозначает какие множества понятий-меток влекут какие следующие метки, которые можно навесить на рассматриваемый объект. Например,

{Ричард} \rightarrow {храбрый, могучий, король} или

{присутствует красный, присутствует зелёный, присутствует синий} \rightarrow {пёстрый}.

"Храбрый", "могучий", "король" - это просто метки. Но высказывание "Ричард — храбрый" заключает в себе не только их, но и весь "лес" стрелочек, который исходит из понятий "король", "храбрый", "могучий". Понятно, что у различных людей эти "леса" стрелочек будут различаться. Однако при более формальной записи понятий однозначность такого "леса" увеличивается.

Из высказывания "Король Ричард" можно вывести, в частности, — "храбрый", "могучий", etc. Далее можно перейти на новый уровень, исходя уже из этих понятий. Например, из понятия "могучий" может следовать (в сниженном варианте) — "грудь колесом", "огромные бiceps" и так далее в меру нашей ассоциативной активности.

Метафоризируемое понятие рассматривается здесь как способ приписывания меток к различным сущностям, как взаимосвязь этого набора меток. Выявляется какие метки с какими связаны в нашей мыслительной модели метафоризируемого понятия, и следующие (второй, третий, четвёртый и т.д.) уровни стрелочек, которые стоят за каждым из понятий-меток в данной понятийной системе. Получается граф из понятий-меток. Этот граф можно наложить на аналогичный граф, составленный из понятий множества-источника метафоры. Естественно, что набор меток в этом графе иной, однако его общая структура сходна. Появляется следующий объект:

{I} \rightarrow M \rightarrow {O},

где M — рассматриваемое понятие, {I} — множество понятий, его подразумевающих, а {O} — множество понятий, им порождаемым.

Таким образом можно рассмотреть граф импликаций, причём считается, что топология путей, проходящих через узел, определяет значение этого узла. Понятийная метка всего лишь указывает на узел в данном ориентированном графе. В случае метафоры и исходное и целевое множества обладают такими графами, показывающими связь значений между собой. На основе этих представлений соответствие между значениями исходной и целевой областей может быть установлено как гомоморфизм графа понятий-меток исходной области на граф понятий-меток целевой области.

В процессе метафоризации определяются графы понятийных меток для исходных множеств. Поиск метафоры — это поиск структуры взаимосвязей, аналогичной структуре целевого пространства. При наложении (не точном, не взаимно однозначном) может стать, что какие-то из нижних уровней будет необходимо удалить. Но может и появиться дополнительный уровень, отсутствующий в целевом пространстве. Именно эта вторая система меток-понятий с её взаимосвязями будет нашей метафорой к первоначально рассмотренному образу. Взаимосвязи между понятиями в метафорической системе могут быть гораздо богаче чем

в первоначальной. При этом некоторые взаимосвязи могут соответствовать тем стрелочкам, которые присутствуют и в первом множестве, но просто не были нами замечены до этапа построения метафоры. А тут они могут оказаться очевидно заметными.

В рассматриваемом примере искомой структурой оказывается “ЛЕВ”, с традиционно приписываемыми ему храбростью, благородством, силой. Отметим, правда, что все эти свойства в принципе отличаются от аналогичных, присущих Ричарду.

Однако, более привычной метафорой является “Ричард — львиное сердце”. Здесь всё просто. Мы хотим проиллюстрировать отвагу короля Ричарда. И метафоризируемое понятие можно выразить в таком виде:

Король Ричард → отважный

Что означает, что как только мы какому-то объекту навесим ярлык “Король Ричард”, то с этой меткой сразу же можно будет ассоциировать отвагу данного короля. При этом понятие “отважный” потянет за собой целый лес других понятий, которые потянут другие понятия и так далее. Во всех более или менее полных словарях присутствуют циклы. Тем не менее, понятие “отважный” слишком сложное понятие. Сложно найти другое понятие с таким же набором вытекающих и втекающих в него стрелочек-взаимосвязей с другим миром. Да и этот мир очень сложно найти. Поэтому для описания отважности, есть только одна естественная альтернатива — взять какой-нибудь образ, который так же включает в себя это понятие. Например,

Львиное сердце → отважное.

Наложение этих структур довольно примитивное. Король Ричард вполне соответствует львиному сердцу. Правильнее (да и просто интереснее) все-таки было бы вместо “сердца”, говорить “лев”. Потому что Лев, например, ко всему прочему царь зверей. (Это соответствует метке Ричарда — король. Отметим, дополнительно, что “Лев — царь зверей” в свою очередь является привычной и потому почти незаметной метафорой.) Соответствие “Ричард — лев” более полное, чем соответствие просто сердцу. Далее, так как в данной модели “Лев” гораздо более богатая метка, чем “Ричард”, начинают проявляться интересные моменты.

Ричард → король, отважный

Лев → царь [король], отважный, рычит, с пышной гривой, хвост с кисточкой.

То есть Ричард вполне может теперь после ассоциирования начать рычать, трясти гривой и вилять хвостом. Что ему, впрочем, совсем не подходит. Но вот боевой клич (рык) — а почему бы и нет? То есть, метафора позволила нам нащупать нечто, что вполне может принадлежать образу короля Ричарда.

В принципе подобным образом можно рассмотреть графы отношений импликации не только для слов, но и для других подобных отношений. Рассмотрим более строгий пример.

Предположим, что у нас есть некая временная последовательность значений неизвестной природы $\{X\}$. Про $\{X\}$ можно сделать следующие выводы:

$\{X\}$ → временная последовательность → набор элементов с одной линейной дискретной координатой и некоторым значением.

(Другое восприятие и описание времени может оказаться излишне сложным и/или спорным.)

Ставится задача представления последовательности в каком-либо виде. Можно рассматривать эту задачу, как обобщение визуализации, куда, в частности, включается и сонификация. Например, попробуем превратить данные значения в ноты и получить мелодию. Это возможно потому, что ноты могут интерпретироваться как значения, а длительность нот в музыке дискретна. Тогда:

Мелодия → временная последовательность нот → набор элементов с одной линейной дискретной координатой и неким значением.

Восприятие $\{X\}$ через подобную метафору может привести к интересным выводам. У последовательности нот есть такое особое свойство — складываться в красивые мелодии. И если наш способ чисто механического преобразования значений $\{X\}$ в ноты породит мелодию,

то, вероятно, в $\{X\}$ есть некое интересное свойство, которое можно обнаружить при помощи метафоры. (В этой связи вспомним идеи по когнитивной, способствующей математическому творчеству визуализации проблем аддитивной теории чисел [21].)

Рассмотрим более подробно ещё одно представление временной последовательности в следующем виде:

набор пикселей \rightarrow набор точек $(x, y) \rightarrow$ набор точек с двумя линейными координатами.

Отметим, что понятийная метка “координата” включает в себя и понятие “значение”. Поэтому веточка стрелочек, связанная в $\{X\}$ с тем, что в каждый момент времени мы имеем значение, вполне укладывается на понятийный граф “набор дискретных точек (x,y) ”.

Воспользуемся этой визуальной метафорой и свяжем время с координатой x , а значение преобразуем в y . В результате получается нечто, имеющее гораздо более богатые свойства, нежели у целевого понятия $\{X\}$. Потому что пиксели, например, обладают свойством геометрической близости, также как упорядоченности по расстоянию от некой прямой. Понятие “точка (x, y) ” вытягивает за собой кроме “значения y в линейной дискретной последовательности на месте x ” ещё целый ряд понятий. Например, “расстояние до окружности радиусом 1 с центром в точке $(0,0)$ равно 30”. Рассматриваемое понятие, благодаря этой метафоре, становится гораздо более богатым. Возможно, что таким образом проявятся некоторые внутренние взаимосвязи в самом $\{X\}$, которые до этого были неизвестны. На следующем этапе возможно их обоснование за счет строгого анализа целевого явления $\{X\}$ на базе его собственных свойств и отношений.

Теперь в свете наших возрений на структуры взаимосвязей опишем процесс порождения метафоры (метафоризации).

Прежде всего происходит (чаще всего неявный) анализ целевого множества будущей метафоры. На основе представлений об основном объекте метафоризации и его свойствах выявляется иерархическая структура взаимосвязей объектов целевого множества и их свойств.

На следующем этапе происходит выбор множества-источника и его главного объекта. Критериев выбора у нас несколько.

Во-первых, у основного объекта области-источника должны иметься свойства, подобные (близкие) свойствам объекта метафоризации. Структура взаимосвязей этого объекта и его свойств должна быть аналогичной структуре взаимосвязей объекта метафоризации и его свойств, хотя бы на первом уровне структурного дерева.

Во-вторых, область-источник должна быть визуализируема. То есть природа области-источника должна быть такова, что её объекты обладают протяженностью, формой, цветом или другими визуально представимыми характеристиками. (Пример — метафора железной дороги для описания ряда функций операционных систем.)

Понятие визуализации в принципе можно расширить, рассматривая в этом плане, например, сонификацию как в [68] и [65], и говорить тогда о звуковом или музыкальном представлении информации. В этом случае необходимо расширить и критерий визуализируемости представления.

“Визуализируемость” (в широком смысле) области-источника обеспечивает интерпретацию, то есть процесс порождения (или восстановления) у пользователя (точнее приёмника метафоры) представлений об изучаемых объектах. Интерпретация является процессом двусторонним (или даже обратным) по отношению к визуализации. (См. предыдущий раздел, а также [10].) Качество метафоры, таким образом, связано с возможностью легкой интерпретации того [визуального] языка, который порождается данной метафорой.

Требование визуализируемости связано также с третьим (давно известным) критерием “хорошей” метафоризации — привычностью, узнаваемостью объектов области-источника. (Или для случая научной метафоры меньшей абстрактностью области-источника [40].) Выше уже указывалось, что понятие привычности и узнаваемости в специализированных системах человеко-компьютерного интерфейса должно увязываться в большей мере не с бытовыми реалиями, а с деятельностью потенциального пользователя в той сфере, для которой создаётся

интерактивная система.

На следующем этапе процесса метафоризации проводится анализ области–источника. На основе анализа взаимосвязей и отношений в рамках области–источника, также как и на основе аналогий с ними происходит анализ и самого объекта метафоризации и его свойств, выявляются отношения объектов в рамках целевой области.

Для поиска новых отношений в целевой области необходимо, чтобы область–источник обладала более богатой и глубокой структурой взаимосвязей чем целевая область.

Вот и еще один признак “удачной” метафоры.

В общем случае ясно, что успех метафоры связан прежде всего с структурой взаимосвязей понятий в области–источнике и с возможностью получить на её базе новое понимание отношений в изначально интересующей нас целевой области.

Двойственность процессов интерпретации и визуализация (или, обобщая, какой–либо иной формы представления) проявляется здесь через метафору. Знаковый процесс в визуальных интерактивных системах (точнее, часть этого процесса, связанная с интерпретацией интерфейса) поддерживается за счёт действия метафоры. Действие метафоры, в частности, реакция пользователя на свойства и отношения, появившиеся у объектов метафоризации, как раз и связаны с наложением богатых структур взаимосвязей понятий области–источника на целевую область.

Мы уже говорили об операционном (помогающем принимать решение и действовать) характере метафор интерфейса и визуализации и их отличии в этом плане от метафор литературы, рекламы и пр. Отсюда вытекает ряд ограничений на выбор метафоры в специализированных системах. В [2] отмечалось, в частности, ошибочность выбора метафор на базе сленга, носящего ярко выраженный профессиональный и/или национальный характер. В среде пользователей, инокультурных по отношению к проектировщику системы, образы, построенные на базе “жаргонных” метафор воспринимаются как простые символы. (Вспомним, кстати, подобное восприятие отечественными программистами метафорического для американских коллег понятия “файл” в конце 60–х начале 70–х годов.)

Наш структурный анализ метафор проясняет причины этого, показывает почему в данном случае не интерпретируются (или интерпретируются неправильно) объекты и операции над ними. Так как инокультурный пользователь не воспринимает сближений, основанных на сленге, то область–источник метафоры для него фактически отсутствует. Вообще, по нашему мнению в случае метафоры интерфейса не следует особенно полагаться на “глагольные” метафоры, где образы строятся на основе представлений отдельных операций, например, “толкать” или “убить” файлы и т.п. И дело не только в том, что действия и процессы часто не имеют очевидных визуальных представлений. Ясно, что метафоризации следует подвергать некоторую систему понятий, связанную с деятельностью потенциального пользователя. Необходимо добиваться общей “операбельности” метафоры, когда можно использовать богатые взаимосвязи и отношения участвующих в операции объектов. Поэтому использование в интерактивных системах классических приёмов метафоризации (таких как “часть вместо целого”, “замена причины процессом”, “замена причины следствием” и т.п.) должно проводиться достаточно осторожно.

В качестве примера рассмотрим приём “часть вместо целого” при использовании известной в русском языке метафоры проверки на [скрытое] свойство “лакмусовая бумажка”. В этой метафоре имеет место замена процесса (“тест на кислотность”) его атрибутом, тем, что тест осуществляет. Лакмусова бумажка является лишь частью испытания на кислотность, и само испытание не перекрывает целиком возможностей использования лакмусовой бумажки [25]. Метафора прекрасно “работает” в литературной речи, но её применение при проектировании интерфейса кажется затруднительным. (Конечно, можно было бы использовать пиктограмму “лакмусовая бумажка” для представления процесса тестирования, но вне соответствующей языковой традиции понять эту метафору не всегда возможно. Да и метафора при подобном применении теряет свои свойства.)

В случае порождения новой метафоры интерфейса на базе подобных приёмов её интерпретация была бы ещё более затрудненной из–за отсутствия привычных структур взаимоот-

ношений понятий, а также в следствии необходимости разгадывания того, какой же приём применён в данном случае.

6. Заключение

Пора подводить итоги нашего исследования. Что получено в результате? Что еще надо сделать, чтобы завершить теорию компьютерной метафоры?

Анализ и вывод формул уже известных метафор задаёт общие правила их генезиса и функционирования, что должно позволить не только оценивать и сравнивать уже известные метафоры, но и искать и/или генерировать новые. На какие же критерии успеха метафор визуализации и интерфейса можно указать после проведенных нами исследований проблемы?

Мы уже отмечали, что классические требования такие, как привычность, полнота, меньшая абстрактность области-источника по сравнению с целевой областью, должны быть дополнены и переформулированы. Да и методы оценки метафоры на основе сравнения результативности работы пользователей с различными (построенными на базе метафор и “безметафорными”) системами не являются удовлетворительными. (Тем более, что по нашему мнению “безметафорной” визуализации не существует [2].)

Выделим набор разноплановых требований к выбору метафор и видов отображения, приведенных в данной работе.

Первым среди них укажем на требование целостности и системности интерфейса и/или визуализации. Это требование связано со знаковой природой и визуализации и человеко-компьютерного взаимодействия в целом. Достаточно очевидно, что их следует описывать как знаковые системы, а не в виде отдельных знаковых ситуаций. Нами показано, что порождение знаковой системы той или иной интерактивной среды происходит за счёт выбора соответствующей метафоры. Задачей проектировщика в связи с этим является описание всего визуального языка, а не его отдельных диалоговых элементов.

Из требования системности вытекает рекомендация о нежелательности создания интерфейса на основе нескольких мелких метафор, описывающих детали и компоненты диалога. Это требование подразумевает также необходимость соотносываться как с другими уже существующими компьютерными метафорами, так и с общими идеями, имеющими место в данной среде, которые можно рассматривать как глобальные метафоры. (Пример такой глобальной метафоры уже упоминался. Это метафора мира, как некоего сверхофиса.) Не стоит использовать метафоры, базирующиеся на жаргонизмы и, тем более, на жаргонное употребление глаголов.

Визуализация рассматривается нами в этой работе, как построение видимого отображения ментальных и/или компьютерных моделей, а интерпретация — как некоторая двойственная к визуализации операция, как построение ментальных (и возможно компьютерных) моделей по видимым изображениям.

Успешная визуализация, построенная на базе “хорошей” метафоры, должна опираться на уже имеющуюся у пользователя модель явлений и/или процессов и хотя бы не противоречить его репрезентативным когнитивным структурам. Возможен анализ видов отображения, базирующихся на соответствующие метафоры, по способу интерпретации — имеет ли в них место непосредственная интерпретация визуальных образов или используются не прямые методы с расшифровкой и истолкованием сложных текстов. Уменьшение сложности интерпретации можно рассматривать как ещё одно условие “хорошей” метафоры для систем компьютерной визуализации.

Изучение действия удачных компьютерных метафор указывает на необходимость “разозначивания” посредством визуализации и/или непосредственного указания на рассматриваемые сущности. То есть метафора успешна, если уменьшает общую сложность компьютерной модели, в том числе сложность пользовательского интерфейса. Необходимо показывать изучаемые сущности, а не рассказывать о них посредством неких графических нотаций. (Отметим, что

этот критерий неприменим к случаям коммуникативной визуализации, когда и допустимо, и необходимо использование визуальных “текстов”, использующих интуитивно ясную графическую нотацию.)

Итак, нами был выписан набор критериев, налагаемых на исходное и целевое множество при порождении метафоры. Среди них — подобие свойств объектов множеств, визуализируемость исходного множества, привычность (узнаваемость) его объектов, а также богатая структура взаимосвязей между ними.

Кроме рассмотренных критериев выбора метафор существуют принципы построения на их базе видов отображения. Среди них такие, как правдивость, лаконичность, выразительность, ясность.

Интерпретация визуализации и интерактивных манипуляций, построенных на базе данной метафоры, восстанавливает (или создает заново) у пользователя некоторые ментальные структуры, в которых представляется картина явлений. Выведено правило, что структура построенного визуального образа не должна противоречить структуре исходного (ментального) объекта. При интерпретации визуального представления объекта не должно возникать отношений, отсутствовавших в ментальном прообразе.

В работах известного американского психолога Б. Тверски (B. Tversky) и её коллег рассматривались общие принципы построения эффективной (не обязательно компьютерной) визуализации. Показано, что эффективность визуализации достигается при следовании двум общим принципам — принципу соответствия и принципу приемлимости. Согласно первому из них структура и содержание визуализации должно соответствовать структуре и содержанию требуемого ментального представления. Согласно второму — структура и содержание визуализации должны быть легко и точно восприняты и поняты [78], [77].

Критерии построения видов отображения, включая рассмотренный в разделе 5 критерий сохранения структур, могут рассматриваться в более общем контексте когнитивных принципов проектирования визуализации.

Таким образом, получен набор правил позволяющих выбрать метафору, также как и построить на её базе правильный набор видов отображения визуальной интерактивной системы. При этом нет необходимости добиваться полноты метафоры — достаточно подобия хотя бы одного свойства основных объектов целевого множества и множества-источника. Привычность метафоры не означает обязательного поиска множества-источника среди бытовых явлений. Так, например, анализ успешных метафор интерфейса показывает, что для сред автоматизации конторского труда и подобных специализированных систем различного назначения привычные множества-источники метафор следует строить на основе анализа деловой и трудовой активности потенциальных пользователей. (В [2], [3] содержится также критика метафор, построенных на основе экзотических понятий.)

В литературе неоднократно рассматривались задачи проектирования метафор, также как и формализации их поиска. Выдвигалась идея автоматизации создания визуальных метафор и соответствующих систем визуализации, делались попытки автоматической и полуавтоматической генерации метафор и их образности на основе моделей прикладных областей и решаемых задач. (См. примеры в [62], [44] [34], [35].) Пока не удаётся решить основную задачу генерации метафор, то есть автоматизировать выбор области-источника метафоры. Однако, сам процесс поиска областей, удовлетворяющих указанным критериям, даёт достаточно однозначные результаты в определении области-источника. Следуя им и добиваясь адекватности в визуализации (определяющей те визуальные свойства, которые наилучшим образом обеспечивают решение конкретной прикладной проблемы данным пользователем или классом пользователей), мы еще более сужаем круг возможных источников в процессе метафоризации.

В области теории компьютерной метафоры важным является уточнение введенного нами ранее понятие “работа метафоры” [2], что послужило основой для анализа действия конкретных метафор интерфейса и визуализации. Базой анализа являются описанные нами методики представления информации с помощью данной метафоры, свойства метафорических объектов

и результаты их интерпретации пользователями. Предложена “формула” метафоры, в которую включаются упрощенные описания исходной и целевой областей, идеи уподобления, результата работы метафоры, а также некоторые “волшебные”, то есть несуществующие в реальности, но полезные (типа “клика мыши”) возможности интерфейса. Целью анализа является выявление структуры метафор, понимание принципов их функционирования, получение основы для сравнения и оценки метафор.

Результат действия метафоры оказывается связанным с появлением некоторого “метафорического” пространства, построенного на основе реальностей исходного и целевого пространств. Именно в этом пространстве существуют метафорические объекты и именно его логике подчиняются их поведение и отношения.

По нашему мнению в рамках развития теории метафоры возможна формализация предложенных понятий, включая понятия формулы метафоры и её действия, на базе некоторой логической модели. Формализация может оказаться важной и в плане практических задач, например, при решении обратной задачи поиска объектов, подходящих для представления при помощи заданной метафоры. Имея выписанную формулу уже существующей метафоры и описание структуры её множества–источника, можно по ним найти объекты соответствующего целевого множества, удовлетворяющего имеющимся критериям.

В нашей работе рассматривались три типа структур:

- репрезентативные когнитивные структуры, существующие у проектировщиков и пользователей визуальных интерактивных систем (авторов и интерпретаторов визуальных текстов);
- структуры видов отображения визуальных интерактивных систем;
- структуры взаимосвязей понятий в области–источнике и целевой области метафоры, возникающие в процессе метафоризации.

Изучение этих структур позволяет как анализировать и оценивать существующие проектные решения, так и научиться создавать системы с необходимыми свойствами.

Интерес к компьютерным метафорам связан с тем, что поиск метафор интерфейса и визуализации являются существенной частью процесса проектирования специализированных и персонализированных интерактивных визуальных сред. В этом плане наши исследования достаточно близки к такой области, как анализ применимости и используемости интерфейса (usability). В литературе по данной (практической!) дисциплине рассматривается необходимость изучения пользователя, его деятельности и активности, построения соответствующих метафор, также как и многие другие проблемы. (Некоторые результаты см. в [41, 48, 70].) Сюда же можно отнести исследования в области моделирования пользователей и изучения психологии восприятия образов на графических носителях [5]. Необходимость комплексного исследования проблем, возникающих в связи с когнитивными аспектами визуализации, отмечена в [42]. Эти проблемы связаны с тем, как люди извлекают значение из изображений, как они понимают изображение и как изображение оказывается наполнено значением. Соответственно занимаются ими такие дисциплины как психофизика, распознавание изображений и семиотика. Можно считать, что нашей задачей является выработка когнитивной составляющей анализа применимости и используемости интерфейса. Изучение структур метафор, метафорических пространств и видов отображения позволило ответить на часть из возникающих вопросов. Однако необходимы следующие этапы изучения данной проблематики на базе привлечения результатов из смежных областей, имеющих отношение к нашей основной теме — поддержке совместной работы человека и компьютера.

Благодарность

Авторы выражают свою благодарность П.А. Васёву, Д.Ю. Горбашевскому, А.И. Зубареву, О.А. Игнатъевой, А.Ю. Казанцеву, Л.Б. Рябининой, чьи идеи по созданию конкретных метафор интерфейса и визуализации очень помогли нам при написании работы.

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. **Авербух В.Л.** Визуализация программного обеспечения. Конспект лекций. Екатеринбург. Мат.–мех. ф–т. УрГУ, 1995.
2. **Авербух В.Л.** Метафоры визуализации // Программирование, 2001. N 5, С. 3–17.
3. **Авербух В.Л.** К теории компьютерной визуализации // Вычислительные технологии Т. 10, N 4, 2005, стр 21–51.
4. **Авербух В.Л.** Метафора интерфейса и метафора визуализации. Какая теория нам нужна? //ГрафиКон'2006, 1–5 июля 2006. Россия. Новосибирск, Академгородок. Труды Конференции. Новосибирск. Институт Вычислительной математики и математической геофизики. 2006. Стр. 262–267.
5. **Авербух В.Л., Авербух Н.В., Перевалов Д.С., Топорков Д.Н., Топоркова И.В.** Постановка проблемы психологического обоснования моделирования пользователей при разработке специализированных визуальных систем // 15–я Международная конференция по компьютерной графике и ее приложениям ГрафиКон'2005 20–14 июня 2005, Россия, Новосибирск, Академгородок Труды Конференции. Новосибирск Институт Вычислительной математики и математической геофизики. Стр. 329–331.
6. **Авербух В.Л., Байдалин А.Ю.** Разработка средств визуализации программного обеспечения параллельных вычислений. Визуальное программирование и визуальная отладка параллельных программ // Вопросы атомной науки и техники. Сер Математическое моделирование физических процессов. 2003. Вып. 4. С. 68–80.
7. **Авербух В.Л., Байдалин А.Ю.** Разработка средств визуализации программного обеспечения параллельных вычислений. Оптимизация программ // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2004. вып. 1. С. 70–80.
8. **Авербух В.Л., Байдалин А.Ю., Васев П.А., Исмагилов Д.Р., Зенков А.И., Манаков Д.В., Перевалов Д.С., Шагубаков М.Р.** Задачи визуализации параллельных вычислений. Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2002. Вып. 3. С. 40–52.
9. **Авербух В.Л., Байдалин А.Ю., Горбашевский Д.Ю., Исмагилов Д.Р., Исмагилов Т.Р., Чернин Р.М.** Разработка видов отображения в специализированных системах компьютерной визуализации // Труды 12 Международной Конференции по Компьютерной графике и Машинному Зрению ГрафиКон'2002. Нижний Новгород, 16–21 сентября 2002 г. Нижегородский Гос. Университет. Нижний Новгород, 2002. Стр. 184–189.
10. **Авербух В.Л., Байдалин А.Ю., Исмагилов Д.Р., Казанцев А.Ю.** Состояние дел в визуализации программного обеспечения параллельных вычислений // 15–я Международная конференция по компьютерной графике и ее приложениям. ГрафиКон'2005 20–14 июня 2005, Россия, Новосибирск, Академгородок Труды Конференции. Новосибирск. Институт Вычислительной математики и математической геофизики. Стр. 179–186.
11. **Авербух В.Л., Байдалин А.Ю., Исмагилов Д.Р., Казанцев А.Ю., Поддубная С.В.** Трехмерное представление графа вызовов функций // Супервычисления и математическое моделирование: Тезисы международного семинара. Саров, 2003. ВНИИЭФ–РФЯЦ Стр. 12–13
12. **Авербух В.Л., Байдалин А.Ю., Казанцев А.Ю., Рябинина Л.Б.** Метафоры и комплексные виды отображения для систем информационной визуализации // Пробл. теорет. и прикл. математики: Тр.36 Регион.мол.конф. Екатеринбург, ИММ УрО РАН, 2005. Стр. 314–315.
13. **Авербух В.Л. Гребенникова И.В., Килина С.В., Рябинина Л.Б.** Специализированная система информационной визуализации для нужд онко–гематологического центра // Мат. методы в медицине и биологии Материалы 3 Урал. науч.–практ. конф., Екатеринбург, 11 мая 2001 г. Екатеринбург, УрМА, 2001, с. 9–11.
14. **Авербух В.Л., Зенков А.И., Исмагилов Т.Р., Манаков Д.В., Пыхтеев О.А., Юртаев Д.А.** Разработка специализированных систем научной визуализации // Алгоритмы и програм. средства парал. вычислений: Сб. науч. тр. / ИММ УрО РАН. Вып. 4, Екатеринбург, 2000, С. 3–23.
15. **Авербух В.Л., Юртаев Д.А.** Методика разработки специализированных систем визуализации на примере задачи построения мостов в линейных дифференциальных играх // Алгоритмы и програмные средства параллельных вычислений, Выпуск 2, Екатеринбург, ИММ УрО РАН,. 1998, стр. 3–9.
16. **Байдалин А.Ю., Исмагилов Д.Р.** Средства представления структур в системах визуализации программного обеспечения // ГрафиКон2006, 1–5 июля 2006. Россия. Новосибирск, Академгородок. Труды Конференции. Новосибирск. Институт Вычислительной математики и математической геофизики. Стр. 271–274.

17. **Васев П.А., Перевалов Д.С.** О создании методов многомерной визуализации // Труды 12-й международной конференции по компьютерной графике и машинному зрению “ГрафиКон’2002” Нижний Новгород, 16 сентября – 21 сентября 2002 года. Нижегородский государственный университет им. Н.И.Лобачевского, 2002, с. 431–437.
18. **Горбашевский Д.Ю., Казанцев А.Ю., Манаков Д.В.** Параллельная фильтрация в системе визуализации параллельных вычислений // ГрафиКон’2006, 1–5 июля 2006. Россия. Новосибирск, Академгородок. Труды Конференции. Новосибирск, Институт Вычислительной математики и математической геофизики. 2006. Стр. 333–336.
19. **Гусев С.С.** Наука и метафора. Л. Издательство ЛГУ, 1984.
20. **Заботин Д.О., Смольянинов А.В.** Среда графического конструирования программ BLS. Версия 1.0 // Рекламно-информационный листок N 59-91. Ленинградский центр научно-технической информации. Л. 1991.
21. **Зенкин А.А.** Когнитивная компьютерная графика. М. Наука. 1991.
22. **Иванов А.Г., Краснов В.П., Кумков С.И.** Расчет констант скоростей элементарных реакций процесса динамического кинетического расщепления // Алгоритмы и средства параллельных вычислений, вып. 4, Екатеринбург, ИММ УрО РАН, 2000. Стр. 99–110.
23. **Исмагилов Д.Р., Шарпан С.В.** Метафора комнаты. Примеры разработки систем компьютерной визуализации // Проблемы Теоретической и прикладной математики. Труды 34 Региональной Молодежной конференции. Екатеринбург, ИММ УрО РАН. 2003. Стр. 264–268.
24. **Лакофф Дж., Джонсон М.** Метафоры, которыми мы живем. М. УРСС, 2004.
25. Рецкер Я.И. Теория перевода и переводческая практика. Очерки лингвистической теории перевода. М. Издательство Р. Валент. 2004.
26. **Самарский А.А.** Математическое моделирование и вычислительный эксперимент // Вестник АН СССР 1979, N 5. Стр. 38–49.
27. **Самарский А.А., Михайлов А.П.** Математическое моделирование. М. Физматлит. 2002.
28. **Чертов Л.Ф.** Знаковость. Опыт теоретического синтеза идей о знаковом способе информационной связи. Санкт-Петербург. Издательство С.-Петербургского Университета. 1993.
29. **Чуприкова Н.И.** Умственное развитие и обучение (к обоснованию системно-структурного подхода). Москва-Воронеж. Издательский дом РАО. 2003.
30. **Abramson D., Foster I., Michalakes J., Sosic R.** Relative Debugger: A New Methodology for debugging Scientific Applications // Communication of the ACM. V. 39, No. 11 (November 1996), pp. 69–77.
31. **Alfert K., Fronk A., Engelen F.** Experiences in 3-Dimensional Visualization of Java Classes Relations // SDPS Journal of Design Process Science, Vol. 5, No. 3 (September 2001), pp. 91-106.
32. **Averbukh V.L.** Towards the Conceptions of Visualization Language and Visualization Metaphor Proceedings of IEEE Symposia on Human Centric Computing Languages and Environments: September 5–7, 2001 Stresa, Italy. IEEE. 2001. P. 390–391.
33. **Averbukh V.L.** The Specialized Systems of Scientific Visualization // AIP Conference Proceedings (ZABABAKHIN SCIENTIFIC TALKS – 2005: International Conference on High Energy Density Physics). August 3, 2006. Volume 849, pp. 481–486.
34. **Averbukh V.L., Konovalov A.V., Tarskikh I.V., Vorzopov V.V.** Analysys of Visual Methaphors and Languages. Toward Prototyping of Software Visualizing Systems // “East–West”. International Conference on Human-Computer Interaction EWHCT’94. St.–Peterburg, Russia, 2–6 August, 1994”. Proceedings. ICSTI. Moscow. 1994. V.I, pp. 244–254.
35. **Barbosa S.D.J., de Souza C.S.** Extending software through metaphors and metonymies // Proceedings of the 5th international conference on Intelligent user interfaces. January 09-12, 2000, New Orleans, Louisiana, United States. Pp. 13-20,
36. **Barr P., Khaled R., Noble J. Biddle R.** A taxonomic analysis of user–interface metaphors in the Microsoft Office Project Gallery // Proceedings of the Sixth Australasian conference on User interface. Volume 40. Australian Computer Society, Inc. Newcastle, Australia, 2005. Pp. 109–117.
37. **Barr P., Noble J., Biddle R.** A Semiotic Model of User–Interface Metaphor // Ch. 13 of Book “Virtual, Distributed and Flexible Organisations”. Publisher Springer Netherlands. 2005. Pp. 189–215
38. **Black M.** Models and metaphors // Studies in language and philosophy. Ithaca, 1962.
39. **Black M.** More about metaphor // Metaphor and thought. Cambridge, 1979. P.19–45.
40. **Brown Th. L.** Making Truth: Metaphor in Science. Urbana Champaign. University of Illinois Press, 2003.

41. **Carlshamre P.** A Field Study in Usability Engineering: Bringing in the Technical Communicators // Technical Report no LiTH-IDA-R-94-44, Dept. Computer and Information Science, Linköping University, S-581 83 Sweden
42. **Duke D.J., Brodli K.W., Duce D.A., Hernan I.** Do You See What I have Mean? // IEEE Computer Graphics and Application. May/June 2005. Pp. 6–9.
43. **Friedman W.H.** A Theory of Metaphors in Information Technology // Proceedings of the Association for Information Systems, Aug., 1998. Pp. 826–828.
44. **Gazendam H.W.M.** Information System Metaphors (Groningen University, NL & Twente University, NL) // <http://www.econ.uba.ar/servicios/publicaciones/journal3/contents/HGazendam/methaphors.htm>
45. **Goguen J.** (1993) Semiotic Morphisms, Representations, and Blending for User Interface Design // <http://www.cs.ucsd.edu/users/goguen/pps/uid.ps>
46. **Goguen J.** (1993) Algebraic Semiotics // <http://www.cs.ucsd.edu/users/goguen/projs/semio.html>
47. **Goosen H.A., Hinz P., Polzin D.W.** Experience Using The Chiron Parallel Program Performance Visualization System. 1997. <http://www.nhse.org/rib/repositories/ptlib/objects/Asset/chiron.html>
48. **Green Th. R. G., Petre M.** Usability Analysis of Visual Programming Environments: A “Cognitive Dimensions” Framework // Journal of Visual Languages and Computing, Volume 7, Number 1, March 1996, pp 131–174.
49. **Greenberg S., Roseman M.** Using a Room Metaphor to Ease Transitions in Groupware / Research report 98/611/02, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, January.
50. **Heath M., Malony A., Rover D.** The Visual Display of Parallel Performance Data // IEEE Computer, V.28, N 11, (November 1995) pp. 21–29.
51. **Hirakawa M., Tanaka M., Ichicawa T.** An Iconic Programming System HI-VISUAL // IEEE Transactions on Software Engineering. Vol. 16 No 10 (October 1990). p.1178–1184.
52. **Hutchings D.R., Stasko J.** QuickSpace: New Operations for the Desktop Metaphor // CHI 2002 Extended Abstracts, ACM Press, pp. 802–803.
53. **Johnson G.J.** Of Metaphor and the Diffulty of Computer Discourse // Communication of the ACM. Vol. 37, No 12 (December 1994). Pp. 97–102.
54. **Kenyon R.A.** Macintosh Instructory Programming: A Guide to Programming the Macintosh Computer using Visual Interactive Programming. London. Mainstay, 1988.
55. **Khaled R., Barr P., Noble J., Biddle R.** System Metaphor in “Extreme Programming”: A Semiotic Approach. Presented at OrgSem 2004, Portugal // <http://www.orgsem.org/papers/13.pdf>
56. **Kuhn T.** The Structure of Scientific Revolutions, 2nd ed., The University of Chicago Press, Chicago, 1970, ch. 4.
57. **Kuhn T.** Metaphor in science. In: Metaphor and thought. Ortony A. ed. Cambridge: Cambridge University Press. 1979. Pp. 533–542.
58. **Kuhn W.** 7+/-2 Questions and Answers about Metaphors for GIS User Interfaces // Cognitive Aspects of Human–Computer Interaction for Geographic Information Systems. (Nyerges, T.L., et al., eds.), Series D: Behavioural and Social Sciences, Vol. 83, Dordrecht, The Netherlands, Kluwer Academic Publishers, pp. 113–122.
59. **Kuhn W., Frank A.U.** A Formalization of Metaphors and Image-Schemas in User Interfaces // Cognitive and Linguistic Aspects of Geographic Space, NATO ASI Series, Kluwer Academic Publishers, 1991. Pp. 419–434
60. **Lakoff, G.** . The contemporary theory of metaphor. In A. Ortony (Ed.), Metaphor and Thought (2nd ed.). Cambridge. Cambridge University Press. (1993) Pp. 202–251.
61. **Lieberman H.** Three-dimensional representation for program execution // Visual Programming Environments; Application and Issues. Los Alamitos: IEEE Comput. Soc. Press. 1990. Pp. 555–560.
62. **Marcus A.** Metaphor design in user interfaces: how to manage expectation, surprise, comprehension, and delight effectively // Human Factors in Computing Systems. CHI 97 Extended Abstracts. Atlanta, Georgia USA, 22–27 March 1997. ACM, 1997, p. 172–173.
63. **Moran Th.P., Zhai Shumin** Beyond the Desktop Metaphor in Seven Dimensions // To be published as a chapter in the book Designing Integrated Digital Work Environments: Beyond the Desktop Metaphor Victor Kaptelinin and Mary Czerwinski (Editors) <http://www.almaden.ibm.com/u/zhai/papers/DesktopMoranZhai.pdf>
64. **Musil S.** Monitoring Parallel Programs with INHOUSE // <http://www.ani.univie.ac.at/ani/research/Monit.html> 35.

65. **Osawa N.** An Enhanced 3-D Animation Tool for Performance Tuning of Parallel Programs Based on Dynamic Models // SPDP 98 Welches Or. USA. Pp. 72–80.
66. **Pelletier F.J.** A History of Natural Deduction and Elementary Logic Textbooks J. Woods, B. Brown (eds.) Logical Consequence: Rival Approaches and New Studies, Vol. 1. (Oxford: Hermes Science Pubs). pp. 105–138. <http://www.sfu.ca/~jeffpell/papers/pelletierNDtexts.pdf>
67. **Pfister H.-R., Schuckmann C., Beck-Wilson J., Wessner M.** The Metaphor of Virtual Rooms in the Cooperative Learning Environment CLear // Cooperative Buildings - Integrating Information, Organization and Architecture. Proceedings of CoBuild '98, Darmstadt, Germany. Pp. 107–113.
68. **Reed D., Scullin W., Tavera L., Shields K., Elford Ch.** Virtual Reality and Parallel Systems Performance Analysis // IEEE Computer, V.28, N 11, (November 1995), pp. 57–67.
69. **Reiss S.P.** Visual languages and the GARDEN System // Visualization in Programming. (Lecture Notes in Computer Science, N 282). Berlin. Springer-Verlag. 1987. Pp. 178–198.
70. **Ritter F. E., Van Rooy D., St. Amant R.** A user modeling design tool for comparing interfaces // Computer-Aided Design of User Interfaces III, Proceedings of the 4th International Conference on Computer-Aided Design of User Interfaces CADUI'2002. Kluwer Academics Publisher, Dordrecht. Pp. 111–118.
71. **Roberts J.C.** Display Models — ways to classify visual representations // International Journal of Computer Integrated Design and Construction. V. 2, N 4 (December 2000), pp. 241–250.
72. **Shu N.C.** Visual Programming Languages: A Perspective and Dimension Analysis // Visual Languages. New York. Plenum Publishing Corporation. 1986. Pp. 11–34.
73. **Spence R.** Information Visualization. L. Addison-Wesley, 2001.
74. **Stubblefield W.A.** Source Selection for Analogical Reasoning An Interactionist Approach / Dissertation, University of New Mexico, 1995 // <http://www.wmstubblefield.com/dissertation/dissertation.pdf>
75. **Tscheligi M., Musil S.** An Information Manipulation Environment for Monitoring Parallel Programs // AVI 1994: Bari, Italy. AVI'94, Proceedings of the Workshop on Advanced Visual Interfaces, June 1–4, 1994, Bari, Italy. ACM, 1994. P. 246–248
76. **Turner M., Fauconnier G.** Conceptual integration and formal expression // Journal of Metaphor and Symbolic Activity, 10 (3) 1995, pp. 183–204.
77. **Tversky B., Agrawala M., Heiser J., Lee P., Hanrahan P., Phan D., Stolte Ch., Daniel M.-P.** Cognitive Design Principles for Automated Generation of Visualizations // Applied Spatial Cognition; From Research to Cognitive Technology. Mahwah, N.J. Lawrence Erlbaum Associates, 2007. Pp. 53–74.
78. **Tversky B., Morrison J.B., Betrancourt M.** Animation - can it facilitate // Int. J. Human-Computer Studies, (2002), 57, pp. 247–262.
79. Visualization in Scientific Computing, Special Issue, ACM SIGGRAPH Computer Graphics, V. 21, N 6, November 1987.
80. **Watch What I Do.** Programming by Demonstration. (Ed. Allen Cypher) MIT Press. Cambridge, Mass. (1993)