

УДК 573.4+001.8

## УВЕЛИЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ РЕСУРСОВ НАУЧНОГО ИССЛЕДОВАНИЯ В БИОЛОГИЧЕСКИХ ОБЛАСТЯХ

*Д.С. Тарасов, Н.И. Акберова*

### Аннотация

Основной целью является разработка архитектуры системы, позволяющей многократно повысить эффективность интеллектуальной работы, выполняемой индивидуумами и малыми исследовательскими группами, работающими с частично структурированными проблемными областями. В качестве одного из основных компонентов такой системы рассматривается взаимодействие *человек – компьютер*, использующее особым образом организованные программные ресурсы.

### Введение

Требования, предъявляемые современной наукой к интеллектуальным возможностям исследования, постоянно растут вследствие роста объемов информации и сложности изучаемых систем. Возникает необходимость в наличии междисциплинарных знаний, а следовательно, наличии достаточно глубоких знаний в смежных областях. «Идеальный ученый должен обладать множеством талантов: он должен уметь продуцировать идеи, выдвигать гипотезы, подвергать их тщательной проверке, строить философскую систему, он должен быть технологом, организатором, популяризатором педагогом, должен заниматься внедрением своих достижений в жизнь и оценкой их практической значимости» [1]. В то же время интеллектуальные возможности разума отдельно взятого индивидуума с развитием прогресса практически не увеличиваются, а иногда даже уменьшаются. Это является одной из главных причин снижения качества научных исследований. Новые идеи, основанные на интеграции знаний из различных областей, труднее воспринимаются научным сообществом. По мнению некоторых исследователей, пик инноваций приходится на 1873 год, после чего темпы научного прогресса постоянно замедляются [2]. Одной из причин этого может служить рассмотренное противоречие между требованиями к интеллектуальным ресурсам и доступными ресурсами.

Существует базовая группа интеллектуальных задач, в которой встречаются наибольшие трудности и которая, соответственно, служит лимитирующим фактором, ограничивающим качество и спектр возможных научных исследований. Эта базовая группа включает:

1. изучение новой предметной области;
2. генерацию новых идей;
3. планирование исследований и систематизацию результатов;

4. организацию совместной работы нескольких человек над решением интеллектуальных задач.

В данной работе рассматриваются возможности для преодоления сложившейся ситуации применительно к решению этих проблем в такой области, как биология.

Еще в середине XX в. было выдвинуто предположение, в соответствии с которым эффективность интеллектуальной работы определяется доступными средствами для манипулирования символами, среди которых можно выделить несколько компонентов [3].

1. Артефакты – физические объекты, разработанные для удобства манипулирования предметами или материалами и, таким образом, для манипулирования символами.

2. Язык – способ, которым индивидуум разделяет общую картину мира на концепции, которые его разум использует для построения модели мира. В язык также входят символы, которые человек сопоставляет с этими концепциями и использует их для сознательного манипулирования символами (мышления).

3. Методология – методы, процедуры, стратегии и т. п., с помощью которых индивидуум организует свою целенаправленную активность (решение проблем).

4. Тренировка и обучение – создание условий, требующихся человеку для получения навыков для эффективного использования средств п. 1–3.

Установлено, что систематический подход к исследованию лимитирующих факторов во всех четырех компонентах может привести к обнаружению способов, существенно повышающих эффективность интеллектуальной активности [4].

Сначала мы очень кратко рассмотрим существующие положение дел по отношению к каждому из рассмотренных компонентов и их роли в решении изучаемых интеллектуальных задач.

#### **Артефакты.**

Среди физических объектов, разработанных для манипулирования символами, наиболее совершенным средством в настоящее время являются компьютеры. Наиболее распространенные интерфейсные средства включают клавиатуру, манипулятор типа «мышь» (touchpad для ноутбуков), монитор на основе ЭЛТ или ЖК. Дополнительные средства ввода и вывода информации включают в себя принтер и сканер. Реже можно найти сенсорный экран, хотя удобство такого рода системы несомненно. Программные и аппаратные средства, реализующие речевой ввод, в настоящее время имеют достаточно высокое качество, но используются редко.

Средством манипулирования символами является также разнообразное программное обеспечение. Когда перед исследователем возникает задача изучить новую предметную область, программы помогают ему лишь косвенно. Интернет и поисковые системы предоставляют доступ к исходной информации, базы данных и электронные картотеки способны эту информацию сохранять. Однако программные средства, оптимизирующие сам процесс изучения, объективно отсутствуют.

Для облегчения процесса создания новых идей разработан ряд программных средств, так называемых “Creativity Computer-Assisted Instruction Systems”. Примерами могут служить такие программы, как Visual Mind (<http://www.visual-mind.com>), Thoughtpath (<http://www.thoughtpath.com>). Показано, что их применение способно реально улучшить результаты [5]. Минусами является то, что эти программы ориентированы в основном на бизнес-приложения, основаны на разных принципах и никак не интегрированы с решением других интеллектуальных задач.

Некоторые средства для планирования исследования и систематизации результатов существуют в узко специализированных программных пакетах.

### **Язык.**

Естественные языки. Языки общения людей, являющиеся результатом общественного развития – это, в целом, языки общего назначения. Однако нужно помнить, что эти языки формировались в определенных условиях, для решения вполне определенных задач и под влиянием доступных физических средств записи, манипулирования и передачи символов. Все письменные языки адаптированы к возможностям физического носителя, используемого для их записи, а устные – к возможностям речевого и слухового аппарата человека.

Специальные искусственные языки существуют во многих областях научной деятельности. Математический символизм успешно используется для решения большого класса задач в естественных науках. Сочинение музыки облегчается использованием нотной записи. Программирование компьютеров является задачей, для которой было создано наибольшее количество искусственных языков.

В целом, для решения задач, которые являются предметом нашего рассмотрения, не было ранее создано каких-либо специальных языковых средств.

### **Методология.**

Существует большое количество различных процедур, созданных для решения самых разнообразных проблем. В рассматриваемой нами области, однако, необходимых процедур либо вовсе нет, либо они чересчур формализованы и неэффективны в силу своей косности и плохой проработанности. Фактически, каждый индивидуум, занимающийся научной деятельностью в частично-структурированных проблемных областях, вынужден сам создавать методы для организации своей работы. Методы работы в малых группах часто формируются спонтанно или на основании сложившихся традиций. Представляется вероятным, что систематический подход, направленный на создание улучшенных методов работы, может привести к значительному увеличению эффективности научного процесса.

Большое число методов, использующих специальные средства, было предложено [6] и позднее частично реализовано [3, 7, 8], однако в силу различных причин они до настоящего времени не нашли широкого применения.

**Тренировка и обучение.**

Следует выделять отдельно обучение применению технических средств, методологии и языку.

**Технические средства.**

Практически каждый студент в совершенстве владеет применением бумаги, карандашей и шариковой ручки. Если же речь идет о применении более совершенных средств манипулирования символьной информацией, таких, как компьютеры, обучение следует признать неудовлетворительным. Во многие курсы информатики не входит, например такая базовая вещь, как эффективное использование клавиатуры. Очевидно, что если человек не в состоянии быстро вводить информацию в компьютер, а испытывает проблемы каждый раз, когда требуется найти нужную кнопку, эффективность его работы значительно снижается. Проблемы с использованием манипуляторов типа «мышь» встречаются реже, но эффективное и правильное использование этого средства все же осваивается не всеми. Сенсорные экраны, средства распознавания речи и рукописного ввода вовсе не демонстрируются.

Если мы рассмотрим обучение использованию специальных программ, упомянутых ранее, то такое обучение широко не применяется.

**Методология.**

К сожалению необходимо признать, что процедуры и стратегии для решения ряда интеллектуальных задач не рассматриваются вообще, не тренируются и соответствующие им навыки. В число задач, способы решения которых остаются без систематического рассмотрения, входят все рассматриваемые нами, то есть:

- изучение новой предметной области;
- генерация новых идей;
- систематизация результатов;
- организация совместной работы нескольких человек над решением интеллектуальных задач.

Можно возразить, что перечисленные операции являются в своей основе творческими и учить им бесполезно. Это одновременно и верно и ошибочно. Дело в том, что даже такие творческие виды деятельности, как рисование картин или сочинение музыки, требуют свободного владения рядом техник, без которых они просто невозможны.

**Язык**

В связи с отсутствием специальных языков, обучение им не практикуется.

**1. Методы (процесс исследования)**

Поскольку решали одну из плохо структурированных задач, мы применили принципы собственных методов к ее решению. Одним из таковых принципов является следующее: если непонятно с чего начинать, следует начать с любой точки (набора утверждений) и последовательно продвигаться к тому, что кажется лучшим решением.

Мы начали со следующего набора утверждений:

1. Возможности компьютера при проведении исследований используются далеко не полностью.

2. Максимальная гибкость при использовании компьютера достигается при создании новых программ.

3. Сами по себе программы являются хорошим средством для выражения мощных идей даже в плохо структурированных областях.

Для проведения первой серии исследований мы взяли проблему организации работ по классификации биологических объектов – грибов *рода Trichoderma*. После изучения вопроса мы пришли к выводу, можно получить ряд выгод от использования специального формального языка описания, который и был построен и применен [9].

В результате проделанной работы по систематике триходермы было сделано следующее заключение:

- описание организма должно включать описание жизненного цикла;
- жизненный цикл можно описывать как компьютерную программу;
- классификацию организмов можно рассматривать как классификацию программ.
- создание баз данных для исследовательских работ может производиться самими исследователями, что способствует повышению качества получаемого продукта.

Эти выводы служат подтверждением положений 2 и 3.

Кроме того, был сделан ряд заключений о том, какие ограничения накладывает использование рассмотренного языка и как их следует устранять.

Были выявлены следующие ограничения и проблемы:

- архитектура языка провоцирует создание недостаточно корректных описаний;
- обучение представляет серьезные трудности;
- отсутствие достаточной гибкости и мощности для выражения определенных идей;
- статический, описательный характер системы.

Созданный новый язык был применен при создании системы для организации популяционных исследований. Первый этап состоял в разработке базы данных для исследований генетической обусловленности гипертонической болезни. В результате выполнения первого этапа были сделаны следующие заключения:

- применение частично формализованных средств представления информации об исследуемой области (компьютерных программ) позволяет улучшить понимание задачи и избежать методических ошибок при планировании и проведении исследований, а также при интерпретации результатов;
- остро ощущается недостаток визуальных средств для работы с системой;
- организация языка недостаточно последовательна и слишком усложнена.

Язык был подвергнут ревизии с целью сделать его принципы более последовательными и упростить работу. Был реализован интерпретатор, позволяющий исполнять программы, написанные на новом варианте языка.

На втором этапе разрабатывалась система для учета и систематизации данных для популяционных исследований. В результате проведенных исследований был сделан вывод о необходимости комплексного подхода, проект которого и представлен в данной работе.

## 2. Результаты: описание ConceptSpace

Предлагается комплекс средств более эффективного манипулирования символами, включающий методы, язык, программную оболочку, и способы тренировки необходимых навыков.

### 2.1. Обзор системы.

Компьютер предоставляет пользователю виртуальную среду, в которой тот может размещать любые объекты-концепции, конструируя их из имеющихся примитивов (рис. 1). Правила конструирования концепций и организации их взаимодействия определяются используемым языком. Необходимость существования правил диктуется рядом соображений. Существование правил дисциплинирует ум, облегчает систематизацию, создает базис для обмена информацией между разными людьми, а также позволяет компьютеру производить различные операции над концепциями.

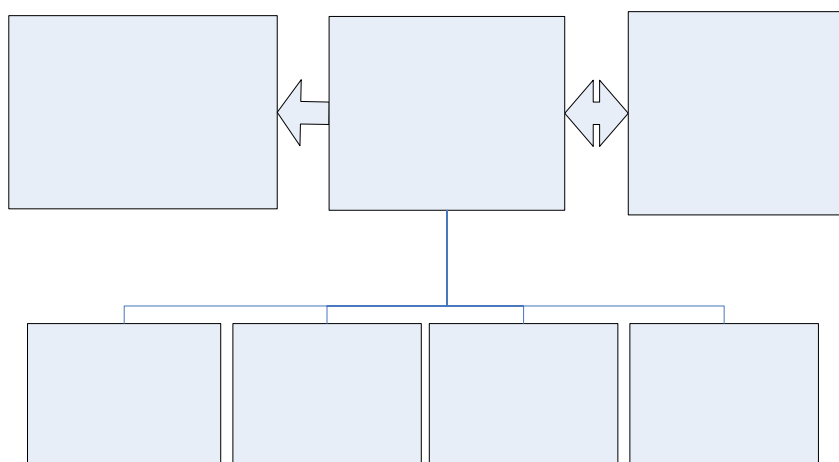


Рис. 1

Компьютер обеспечивает хранение информации, средства визуального представления концептуальных пространств и средства навигации.

Основным способом представления визуальной информации о концепциях являются *концептуальные сети*.

### 2.2. Программная компонента: Концептуальные сети.

Представим себе, что мы изучаем концепцию «вещества». Мы начинаем с главного узла, который называется «вещество», и видим, что от него отходит множество ветвей-связей (рис. 2).

Концептуальное пространство (КП) – это место где пользователь может располагать любые

Виртуал среда про



Рис. 2

В соответствии с нашими интересами мы выбираем определенный пункт, скажем «Формы вещества».

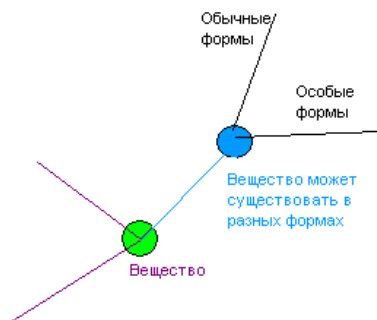


Рис. 3

Допустим, нас интересуют «Особые формы вещества» (рис. 3). Мы двигаемся дальше по этой связи.

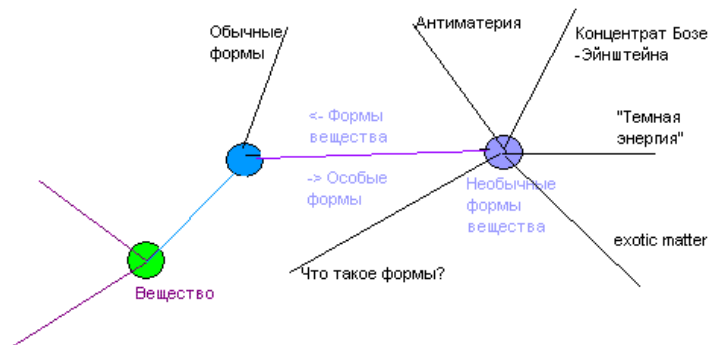


Рис. 4

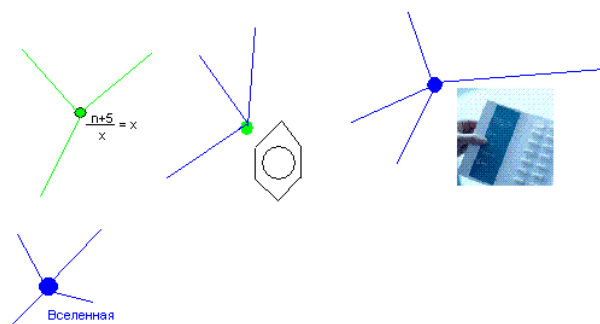


Рис. 5

Здесь мы можем видеть ссылки на особые формы вещества, ссылку на «Формы вещества» и формы вообще. С узлом «Необычные формы вещества» ассоциирован текстовый комментарий, который можно видеть на рис. 4.

**2.2.1. Концепции и связи.** Что содержится в узлах концептуальной сети? Логично предположить, что концепции. Что такое концепции, и какими они могут быть?

Любой блок информации, который может быть помещен в концептуальное пространство, есть концепция.

Концепция может представлять (рис. 5):

1. текстовый блок – набор предложений на естественном языке;
2. математическую формулу;
3. химическую формулу;
4. картинку, анимацию, звук;
5. фрагменты компьютерной программы;
6. результаты выполнения фрагментов программы.

**2.2.2. Концептуальные сети и гипертекст: сходства и различия.** Мы видим, что система напоминает переход по гиперссылкам в гипертекстовых системах. Сравним эти две системы по нескольким критериям (табл. 1).

**2.2.3. Трехмерные и  $n$ -мерные концептуальные пространства.** До сих пор все примеры КС приводились с использованием двухмерной проекции. Однако нет никакой причины, почему мы не можем использовать трехмерные проекции концептуальных пространств.

### 2.3. Язык.

В предыдущем рассмотрении мы не учитывали существование специальных правил выделения концепций, не определяли точно виды концепций и связей между ними. Все эти средства входят в *язык концептуальных пространств*. В качестве основы для такого языка мы предлагаем использовать язык ConceptSphere, разработанный нами в процессе работы над системой для организации популяционных исследований. Дальнейшее изложение представляет собой формальное описание данного языка.



Табл. 1

Сравнение КС и гипертекста

Гипертекст	КС
Навигация осуществляется посредством ссылок, вставленных в текст.	Навигация осуществляется с помощью связей, прямого перемещения (прокрутки), панорамирования (ближе, дальше).
Общая картина не видна.	Общая картина видна посредством изображения узлов и связей.
Единицы информации могут быть очень велики.	Единицы информации обычно невелики.
Создание гипертекстовых ссылок обычно является делом специалистов, процессы создания ссылок и просмотра разделены: разные программы.	Создание новых связей является естественным процессом.

**2.3.1. Общие положения.** Все в языке является объектами. Объекты взаимодействуют друг с другом только при помощи сообщений. Сообщениями могут обмениваться только связанные (соединенные) объекты. Связывание является механизмом реализации инкапсуляции данных. Объекты могут размещаться в разных пулах. Пулы играют роль сред периода исполнения (баз данных в Прологе). Непосредственно связываться могут только объекты, находящиеся в одном пуле. Пулы могут быть вложенными друг в друга или просто соединяться между собой.

**2.3.2. Методы.** Каждый объект реализует определенный интерфейс, определяющий набор сообщений, которые могут обрабатываться данным объектом. Описание сообщения вместе с набором процедур, описывающих реакцию объекта на прием данного сообщения, будет называться методом объекта.

Описание сообщения включает в себя название сообщения и перечень классов, которые могут быть переданы в качестве аргументов. Само сообщение состоит также из названия и списка ссылок на объекты-аргументы. В результате выполнения метод возвращает объект.

Метод включает в себя фрагмент кода, привязанный к определенному сайту связывания. Вызов метода приводит к получению объекта, возвращаемого методом.

*Использование.*

```
Account.Deposit(Amount=10,Unit=Dollars)
```

```
Account.Deposit 10.+ 5 Dollars
```

Единственный способ заставить объект сделать что-либо – это вызвать его метод. Таким образом, программа состоит из последовательности различных указаний различным объектам.

*Структура методов.*

Метод состоит из фрагмента кода и аргументов.

Фрагмент кода – последовательность вызовов методов, разделенных символом ‘;’ и заключенных в фигурные скобки:

```
{console.Write Hello!;console.Write This is me!;}
```

Аргументы являются локальными сайтами связывания (переменными) методов, к которым привязываются аргументы, употребленные при вызове метода. Аргументы причисляются через запятую и отделяются от тела метода знаком |.

Каждый метод имеет локальный сайт связывания Result, который используется для возвращения результата работы метода:

```
{Number ToAdd; Number Result|Result.= ToAdd.+ 5;}
```

**2.3.3. Сайты связывания (переменные экземпляров).** Каждый объект может включать в себя набор переменных экземпляров (сайтов связывания), к которым могут привязываться другие объекты. Один объект может одновременно быть связанным с несколькими сайтами связывания. Сайт связывания имеет тип и может связывать объекты определенного класса или классов-потомков данного класса.

**2.3.4. Тело метода.** Тело метода состоит из последовательности операций отсылки сообщений объектам. Части тела метода также являются объектами и могут отсылаться в качестве аргументов сообщений.

Тело метода также может содержать локальные сайты связывания

```
<Classname> <SiteName>
```

```
Contains Geographic Region MyRegion
```

**2.3.5. Связывание и отсылка сообщений.** Связывание объекта из одной переменной состояния с другой переменной состояния:

```
BindingSite.= BindingSite1
```

Связывание объекта, создаваемого методом:

```
MyAccount.= BankAccount.New
```

*Способы связывания объекта из пула (унификация).*

Связывание любого объекта данного класса

```
Pool.Bind <classname>
```

Связывание объекта, возвращающего определенные ответы на сообщения,

```
(Bind <classname>
 {<Conditions> }
 )
```

*Отсылка сообщений связанному объекту (вызов методов)*

```
BindingSite.MessageName Param1 Param2 ParamN;
```

```
BindingSite.MessageName(Name='newname', Color = new Blue);
```

### 2.3.6. Создание нового объекта

<модификаторы> <ClassName>.New

geographic region.New

(number of Color).New

Освобождение объекта.

Сайты связывания также являются объектами. Они имеют методы release и destroy. Связывание нового объекта автоматически освобождает данный объект.

**2.3.7. Class.** Класс – это особый объект. Каждый пул может содержать только один экземпляр данного класса. Класс содержит спецификацию для создания какого-либо объекта. Класс содержит спецификацию методов и спецификацию свойств, методы, обеспечивающие редактирование спецификаций и создание объектов по заданным спецификациям. Класс специфицирует сам себя. Кроме того, класс отвечает за создание других классов, реализацию механизма наследования.

*Методы класса:*

New – создает экземпляр объекта данного класса;

KindOf – создает новый класс – наследник от класса-аргумента;

Method – добавляет метод к спецификации объекта класса;

RemoveMethod – удаляет метод из спецификации объекта класса;

(?)GetMember – возвращает спецификацию методов.

*Использование:*

Contains class Account;

Ok

Account.= Class.kindof Object;

Ok

Console.List Account

Account.= Class.kindof Object;

{

}

Account.Contains Integer Money;

Account.Contains Method Deposit;

Account.Deposit.=Method.new {integer toAdd|self.money.add toAdd};

Account.Method Deposit {integer toAdd|self.money.add toAdd};

Ok

Account.Deposit(10) → ошибка

Account.Money → ошибка

Contains Account MyAccount;

Ok

MyAccount.= Account.new;

Ok

MyAccount.Deposit 10

*Ok*

MyAccount.Money → 10  
(Account Money=10).Money → 10

**2.3.8. Object.** Базовый класс. Всякий другой класс является наследником класса object (кроме class, bindingsite).

Методы:

*ForEach*

Реализация перебора в пуле через связывание.

(спецификация\_связывания).foreach(BindTo=BindingSite, CodeFragment={<statements>})

(geographic region).ForEach(MyRegion, {MyRegion.Name:='Tatarstan'})

(integer

Value>1&Value<6).ForEach(I, {m:=\*(I,10)})

*Is(classname class, YES Code, No Code)* – определяет, является ли объект наследником класса class, выполняет код YES или No в зависимости от результата.

== <Object>:Boolean

Определяет отношение объекта к объекту-аргументу. Возвращает объект типа Boolean.

*ObjectList* – список объектов

Методы

Add(Object) – добавляет новый объект в список;

Remove(Object)

Remove(Integer) – удаляет объект из списка;

Get(Integer) – возвращает требуемый объект;

IsBefore(Object, Object):Boolean – определяет, находится ли один объект до другого в списке;

For(Object, Code) – выполняет операцию Code для каждого объекта Object

Boolean

Class True kindof Boolean

Class False kindof Boolean

*Pool*

Пул – это особый объект. Он отвечает за хранение находящихся в нем объектов, классов и глобальных переменных (сайтов связывания). При создании объекта методом New или класса методом KindOf класс вызывает метод пула Add и добавляет туда созданный объект. Пул также содержит методы Remove и Bind, которые отвечают за реализацию связывания.

Метод Bind является методом пула по умолчанию. Он возвращает объект.

*Аргументы*

Модификаторы Название класса.

Модификаторы Название класса Фрагмент кода. Фрагмент кода осуществляет проверку дополнительных требований к связываемому объекту. Должен возвращать значение типа Boolean.

### **Number.= Object.KindOf**

Объект-число

Методы

(метод класса) New <Number>

Создает объект-число с заданным значением

+ <Number> – возвращает сумму данного числа и числа-аргумента

– <Number> –возвращает разность данного числа и числа-аргумента

\* <Number> –Возвращает произведение данного числа и числа-аргумента

/ <Number> – Возвращает результат деления данного числа и числа-аргумента

Сравнение чисел.

> <Object>:boolean

< <Object>:boolean

Переопределяет метод ==

### **Пример:**

Contains Number MyMoney

*Ok*

MyMoney.= Number.New 56 (или тоже самое MyMoney.= 56)

*Ok*

MyMoney.= MyMoney.+ 4

*Ok*

Console.List MyMoney

60

### **String.= Object.KindOf**

Объект-строка

Методы

(метод класса) New <'string-value'>

Создает объект-строку с заданным значением

Переопределяет метод ==

**2.3.9. Модификаторы** (Modifier kindof class). Модификаторы можно применять совместно с идентификаторами классов для быстрого конструирования новых классов. Результатом применения модификаторов являются новые классы.

**2.3.10. Произвольные отношения** (Класс Relation, модификатор Relational). При составлении программы часто возникает необходимость добавить в нее отношения отличные от отношений наследования и включения. Класс Relation служит для установления такой связи.

*Свойства*

Relational Object<> //Объект связи

Relational Object Subject //Субъект связи

Конструктор New(object, subject) – создает новую связь и добавляет к списку Relations объекта и субъекта ссылки на данную связь.

Модификатор Relational добавляет в объект свойство number of relation Relations

*Использование*

Связь likes

Отношение 1 к 1

Утверждение John likes Jenn выражается как

Likes.New(John,Jenn); – связь образуется в пуле, где находятся объекты John и Jenn

(likes subject=Jenn).object возвращает объект John

(John.Relations.Items[0] as Likes).subject = Jenn

Отношение 1 к многим

Likes.New(John,[(computer speed>1000 Weight<5)]) //John likes computer with speed higher that 1000 and Weight less than 5

(likes object=John subject=Computer.New(speed=1300 weight=3))

*Ролевые модификаторы*

Модификатор Role kindof modifier, Relation

Модификатор роли добавляет в объект поддержку ролей. Роль – это форма связи между двумя или несколькими объектами, наличие или отсутствие которой определяет изменения в иерархической принадлежности объекта (при выполнении роли объект наследует свойства объекта этой роли). Так, например, «секретарь» является ролью, которую один человек выполняет по отношению к другому в определенных обстоятельствах.

*Использование*

Утверждение английского: James is father of John

James = (Human, father of John).New

James.is (father of John) возвращает true

(James.as father of John).execute

(father of John) возвращает James

### 2.3. Методы работы.

Список описанных здесь процедур не претендует на полноту. Его цель – показать способы применения предлагаемой системы при решении ряда задач, дать общее представление и продемонстрировать возможные преимущества.

**2.3.1. Изучение новой предметной области (формирование представления о предмете).** Когда перед человеком ставится задача изучить некоторую незнакомую ему проблему, он часто оказывается в затруднительном положении. Вот несколько примеров.

- Изучить имеющуюся литературу по вопросу о (*заболеваниях сосудов, устройстве подводных аппаратов, выращивании цветов и т.п.*). Это самый тяжелый случай, поскольку непонятно, ни что требуется, ни каким образом контролировать результат.

- Подготовить доклад (обзор) по некоторой проблеме.
- Найти ответ на какой-либо конкретный вопрос (*как спят жирафы, какую площадь надо покрыть солнечными батареями, чтобы обеспечить энергетические потребности города*).

Методы решения подобных задач нигде и никак не изучаются. Предполагается, что студент обучится этому самостоятельно посредством выполнения различных заданий. Разумеется, как правило, ничего подобного не происходит. Это равносильно тому, что дать человеку пианино и сказать, что он должен обучиться музыке, дать ему задание сыграть несколько довольно сложных музыкальных произведений. Как вы думаете, какой процент испытуемых обучится музыке самостоятельно и сколько людей приобретут в процессе отвращения к игре на пианино?

Изучить литературу по какому-либо вопросу означает выявить важные для данной области сущности, взаимоотношения между ними, а также установить, кем и когда были определены данные отношения и сущности.

#### *Процедура.*

Первый вопрос, который всегда возникает у любого человека, – с чего начинать. Мы предполагаем, что начинать можно с любой точки. Необходимо добыть какую-либо информацию о рассматриваемой проблеме. Одинаково подойдет короткая статья в энциклопедии, большая обзорная статья в журнале или работа, посвященная каким-либо частным вопросам. Берем, прежде всего, то, что есть под рукой.

Заносим наименование источника в концептуальное пространство.

Читаем источник, выделяем ключевые сущности, связываем их с источником отношением содержания и между собой подходящими соотношениями. После этого двигаемся к следующему источнику. Для выбора следующего источника можно воспользоваться полученной информацией, например, найти источник, разъясняющий значение определенной сущности. Далее мы создаем новый источник, размещаем его в пространстве и связываем его с предыдущим источником соотношением следования. Продолжаем выполнять операцию, пока не будет достигнут нужный уровень понимания проблемы.

#### *Преимущества процедуры:*

- проделанную работу можно оценить количественно;
- вся информация фиксируется на компьютере для дальнейшего использования;
- полученную сеть легче передать другому человеку;
- позволяет не возвращаться несколько раз к одному и тому же, что ускоряет работу;
- процедура может быть выполнена даже человеком, не имеющим большого опыта в решении данной проблемы.

**2.3.2 Генерация идей и систематизация результатов.** Организация исследовательского процесса всегда включает в себя генерацию новых идей, их систематизацию и планирование. Сюда включаются такие задачи, как:

- придумать способ исследования некоторой проблемы (*влияние генов на заболевания, поиск лучшей классификации некоторых организмов...*);
- создать план исследования, план написания статьи, доклада...

#### *Процедура.*

Придумывается какая-либо мысль, имеющая отношение к делу. Эта мысль помещается в концептуальное пространство. Придумывается следующая мысль, связывается с предыдущей отношением последовательности. Операция повторяется, пока не иссякнет список мыслей. После этого все размещенные концепции сортируются в группы, групповым концепциями также даются названия. Определяется, какие части проблемы остались за рамками рассмотрения, и процедура повторяется сначала до тех пор, пока не будет достигнуто требуемое решение.

#### *Преимущества процедуры:*

- организует процесс поиска решения;
- помогает избежать ошибок и учесть большее количество обстоятельств;
- допускает организацию работы нескольких человек.

**2.3.3. Организация продуктивной совместной работы нескольких человек.** Если к определенной интеллектуальной работе вместо одного человека привлечь двух, иногда эффективность работы может вырасти даже более чем в два раза. Однако чаще на практике наблюдается совсем другая ситуация – эффективность возрастает незначительно или даже снижается.

Любая проблема, описанная в предыдущих двух пунктах, может в принципе решаться совместно. При этом один человек в группе фиксирует результаты общей работы.

## **2.4. Обучение.**

### *Ключевые навыки.*

Обучение использованию системы означает, что обучаемый одновременно учится использовать программную оболочку, язык и процедуры решения задач. Нашей задачей является найти наиболее эффективный способ организации такого обучения. Суммируя все отдельные элементы системы, мы получаем:

- артефакты:
  - использование аппаратных средства взаимодействия с компьютером,
  - использование современных графических интерфейсов,
  - язык,
  - язык ConceptSphere, по сути являющийся языком программирования;
- процедуры:
  - исследование новой предметной области,
  - генерация идей и систематизация результатов,
  - организация совместной работы нескольких человек.

### *Методы обучения.*

Разработка новых систем организации интеллектуальной деятельности останется пустой тратой времени, если новые подходы не будут опираться на качественно иные методы обучения.



Если мы зададимся вопросом, почему некоторые предметы представляют особые трудности для изучения, первым ответом будет «потому что эти предметы сложнее остальных». Такой ответ, однако, неверен. Проблему представляет не сущность изучаемого предмета, а форма обучения. В прошлом, когда возможности манипулирования символами были ограничены использованием письменных средств, мы не имели особого выбора в вопросе формы обучения. Сегодня компьютеры предоставляют нам значительно большую гибкость, однако эта гибкость остается невостребованной, потому что значительно проще переносить старые формы обучения на новый носитель, чем создавать что-то принципиально новое [10].

Пожалуй не вызывает возражений утверждение о том, что наиболее естественной для человека формой обучения являются сказки и игры. Используя только эти две формы, дети обучаются огромному спектру знаний и навыков, не теряя при этом интереса, а значит, и эффективности обучения. Системное обучение в школе и вузе основано на других принципах в силу ряда причин, и недостаток этих принципов очевиден – процент детей, с интересом посещающих школу, так же, как и студентов, которым интересны занятия в вузе, ничтожно мал. Известный психолог А. Маслоу писал: «Детей не надо учить любопытству. Детей можно отучить от любопытства, и именно эта трагедия и разворачивается в наших школах» [1].

Рост производительности компьютеров, объема памяти, а также пропускной способности локальных и глобальных сетей, совместно с современными средствами разработки программного обеспечения позволяет нам предпринять попытку преодоления существующих ограничений. Для успеха здесь необходимо не только избавиться от существующих стереотипов, но провести обширные исследования в областях, находящихся на стыке нескольких, на первый взгляд, не связанных между собой дисциплин.

**2.4.1. Сценарное обучение.** Игровое и имитационное обучения с помощью компьютерных средств приобретают популярность в последнее время, хотя значительная часть программных продуктов для этих целей создается для обучения в различных сферах бизнеса [11]. В силу того, что предлагаемая система ориентируется на взаимодействие человека и компьютера, игровой и имитационный способы обучения представляются нам идеальным методом.

Мы предлагаем использование специальной многопользовательской игры-стратегии, основанной на том, что задачей каждого из игроков является получение или сокрытие определенной информации от других. Игровая среда представляет собой концептуальное пространство, специальным образом адаптированное для ее проведения. В ходе игры обучающиеся объединяются в команды, добывающие или защищающие информацию, и постепенно вырабатывают все описанные выше навыки.

### **Заключение**

На основе предыдущего опыта исследований предложен комплекс средств более эффективного манипулирования символами, включающий методы, язык, программную оболочку и способы тренировки необходимых навыков. Предпо-

лагается, что данный комплекс способен значительно повысить эффективность интеллектуальной работы исследователя в ряде предметных областей биологических исследований.

### Summary

*D.S. Tarasov, N.I. Akberova.* Computer aided creativity and intellect augmentation using semi-formal problem oriented environments.

This paper presents development of basic architecture of the system that allows significant amplification of efficiency of intellectual work performed by individuals and small groups in the field of biology. Human-computer interaction is considered as a main component of this system.

### Литература

1. *Maslow A.H.* Motivation and personality. – N. Y.: Harper, 1970. – 411 p.
2. *Jonathan H.T.* A possible declining trend for worldwide innovation // *Technological Forecasting & Social Change*. – 2005. – No 72. – P. 980–986.
3. *Engelbart D.C.* Augmenting human intellect: a conceptual framework / Summary report. – Stanford Research Institute, Menlo Park, California, USA. – 1962.
4. *Engelbart D.C., Watson R.W., Norton J.C.* The augmented knowledge workshop // *AFIPS Conf. Proc.* – 1973. – P. 9–21.
5. *Chun-Yang C., Ying-Yao C., Yi-Hui H., Tung-Ching L., Pei-Chen S.* Creativity computer-assisted instruction systems (CCAI) in Taiwan: system development & an experimental study // Paper presented at the AARE Conf. – Fremantle, 2001. – <http://www.aare.edu.au/01pap/cha01335.pdf>.
6. *Vannevar B.* As we may think // *The Atlantic Monthly*. – 1949. – No 7 – P. 101–108.
7. *Engelbart C.A.* Research center for augmenting human intellect // *AFIPS Conf. Proc. of the 1968 Fall Joint Computer Conference*. – San Francisco, 1968. – P. 395–410.
8. *Engelbart D.C., Lehtman H.* Working together // *BYTE Magazine*. – 1988. – No 12. – P. 245–252.
9. *Тарасов Д.С., Тухбатова Р.И., Акберова Н.И., Алимова Ф.К.* Формат представления биологических описаний гриба и его применение на примере рода *Trichoderma* // *Вестн. Татарстанского отделения Российской экологической академии*. – 2005. – № 2(24). – С. 44–49.
10. *Kay A.* Revealing the elephant: the use and misuse of computers in education // *Educom Review*. – 1996. – No 4. – P. 22–28.
11. *Faria A.J., Wellington W.J.* A Survey of simulation game users, former-users, and never-users // *Simulation & Gaming*. – 2004. – No 2. – P. 178–207.

Поступила в редакцию  
23.09.05

---

**Тарасов Денис Станиславович** – аспирант кафедры генетики Казанского государственного университета.

E-mail: [dtarasov@mntech.v.rusonyx.ru](mailto:dtarasov@mntech.v.rusonyx.ru)

**Акберова Наталья Ивановна** – кандидат биологических наук, старший научный сотрудник кафедры биохимии Казанского государственного университета.

E-mail: [nakberov@ksu.ru](mailto:nakberov@ksu.ru)