

# **Трёхмерные методики визуализации программного обеспечения параллельных и распределенных вычислений**

**В.Л. Авербух, А.Ю. Байдалин,  
Д.Р. Исмагилов, А.Ю. Казанцев**

**Институт математики и механики УрО РАН**

[averbukh@imm.uran.ru](mailto:averbukh@imm.uran.ru)

## Визуализация программного обеспечения

Под *визуализацией программного обеспечения* понимается совокупность методик использования графики и средств человеко-машинного взаимодействия, применяемых для лучшего уяснения понятий и эффективной эксплуатации программного обеспечения ЭВМ, а также для спецификации и представления программных объектов в процессе создания программ.

## **Визуализация программного обеспечения параллельных вычислений**

***Визуализация программного обеспечения параллельных вычислений*** включает в себя исследования и разработки визуальных языков параллельного программирования, визуальных отладчиков правильности и систем настройки, отладки, измерения и анализа производительности параллельных программ.

# Визуализация программного обеспечения параллельных вычислений

## *Визуализация*

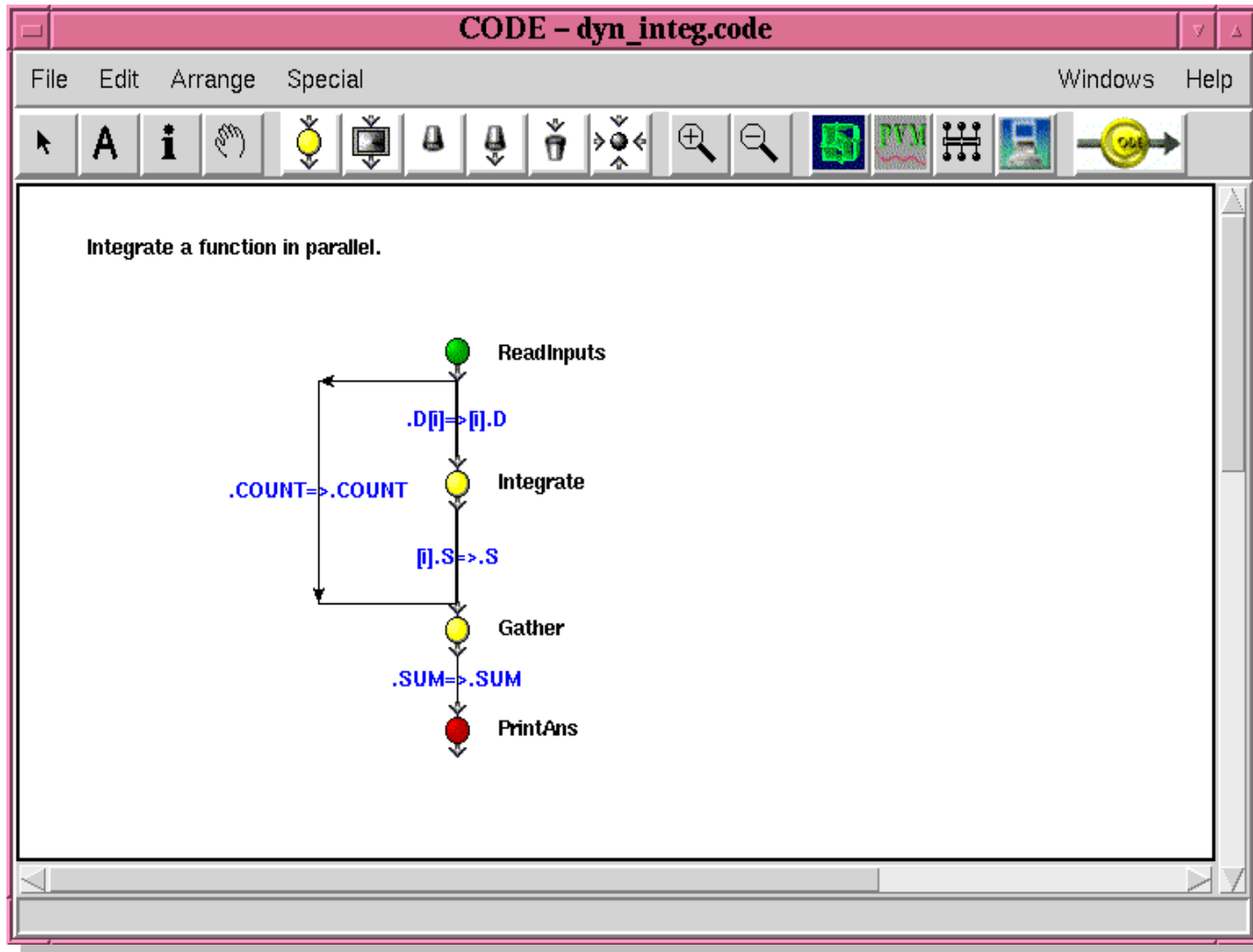
интеллектуальную программирования разнообразных считаются, что анимации визуализации программного обеспечения

помогает сложность за счёт методик. повышает при представлении

понижить параллельного использования Традиционно трёхмерности и эффективность сущностей

Традиционно считается, что использование трёхмерности и анимации повышают эффективность визуализации при представлении сущностей программного обеспечения.

# Визуальные языки параллельного программирования

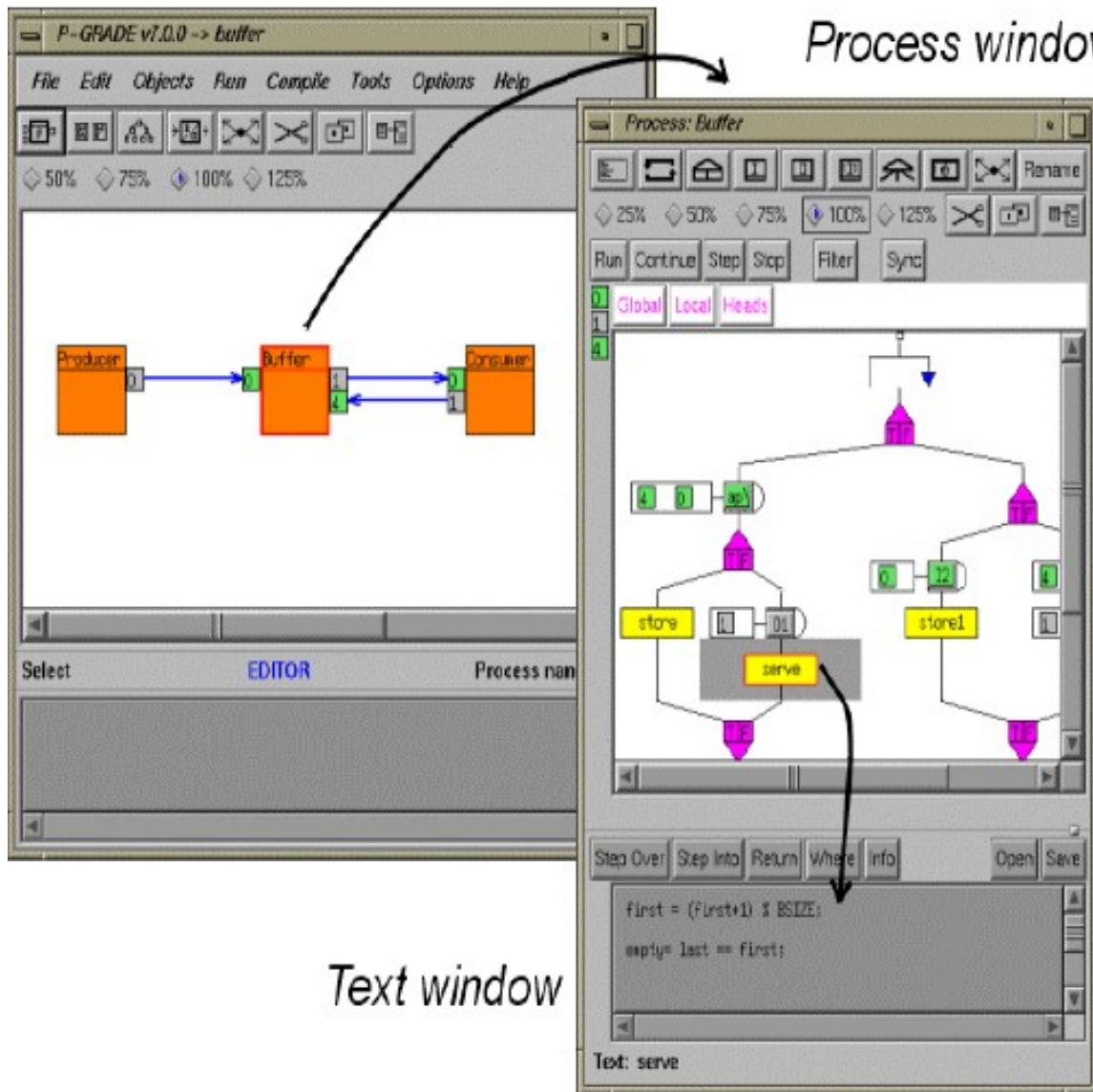


Параллельная программа на языке CODE задаётся направленным графом, в котором потоки данных на дугах связывают узлы, представляющие последовательные программы.

# Визуальные языки параллельного программирования

Application window

Process window

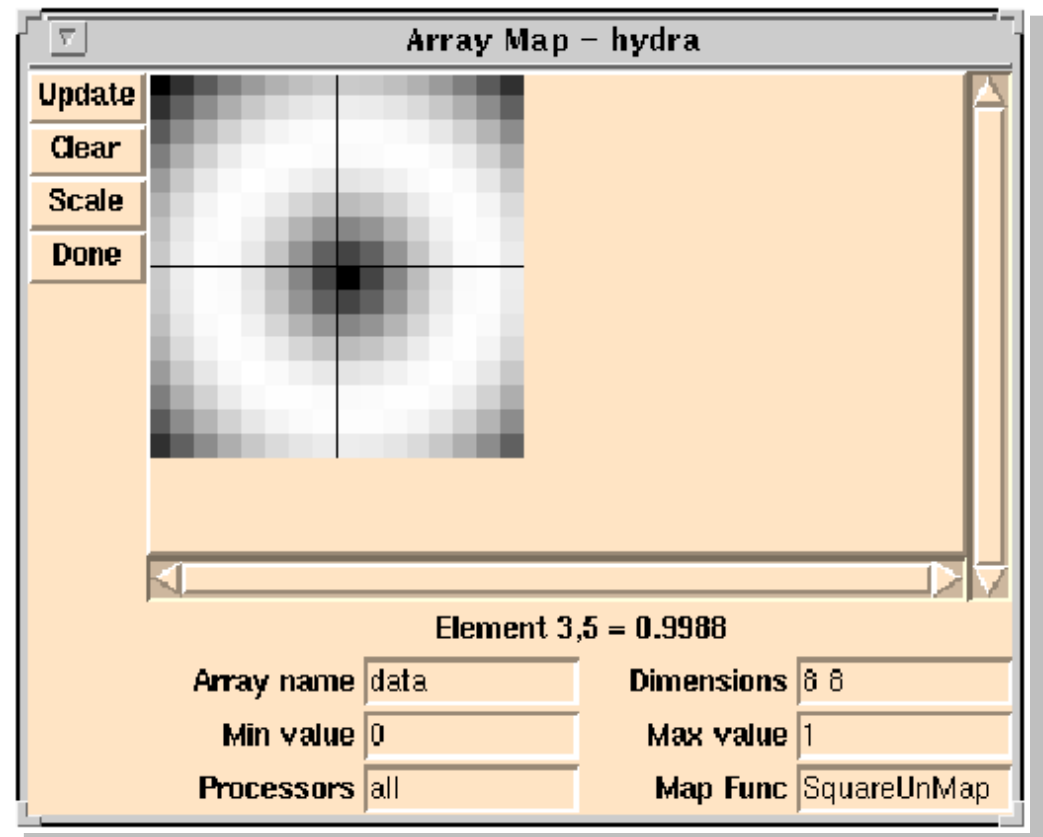
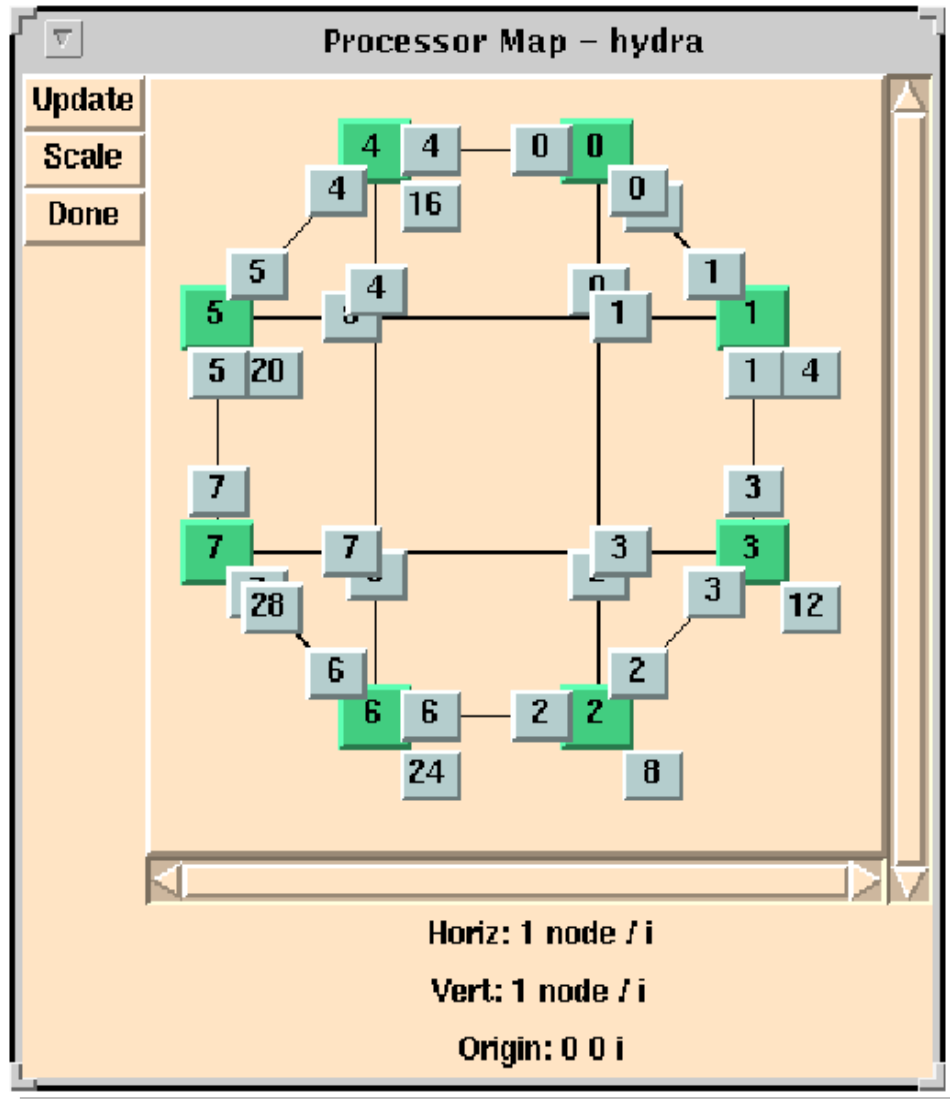


Text window

Визуальный язык программирования GRAPNEL (GRAPhical Process's NEt Language) предназначен для проектирования распределенных и параллельных программ.

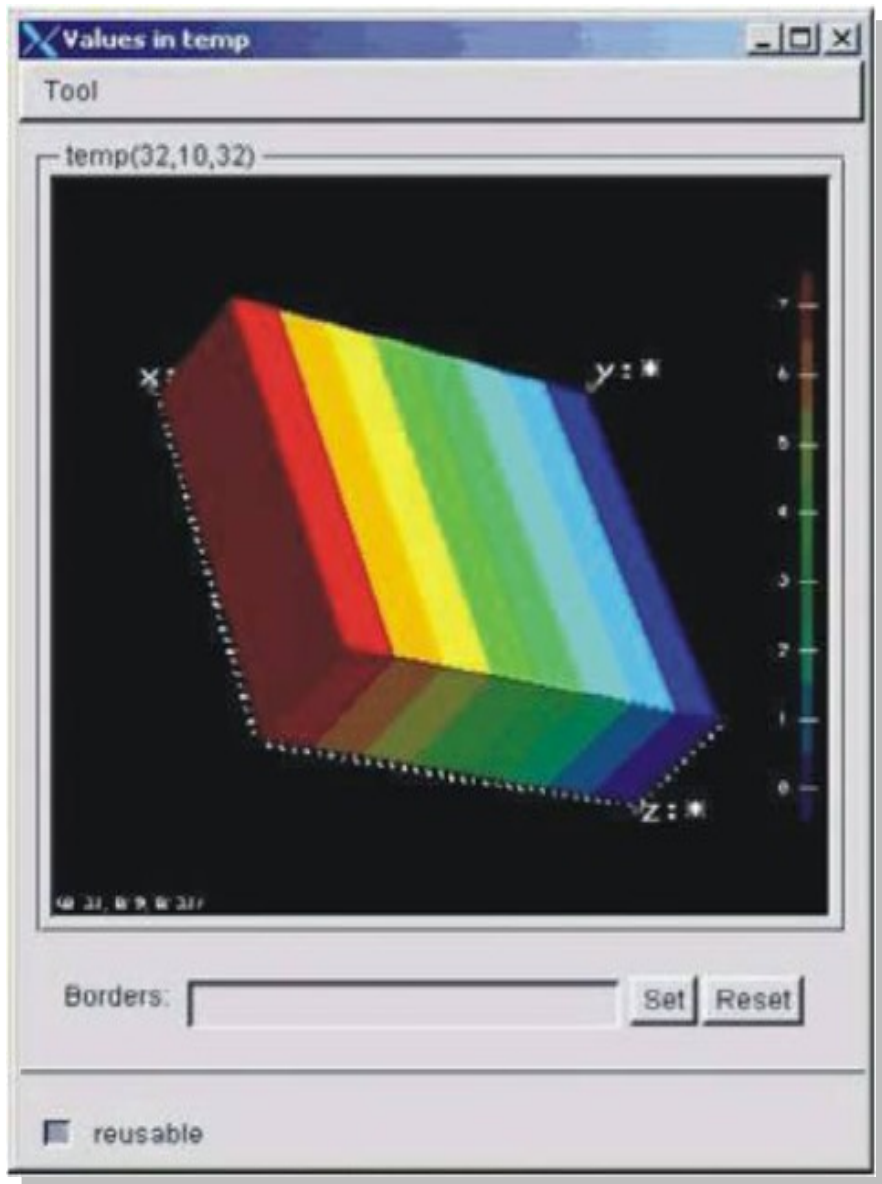
# Отладка правильности параллельных программ

## Отладчик Panorama



# Отладка правильности параллельных программ

## Отладчик MAD

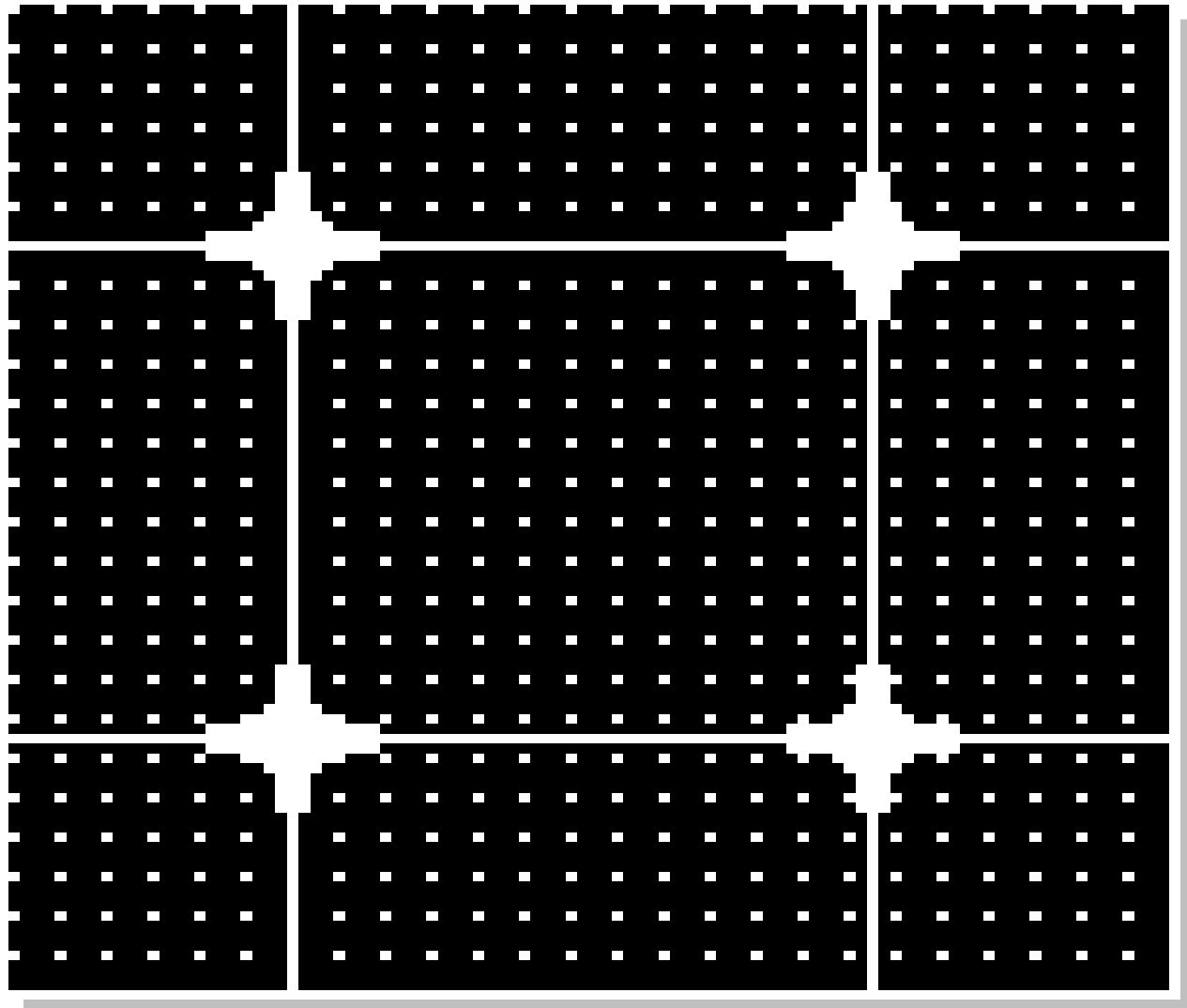


**Data distribution  
among processors**



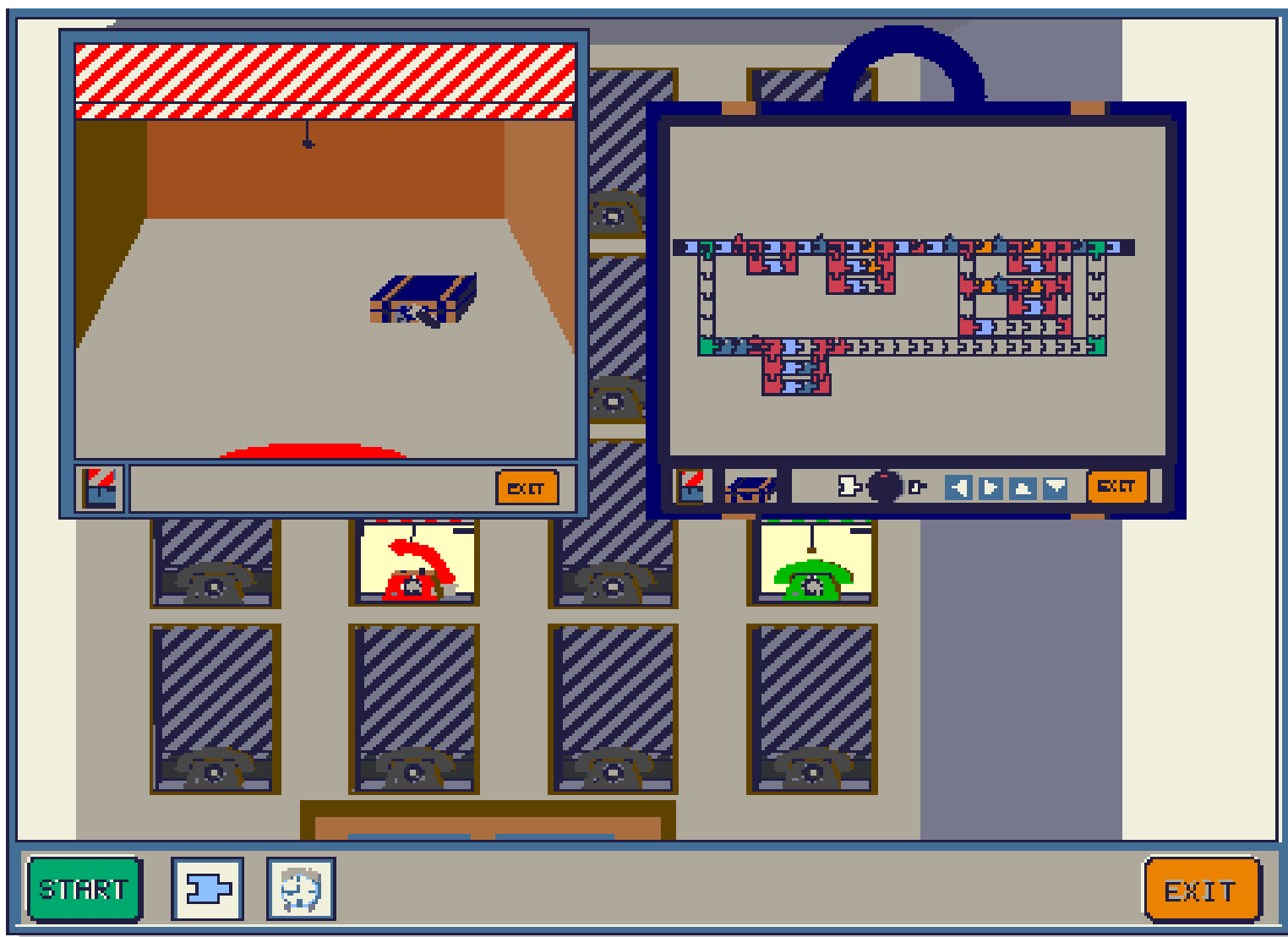
# Отладка правильности параллельных программ

Сравнительная отладка. Отладчик *Guarde*

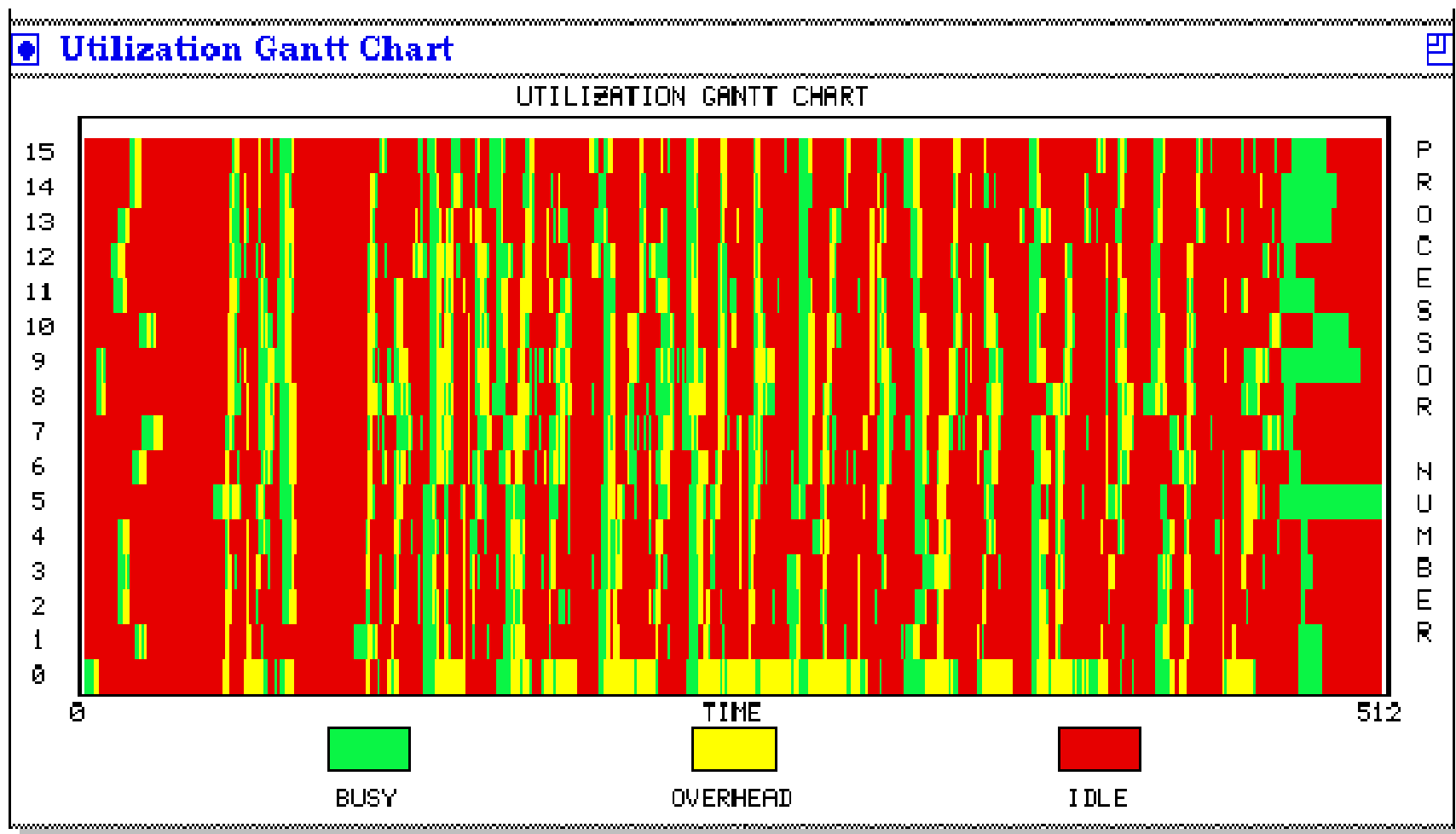


# Отладка правильности параллельных программ

## Система INHOUSE



# Отладка, настройка, измерение и анализ производительности параллельных программ



# Отладка, настройка, измерение и анализ производительности параллельных программ

## Общий вид визуализатора для DVM

The screenshot displays the DVM performance visualization interface. The main window shows a Fortran code snippet with a MAP statement. Several sub-windows provide performance data:

**Файл данных производительности -- D:\tmp\2\stat39.txt**

```

c.313 continue
c.213 continue
c print*, 'dan0_vk', vk(7,1,1)

nump=NUMBER_OF_PROCESSORS()
print *, 'number of processors =', nump
j=1
ndk1=nump/ld
ndk2=mod(nump,ld)
ndk=ndk1
if(ndk1.ne.0 .and. ndk2.eq.0) then
  ndk=ndk-1
endif
c print*, 'm4'
DO i=1,ld
CDVM$ MAP TASKDK(i) ONTO PROC(j+j+ndk)
print *, 'line number :', i, ' processors', j, ':', j+ndk
j=j+ndk+1
if(ndk1.eq.0) then
  if(j.gt.nump) then
    j=1
  endif
elseif(ndk2.ne.0 .and. i.eq.ndk2) then
  ndk=ndk-1
endif
ENDDO

times1 = dvtime()
CDVM$ INTERVAL 1
  
```

**Коммуникационные характеристики интервалов**

Processors->	33	34	35	36	37	38	39
Communics	0	0	0	0	0	0	0
I/O	0	0	0	0	0	0	0
Reduction	7,3606	11,3888	8,0152	8,7955	11,0763	11,1589	11,0761
Shadow	0	0	0	0	0	0	0
Remote	0	0	0	0	0	0	0
Redistrib	0	0	0	0	0	0	0
Real Synchro							
I/O	0	0	0	0	0	0	0
Reduction	0	0	0	0	0	0	0
Shadow	0	0	0	0	0	0	0
Remote	0	0	0	0	0	0	0
Redistrib	0	0	0	0	0	0	0

**Временные характеристики выполнения интервала**

Processors->	34	35	36	37	38	39
Useful Time	23,6893	26,9829	26,2783	23,8943	23,8696	23,2179
User CPU	23,6835	26,9759	26,2693	23,8863	23,8608	23,1964
System CPU	0,0058	0,007	0,0089	0,0079	0,0088	0,0215
I/O Call	0	0	0,0001	0,0001	0	0
Lost Time	30,242	26,9484	27,653	30,037	30,0616	30,7134
User Insuf.	18,7424	18,7174	18,7384	18,7484	18,7891	18,7423
Sys. Insuf.	0,1085	0,118	0,117	0,1161	0,1123	0,6672
Communic.	11,3888	8,0152	8,7955	11,0763	11,1589	11,0761
Real Synchr.	0	0	0	0	0	0
Idle CPU	0,0023	0,0978	0,0021	0,0962	0,0013	0,2277

**FormWallView**

Характеристики выполнения интервалов по процессорам

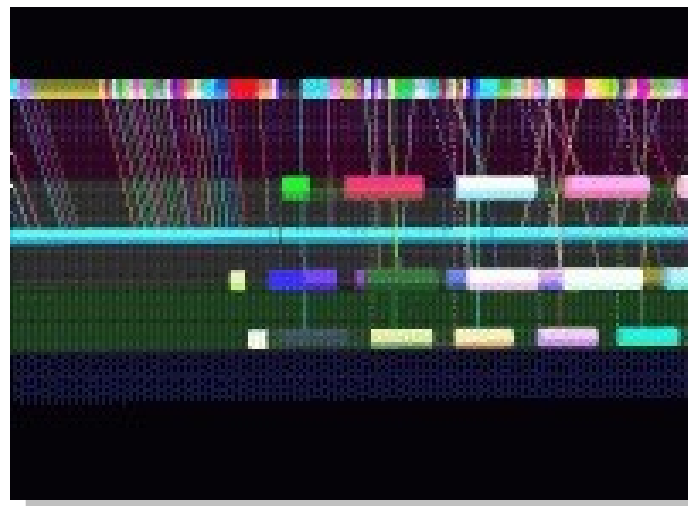
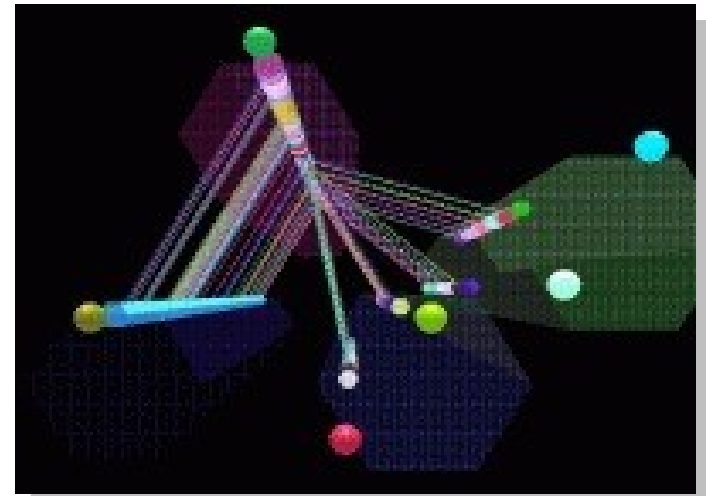
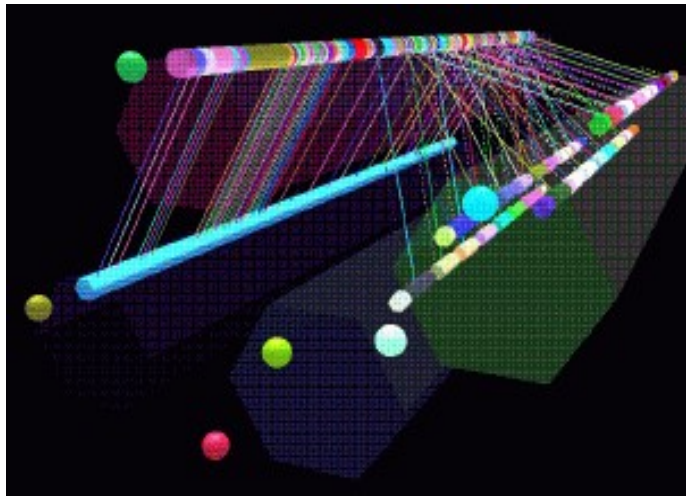
Высота: 53,9314

Легенда:

- Use CPU
- System
- IDCall
- Usr Insuf
- Sys Insuf
- Communic
- Idle Time

# Отладка, настройка, измерение и анализ производительности параллельных программ

## VisuaLinda



## *Метафоры визуализации*

Под метафорой будем понимать основную идею уподобления сущностей моделируемой области набору визуальных объектов, составляющих виды отображения.

## Метафора комнаты

### *Свойства:*

- ♦ *Способность содержать какие-либо объекты внутри себя.*
- ♦ *Ограничение контекста восприятия.*
- ♦ *Замкнутость.*
- ♦ *Включение в структуру.*
- ♦ *Естественность метафоры.*

# Метафора комнаты

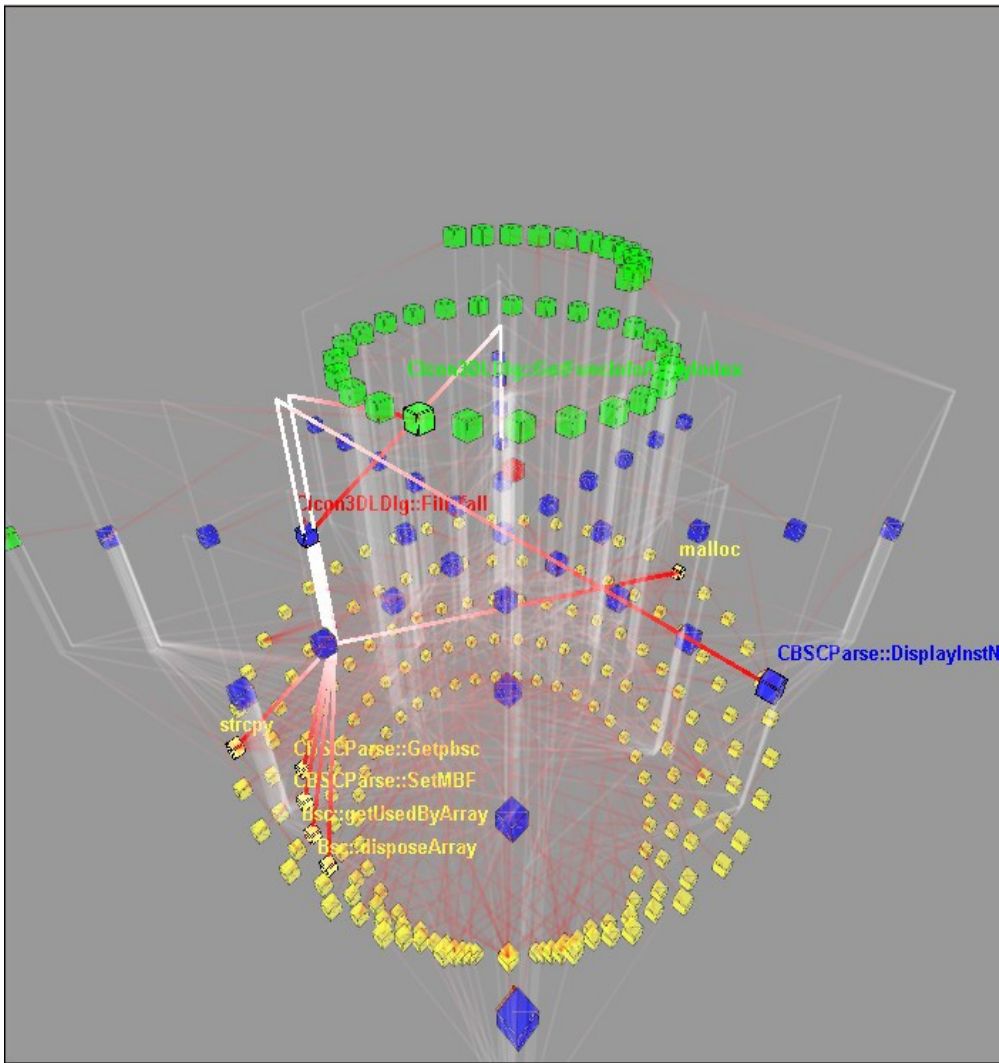


Diagram illustrating a room metaphor for a software function. The room is red, and the whiteboard displays the following code:

```
bool cristate=false;
if (GetKeyState(VK_CONTROL)<0)
  cristate=true;
if (cristate)
  switch (nChar) {
  case 'O': CIcon3DLDlg::OnFileOpen();cristate=false;break;
  case 'C': CIcon3DLDlg::OnFileClose();cristate=false;break;
  case 'F':
    if (flist_not_prepared) break;
    DialogBox(hinstance,MAKEINTRESOURCE(IDD_FINDDIALOG),m_hWnd,
      FindDialogProc);
    cristate=false;
    break;
  case VK_F12: cristate=false;PostQuitMessage(0);break;
  //Сработает обработчик DestroyWindow и вызовет OnExit()
  case VK_F11: cristate=false;break; //for DEBUG needs
  }
CDialog::OnKeyDown(nChar, nRepCnt, nFlags);
```

Information box (Функция пользователя):

- Имя: `CIcon3DLDlg::OnKeyDown`
- Родитель: `CIcon3DLApp::InitInstance`
- Общее время: Unknown
- Определение: Unknown

On the floor, there are two cyan boxes labeled `OnKeyDown` and `PostQuitMes`.



## Метафора ландшафта, метафора географического или городского пространства

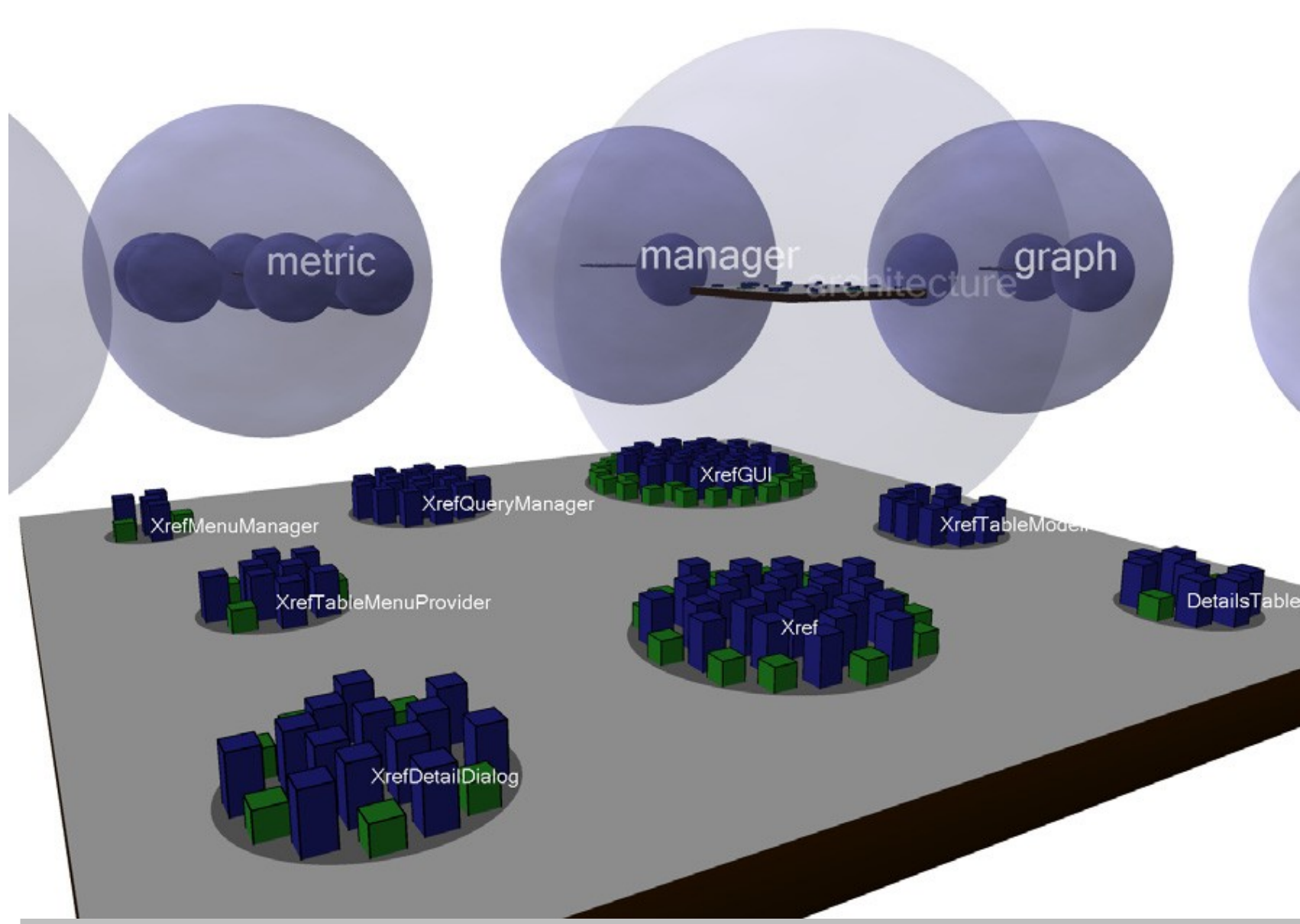
### *Свойства:*

- ♦ *Неограниченный контекст.*
- ♦ *Естественность метафоры.*
- ♦ *«Вложенность» ландшафтов, организация внутренней структуры.*

# Метафора ландшафта, метафора географического или городского пространства



# Метафора ландшафта, метафора географического или городского пространства



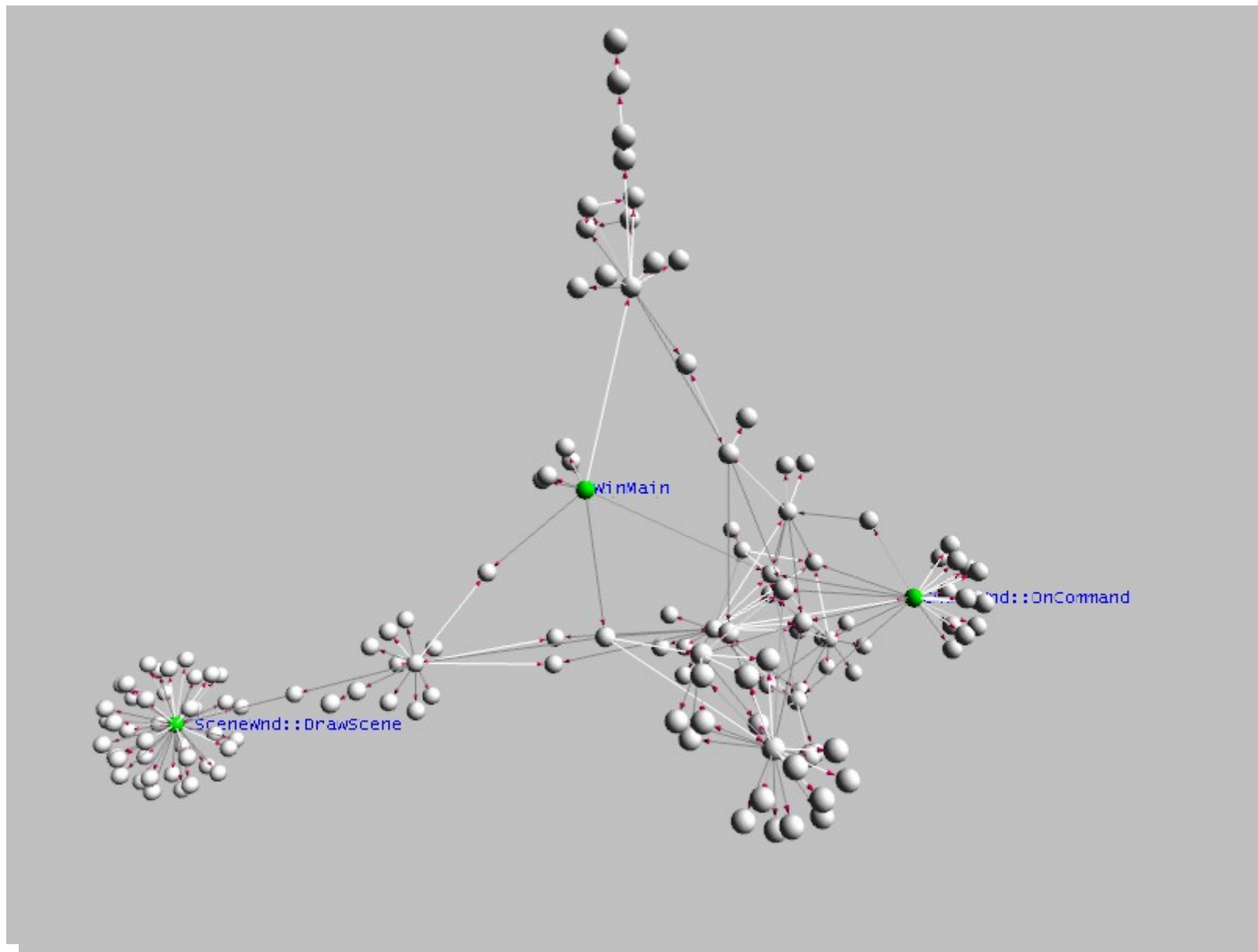
## Метафоры молекулы

<http://sourceforge.net/projects/callgraph/>

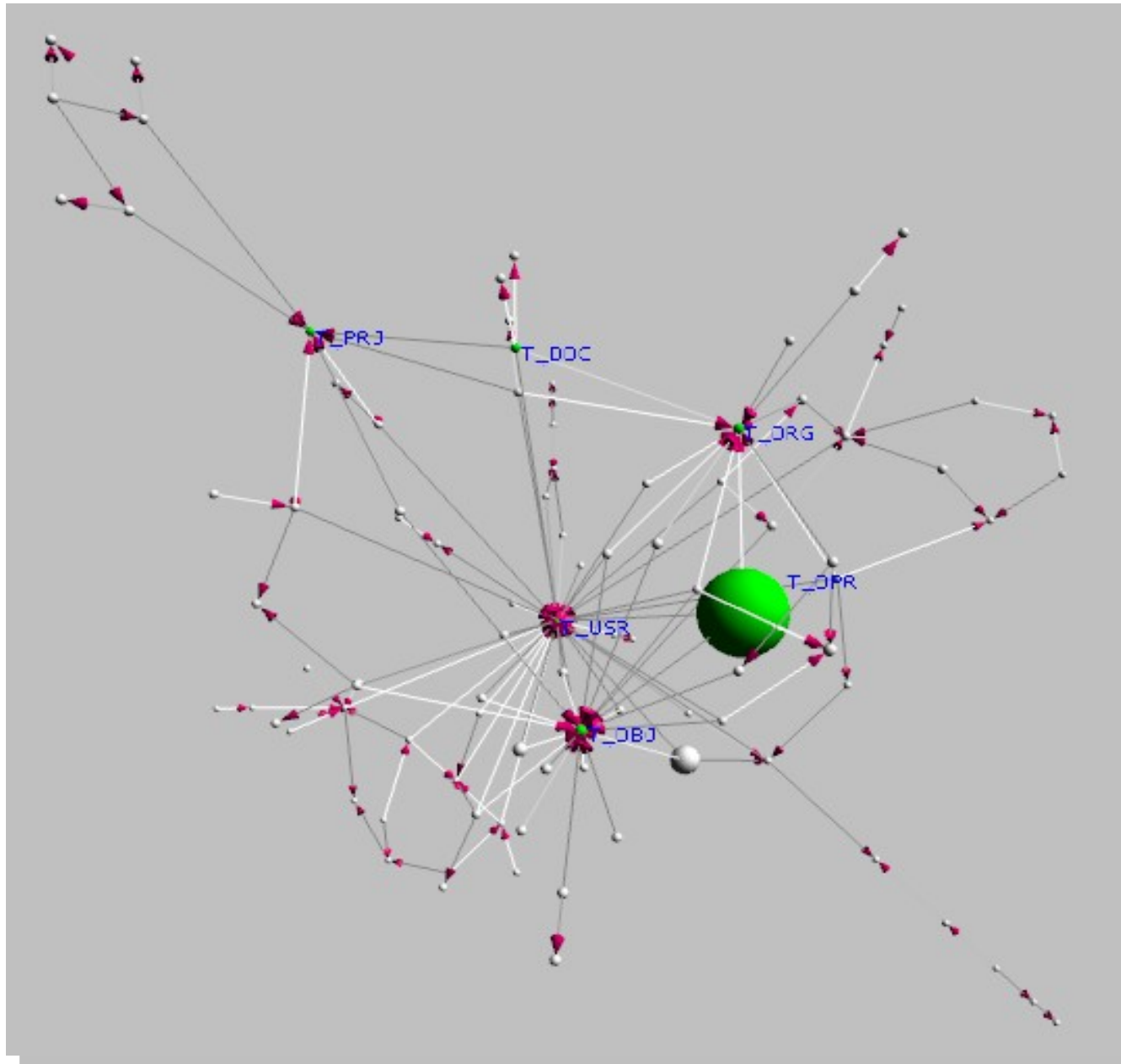
В этой метафоре структура молекулы отражает структуру исходного графа вызовов. Между атомами (отображающими функции – вершины графа) введено два типа взаимодействия – упругое между связанными вершинами и электростатическое между всеми вершинами-атомами. Электростатическое взаимодействие отражает временные характеристики вызовов функций, тогда как упругое – количество вызовов. Время работы функции можно показывать как размер вершины. Возможно использование цвета для выделения/подсветки интересующих особенностей визуализируемого графа. Анимация (вращение молекулы) позволяет изучить структуру графа. Алгоритм визуализации графа позволяет отобразить графы с сотнями вершин.

# Метафоры молекулы.

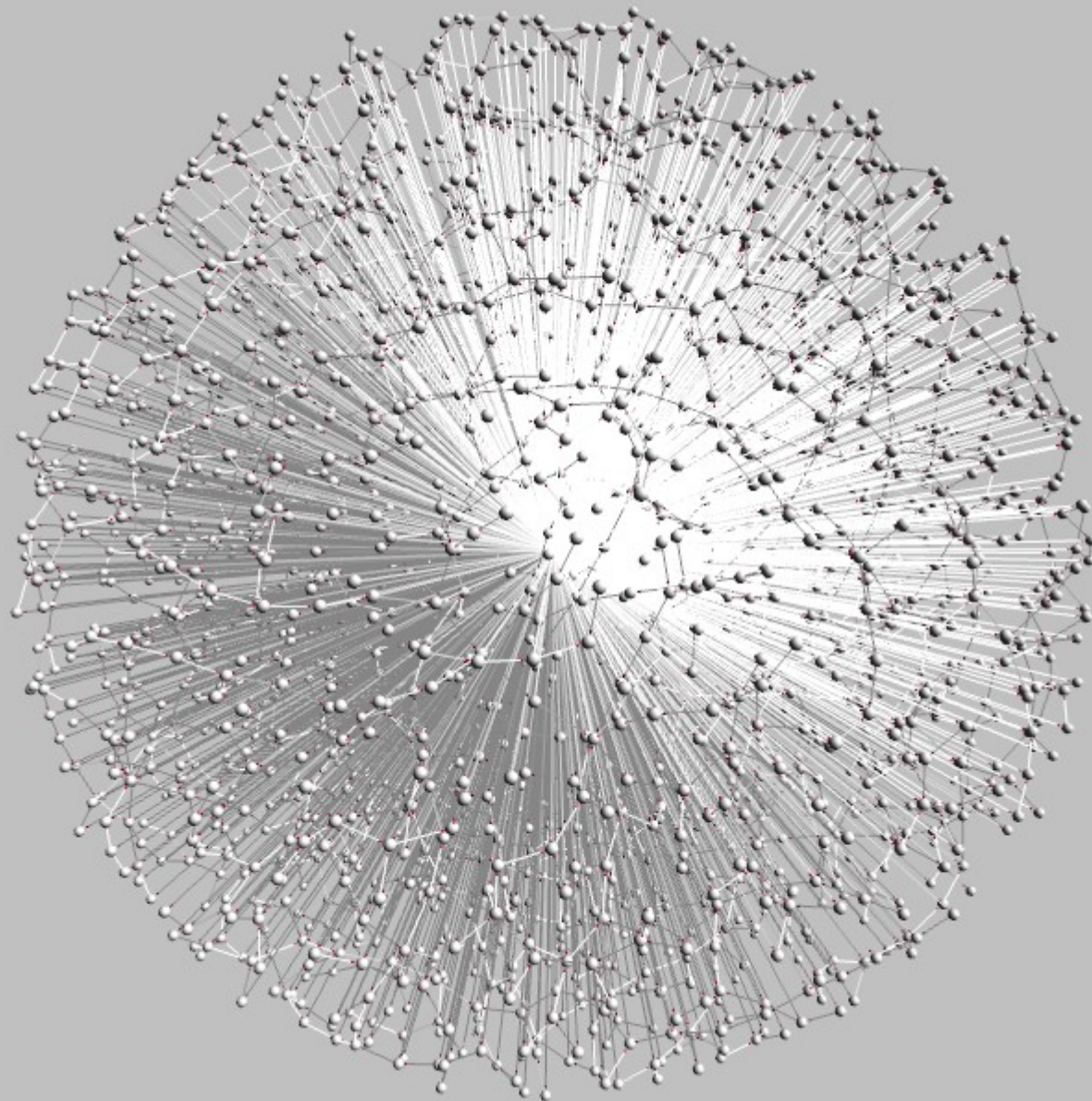
## Граф вызовов программы рисования графа вызовов



# Метафоры молекулы. Структура прикладной базы данных



**Метафоры молекулы.  
3000 вершин, соединенных с центральной**



# *Метафоры молекулы. Структура, похожая на молекулу*





# Метафоры молекулы. Дерево



- ♦ *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.
- ♦ *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.
- ♦ *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. С. 12-13.
- ♦ *Averbukh V., Bakhterev M., Baydalin A., Ismagilov D., Trushenkova P.* Interface and Visualization Metaphors // Lecture Notes in Computer Science. V. 4551, Springer Berlin / Heidelberg, 2007. Pp. 13-22