

Two-dimensional C++

Oben mitte: <fractool-Logo>

Oben rechts: Hochschule Darmstadt – University of Applied Sciences, ZFE – Zentrum für Forschung und Entwicklung

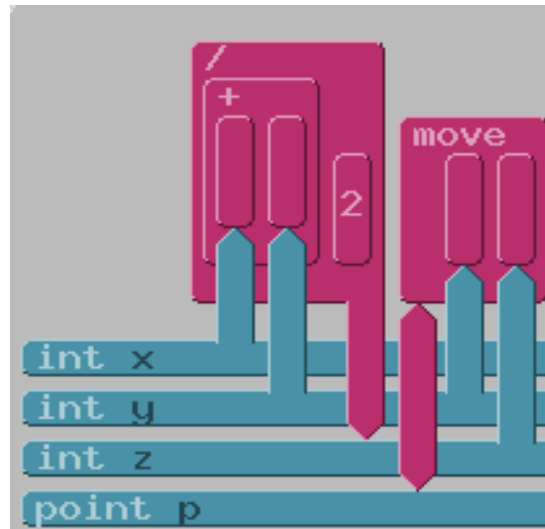
Unten links: Contact: Johannes Reichardt, Hochschule Darmstadt, Computer Science Department, Haardtring 100, D-64295 Darmstadt, j.reichardt@fbi.h-da.de,
www.fractool.com

General Features

Classification of ROMAN, G.-C. and COX, K.C. 1993.

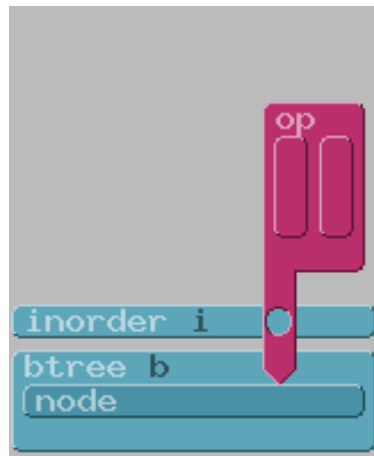
- Visual programming and modeling language, general-purpose and imperative.
- The *scope* of the language is the static description of code, data, flow control and of the surrounding module structure.
- Its *abstraction level* is close to that of C++ including templates and the STL.
- The *visualization* is specified by editing the boxes, bars, arrows and textual labels on a drawing area with relatively fixed editing positions.
- The *interface* of the visualization is the drawing area itself, enabling further editing or navigating through the program by clicking on the bars, boxes and labels or by defining views via textual or topological queries.
- The *presentation* of the visualization is by a (still hypothetical) wall-projection of the virtual drawing area with selected areas drawn on working stations for more detailed work. High-resolution display of large coherent graphs is an as yet unresolved problem.

Arrows expressing element, argument and assignment relations with generated code



```
int      x,y,z;  
point   p;  
  
z=(x+y)/2;  
p.move(y,z);
```

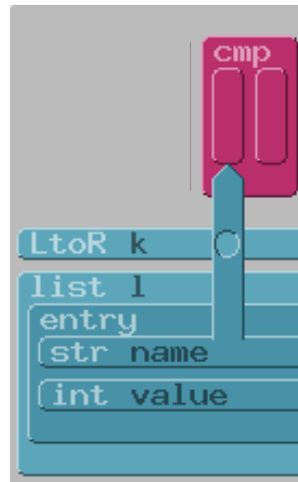
Container-access via iterator, with generated code



```
inorder i;  
btree b;
```

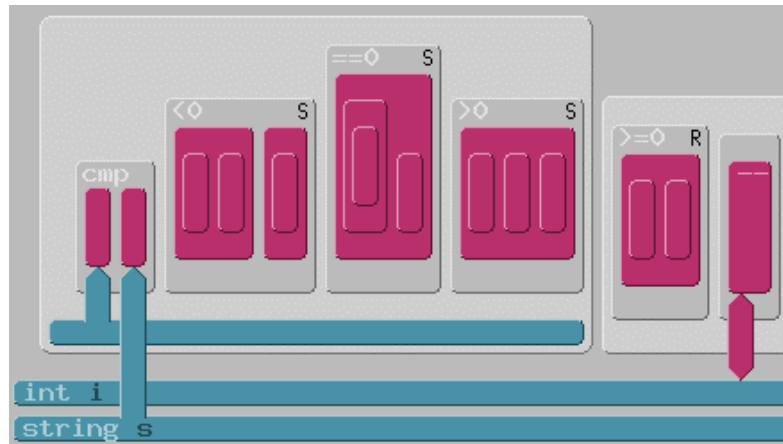
```
b[i].op(..., ...);
```

Direct access to Struct-elements, with generated code



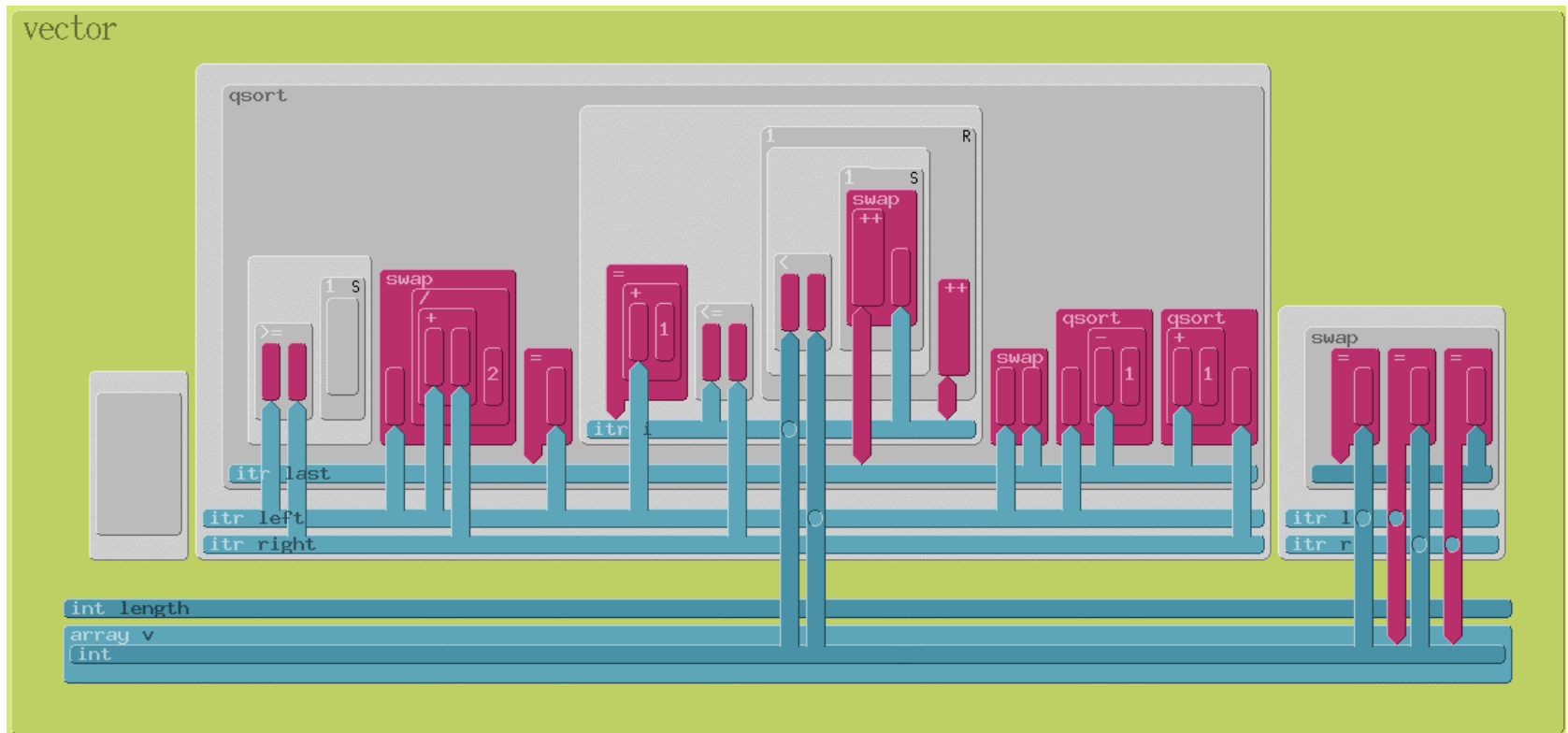
```
LtoR k;  
typedef struct {  
    str name;  
    int value;  
} entry;  
list<entry> l;  
  
cmp(l[k].name, ...);
```

Flow Control, with generated code

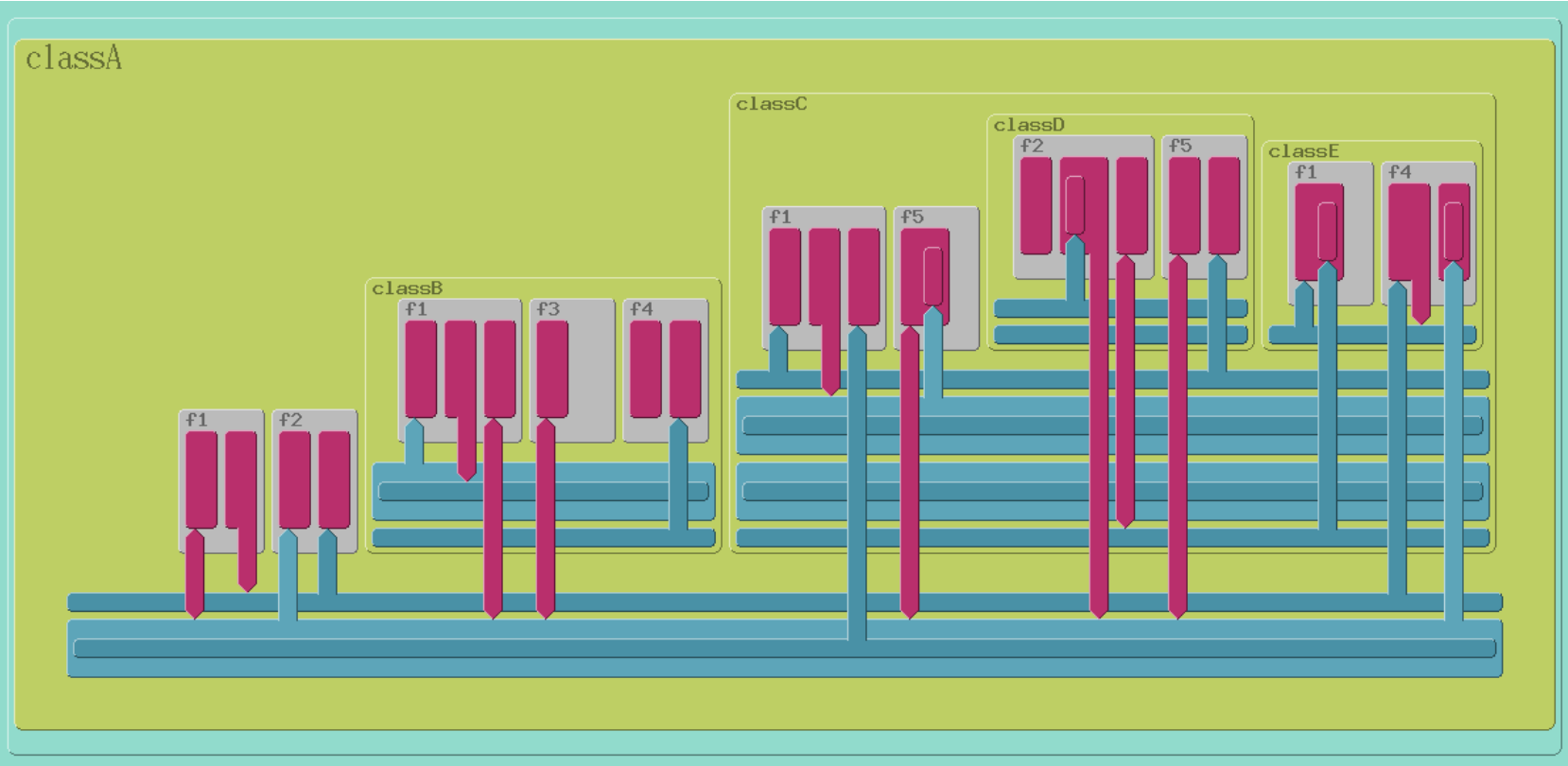


```
if (cmp(...,s)<0){...} else
if (cmp(...,s)==0){...} else
if (cmp(...,s)<0){...};
do {...} while ((i--)>=0);
```

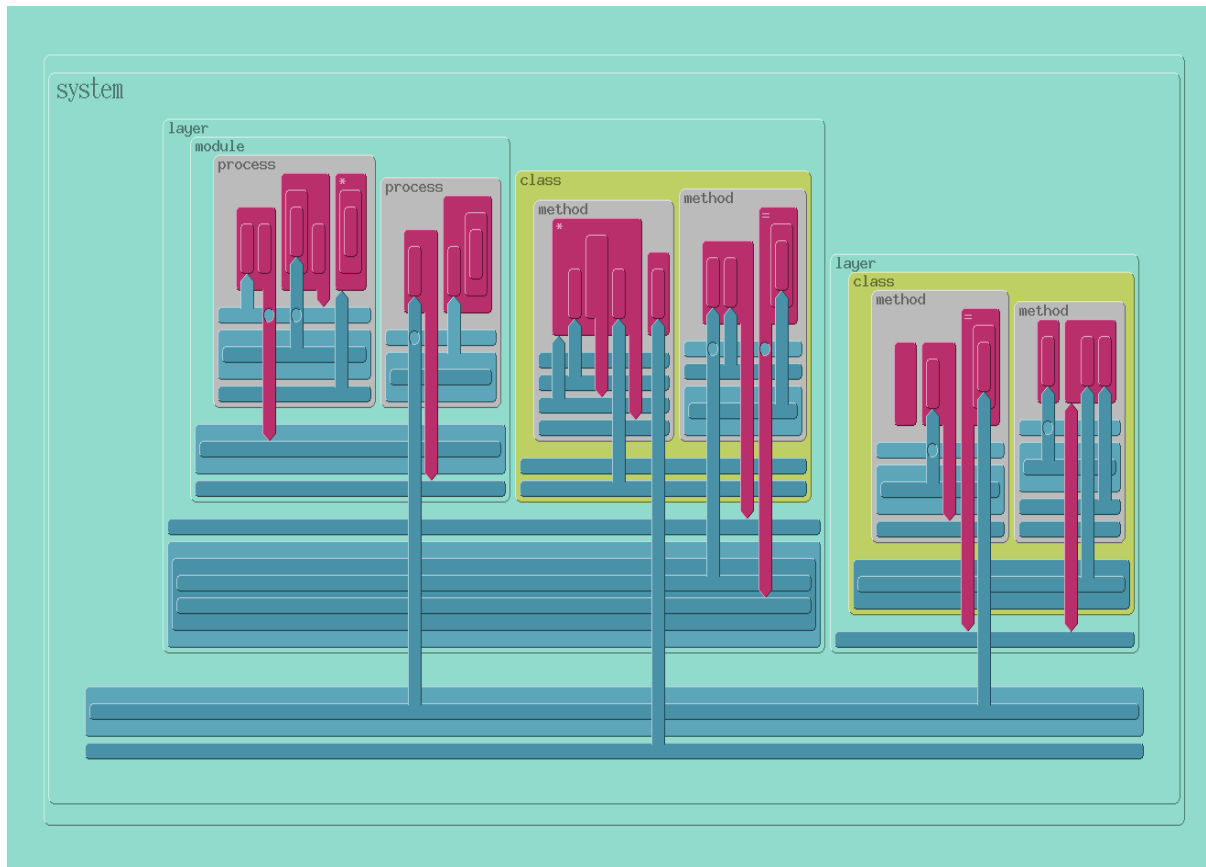
Class *vector* with methods *qsort* and *swap*



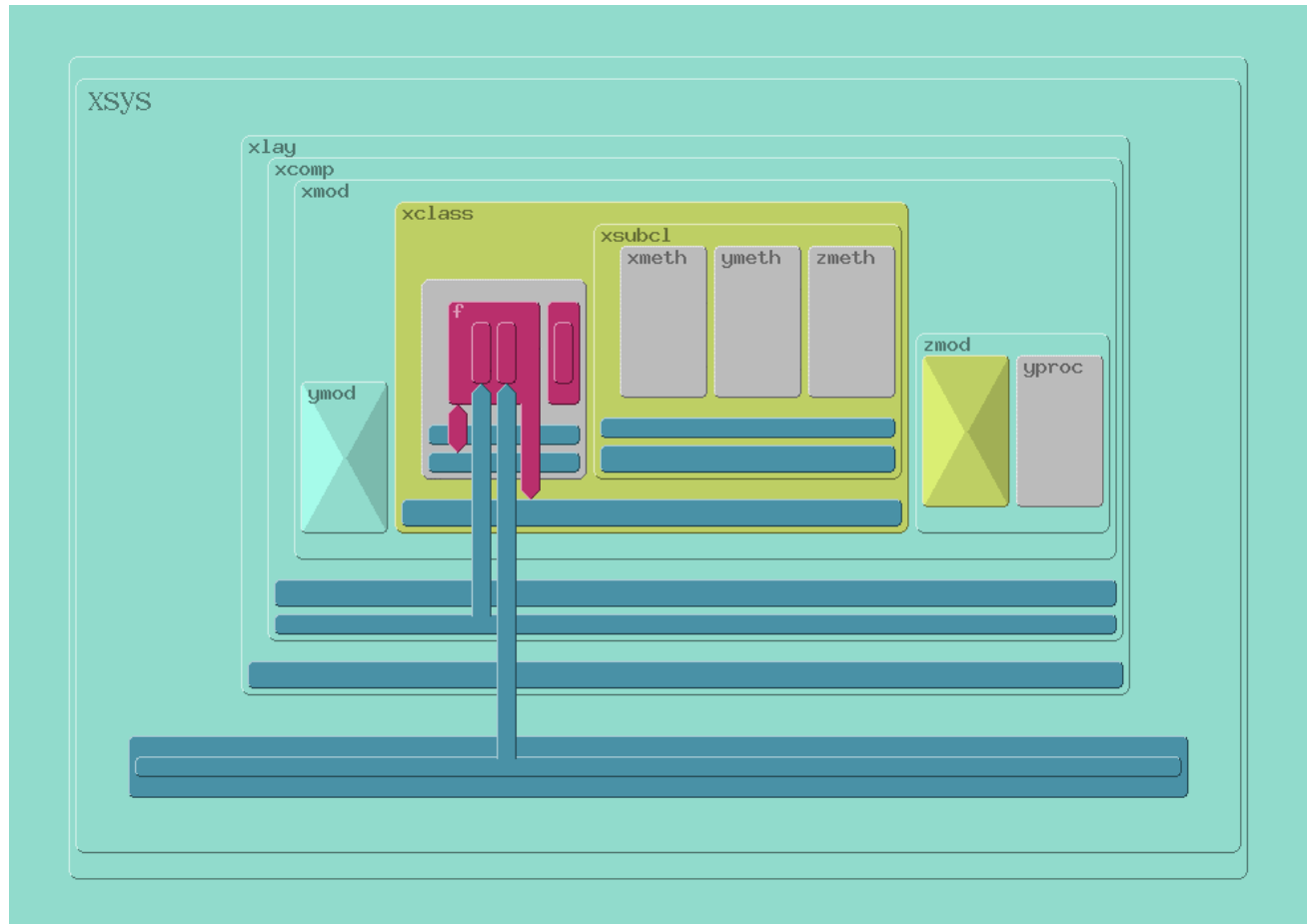
Class Hierarchy



System Structure



Fractal view of a module



Design Goals

- Support of program comprehension by resemblance.
- Resemblance established by structure-preserving (homomorphic) mappings:
 - o Algorithmic level: Process on objects mapped onto states and state transitions along a time axis.
 - o Architectural level: Concepts of the application domain and their interrelationships mapped onto class and module definitions and their geometrical neighbourhood (horizontal, vertical, nested) in the two-dimensional design space.