XXL

A Visual + Textual Environment for Building Graphical User Interfaces

Eric Lecolinet

Ecole Nationale Supérieure des Télécommunications (ENST)
Dept. INFRES, 46 rue Barrault, 75013, Paris, France.

elc@enst.fr - www.enst.fr/~elc

Context

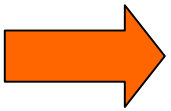
- **Model-based interface development systems**
  - Still laboratory tools
  - Despite their potential
- **"Classical" interactive GUI builders**
  - Quite widespread
  - In spite of their limitations
- **A few possible reasons**
  - Large set of interface primitives ("*Widgets*", "*Controls*")
  - Easiness and intuitiveness ("*Visual Programming*")
  - ➨ Highly customized GUIs

XXL Approach

- **Textual + Visual development tool**
  - Not a high-level M-B approach
  - *"Missing link"* between interactive GUI Bs and high-level M-B tools
- **Textual + Visual equivalence**
  - Core idea: unify ***textual*** and ***visual*** programming:
    - Specification language
    - Interactive GUI builder
    - Free combination of both modes

Textual+VisualEquivalence

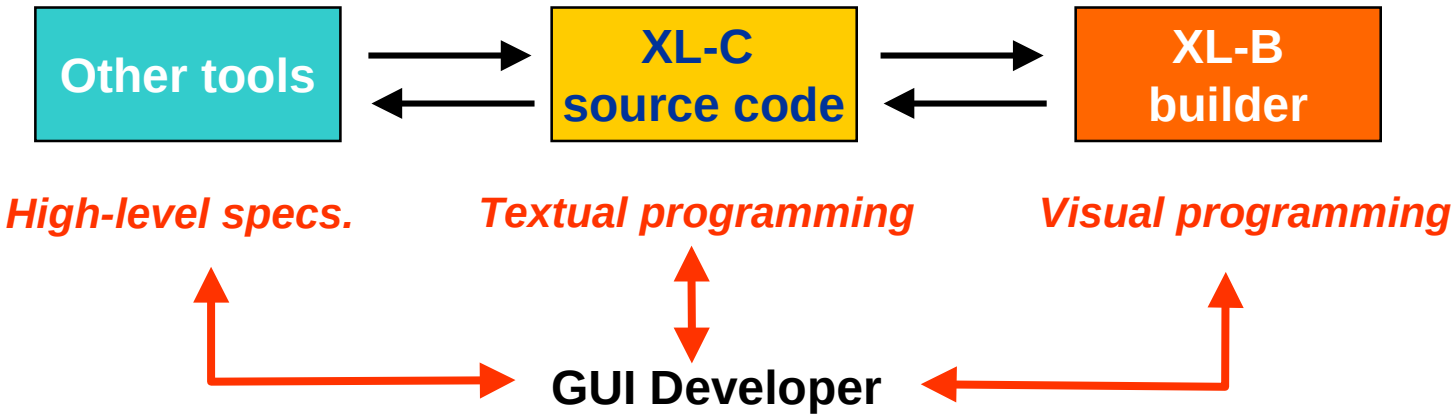
- ***XL-C*specificationlanguage**
  - Mostlydeclarative
  - Caneitherbe ***interpreted*** or ***compiled*** :
    - *standard*C/C++sourcecode(Csubset)
- ***XL-B*interactiveGUIbuilder**
  - Canreeditandmodify *preexisting*XL-Csourcecode
    - Notlimitedtothesourcefilesititselfproduced
    - Onlygeneratesstandard(XL-)Ccode



Editable C source code can be produced or modified by programmers or by other tools

Consequences (1)

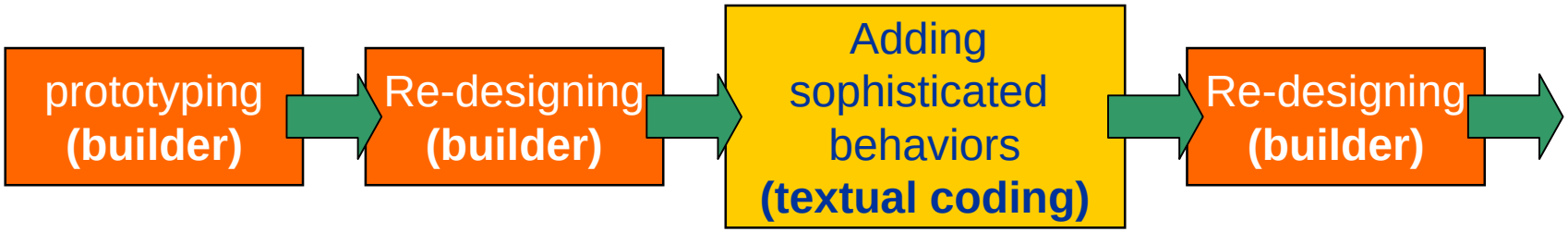
- **Open system**
  - The XXL builder can cooperate with higher-level tools



  - Example:
    - fine tuning of XL-C specifications produced by another tool

Consequences (2)

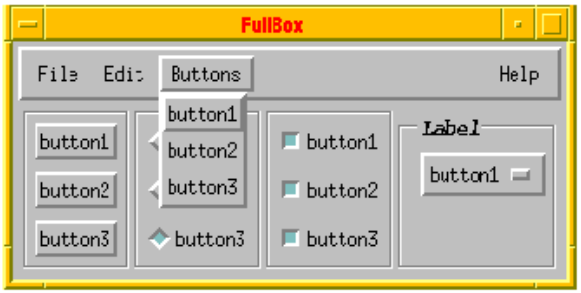
- **Truly iterative** development scheme
  - The builder can be used at any stage:
    - From the prototype to the final product...



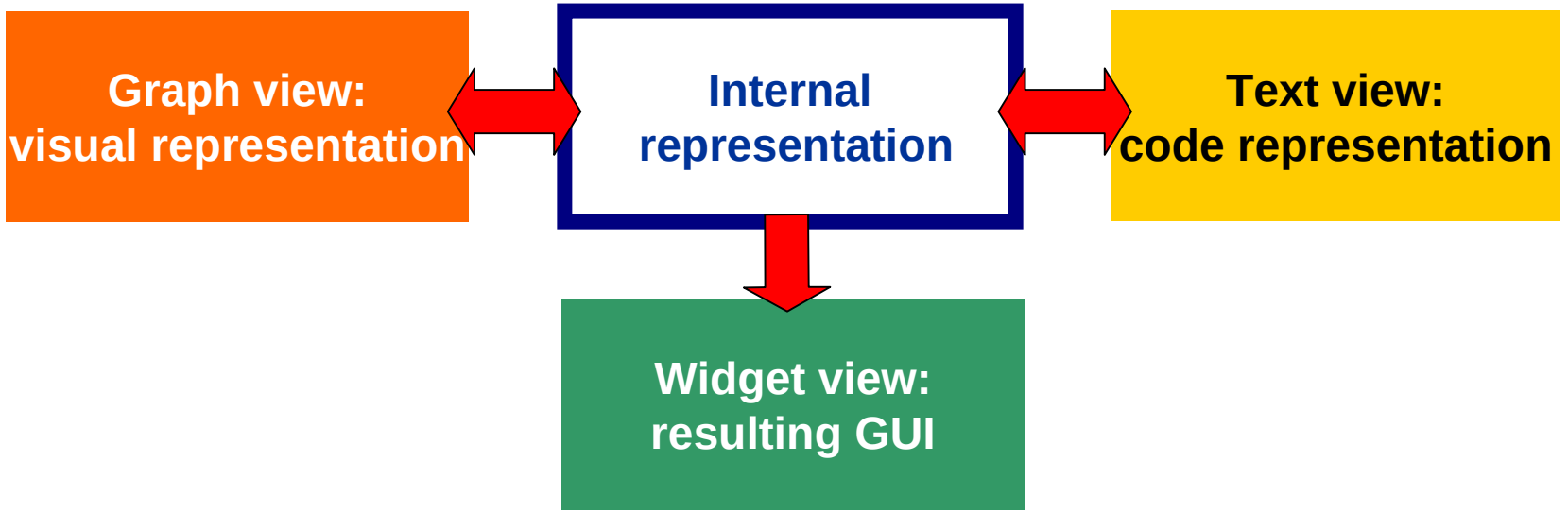
- **No strong separation** between presentation and GUI control:
  - GUIs that evolve dynamically at run-time...
  - Highly customized components

The XXL Model

- **Generic OO model**
- **4 meta-classes:**
  - **Graphical** objects ➡ encapsulate **Motif** widgets
  - **Control** objects ➡ repetitions, conditions, callbacks
  - **Structuring** objects ➡ interfaces and sub-interfaces
  - **Property** objects ➡ appearance and native behaviors
- **Generic G-Objects**
  - Actual widgets + implicit behaviors derived from **context**
    - Higher level of abstraction
    - Recursive changes



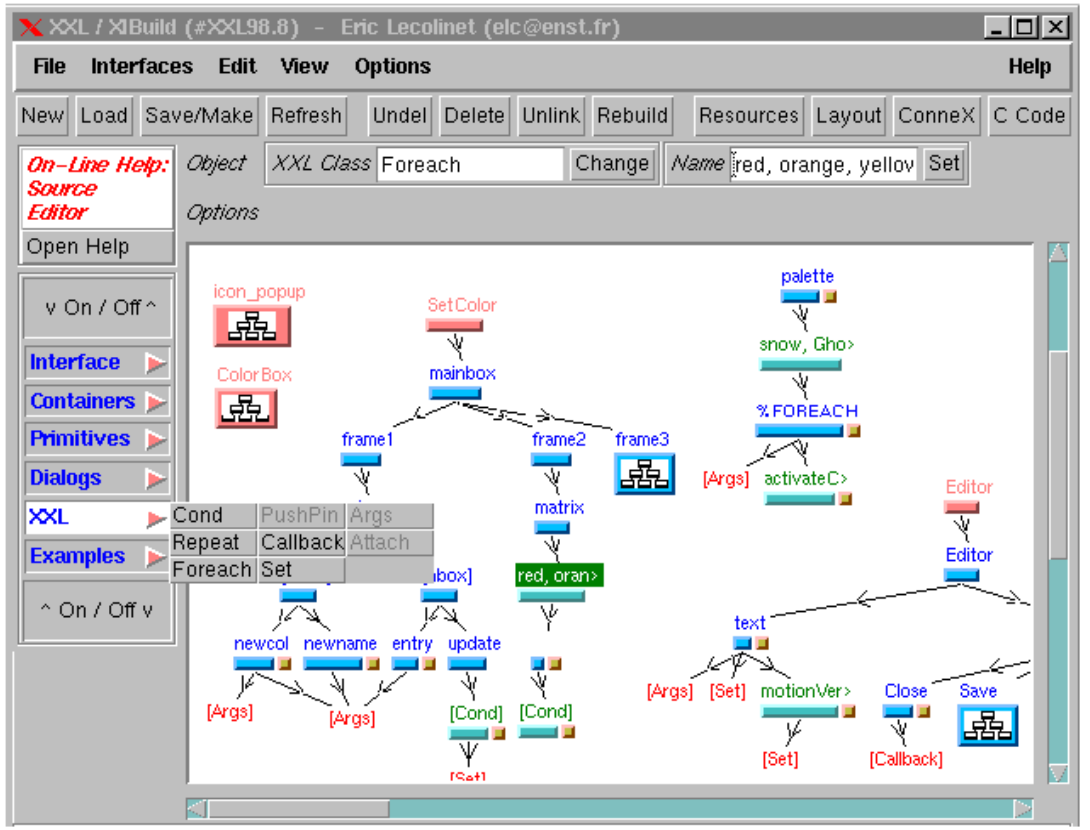
Three View Edition



- All views are **linked together** and are **incrementally updated**
- All objects have a **visual representation** :
  - In the graph view
  - Even non graphical objects

GraphView

- **Hidden+visibleparts**
  - *Concrete* and *abstract* objects represented and manipulated *in the same way*
  - Specification of the GUI in a *direct manipulation style*
- **Iterativedesign**
  - Widget view updated *“on the spot”*
  - The GUI can be deeply changed *at any stage*



TextView

- **Textual+VisualEquivalence**
  - C Code changed *“on the spot”* when Graph View is modified (and vice versa)
  - C Code can be changed *interactively*
- **Fullintegrationbetween Visual andTextualprogramming**
- **ProgramLiveliness**
  - The GUI can be modified *while the program is running*

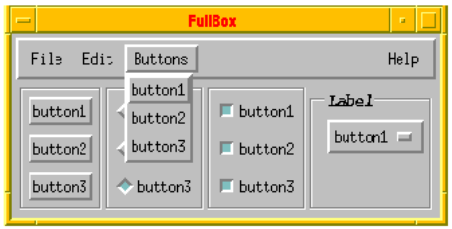
XL-C Specification Language

- **Standard ANSI C**
  - Can be **compiled** or **interpreted**
  - No strong separation between GUI code and other C functions
- **Intuitiveness of GUI builders + expressiveness of progr. languages**
- **Reverse interpretation**
  - Dynamical correspondence between run-time GUI objets and source code
- **The builder can deal with preexisting C source code**

```
XlObj open_dialog =
  (FileDialog, "open_dialog",
    (Args,"dialogTitle","Open Image",o),
    (Callback,"",OpenProc,NULL,o),
   o);

XlObj file_menu =     //each button opens a menu
  (Menu "file_menu",
    (Button, "Open", open_dialog,o),
    (Button, "Save", save_dialog,o),
   ....etc .....
   o);

XlObj menubar =
  (MenuBar, "menubar",
    (Button, "File", file_menu,o),
    (Button, "View", view_menu,o),
   ....etc .....
   o);

(HBox,"",
  (Label,"newcol",o), (Label,"newname",o),
o),
(HBox,"",
  (TextField,"entry",o),
  (Button,"update",
    (Cond,"",
      (Set,
       "~*newname.labelString","{~*entry.value}
       "~*newcol.background","{~*entry.value}",
     o,)o),
```

Combinations and "second thoughts"

- **Frequent reproach against GUI builders**
  - They require taking decisions that fix the presentation too early
- **XXL**
  - Adaptive layout:
    - Constraints rather than absolute coordinates
  - Graph view: powerful way:
    - To change UI structure
    - To re-combine various interfaces
  - Generic G-Objects:
    - Basic presentation and behavior derived from **context**
  - Recursive class changes
    - Actual widget classes modified contextually and recursively

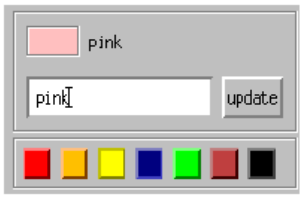


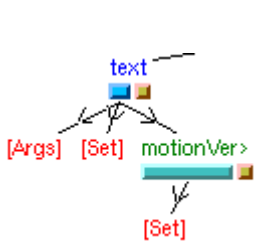

```
(VBox, "menu",
    (Button, "button1", o),
    (Button, "button2", o),
    (Button, "button3", o),
o)

(PulldownMenu, "menu",
    (Button, "button1", o),
    (Button, "button2", o),
    (Button, "button3", o),
o)
```

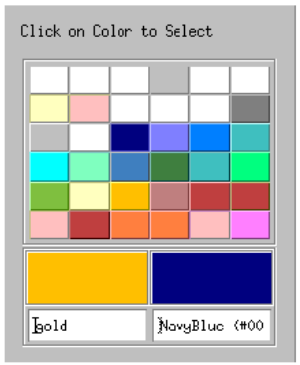
Control Objects

- **Conditional evaluation**
  - **Cond** object
  - For specifying basic behaviors in a **declarative way**
  - Can be triggered by events or **active values**
- **Repetitions**
  - **Foreach** object

```
(HBox,"",
  (Label,"newcol",o), (Label,"newname",o),
o),
(HBox,"",
  (TextField,"entry",o),
  (Button,"update",
    (Cond,"",
      (Set,
        "~*newname.labelString","{~*entry.value}
        "~*newcol.background","{~*entry.value}",
      o,)o),
```





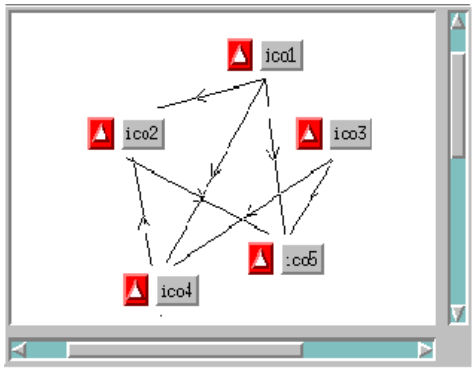



```
(Foreach, "red, orange, yellow, navy, gree
  (Button, "  ",
    (Args, XmNbackground,"%FOREACH",o),
    (Cond, "",
      (Set,
        "~*entry.value", "%FOREACH",
        "~*newname.labelString", "%FOREACH
        "~*newcol.background",   "%FOREACH
      o),
    o),
  o),
o),
```

OtherFeatures

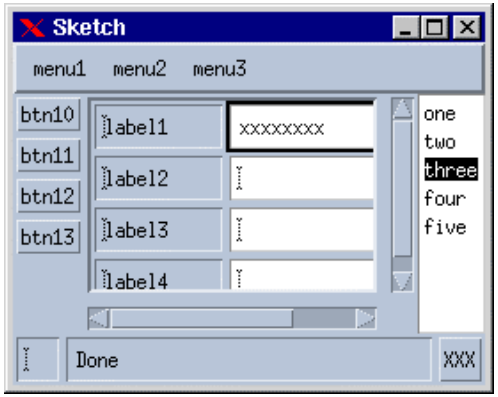
- **Sub-interfaces**
  - Multipleinstantiations
  - Parameterization
  - Interfacescanmakereference toother(sub-)interfaces
- **Directmanipulation**
  - **MoveHandle, DLink**, etc.
- **Migratoryinterfaces**
  - Dynamidoadingthroughthe network

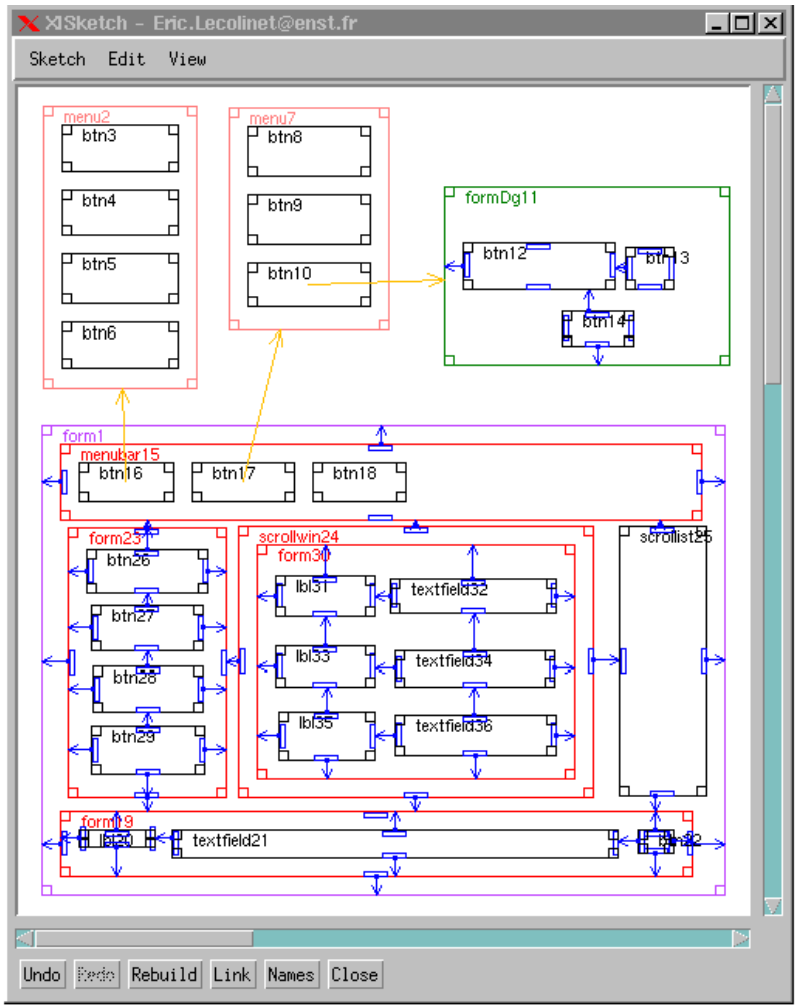
```
XlObj icon =
  (Interface, "icon",
    (HBox, "%INSTANCE",o),
      (ArrowButton, "b", o),
        (Button, "%INSTANCE",
          (MoveHandle, o),
    o),o),o),

XlObj canvas =
  (Canvas, "canvas",
    (Instance, "ico1", icon, o),
    (Instance, "ico2", icon, o),
    (Instance, "ico3", icon, o),
    (DLink, "ico1", "ico2", o),
    (DLink, "ico1", "ico4", o),
    .......
```



Sketch or Drawing

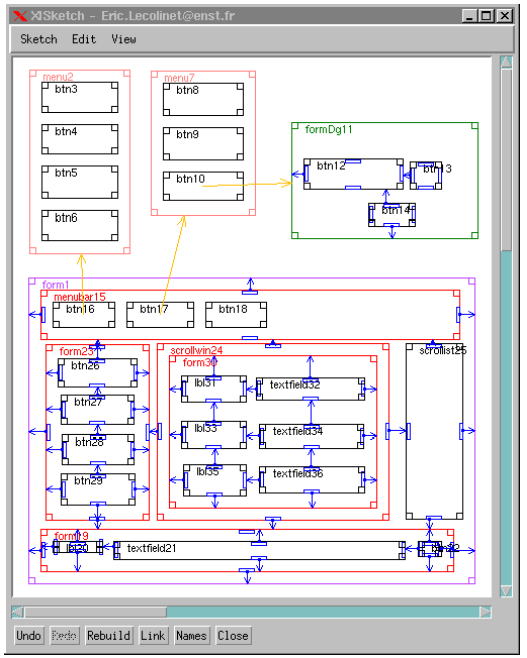
- **Early stages of design**
  - Rule: let the user focus on the global design!
- **Sketch drawing**
  - **Constrained** drawing with the mouse

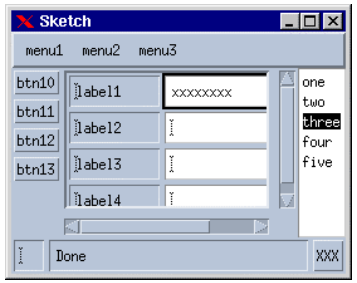




Sketching

- **Object classes and Layout constraints**
  - Are automatically deduced by the system
  - Can be changed interactively
- **Adaptive layout**
  - Constraints rather than absolute coordinates
  - Not WYSIWYG:
    - Drawing = *logical* representation
- **Iterative design**
  - Widget, Graph and Text views incrementally created
  - Can then be modified / refined at will



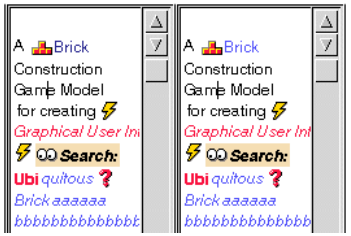


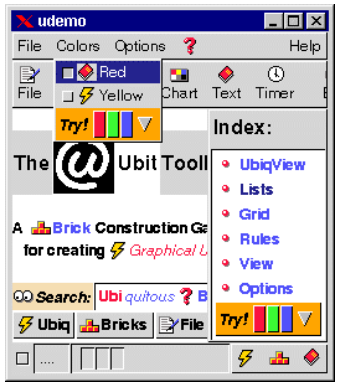
CurrentStatus

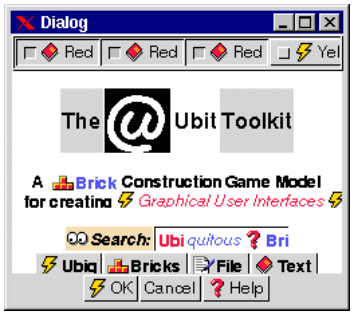
- **Currentstatus**
  - Implementedandfreelyavailable
  - BasedonXWindow/Motif1.1
- **Availability**
  - GNUGPL
  - **http://www.enst.fr/~elc**
- **Currentapplications**
  - Varioustoolsandstudents'projects:
    - SimplifiedHTMLbrowser,imageandtextedition, Website/hypermediavisualization,etc...

Ubit toolkit/Futurework

- **A new GUI toolkit**
  - Based on a construction game metaphor
  - **"Basic bricks"** that can easily be combined
- **Goals**
  - Simplicity, compactness
  - Versatility, application-specific components
  - GUI control, multiple views, synchronization
  - Toolkit flexibility







➡ **A tool for exploring new research directions**

➡ **http://www.enst.fr/~elc**
and Interact'99 paper