ICDL - ESUG 2007 - Lugano, Switzerland

Bridging the Gap Between Morphic Visual Programming and Smalltalk Code

Noury Bouraqadi
École des Mines de Douai
France
http://csl.ensm-douai.fr/noury

Serge Stinckwich
GREYC - CNRS / Univ. de Caen
France
http://www.iutc3.unicaen.fr/serge

Context: Visual Programming in Squeak

- GUI model named Morphic
  - Introduced first in Self
  - All GUI elements are objects named Morphs
    - e.g. buttons, windows, menus, frames, …
  - Morph support visual manipulation
    - e.g. Move, rotate, resize, clone, colors, layout…
- EToys: A visual scripting on top of Morphic
  - Expressions = Tiles
  - Tiles may be assembled to get complex expressions
  - Scripts = sequences of tiles
    - May be reactive or run in a loop

Motivations

- Etoys is not suitable for complex applications
  - Restricted set of expressions (messages)
  - Limited reuse possibilities

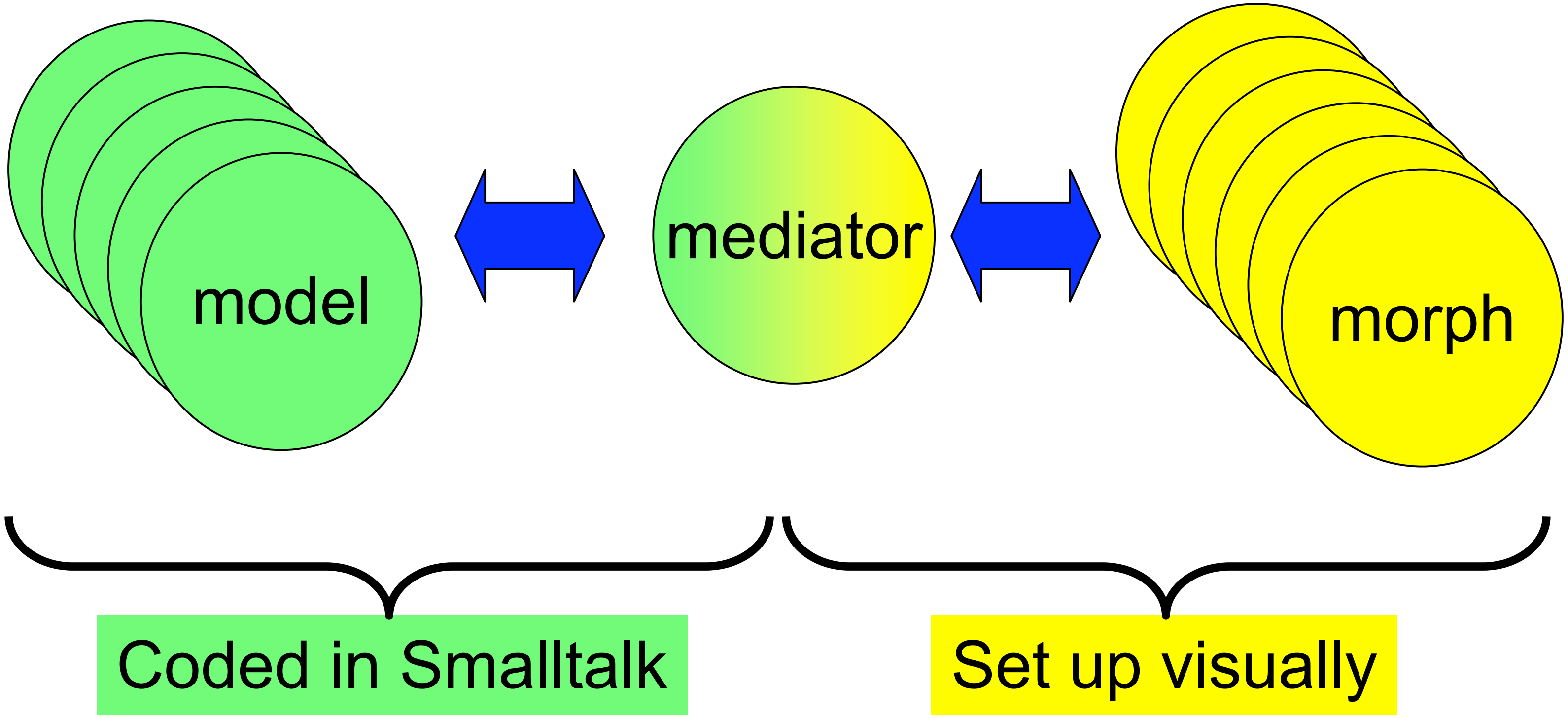
- GUI construction in Squeak is programmatic
  - Pure Smalltalk code
  - No direct connection to Morphic and Etoys visual programming capabilities

Requirements for a GUI Construction Environment

1. Visual GUI development by **direct manipulation**
   - GUI parts should be live application objects (not descriptions!)
2. Smalltlak "traditional" tools for business code
   - Smalltalk has a many powerful programming tools
3. Support GUI reuse
   - Reuse and compose existing GUIs
4. Storage of the GUI with the code
   - Avoid project scattering
5. Support GUI versioning
   - Allow rollbacks during project life-time

Overview of Easy Morphic GUI (EMG)

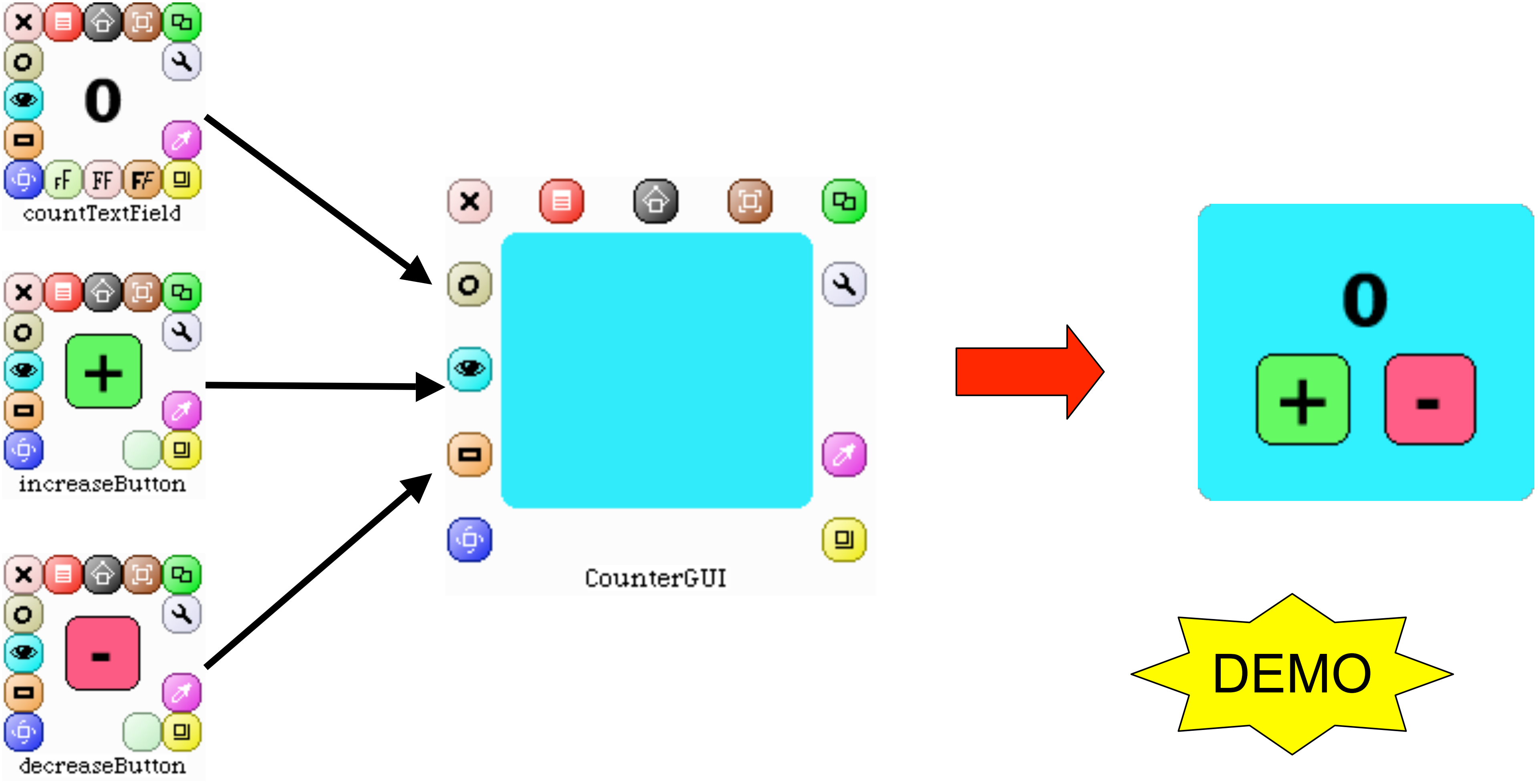
- EMG = framework + Morphic extension
- Development using EMG



Demo
Steps for building a counter with a GUI

1. Building a model class: Counter
2. Building a mediator class: CounterGUI
   - Subclass of EMGGuiMorph
   - Both a mediator and a morph
3. Set up visually the prototypical instance of CounterGUI
   - CounterGUI is a container for other morphs

Demo
Step 3: GUI visual assembling



DEMO

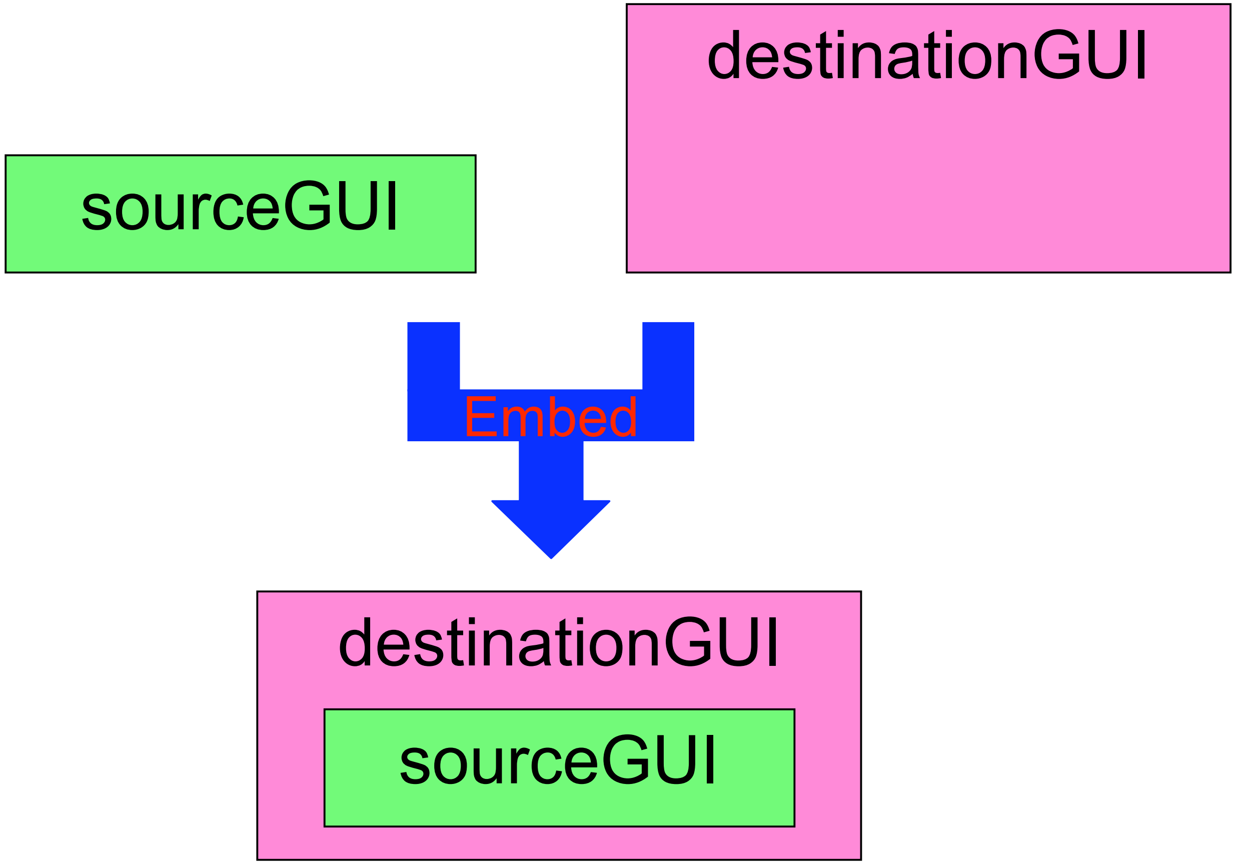
EMG Visual Reuse Operators

- Two levels of development
  - Smalltalk code
  - GUI visual construction

- Need for an extra reuse level beside Smalltalk reuse facilities (inheritance, composition)
  - Subclassing a mediator class don't lead to GUI reuse !
    - Reminder : Mediator is also a container of morphs

- Introduction of two visual reuse operators
  - Embed
  - Clone

EMG Visual Reuse Operators
Embed operator



EMG Visual Reuse Operators
Clone operator

- Static-Clone<sourceGUI, destinationGUI>
  - Empty destinationGUI
  - Copy submorphs of sourceGUI into destinationGUI
  - Link new submorphs to destinationGUI

- Dynamic-Clone<sourceGUI, destinationGUI>
  - Static-Clone
  - Update destinationGUI whenever sourceGUI changes

Example of reuse by clone
Building a circular counter with a GUI

1. Building a model class : CircularCounter
   - Circular counter = a counter which value loops between a min and a max
2. Building a mediator class CircularCounterGUI
3. Use the clone operator to set up the prototypical instance of CircularCounterGUI

DEMO

EMG Relies on Different Design Patterns

- Mediator
  - Link betwen model objects and morphs
- Prototype
  - Each mediator class holds a GUI prototype
- Null Object
  - A null object is used instead of missing morphs during early GUI development stage
- Observer
  - Allow model changes reflect on the GUI

Summary
EMG Does Meet our Requirements

1. Visual tools for GUI by **direct manipulation**
   - Use of Morphic and Etoys == direct manipulation of GUI objects
2. Smalltalk "traditional" tools for business code
   - **Model objects are created and referenced** by GUI objects
3. Support GUI reuse
   - Introduction of 2 reuse operators : "clone" and "embed"
4. Storage of the GUI with the code
   - A serialized version of a GUI is stored as string in a class method
5. Support GUI versioning
   - Use of existing method version management system
   - Use of existing project management system

Perspectives

- Improve reliability and efficiency
  - On-going work
- Better Integration with Morphic and Etoys
  - All morphs should support visual manipulation
    - Menus, Halos, …
- Visualization of references between morphs
  - Ease understanding and maintenance
- Have a visual representation of model objects
  - Integration with Morphic Wrappers ?

Easy Morphic GUI

- Mix the best of both worlds
  - Smalltalk + Morphic and Etoys
  - Developers use the most appropriate tool
- Available for free download at:
  - http://csl.ensm-douai.fr/EasyMorphicGUI
- Submtited to the ESUG Awards 2007
  - Vote for EMG