Interactivity in Active Objects Model

Dorian Gorgan, Vasile Sebastian Cornea
Computer Science Department
Technical University of Cluj-Napoca
e-mail: Dorian.Gorgan@cs.utcluj.ro
Contents

- Introduction
- Visual Programming
- Active Objects Model
- Interactors
- Experiments
- Results
- Conclusions
Computational Models

- Procedural
- Object oriented
- Rule based
- Logic
- Constraint based
- ...

- Dynamical simulation
- Concurrency
- Parallelism
- Adaptive behavior
Program Development Issues

- Description along a linear address space
- N-dimensional parallel evolution
- Concurrency
- Cooperation
- Adaptive behavior
- Visual programming techniques
  - Graphical presentation
  - Application entities manipulation
  - Complex space navigation
  - Dynamic relationships
  - Data structures
  - Programming control structures
Linear address space

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>45A62</td>
<td>( A = \cos(\alpha\sqrt{m+n}) );</td>
</tr>
<tr>
<td>45A62</td>
<td>mov ax,...</td>
</tr>
<tr>
<td>45A63</td>
<td>shr dl,...</td>
</tr>
<tr>
<td>45A65</td>
<td>add al,...</td>
</tr>
<tr>
<td>45A67</td>
<td>jmp 45A63h</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Program Counter
Value Space Mapping

- Attribute value space is mapped into 2D/3D virtual space
- **Trajectory** = mapping the evolution throughout the value space into the evolution throughout the virtual space
- **Position** = explicit value space position where/particular functionality is achieved
Active Objects Model - AOM

- Active Objects Model
- Virtual Space Modeling
- Visual Programming
AOM – Development Environment
AOM – Development Environment
Concerning on AOM Interactivity

- **User Interaction Techniques**
  supported by AOM into GUI
  - Keyboard events
  - Mouse events
  - Real time events

- **Access to Computer Resources**
  - Files, directories
  - Video, audio
  - Internet communication

- **Real Process Control**
AOM Functional Levels

User

Graphical User Interface

AOM Model

AOM Platform

Computer Resources

Internet
AOM Entities

- Active Entities:
  - Active object (agent)
  - Variable
- Passive Entities:
  - Behavior
  - Trajectory
  - Explicit trajectory position (ETP)
  - Rule
  - Action
  - Expression
  - Presentation: graphics, audio, video
  - Interactor
  - Event
  - Active zone
AOM – Entity Relationships

Active Object
  └── Behavior Processor
      ├── Server Processor
      │    └── Presenter Processor
      │         └── Interaction Processor
  └── Bhv
      └── Trj
          └── Presentation
              └── ETP
                  ├── ETP 1
                  │     └── ETP 2
                  │         └── ETP n
Interactor I1{
    position (175,230);
    mouse_event M1 {...}, M2 {...};
    keyb_event K1{...};
    timer T1 {...};
    stream S1 {...};
};
Mouse Event

```plaintext
mouse_event Lbdown{
  type LEFTBDOWN;
  zone Zone1{
    position(32,57);
    visibility 1;
    graphics G1{
      position (10,7);
      select brush,br1{
        color RGB(0,255,0);
      };
      ellipse (0,0,50,50);
    };
  }
  do{
    type UNCOND;
    rule R1{
      action A1{
        set agent(AS).interactor(IM).
        mouse_event(Lbup).
        zone(Zone1).graphics(G1).
        brush(brsh1).color_green,0;
      };
    }
  }
}
```
Mouse Event Based Interaction
Keyboard Event

```
keyb_event K1{
    position(1,1);
    type KEYDOWN;
    key ANY;
    do{
        type UNCOND;
        rule R1{
            position(4,4);
            action Act01{
                set agent(Ts).presentation(pr).
                graphics(G1).drawtext,
                "Value: " + agent(Ts).
                interactor(I1).
                keyb_event(K1).value;
            }
        };
    }
}
```
Keyboard Event Based Interaction
Timer Event

timer T1{
    position(1,5);
    start 0.0;
    end 2.0;
    steps 0.01;
    loop TRUE;
    do{
        type UNCOND;
        rule R1{
            position(4,4);
            action Act{
                set agent(Clock).presentation
                (Hour).graphics(G1).drawtext,
                agent(Clock).interactor(I1).
                timer(T1).tmrminute + ":" +
                agent(Clock).interactor(I1).
                timer(T1).tmrsecond;
            }
        }
    }
}
Timer Event Based Interaction
Interaction Technique - Menu
Complex Interaction Techniques
Counter – Simple Application
Process Controller
Internet based – Image visualization
Conclusions

- Complex interactors
- Hierarchical definition
- Specialization
- Software platform independence
- Device independence
- Experiments
- Task modeling