



OpenOffice.org
Conference 2008 Beijing
世界开源大会



Overview of SMIL Animation and Implementation in IBM Lotus Symphony

Clarence Guo

Software Engineer/IBM



Agenda

- **What is SMIL**
- **What's new in SMIL 3.0**
- **Characters of SMIL**
- **Animations in Symphony**
- **Looking forward to the animation in Symphony**
- **QA**



Agenda

- **What is SMIL**
 - **SMIL ≠ smile**
 - **Bird's eye on SMIL**
 - **A Sample for the animation**
 - **Main Elements**
 - **Recommended 5-layer structure for the animation**
- **What's new in SMIL 3.0**
- **Characters of SMIL**
- **Animations in Symphony**
- **Looking forward to the animation in Symphony**
- **QA**



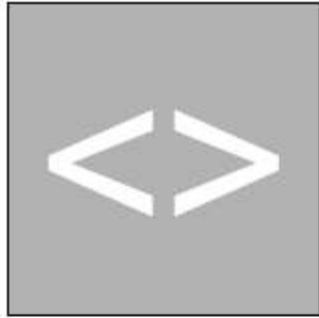
SMIL



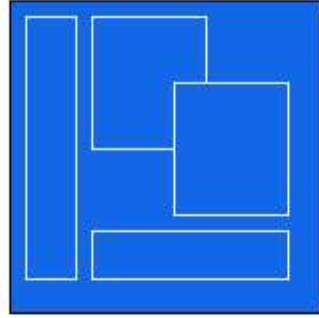
- Synchronized Multimedia Integration Language
- 1.0 published by W3C for web-based multimedia support in 1998, 2.0 in 2000, 3.0 in Oct. 2008
- A standard language that is also recommended by OASIS (Organization for the Advancement of Structured Information Standards) and it is integrated into ODF
- Defines an XML-based language to get powerful support for interactive multimedia
- Sponsors: RealNetworks, IBM, Intel, Microsoft, Macromedia, AOL, Nokia, Panasonic, Philips...
- Products: IE, RealOne, QuickTime...



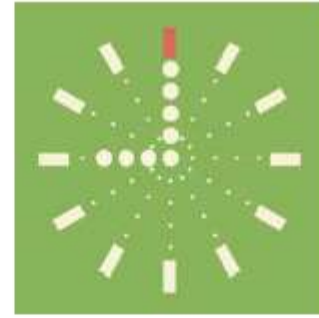
SMIL – Bird's Eye



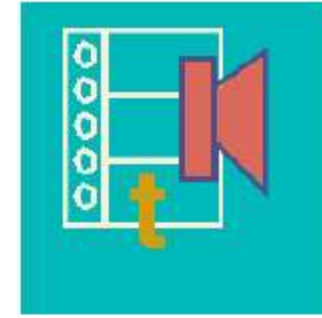
Structure



Layout



Timing
Time Manipulation



Media Objects



Transitions
Animations



Interaction



Content Control



Linkings



SMIL – a Sample



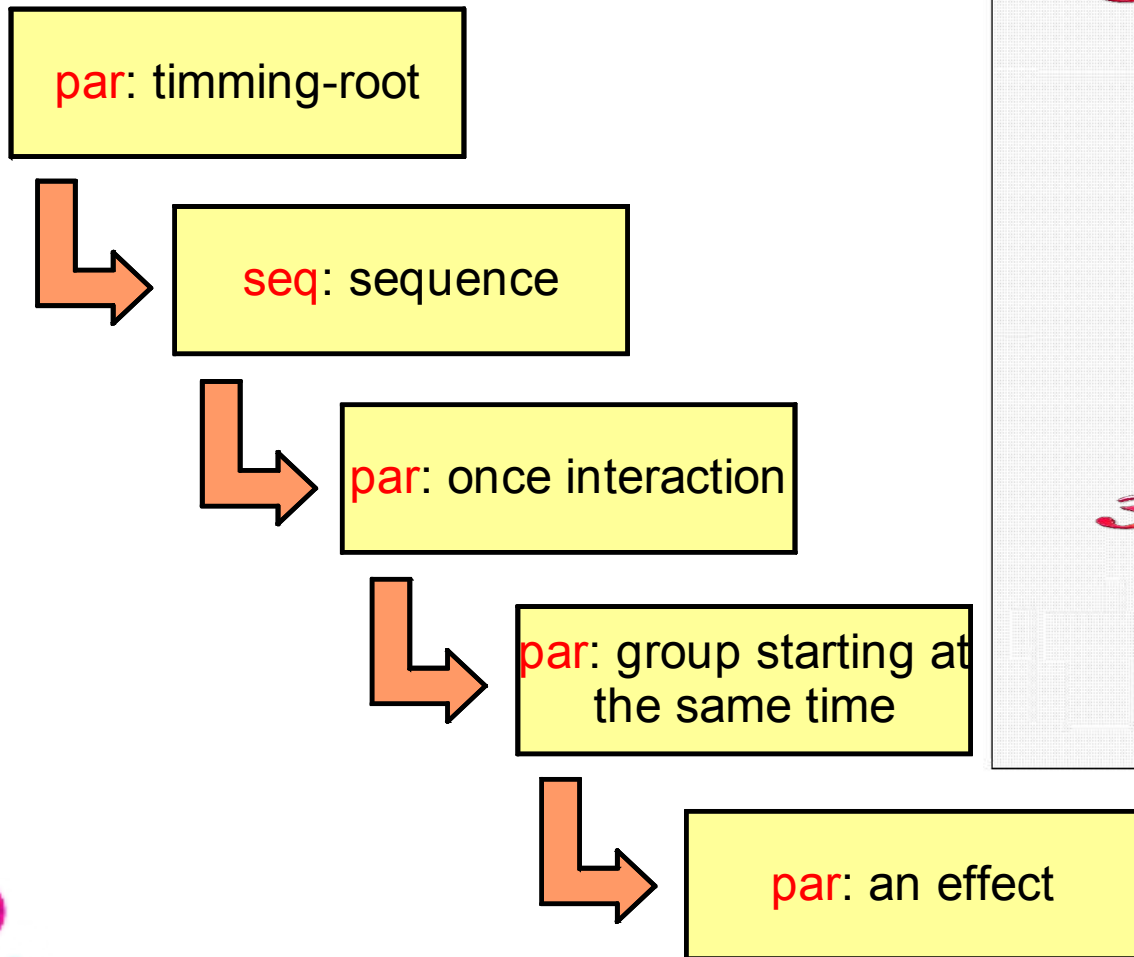
```
- <anim:par presentation:node-type="timing-root">
- <anim:seq presentation:node-type="main-sequence">
- <anim:par smil:begin="indefinite" smil:fill="hold">
- <anim:par smil:begin="0s" smil:fill="hold">
- <anim:par presentation:node-type="on-click" presentation:preset-id="Whirlpool" presentation:preset-class="emphasis"
  smil:begin="0s" smil:fill="hold">
  <anim:animateTransform smil:begin="0s" smil:dur="2s" smil:fillDefault="hold" smil:fill="hold" smil:targetElement="id1"
    smil:by="360" svg:type="rotate" />
  </anim:par>
</anim:par>
- <anim:par smil:begin="indefinite" smil:fill="hold">
- <anim:par smil:begin="0s" smil:fill="hold">
- <anim:par presentation:node-type="on-click" presentation:preset-id="Star (4 Point)" presentation:preset-
  class="motion-path" smil:begin="0s" smil:fill="hold">
  <anim:animateMotion smil:begin="0s" smil:dur="2s" smil:fillDefault="hold" smil:fill="hold" svg:path="M 0 0 L 0.091 -
    0.0453 L 0.125 -0.16655 L 0.158 -0.0453 L 0.249 0 L 0.158 0.0453 L 0.125 0.16655 L 0.091 0.0453 L 0 0 Z"
    smil:calcMode="paced" smil:targetElement="id2" />
  </anim:par>
</anim:par>
- <anim:par smil:begin="indefinite" smil:fill="hold">
- <anim:par smil:begin="0s" smil:fill="hold">
- <anim:par presentation:node-type="on-click" presentation:preset-id="Change Fill Color" presentation:preset-
  class="emphasis" smil:begin="0s" smil:fill="hold">
  <anim:animateColor smil:begin="0s" smil:dur="2s" smil:fillDefault="hold" smil:fill="hold" smil:targetElement="id3"
    smil:attributeName="fillColor" smil:additive="replace" smil:to="#cc66ff" anim:color-interpolation="rgb" anim:color-
    interpolation-direction="clockwise" />
  </anim:par>
</anim:par>
</anim:seq>
</anim:par>
```

SMIL – Main Elements

- **animateTransform**: animates a transformation attribute, e.g., scale, rotate, skew, etc.
- **animateMotion**: animates along a path
- **animateColor**: specifies an animation of a color attribute
- **transitionFilter**: animates the progress of changing a polygon
- **animate**: animates numeric attributes, e.g., x, y, width, height, etc., as well as non-numeric attributes
- **set**: animates simple attributes, e.g., font name, font size, etc.
- **seq**: represents sequence (main or interactive)
- **par**: timing container/Synchronization



SMIL – 5-layer structure for Animation



```
1<anim:par> <!-- timing root-->
2  <anim:seq> <!-- main sequence-->
3    <anim:par smil:begin="indefinite">
4      <!-- first user interaction -->
5        <anim:par smil:begin="0s" smil:dur="4s">
6          <!-- effect a -->
7            <!-- nodes for effect a-->
8          </anim:par>
9          <!-- effect b -->
10         <!-- nodes for effect b-->
11       </anim:par>
12     </anim:par>
13     <anim:par smil:begin="4s">
14       <!-- effect c -->
15       <!-- nodes for effect c-->
16     </anim:par>
17   </anim:par>
18   <anim:par smil:begin="indefinite">
19     <!-- second user interaction-->
20     <anim:par smil:begin="0s" smil:dur="4s">
21       <!-- effect d -->
22       <!-- nodes for effect d-->
23     </anim:par>
24   </anim:par>
```

The code is annotated with numbers 1-5 and labels Group 1, Group 2, and Group 3. Group 1 is the first user interaction, Group 2 is the second user interaction, and Group 3 is the third user interaction. The layers are: 1. <anim:par> (timing root), 2. <anim:seq> (main sequence), 3. <anim:par smil:begin="indefinite"> (once interaction), 4. <anim:par smil:begin="0s" smil:dur="4s"> (group starting at the same time), 5. <anim:par> (an effect).



Agenda

- What is SMIL
- **What's new in SMIL 3.0**
- **Characters of SMIL**
- **Animations in Symphony**
- **Looking forward to the animation in Symphony**
- **QA**



What's New in SMIL 3.0 (part I)



- Three new functional areas
 - smilText - a new media type that provides a text container element for defining the timed text, and a set of additional elements and attributes to control timed text rendering
 - State - Provides a mechanism for the author to create more complex control flow
 - DOM - Better DOM support for programmable content control



What's N

```
<seq>
  <audio src="chapter1.mp3" />
  <setvalue ref="lastPlayed" value="1" />
  <audio src="chapter2.mp3" />
  <setvalue ref="lastPlayed" value="2" />
  <audio src="chapter3.mp3" />
  <setvalue ref="lastPlayed" value="3" />
</seq>

<smil>
  <head>
    <state xml:id="stateid">
      <data xmlns="">
        <lastPlayed>0</lastPlayed>
      </data>
    </state>
    <submission xml:id="subid" action="http://www.example.com/savexmldoc" method="put" />
  </head>
  <body>
    <par>
      <send submission="subid" begin="stateid.stateChange(lastPlayed)" restart="always" />
      ...
      <seq end="... some interactive end condition ..." >
        <seq expr="lastPlayed &lt; 1">
          <audio src="chapter1.mp3" />
          <setvalue ref="lastPlayed" value="1" />
        </seq>
        <seq expr="lastPlayed &lt; 2">
          <audio src="chapter2.mp3" />
          <setvalue ref="lastPlayed" value="2" />
        </seq>
        <seq expr="lastPlayed &lt; 3">
          <audio src="chapter3.mp3" />
          <setvalue ref="lastPlayed" value="3" />
        </seq>
      </seq>
    </par>
  </body>
</smil>


</switch>
```



- |

es
al
e
ion
re

renders `` Saves author defined state or to transmit it to an external server

- **DOMTimingMethods**: Contains DOM methods to start and stop parts of a presentation during playback, and also DOM events that may be used to influence a presentation



What's New in SMIL 3.0 (part III)

- Revised some functional areas
 - Content Control
 - Layout
 - Linking
 - Media Object
 - Timing and Synchronization
 -
- No change for animation



Agenda

- What is SMIL
- What's new in SMIL 3.0
- **Characters of SMIL**
- **Animations in Symphony**
- **Looking forward to the animation in Symphony**
- **QA**



Characters of SMIL

- An open and free architecture
- Clear responsibility definitions
- Easy to learn and use
- Integrates/conforms to other standards, e.g., XML, SVG, DOM
- Synchronously play multiple multimedia objects that are deployed at different sites
- Complex control flow in terms of different situations
- Easy to convert to/from other languages, e.g., HyTime
- A declarative language; Is Scripting required by experts?

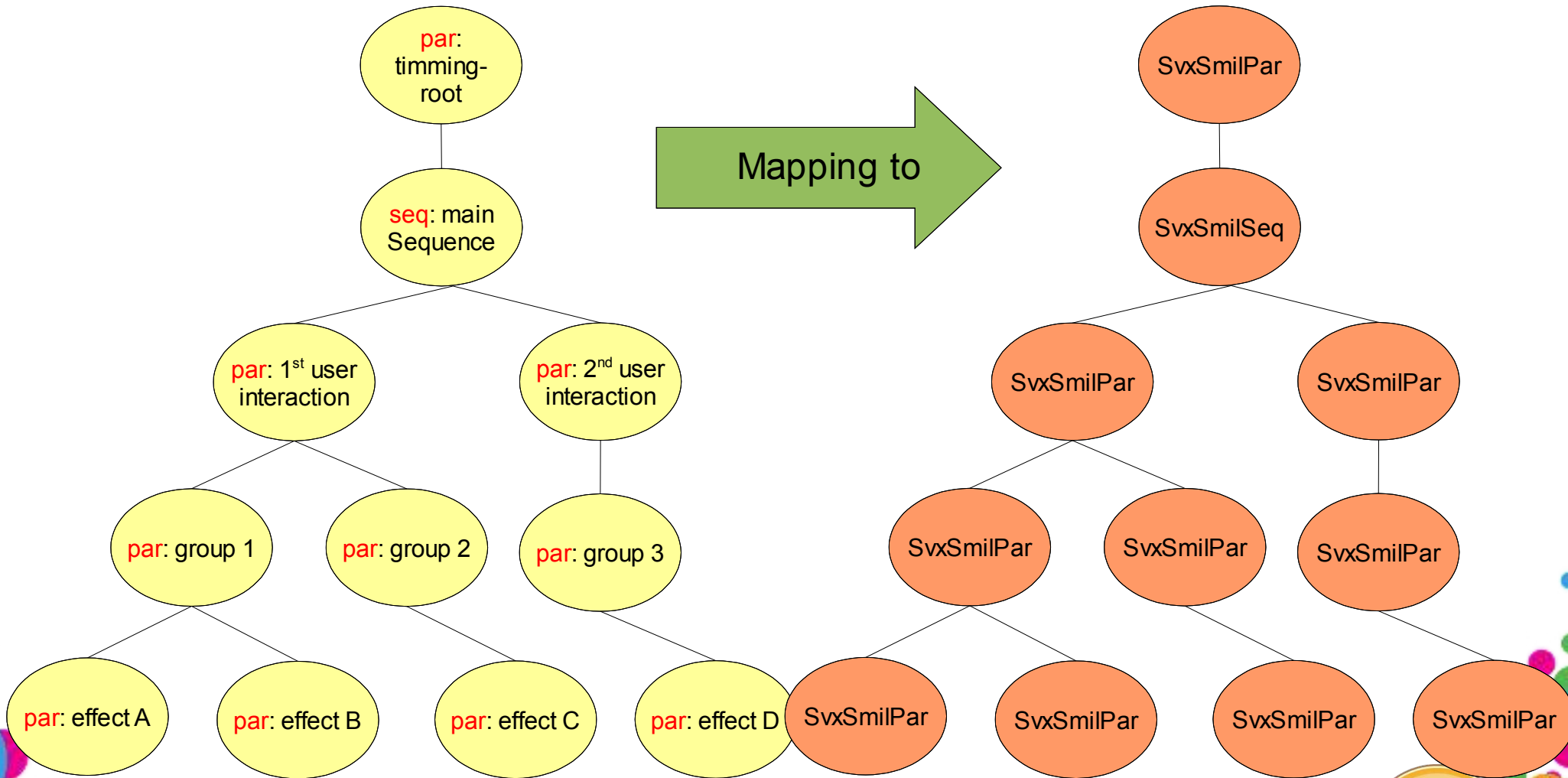


Agenda

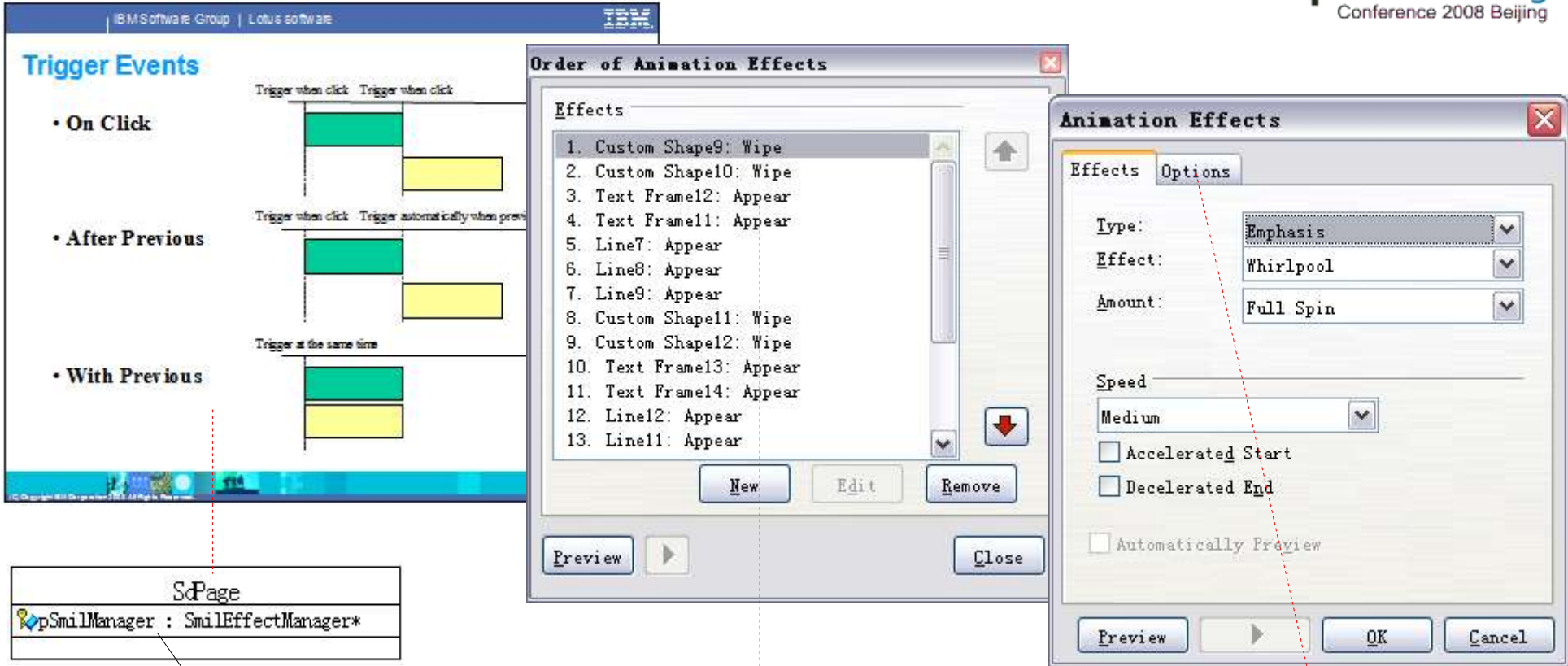
- What is SMIL
- What's new in SMIL 3.0
- Characters of SMIL
- **Animations in Symphony**
 - Design for model
 - Design for UI
 - Timing and Drawing
 - Design for copy/paste
- Looking forward to the animation in Symphony
- QA



Animations in Symphony - Design for model



Animations in Symphony - Design for UI

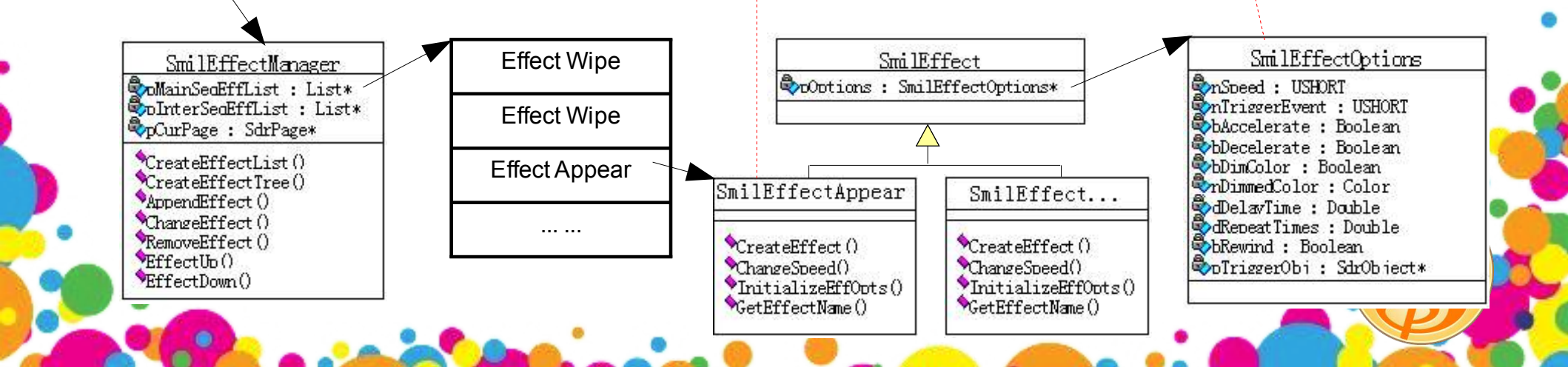


The screenshot displays the Symphony software interface with three windows open:

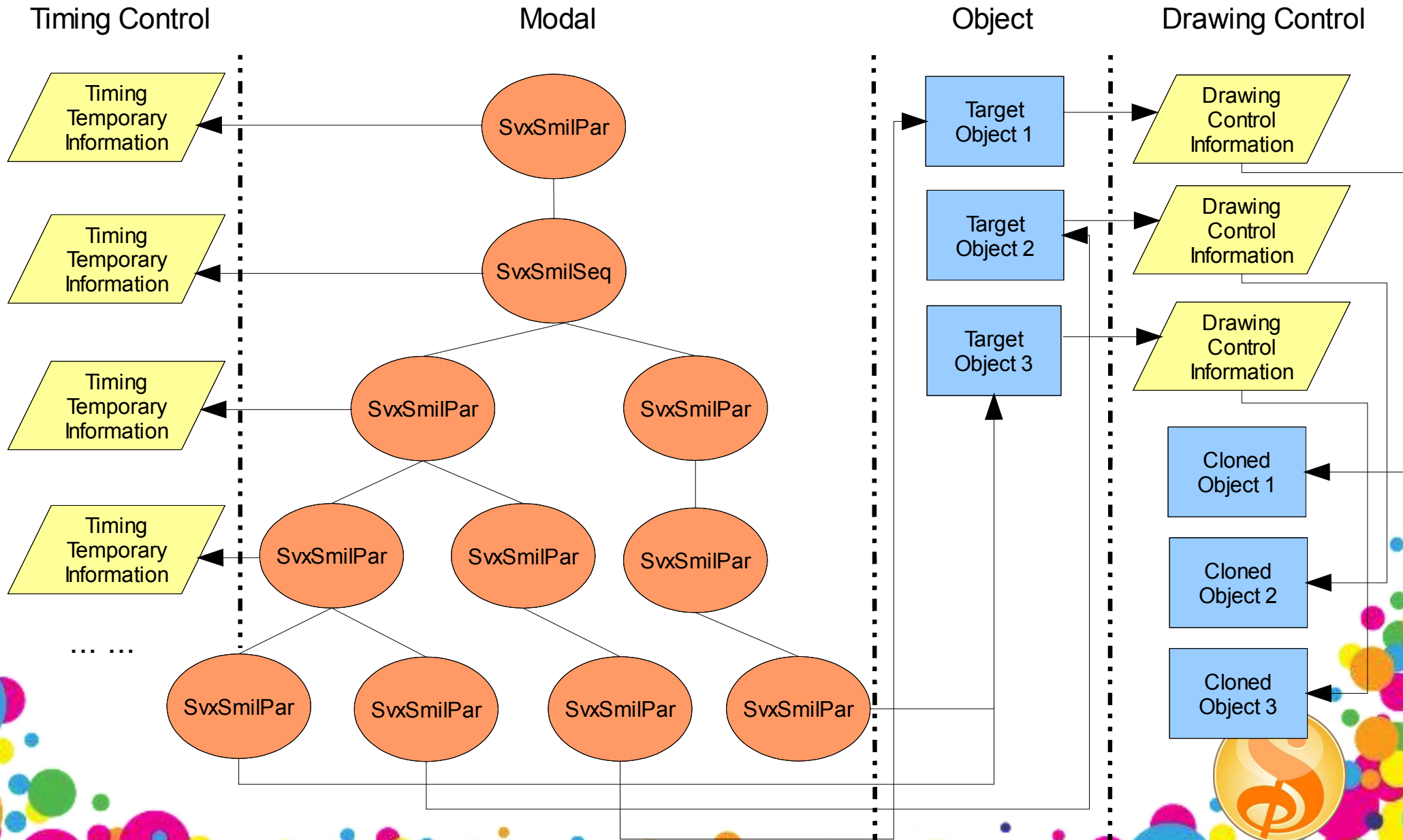
- Trigger Events:** Shows three event types:
 - On Click:** Trigger when click (green bar), Trigger when click (yellow bar).
 - After Previous:** Trigger when click (green bar), Trigger automatically when prev (yellow bar).
 - With Previous:** Trigger at the same time (green bar), Trigger at the same time (yellow bar).
- Order of Animation Effects:** Lists 13 effects in a sequence:
 - Custom Shape9: Wipe
 - Custom Shape10: Wipe
 - Text Framel2: Appear
 - Text Framel1: Appear
 - Line7: Appear
 - Line8: Appear
 - Line9: Appear
 - Custom Shape11: Wipe
 - Custom Shape12: Wipe
 - Text Framel3: Appear
 - Text Framel4: Appear
 - Line12: Appear
 - Line11: Appear
- Animation Effects:** Shows configuration for an effect:
 - Type: Emphasis
 - Effect: Whirlpool
 - Amount: Full Spin
 - Speed: Medium
 - Accelerated Start:
 - Decelerated End:
 - Automatically Preview:

Below the screenshot is a class diagram showing the following classes and their relationships:

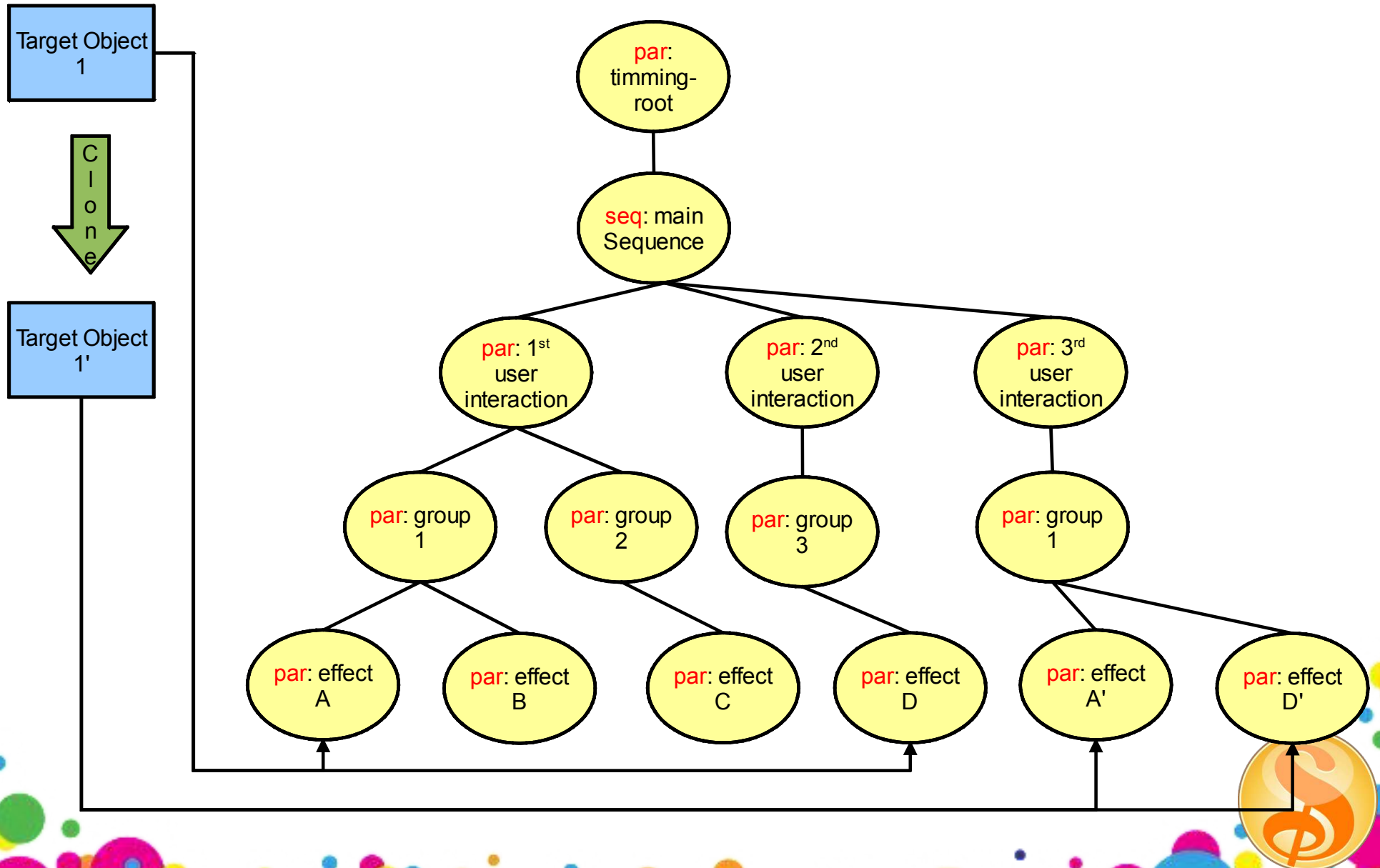
- SdrPage** (Class): Contains `pSmilManager : SmilEffectManager*`.
- SmilEffectManager** (Class): Contains `mMainSeqEffList : List*`, `mInterSeqEffList : List*`, and `pCurPage : SdrPage*`. Methods include `CreateEffectList()`, `CreateEffectTree()`, `AppendEffect()`, `ChangeEffect()`, `RemoveEffect()`, `EffectUp()`, and `EffectDown()`.
- Effect Wipe** (Class): Inherits from **SmilEffectManager**.
- Effect Appear** (Class): Inherits from **SmilEffectManager**.
- SmilEffect** (Class): Contains `mOptions : SmilEffectOptions*`. Methods include `CreateEffect()`, `ChangeSpeed()`, `InitializeEffOpts()`, and `GetEffectName()`.
- SmilEffectAppear** (Class): Inherits from **SmilEffect**. Methods include `CreateEffect()`, `ChangeSpeed()`, `InitializeEffOpts()`, and `GetEffectName()`.
- SmilEffect...** (Class): Inherits from **SmilEffect**. Methods include `CreateEffect()`, `ChangeSpeed()`, `InitializeEffOpts()`, and `GetEffectName()`.
- SmilEffectOptions** (Class): Contains `mSpeed : USHORT`, `mTriggerEvent : USHORT`, `mAccelerate : Boolean`, `mDecelerate : Boolean`, `mDimColor : Boolean`, `mDimmedColor : Color`, `mDelayTime : Double`, `mRepeatTimes : Double`, `mRewind : Boolean`, and `mTriggerObj : SdrObject*`.



Animations in Symphony - Timing and Drawing



Design for copy/paste



Agenda

- What is SMIL
- What's new in SMIL 3.0
- Characters of SMIL
- Animations in Symphony
- **Looking forward to the animation in Symphony**
- QA



Looking forward to the animation in Symphony

- Interoperability
- Performance
- Rich and colorful effects
- New UI



Ref Link:

<http://www.w3.org/TR/2008/PR-SMIL3-20081006/>



Let's SMIL/Smile



Agenda

- What is SMIL
- What's new in SMIL 3.0
- Characters of SMIL
- Animations in Symphony
- Looking forward to the animation in Symphony
- QA





Thanks!

凝聚全球力量 绽放开源梦想

www.OOobeijing2008.com

