Developing a Statechart-to-ECMAScript Compiler Optimized for SVG User Interface Development for the World Wide Web

Jacob Beard, Hans Vangheluwe
September 1st, 2010
SVG Open 2010
Questions

• **What** is scxml-js?
• What **problems** does it solve?
• How do I **use** it?
scxml-js is...

- A **tool** to facilitate development of **behaviourally rich, web-based User Interfaces (UIs)**
- Well-suited to UIs with complex, **state-based** behaviour requirements
Problem: UI development is hard

- Behaviour **autonomous** or **reactive**, and possibly **real-time**
- Behaviour **differs** between UI components
- Components operate **concurrently**
- Behaviour specification by means of **timeouts**
- **Scope**-delimited behaviour
Solution

Model every aspect of the system-to-be-built explicitly, at the most appropriate level of abstraction, using the most appropriate formalism(s).
Statecharts (David Harel '87)

- Visual language
- Finite State Automata extended with:
  - Hierarchy
  - Concurrency
  - Syntactic Sugar (History, Junctions, etc.)
- Semantics first formalized in “The STATEMATE Semantics of Statecharts”, Harel '96
- Meant to be intuitive
- One of the UML notations (behavioural)
Statecharts Concepts
Applications

Embedded software

Web servers [Mongrel]

Web frameworks

User Interfaces
scxml-js is...

A Statechart-to-ECMAScript compiler optimized for UI development for the World Wide Web
scxml-js is...

```xml
<?xml version="1.0"?>
<scxml>
  <state id="a">
    <transition event="t1"
      target="b">
      <assign location="foo"
        expr="bar"/>
      <script>
        print(foo);
      </script>
    </transition>
  </state>
  <state id="b"/>
</scxml>
```

```javascript
function StatechartExecutionContext() {
  var self = this;
  // system variable declarations
  var _event = {
    name: undefined,
    data: undefined
  },
  _name = "",
  _sessionid;
  var _x = {
    _event: _event,
    _name: _name
  };
  ...
}
```
SCXML is...

- Draft specification by W3C
- XML
- Human-Usable Textual Notation
  - *Although you still want tool support*
- scxml-js aims to comply with SCXML syntax and semantics
scxml-js is...

- Apache Commons Sandbox project
- Google Summer of Code 2010 project
- Open Source Software (Apache license)
Example: Drag-and-Drop

Example: Drag-and-Drop

- **Events**
  - Mousedown
  - Mousemove
  - Mouseup

- **States**
  - Dragging
  - Idle
Drag-and-Drop Statechart

initial_default \(\xrightarrow{\text{init}}\) idle \(\xleftarrow{\text{mouseup}}\) dragging

\(\xrightarrow{\text{mousedown}}\) mousemove
Drag-and-Drop Demo 1

- Link to demo
Drag-and-Drop Demo 1

- Query DOM for SCXML elements
- Compile SCXML content to ECMAScript
- Eval to produce a constructor function
- Instantiate and initialize statechart instance
- Hook up DOM events to statechart dispatcher
Just-in-Time Compilation

```xml
<?xml version="1.0"?>
<scxml>
  <state id="a">
    <transition event="t1"
      target="b">
      <assign location="foo"
        expr="bar"/>
      <script>
        print(foo);
      </script>
    </transition>
  </state>
  <state id="b"/>
</scxml>
```

```javascript
function StatechartExecutionContext() {
  var self = this;
  // system variable declarations
  var _event = {
    name: undefined,
    data: undefined
  },
  _name = "",
  _sessionid; 
  var _x = {
    _event: _event,
    _name: _name
  };
  ...
}
```
Ahead-of-Time Compilation

XML

<scxml>
  <state id="a">
    <transition event="t1" target="b">
      <assign location="foo" exp="bar"/>
      <script>
        print(foo);
      </script>
    </transition>
  </state>
  <state id="b"/>
</scxml>

scxml-js

function StatechartExecutionContext() {
  var self = this;
  //system variable declarations
  var _event = { name: undefined, data: undefined },
      _name = ",
      _sessionid;
  var _x = { _event: _event,
             _name: _name
           };
  ...
}
Drag-and-Drop Demo 2

- Link to demo
Inkscape Behaviour Spec

- Described [here](#)
Inkscape Demo

- Drawing Tool with graphical debugger
Other Applications we developed

- Browser Game AI
  - Tank Wars
Other Applications we developed

- Modal Text Editor
- Bespin and Keyboard Manager
Other Applications

Have ideas?

*Let us know!*
Future Work

- Improve compiler
- Catalogue of Design Patterns for modelling Web UI behaviour
- High-level domain-specific language may emerge
References

• Read the paper
• Visit the project page
  • Check out the code
  • Try the demos
Thanks for attending!