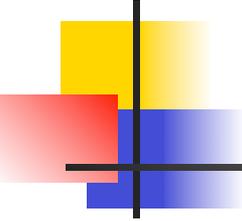




Michael Bodell

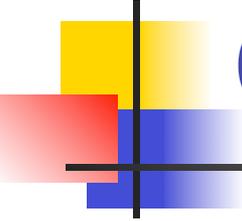
[bodell@tellme.com](mailto:bodell@tellme.com)



# Prologue (VXML 2.0/2.1)

---

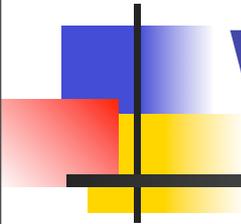
- VoiceXML 2.0/2.1 is a standard out of the Voice Browser Working Group of the W3C
- VXML is to networked phone browsers as HTML is to internet web browsers
- VXML leverages all the same web technologies
- VXML answers billions of phone calls with thousands of different applications



# Overview

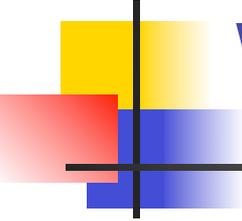
---

- What is SCXML
- Motivation behind SCXML
- Theory of SCXML
- SCXML in a voice-only context
- SCXML in a multimodal context
- Q & A



# What is SCXML?

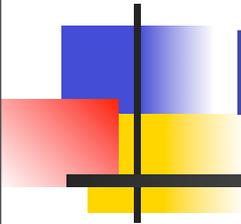
---



# What is SCXML?

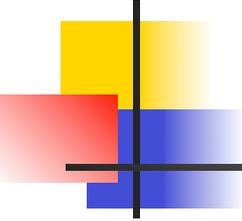
---

- SCXML is State Chart XML
- A W3C standard from the Voice Browser WG
- The Multimodal WG is also planning on using SCXML as a core part of its work



# Motivation behind SCXML

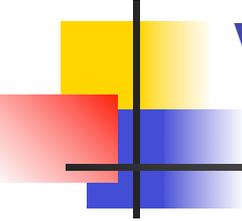
---



# Motivation behind SCXML

---

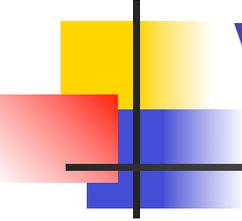
- VBWG VXML 3.0 architecture
- MMIWG overall multimodal architecture



# VBWG Motivation

---

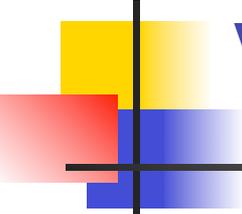
- VXML 2.0/2.1 is a great specification that meets most of the WG's needs
- But want improvements in the following:
  - Reusability
  - Faster, cheaper application development
  - Integrate with other W3C standards
  - Specifically work in a multimodal context
- Solution involves new architecture for VXML 3.0



# VXML 3.0 Architecture

---

- Three tiered “DFP architecture” with:
  - “D” is for data
  - “F” is for flow
  - “P” is for presentation
- Similar to a traditional Model-View-Controller architecture
- Can pick and choose different languages to solve each tier – but VBWG has some languages in mind



# VXML 3.0 Architecture

---

## VXML 3.0 DFP Architecture

### DATA

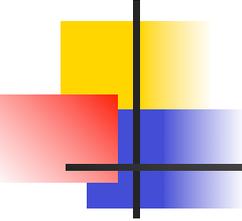
XML Data (W3C DOM, XPath, XForms)

### FLOW

SCXML

### Presentation Components

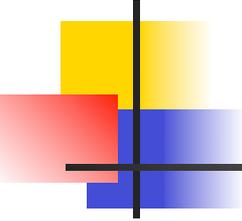
VXML 3.0 Dialog Widgets, VXML 2.0 Dialog Widgets



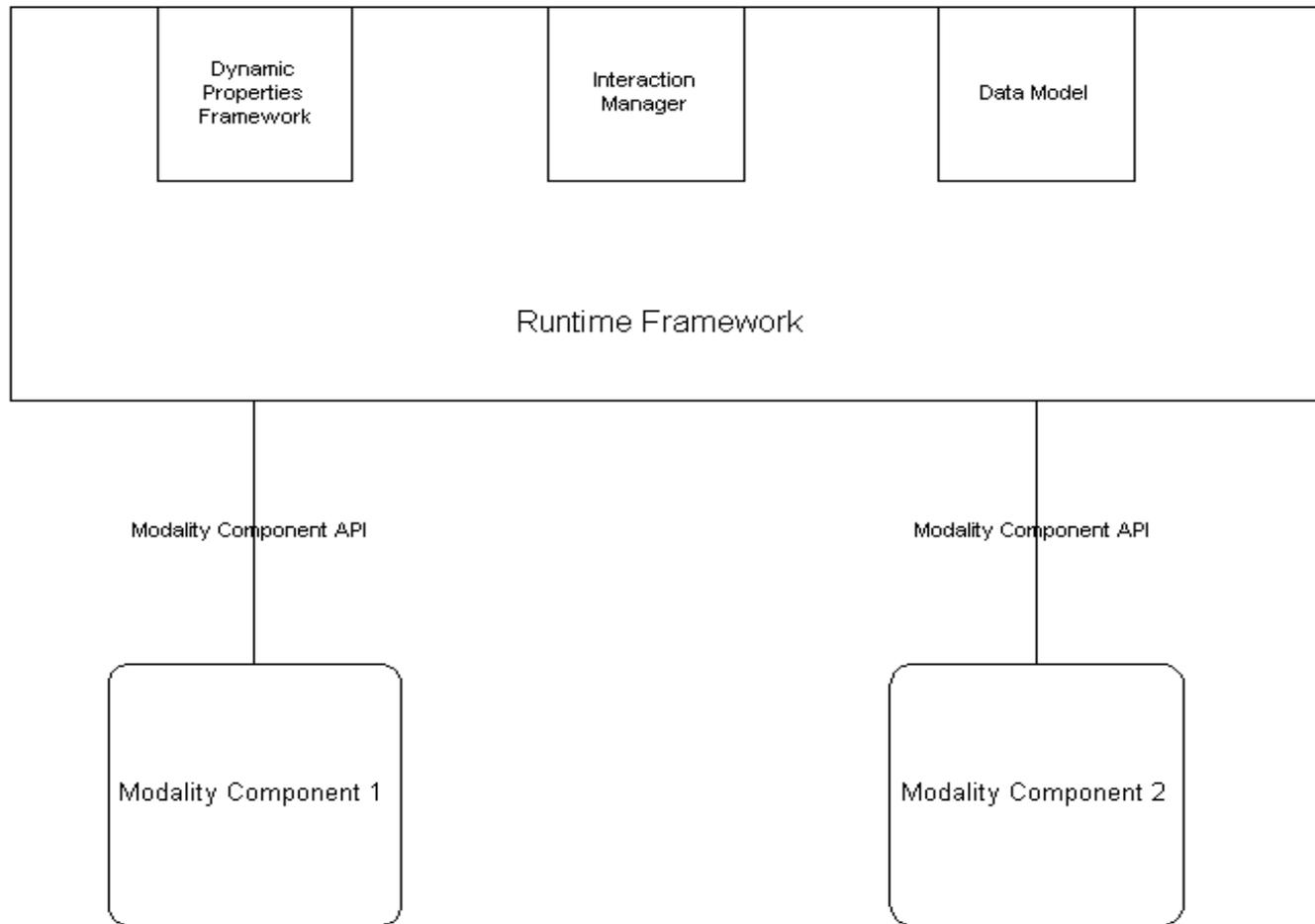
# MMI Motivation

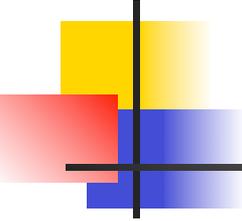
---

- Need a way to mix different W3C languages together
- Want to incorporate existing and future languages
- Need a way to synchronize between input from multiple modalities
- Interaction manager is the piece that solves this problem and SCXML is the authoring interface to the interaction manager



# MMI Architecture

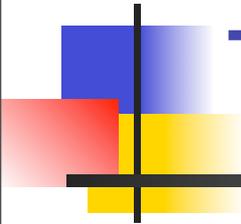




# Bringing it all together

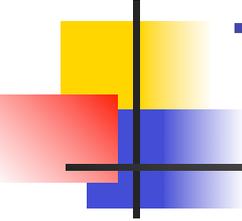
---

- SCXML fits as the “flow controller” in the VBWG DFP architecture
- SCXML fits as the “interaction manager” in the MMIWG Multimodal architecture
- SCXML also has interest from other W3C working groups like XHTML, XForms, SVG, etc.



# Theory of SCXML

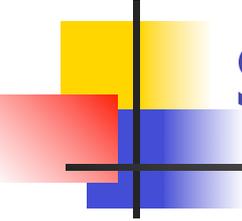
---



# Theory of SCXML

---

- SCXML is based on David Harel's statecharts
- Harel statecharts are a mathematical representation of state machines with a few powerful wrinkles
- Harel statecharts are the underpinning of UML state semantics



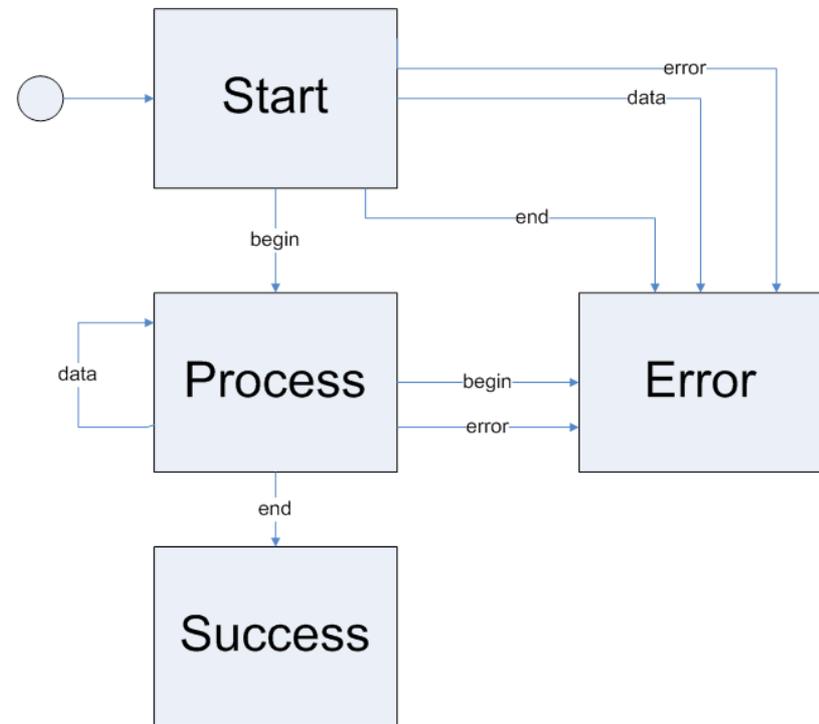
# statechart semantics

---

- statecharts have all the traditional statemachine semantics:
  - states
  - transitions
  - events
  - conditions
- Important advanced features of statecharts are:
  - hierarchical states
  - parallel states
  - action handlers

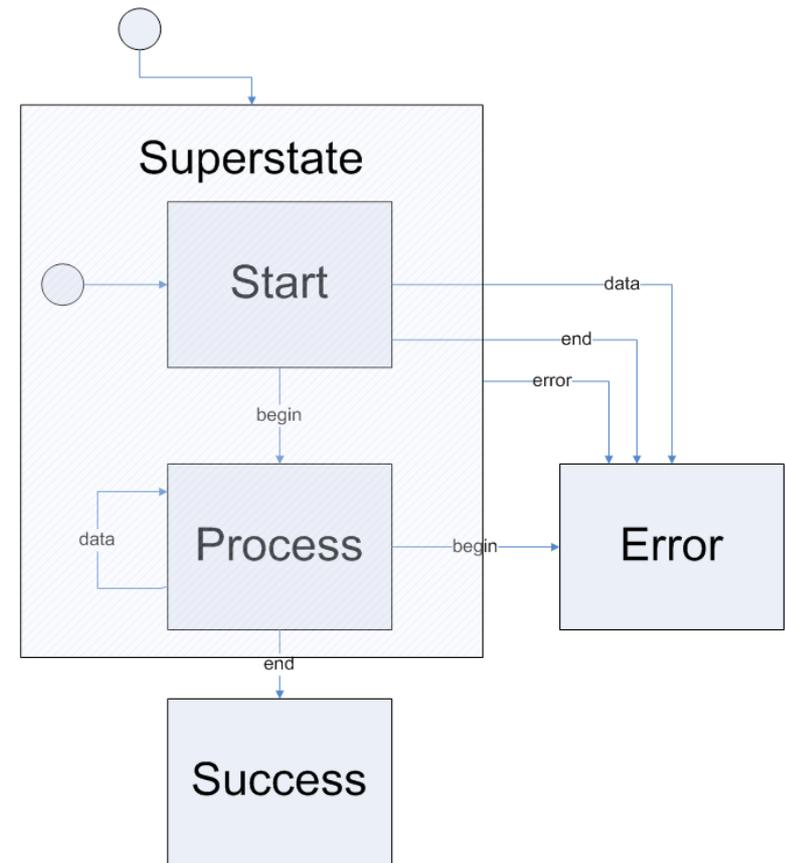
# Basic statechart example

- Dot represents the initial state
- Normal processing gets a begin event, 0 or more data events, and an end event
- out of sequence events or error events cause transitions to the error state

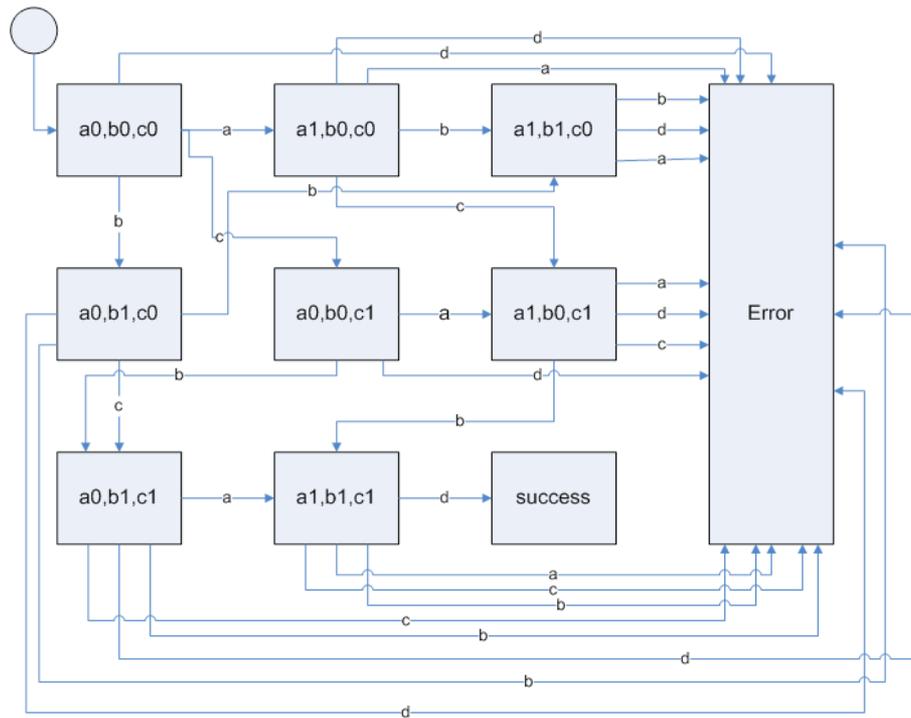


# Hierarchical statechart example

- Superstate incorporates two other states
- This factors out one of the error arcs
- Note:
  - statecharts are not like Venn diagrams, one can't be in Superstate but not be in either Start or Process
  - one can transition to states inside the superstate directly, or one can transition to the superstate itself, which then goes to the initial state
  - in superstates there can be the idea of a history state that remembers which substate you were in last time

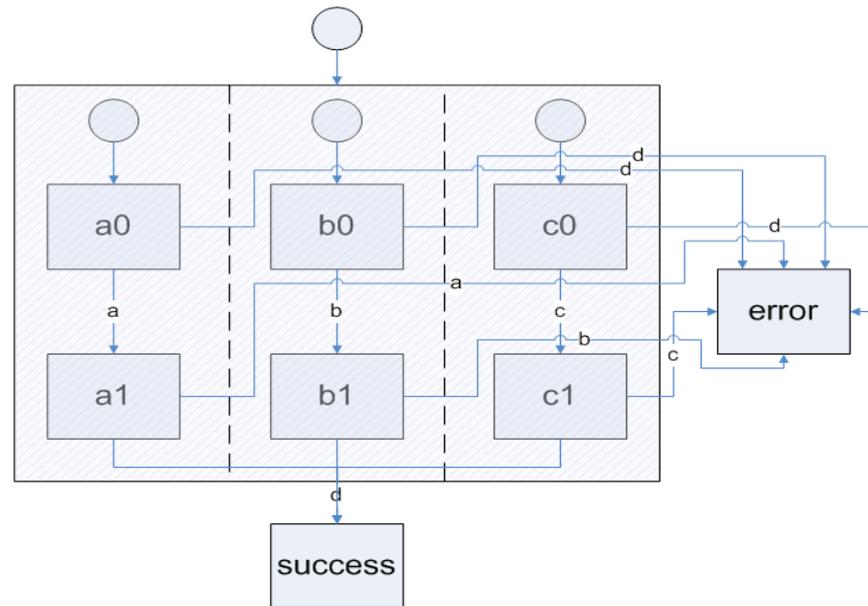


# Parallel statechart example

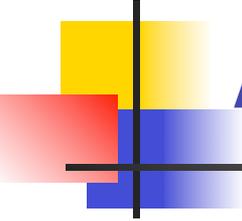


- Simple statechart to accept exactly one of a, b, and c – in any order – followed by a d, with any other thing (d before c, 2 b events, etc.) as an error

# Parallel statechart example



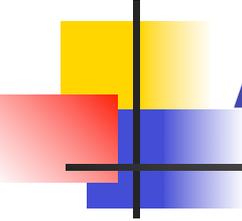
- With parallel regions what was 10 lowest level states and 30 transitions becomes 8 lowest level states and 10 transitions
- Note parallel regions are just shorthand for this type of cross product simplification. They don't necessarily imply parallel execution



# Action handlers

---

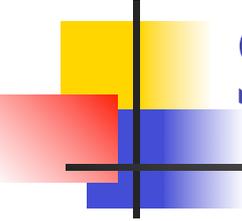
- Statecharts allow actions to be triggered
- Actions handlers can be triggered in 4 ways:
  - on exit of a state
  - on a transition
  - on entry to a state
  - while in a state
- Typical things that one does in action handlers can include:
  - cause another event to fire
  - update the data model
  - interact with the outside world



# Action handlers

---

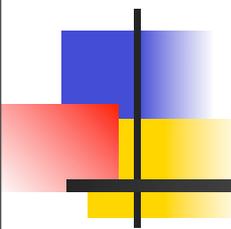
- The order that the action handlers happen is:
  1. all the on exit handlers are triggered for each state that you are exiting starting with the most deeply nested state
  2. the handlers on the transition are triggered
  3. all the on entry handlers are triggered for each state that you are entering starting with the highest level state
  4. while you are then in each state the while handler is triggered



# SCXML semantics

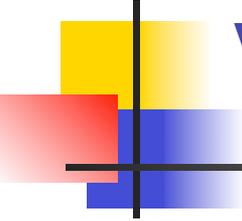
---

- SCXML basically follows the Harel semantics with a few small changes:
  - while handler is replaced with `<invoke>` which can only be in a “simple” state
  - when multiple transitions are eligible at the same level SCXML uses document order to resolve them instead of leaving it indeterminate



# SCXML in a voice-only context

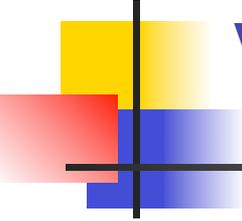
---



# Voice-only SCXML uses

---

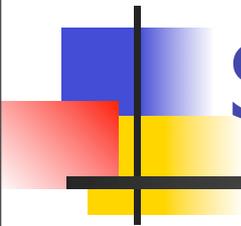
- Continue using normal VXML 2.0/2.1 tags and execution inside each individual `<form>`, i.e., intradialog flow stays the same
- Rather than using `<goto next="URI"/>` and `<submit>` instead at `<form>` completion return control, and collected data, to SCXML which selects the next `<form>` to execute, i.e., interdialog flow is using SCXML



# Voice-only SCXML uses

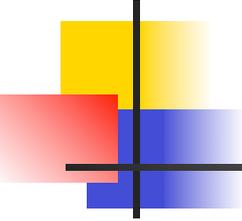
---

- In the voice-only context it is less likely that parallel will be as useful
- More likely that the SCXML flow control will adapt something like a VISIO flow chart a UI designer would make with each “box” becoming a state which invokes a VXML 2.x dialog and the arrows joining them would be the transitions



# SCXML in a multimodal context

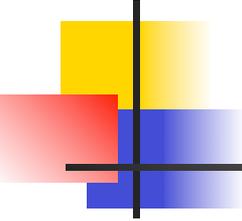
---



# Multimodal SCXML use cases

---

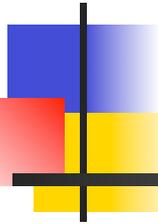
- Could use SCXML to do sequential multimodal (I.e., where you use multiple modalities but only one at a time) by having a simple state machine where each state choose the modality it wanted to invoke.
- Primary focus of MMIWG is on simultaneous multimodal (I.e., not just voice-in and data-out but multiple modalities simultaneously)



# Multimodal SCXML use cases

---

- Likely here parallel would get used much more
- Main idea is that at a high level you may want to collect a sequence of information (I.e., acct #, pin #, action) and at each stage you could use parallel to allow the information to be collected from any modality (voice, visual, ink, etc.)
- Once input is collected from any modality all modalities prompt for the next piece of information



Q & A

---



backup

---