PB138 — XML Applications

W3C Voice Browser Activity

- Standards for Voice and Dialogue applications
  - VoiceXML
  - SRGS
  - SISR
  - SSML
  - PLS
  - Call Control XML
  - State Chart XML
  - ...
- W3C Recommendations

VoiceXML

- Language for dialogue applications development.
- Specification
- Primary targeted to phone applications.
  - telephone support automation
  - railways/bus schedules information
  - ticket reservation
  - ...
- Describes algorithm for dialogue flow control (dialogue strategy)
- Alternatively can be described by finite state automaton with output (Mealy automaton)
  - SCXML
- W3C standard W3C (present version 2.1, version 3.0 in state of Working Draft)

VoiceXML - processing

- Application needs to be run on VoiceXML platform or using VoiceXML interpreter.
  - desktop platforms - OptimTalk, publicVoiceXML, JVoiceXML, ...
  - opensource on-line - Asterisk+VoiceGlue, Asterisk+OpenVXI, ...
  - on-line commercial:
    - Aspect Prophecy
    - ...
  - VoiceXML forms in XHTML documents

        - using namespaces (formerly W3C submission XHTML+Voice profile 1.0)
        - Support in Opera a Firefox web browsers.
    - …

VoiceXML - example

Figure: VoiceXML example

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
   <form id="pizza-mixed">
      <grammar src="pizza.grxml"/>
      <initial name="pizzaall">
       <prompt>Welcome to FI pizzeria</prompt>
       <nomatch count="2"><assign name="pizzaall" expr="true"/></nomatch>
       <noinput count="2"><assign name="pizzaall" expr="true"/></noinput>
      </initial>
      <field name="kind">
       <prompt>What kind of pizza do you want?</prompt>
       <nomatch>We have salami, mozzarela and appolo pizza</nomatch>
       <noinput>We have salami, mozzarela and appolo pizza</noinput>
       <grammar src="pizza.grxml#kind"/>
      </field>
      <field name="topping">
       <prompt>What topping do you want?</prompt>
       <nomatch>We offer ketchup and chilli.</nomatch>
       <noinput>We offer ketchup and chilli.</noinput>
       <grammar src="pizza.grxml#topping"/>
      </field>
      <field name="drink">
       <prompt>What do you want to drink?</prompt>
       <nomatch>Select one of coke, sprite and water</nomatch>
       <noinput>Select one of coke, sprite and water</noinput>
       <grammar src="pizza.grxml#drink"/>
      </field>
      <field name="ack">
       <prompt>Did you ordered <value expr="kind"/> pizza with <value
       expr="topping"/> and <value expr="drink"/>?</prompt>
       <grammar src="yesno.grxml"/>
      </field>
      <filled>
       <if cond="ack=='yes'">
            <prompt>Order submitted</prompt>
       <else/>
            <clear namelist="kind topping drink ack"/>
       </if>
      </filled>
   </form>
</vxml>
```

SRGS (Speech Recognition Grammar Specification)

- Standard for description of context free grammars.
  - describes the accepted inputs of particular VoiceXML fields

- Specification
- Part of W3C Voice Browser Activity standards
- Present version 1.0
- SRGS - motivation
  - User's voice input needs to be recognized - continues speech recognition.
  - success rate 50-99 %
- Possibilities how to improve success rate:
  - improve the language model
  - problem domain restriction
  - improve the user model
- Problem domain restriction + language model improvement = SRGS.

SRGS - example

Figure: SRGS grammar referenced in the previous VoiceXML example (pizza.grxml)

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar root="mixed" xml:lang="en_US">
<rule id="mixed">
   <item>
      <ruleref special="GARBAGE"/>
      <ruleref uri="#kind"/> pizza <ruleref special="GARBAGE"/>
      <ruleref uri="#topping"/> and <ruleref uri="#drink"/>
   </item>
   <tag>
   {
    out.kind=rules.kind;
    out.topping=rules.topping;
    out.drink=rules.drink;
   }
   </tag>
</rule>
<rule id="kind">
   <one-of>
    <item>salami</item>
    <item>mozzarela</item>
    <item>polo</item>
   </one-of>
</rule>
...
</grammar>
```

SISR (Semantic Interpretation for Speech Recognition)

- Purpose:
  - What is the meaning of recognized input?
- Language for derivation of the recognized inputs semantic.
- Based on ECMAScript.
- Used in speech recognition grammars (see previous slide).
- SISR 1.0 Specification

SSML (Speech Synthesis Markup Language)

- link: Speech Synthesis Markup Language
- W3C Standard
- present version 1.1 (September 2010)
- Used to describe prosody characteristics of synthesized speech.
  - loudness
  - prosody
  - emphasis
  - speech rate
  - voice kind (male, female, neutral)
  - ...
- Contains markup for description of pronunciation of foreign words.
  - IPA (International Phonetic Alphabet) can be utilized.

SSML - example of loudness and breaks

Figure: SSML Breaks and loudness control example

```
<?xml version="1.0" encoding="utf-8"?>
<speak version='1.1' xmlns="http://www.w3.org/2001/10/synthesis"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/TR/speech-synthesis11/synthesis.xsd">
   <prosody volume="loud">
      Dobre rano.<break/>
   <prosody>
   <prosody volume="default">
      Jak se mate?
   </prosody>
</speak>
```

SSML - example of intonation modeling

Figure: SSML Intonation modeling

```
<speak ...>
   <prosody contour="(0%,50Hz) (75%, +10%) (80%, +20%) (90%,+30%)">
   Mas se dobre?
   </prosody>
</speak>
```

PLS (Pronunciation Lexicon Specification)

- Pronunciation Lexicon Specification
  - W3C standard
  - Actual version - 1.0, October 2008
- Developed for description of pronunciation of words, abbreviations, etc.
- Used for:
  - Speech synthesis (SSML) - pronunciation of
    - foreign words
    - abbreviations
    - number values
    - ...
  - Speech recognition (SRGS) - PLS allows to describe different pronunciations of some words (needed to be correctly recognized).

PLS Structure

- Root element - lexicon

    - contains one or more lexicon entries - lexeme element
        - contains:
            - one or more word notations - grapheme element
            - one or more word pronunciation - phoneme element
                - pronunciation may be written using IPA, SAMPA, etc

PLS - example

Figure: PLS pronunciation example

```
<?xml version="1.0" encoding="utf-8"?>
<lexicon version="1.0"
    xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
      http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
    alphabet="ipa" xml:lang="cs-CZ">
   <lexeme>
   <grapheme>CSR</grapheme>
   <phoneme>t  'e: 'es 'er</phoneme>
   <phoneme>t  'eska: r'epubl,ika</phoneme>
   </lexeme>
</lexicon>
```

Call Control XML

- Voice Browser Call Control eXtensible Markup Language
- Provides declarative markup to describe telephony call control
    - directing calls to corresponding application/human
    - merging multiple calls into a conference call
    - the ability to place outgoing calls
    - handling for a richer class of asynchronous events
    - handling the outside call queue for VoiceXML
    - etc.

State Chart XML

- W3C Recommendation (September 2015) of event-based state machine.
- General-purpose event-based state machine language.
- Based on:

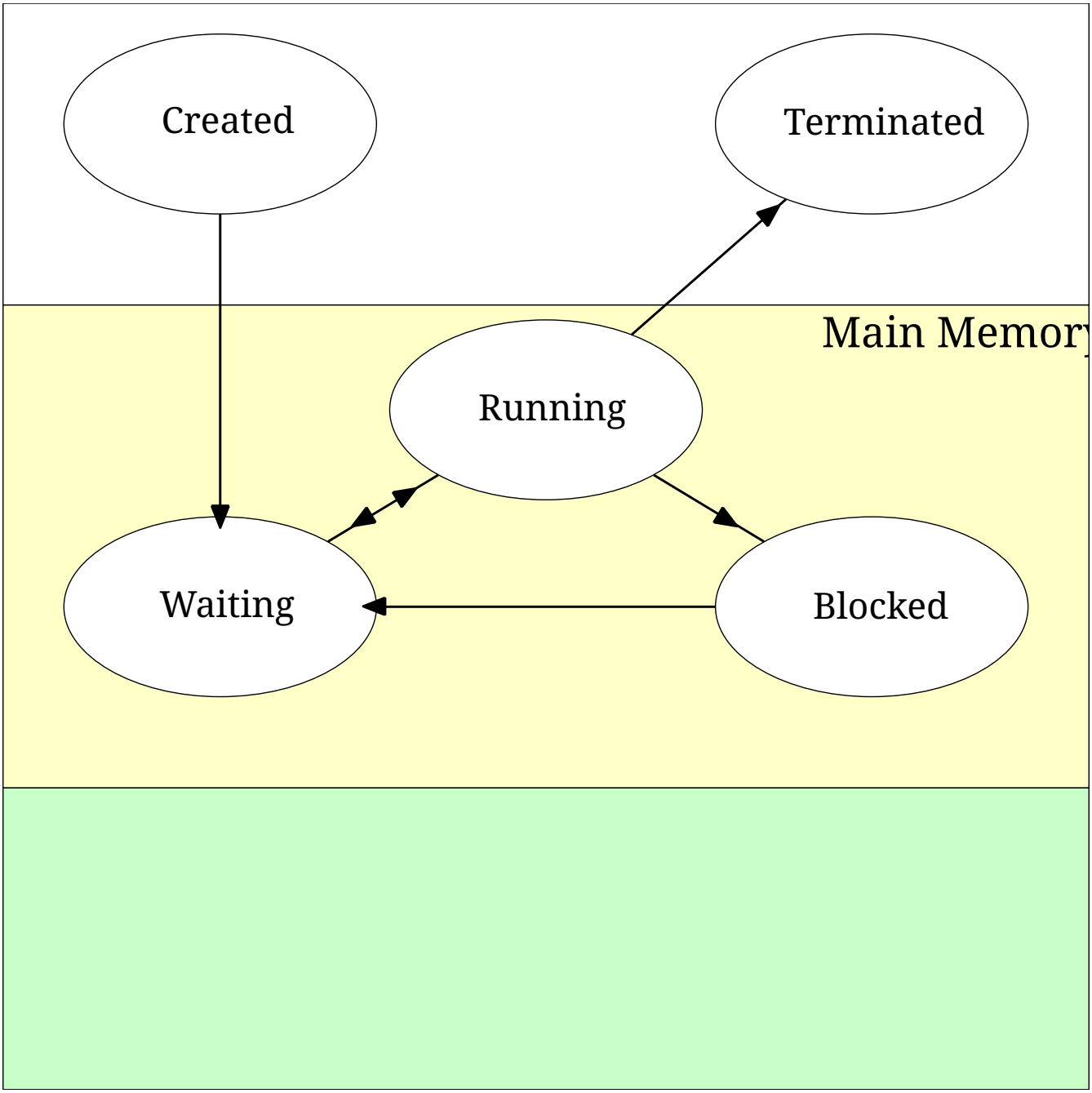
    - CCXML
    - Harel State Tables (included in UML for example)

State Chart XML - Relation to Dialogue

- Dialogue can be modeled using Mealy Automaton.
    - Mealy automaton - finite state automaton with an output function.
    - States of the automaton corresponds to the states of the dialogue.
    - Transition is function of the user input.
    - Output function is the dialogue system response.
- Mealy automaton can be described using the SCXML (see example)

SCXML - Demo

Example 1: Process planing demo



(if the image does not show, click here - Process state diagram)

SCXML - Demo

Example 1: Corresponding SCXML

```
<?xml version="1.0" encoding="UTF-8"?>
<scxml version="1.0" xmlns="http://www.w3.org/2005/07/scxml">
 <initial>
  <transition target="Created" type="external"/>
 </initial>
 <state id="Created">
  <transition target="Waiting" event="enqueue"/>
 </state>
 <state id="Waiting">
  <transition target="Running" event="assign"/>
 </state>
 <state id="Running">
  <transition target="Blocked" event="wait for resource"/>
  <transition target="Waiting" event="timeout"/>
  <transition target="Terminated" event="terminate"/>
 </state>
 <state id="Blocked">
  <transition target="Waiting" event="resource available"/>
 </state>
 <final id="Terminated"/>
</scxml>
```

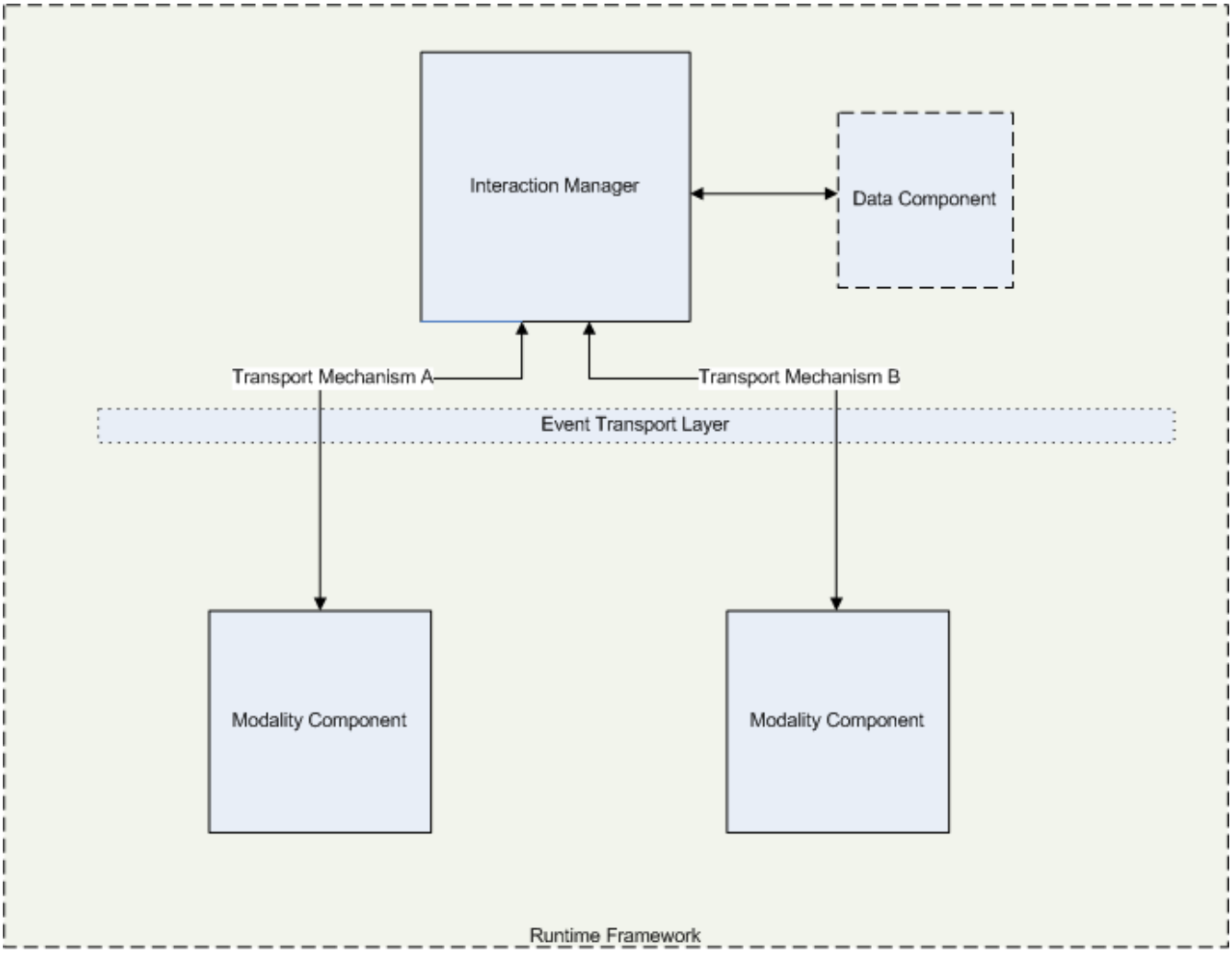
W3C Multimodal Interaction WG Standards

- MultiModal Interaction Working Group
- Used to create Multimodal dialogue interfaces
  - Multimodal interface allows comunition using multiple simultaneous channels
    - voice, graphics, text, emotions, etc.
- Standards:
  - Multimodal Architecture Specification
  - Extensible Multi-Modal Annotaions
  - InkML
  - Emotion Markup Language EmotionML

Multimodal Architecture Specification

- Specification describes:
  - architecture of multimodal user interfaces
  - interfaces between components of the UI
  - protocols for components communication.

Architecture diagram



(if the image does not show up, click here).

Extensible Multi-Modal Annotations (EMMA)

- The language can be used for:
  - data interchange between modality components
  - anotations of user inputs
  - provides metadata for user inputs.
- For details and examples see specification.

EMMA Demo

```
<emma:emma version="1.0"
    xmlns:emma="http://www.w3.org/2003/04/emma"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2003/04/emma
     http://www.w3.org/TR/2009/REC-emma-20090210/emma.xsd"
    xmlns="http://www.example.com/example">
  <emma:one-of id="r1" emma:start="1087995961542" emma:end="1087995963542"
     emma:medium="acoustic" emma:mode="voice">
    <emma:interpretation id="int1" emma:confidence="0.75"
    emma:tokens="flights from boston to denver">
      <origin>Boston</origin>
      <destination>Denver</destination>
    </emma:interpretation>
    <emma:interpretation id="int2" emma:confidence="0.68"
    emma:tokens="flights from austin to denver">
      <origin>Austin</origin>
      <destination>Denver</destination>
    </emma:interpretation>
  </emma:one-of>
</emma:emma>
```

InkML

- The language for representing inputs made by electoronic pen, stylus, touch-screens, etc.
- Not designed to represent semantic of the ink input.

InkML Demo - Input

[Ink Input demo] (if image does not show up, click here).

InkML Demo - Markup

```
<ink xmlns="http://www.w3.org/2003/InkML">
   <trace>
      10 0, 9 14, 8 28, 7 42, 6 56, 6 70, 8 84, 8 98, 8 112, 9 126, 10 140,
      13 154, 14 168, 17 182, 18 188, 23 174, 30 160, 38 147, 49 135,
      58 124, 72 121, 77 135, 80 149, 82 163, 84 177, 87 191, 93 205
   </trace>
   <trace>
      130 155, 144 159, 158 160, 170 154, 179 143, 179 129, 166 125,
      152 128, 140 136, 131 149, 126 163, 124 177, 128 190, 137 200,
      150 208, 163 210, 178 208, 192 201, 205 192, 214 180
   </trace>
   <trace>
      227 50, 226 64, 225 78, 227 92, 228 106, 228 120, 229 134,
      230 148, 234 162, 235 176, 238 190, 241 204
   </trace>
   <trace>
      282 45, 281 59, 284 73, 285 87, 287 101, 288 115, 290 129,
      291 143, 294 157, 294 171, 294 185, 296 199, 300 213
   </trace>
   <trace>
      366 130, 359 143, 354 157, 349 171, 352 185, 359 197,
      371 204, 385 205, 398 202, 408 191, 413 177, 413 163,
      405 150, 392 143, 378 141, 365 150
   </trace>
</ink>
```

See InkML Specification.

Emotion Markup Language (EmotionML)

- EmotionML main goals:
  - representation of emotions in multimodal interface to improve pragmatic derivation from inputs
    - emotions can change the pragmatic of the utterance
  - allow emotional anotations (manual/automatic) of input
  - allow emotional anotations of output that can be used by TTS for example.

Emotion Markup Language Demo

```
<emotionml version="1.0" xmlns="http://www.w3.org/2009/10/emotionml"
category-set="http://www.w3.org/TR/emotion-voc/xml#everyday-categories">
Hello and good afternoon.
<emotion><category name="angry"/>
What was that all about?
</emotion>
<emotion><category name="happy"/>
Nice to see you again!
</emotion>
<emotion><category name="sad"/>
Yeah I also had something else in mind than this.
</emotion>
<emotion><category name="content"/>
Well at least there is something nice to see.
</emotion>
<emotion dimension-set="http://www.w3.org/TR/emotion-voc/xml#pad-dimensions">
  I'm calm.
    <dimension name="arousal" value="0.3"/><!-- lower-than-average arousal -->
    <dimension name="pleasure" value="0.9"/><!-- very high positive valence -->
    <dimension name="dominance" value="0.8"/><!-- relatively high potency    -->
</emotion>
<emotion dimension-set="http://www.w3.org/TR/emotion-voc/xml#pad-dimensions">
  I'm nervous!
    <dimension name="arousal" value="0.9"/><!-- high arousal -->
    <dimension name="pleasure" value="0.2"/><!-- negative valence -->
    <dimension name="dominance" value="0.2"/><!-- low potency    -->
</emotion>
</emotionml>
```

► images/emoml.wav *(audio)*