

# Dialogové systémy

Luděk Bártek

Fakulta informatiky  
Masarykova univerzita

podzim 2019

# Obsah

- 1 PLS
- 2 CCXML
- 3 State Chart XML

# Pronunciation Lexicon Specification (PLS)

- Standard W3C
- Aktuální verze 1.0 (říjen 2008)
- Definuje značkování pro specifikaci slovníků výslovnosti pro podporu syntézy a rozpoznávání řeči.
- Specifikace na stránkách W3C Specifikace na stránkách W3C

# Základní elementy PLS

- Kořenový element - lexicon
  - atributy - xmlns - specifikace jmenného prostoru (<http://www.w3.org/2005/01/pronunciation-lexicon>)
  - xml:lang - jazyk dokumentu
  - version - verze dokument (1.0)
  - alphabet - abeceda použitá pro fonetický přepis
- lexeme - obsahuje popis pro jednu lexikální jednotku (slovo, zkratku, . . .
  - musí obsahovat aspoň jeden dceřiný element grapheme
- phoneme - obsahuje fonetický přepis dané lexikální jednotky (většinou se používá IPA).

# Ukázka lexikonu

```
<?xml version="1.0" encoding="utf-8"?>
<lexicon
  version="1.0"
  xmlns="..."
  alphabet="ipa"
  xml:lang="en-US">
  <lexeme>
    <grapheme>color</grapheme>
    <phoneme>
      k´ ʌə
    </phoneme>
  </lexeme>
</lexicon>
```

# Ukázka propojení lexikonu a gramatiky

- XML formát SRGS

```
<grammar xmlns="..." xml:lang="en" version="1.0">  
  <lexicon  
    uri="http://www.example.com/lexicon.file"/>  
  <lexicon  
    uri="http://www.example.com/strange-city-names.  
    type="media-type"/>  
  ...  
</grammar>
```

# Ukázka propojení lexikonu a gramatiky

- ABNF formát SRGS

```
#ABNF V1.0 ISO-8859-1;  
language en-US;  
lexicon <http://www.example.com/lexicon.file>;  
lexicon <http://www.example.com/strange-city-names  
      <media-type>;  
...
```

## Ukázka použití lexikonu v SSML

```
<speak version="1.1" xmlns="..." xml:lang="en-US">  
  <lexicon uri="lexicon.pls" xml:id="pls"/>  
  <lexicon uri="strange-words.file" xml:id="sw"  
    type="media-type"/>  
  <lookup ref="pls"> tokens here are looked up in  
    lexicon.pls  
    <lookup ref="sw"> tokens here are looked up first  
      strange-words.file and then, if not found, in  
      lexicon.pls  
    </lookup>  
  tokens here are looked up in lexicon.pls  
</lookup>  
tokens here are not looked up in lexicon documents  
</speak>
```



# Call Control XML

- Slouží k ovládání a řízení telefonních hovorů v průběhu interaktivních hlasových služeb.
- Může být doplňkem dialogových systémů na bázi VoiceXML, který poskytuje pokročilé telekomunikační funkce:
  - konferenční hovory
  - umožňuje přiřazení VoiceXML interpretrů jednotlivým příchozím hovorům,
  - ...
- Může být použit dialogovým systémem pro ovládání spojení uživatele a DS.
- Platformy, které CCXML implementují mohou použít libovolnou z definic ovládání telefonního spojení (např. JAIN Call Control, ECMA CSTA, ...)

# Architektura systému využívajícího CCXML

## CCXML System Architecture

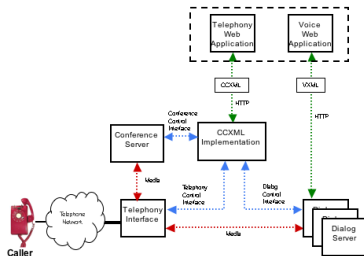


Figure: Architektura CCXML Aplikace (převzato ze specifikace CCXML)

# Struktura CCXML Aplikace

- Aplikace se skládá z dokumentů, které popisují ovládání a správu:
  - CCXML sezení (session) – zpracování CCXML dokumentu, který může být identifikován a odkazován.
  - Spojení – hovor nebo zdroj, který interaguje s hlasovým dialogem.
  - Konferenční objekt – slouží k sestavení a řízení konferenčních hovorů.
  - Dialog – může interagovat s různými spojeními a konferenčními objekty.
- Manipulace s těmito entitami je definována v CCXML, které umožňuje i zpracování asynchronních událostí.

# Životní cyklus CCXML aplikace

- Začátek
  - Příchozí telefonní hovor.
  - Zpracování elementu *createccxml* CCXML aplikací.
  - Požadavek na nové sezení z vnějšku.
- Pro vytvoření sezení je nutné znát:
  - URI počátečního CCXML dokumentu
  - parametry, které mohou ovlivnit získání tohoto dokumentu.

## Ukázka CCXML dokumentu

```
<?xml version="1.0" encoding="utf-8"?>
<ccxml version="1.0"
  xmlns="http://www.w3.org/2002/09/ccxml">
  <var name="state0" expr="'init'"/>
  <eventprocessor statevariable="state0">
    <transition state="init" event="connection.al
<accept/>
    </transition>
    <transition state="init"
      event="connection.connected">
      <log expr="'Prijate volani'"/>
      <dialogstart src="dialog.vxml"/>
      <assign name="state0" expr="'active'"/>
    </transition>
```

# Ukázka CCXML dokumentu

pokračování

```
<transition state="active" event="dialog.exit"  
  <log expr="'Ziskana hodnota: '+event$.values  
  <exit/>  
</transition>  
</eventprocessor>  
</ccxml>
```

## Použitá literatura

- Specifikace CCXML verze 1.0 z července 2011 na stránkách W3C

# SCXML

- State Chart XML – standard W3C Voice Browser Activity – září 2015
- Slouží k specifikaci konečných automatů (Mealyho automatů).
- Plánován jako jazyk pro zápis dialogové strategie ve VoiceXML 3.0.
- Principiálně vychází ze CCXML



# Konečný automat

- Konečný automat  $K = (S, \Sigma, \delta, q_0, Q_f)$ 
  - $S$  – konečná množina stavů
  - $\Sigma$  – konečná neprázdná množina vstupních symbolů (abeceda)
  - $\delta$  – přechodová funkce
    - deterministický automat –  $\delta : S \times \Sigma \rightarrow S$
    - nedeterministický automat –  $\delta : S \times \Sigma \rightarrow 2^S$
  - $q_0$  – počáteční stav
  - $Q_f$  – množina koncových stavů
- Mealyho automat  $M = (S, \Sigma, Y, \delta, \psi, q_0, Q_f)$ 
  - $Y$  – neprázdná výstupní abeceda
  - $\psi$  – výstupní funkce:  $S \times \Sigma \rightarrow Y$

# SCXML

## Základní elementy

- *scxml* – kořenový element, slouží jako kontejner pro elementy popisující stavy a přechody mezi nimi.
- *state* – stav konečného automatu
- *parallel* – definice paralelních dceřiných stavů
  - Jsou aktivní současně s rodičem.
- *transition* – definice přechodové funkce
- *initial*, *final* – definice počátečního/koncového stavu
- *onentry*, *onexit* – ošetření události při aktivaci/deaktivaci stavu.
- ...

# Ukázka zápisu konečného automatu v SCXML

```
<?xml version="1.0" encoding="utf-8"?>
  <scxml xmlns="http://www.w3.org/2005/07/scxml"
    xmlns:xi="http://www.w3.org/2001/XInclude"
    version="1.0"
    initial="Main"
    datamodel="ecmascript">
    <state id="main">
      <initial>
        <transition target="Start"/>
      </initial>
    </state>
  </scxml>
```

# Ukázka

## pokračování

```
<state id="Start">
  <onentry>
    <log>Vítejte v registraci IS MU</log>
  </onentry>
  <transfer event="AllData" target="process"/>
  <transfer event="course" target="conclusion"/>
  <transfer event="concl" target="courseState"/>
</state>
<final id="process"/>
```

# Ukázka

## pokračování

```
<state id="conclusion">
  <onentry><log>Způsob ukončení</log></onentry>
  <transition target="process"/>
</state>
<state id="courseState">
  <onentry><log>Předmět</log></onentry>
  <transition target="process"/>
</state>
</state>
</scxml>
```