

Standardy W3C Voice Browser Activity

Luděk Bártek

Fakulta informatiky
Masarykova univerzita
Brno

podzim 2023

Obsah

- 1 W3C Voice Browser Activity
- 2 VoiceXML
- 3 Speech Recognition Grammar Specification
- 4 Semantic Intepretation for Speech Recognition
- 5 PLS
- 6 Ostatní standardy W3C VoiceBrowser Activity

Úvod

- 1876 - udělen patent na telefon A. G. Bellovi
- WWW
 - 1989 - článek HyperText and CERN (Tim Burns Lee) koloval po CERNu k připomínkám
 - Vánoce 1990 - demonstrován řádkový webový prohlížeč a editor
 - 1991 - všeobecná dostupnost WWW na počítačích v CERNu
 - 1994 - první setkání W3 konsorcia (www.w3.org)

Historie

- W3C Voice Browser Working Group
 - založena 1999
 - cíl - návrh standardů umožňujících přístup k WWW pomocí hlasu a telefonu
 - zastoupeny firmy jako:
 - HP
 - Nuance Communications
 - Lucent Technologies
 - Motorola
 - ScanSoft
 - IBM
 - Tellme Networks
 - Vocalocity

Standardy W3C Voice Browser Activity

- VoiceXML
- Speech Recognition Grammar Specification (SRGS)
- Speech Synthesis Markup Language (SSML)
- Semantic Interpretation for Speech Recognition (SISR)
- Pronunciation Lexicon Specification (PLS)
 - slouží k popisu fonetických informací pro rozpoznávání a syntézu řeči
 - výslovnost zkratk, místních jmen, ...
- Call Control XML (CCXML)
- State Chart XML (SCXML)

Základní informace

- Jazyk pro popis dialogových rozhraní
- Cíl - přinést výhody webového vývoje a doručování obsahu do interaktivních hlasových aplikací
- vývoj započat 1995 - AT&T Phone Markup Language
- 1998 - konference hostovaná W3C na téma hlasového procházení WWW - předvedeny jazyky PML, VoxML, SpeechML, TalkML, VoiceHTML, ...
- 1999 - založeno VoiceXML Forum - spojení sil při vývoji jazyka pro značkování dialogů
- 2000 - VoiceXML 1.0, krátce na to přijato jako standard W3C
- Aktuální verze:
 - doporučení 2.1 (červen 2007)
 - pracovní návrh 3.0 (prosinec 2010) – zahrnuje podporu pro

Architektura VoiceXML aplikací

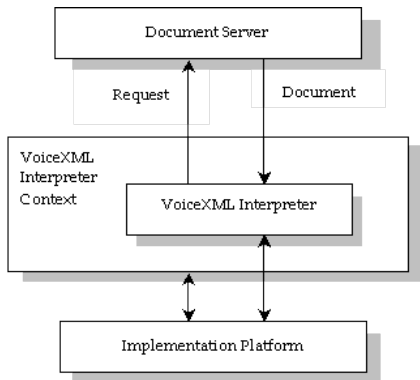


Figure: Převzato ze specifikace VoiceXML 2.0

Struktura VoiceXML aplikací

- VoiceXML dokument(y)
 - formuláře - konečně stavové automaty.
 - Uživatel se nachází v jednom z konverzačních stavů.
 - Přejechy definovány pomocí URI - odkazují na další krok dialogu.
 - URI - Uniform Resource Identifier
 - jednoznačná identifikace zdroje (souboru, obrázku, ...) na Internetu
 - rozšíření URL (URL je odkaz na soubor, cíl URI nemusí existovat).
 - Dialog končí, pokud tento přechod není definován.
- Dva druhy dialogů:
 - formuláře - definuje proces pro získání hodnot sady položek
 - menu - poskytuje uživateli sadu možností a odkazů na pokračování dialogu

Struktura VoiceXML aplikací

- Subdialogy
 - obdoba funkcí
 - slouží k opětovnému provádění jisté části dialogu a vrácení získaných hodnot.
- Sezení - začíná v okamžiku, kdy uživatel zahájí interakci s VoiceXML interpretrem a končí, když je komunikace ukončena buď uživatelem, VoiceXML dokumentem nebo kontextem dialogu.
- Aplikace - sada dokumentů, které sdílejí kořenový dokument

VoiceXML formulář

- Základní komponenta VoiceXML dokumentů.
- Obsahuje:
 - sadu položek
 - deklarace proměnných nepatřících položkám
 - ošetření událostí.
- Základní atribut - id
 - název formuláře
 - lze se pomocí něj na formulář odkazovat
 - musí být unikátní.

Řízení průběhu dialogů

Form Interpretation Algorithm - FIA

- Zpracování formuláře - FIA
 - 1 Výběr 1. nezadaného vstupního pole formuláře a jeho zpracování
 - 1 Výběr a přehrání výzev u tohoto vstupního pole (elementy prompt).
 - 2 Získání uživatelských odpovědí, které naplní jednu nebo více položek a nebo vyvolání a zpracování události(žádost o náповědu, nesrozumitelný vstup, nezadaný vstup, ...).
 - 3 Zpracování sekcí *filled* u všech zadaných položek.
 - 2 Pokud existuje nezadané vstupní pole pokračování krokem 1, jinak zpracování události *filled* u formuláře.

Ukázkový VoiceXML formulář

```
<vxml version="2.0" ...>
<form id="hello">
  <block name="hello">
    <prompt>Welcome to the VoiceXML!.</prompt>
  </block>
  <field name="greeting">
    <prompt>Hello.</prompt>
    <grammar root="greeting" src="greeting.grxml"/>
    <noinput>
      <prompt>Tell mi something nice, like hello, hi,
        good day.</prompt>
    </noinput>
  </field>
</form>
```

Ukázkový formulář

Pokračování

```
<nomatch>
  <prompt>I didn't understand you, but thanks
  anyway. </prompt>
  <exit/>
</nomatch>
<noinput count="2">
  <prompt>
    You don't want to speak to me. Good bye.
  </prompt>
  <exit/>
</noinput>
</field>
```

Ukázkový formulář

Dokončení

```
<filled>  
  <prompt>  
    You said <value expr="greeting"/>  
  </prompt>  
</filled>  
</form>  
</vxml>
```

Položky formuláře

- Vstupní položky
 - field
 - record
 - transfer
 - object
 - subdialog.
- Vstupním položkám odpovídají proměnné s názvem odpovídajícím hodnotě atributu name příslušné vstupní položky.
- Řídící položky
 - block
 - initial.
- Provádění položky lze omezit pomocí atributu cond.
 - pravdivostní výraz v jazyce ECMAScript s podmínku pro zpracování pole/provádění bloku.

Element field

- Představuje vstup od uživatele.
- Atributy:
 - name - jméno pole
 - přístup k výsledné hodnotě pomocí stínové proměnné s tímto jménem.
 - expr - případná počáteční hodnota, lze použít výrazy jazyka ECMAScript
 - cond - podmínka nutná pro zpracování vstupu
 - více viz specifikace.

Element field

Dokončení

- Obsah:
 - případná výzva s popisem vstupu (element prompt)
 - gramatika - popisuje množinu akceptovatelných vstupů
 - ošetření událostí
 - noinput
 - nomatch
 - filled
 - ...
 - ...

Ukázka použití elementu field

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/v
<form id="main">
  <field name="name">
    <prompt>Zadejte Vaše jméno</prompt>
    <grammar src="..." type="application/xml+srgs"/>
    <noinput>
      Zadejte prosím Vaše křestní jméno.
    </noinput>
    <nomatch>
      Je mi líto, ale zadané jméno není v kalendáři.
    </nomatch>
  </field>
```

Ukázka použití elementu field

Dokončení

```
<filled>  
  <submit next="URL aplikace" namelist="name"/>  
</filled>  
</form>  
</vxml>
```

Element record

- Umožňuje systému nahrát zprávu.
- Lze využít např. pro dialogový záznamník.
- Atributy:
 - name
 - expr
 - cond
 - beep - má-li se před začátkem nahrávání přehrát zvukový signál
 - maxtime - maximální délka nahrávky
 - type - mime-type výsledné nahrávky; musí být podporována VoiceXML platformou
 - ...
- Obsah:
 - případná výzva s popisem vstupu
 - ošetření událostí

Ukázka použití elementu record

```
<?xml version="1.0" encoding="utf-8"?>
<vxml version="2.0" xmlns="...">
  <form id="zaznamnik">
    <record name="zaznam" beep="true" maxtime="30s"
      type="audio/x-wav">
      <prompt> Bohužel zde nikdo není. Po zaznění
        signálu můžete zanechat vzkaz. </prompt>
      <noinput> Bohužel nic neslyším. Zkuste to znovu.
      </noinput>
      <catch event="connection.disconnect.hangup">
        <submit next="http://some.uri.cz/zaznamnik"/>
      </catch>
    </record>
```

Element subdialog

- Slouží k vyvolání dialogu, řešícího dílčí problém.
- Element subdialog.
- Jeden a tentýž subdialog lze volat opakovaně.
- Elementy:
 - subdialog - volání dílčího dialogu
 - param - definice hodnoty parametru
 - filled - kód, který se má provést po návratu z dílčího dialogu.
- Atributy
 - name - jméno volaného dílčího dialogu
 - src - URI dokumentu, který obsahuje kód dialogu.
- Kód subdialogu - formulář, ukončený elementem return.

Ukázka subdialogu

```
<?xml version="1.0" encoding="utf-8"?>
<vxml version="2.0" xmlns="...">
<form id="demo">
  <subdialog name="greeting" src="#say_hello">
    <param name="param1" expr="'ahoj'"/>
    <filled>
      <prompt> Hodnota subdialogu je
        <value expr="greeting.great"/></prompt>
    </filled>
  </subdialog>
  <filled>
    <prompt>Řekl jste <value expr="greeting.great"/>
    </prompt>
  </filled>
</form>
```

Ukázka subdialogu

```
<form id="say_hello">
  <var name="param1"/>
  <field name="great">
    <prompt><value expr="param1"/></prompt>
    <grammar root="pozdrav" src="pozdrav.grxml"/>
    <noinput count="2">
      <prompt>
        Na pozdrav jste mi neodpověděl. Nashledanou.
      </prompt>
    <return/>
  </noinput>
```


Ukázka subdialogu

Dokončení

```
<nomatch>
  <prompt>
    Bohužel jsem Vám nerozuměl, ale stejně dekuji.
    Nashledanou.
  </prompt>
  <return/>
</nomatch>
</field>
<filled>
  <return namelist="great"/>
</filled>
</form>
</grammar>
```

Element block

- Obsahuje proveditelný obsah.
- Provádí se pokud:
 - má hodnotu 'undefined' (nebyl dosud navštíven)
 - atribut cond se vyhodnotí jako true.
- Struktura - viz předchozí příklady.
- Atributy:
 - name - jméno bloku
 - expr - iniciální hodnota proměnné formuláře
 - cond - podmínka omezující provádění bloku.

Element initial

- Typické využití - dialogové strategie se smíšenou iniciativou.
- Umožňuje uživateli zadat více informací naráz.
- Na rozdíl od ostatních vstupních elementů nemůže obsahovat:
 - gramatiku - využívá se gramatika formuláře (viz ukázka na dalším slidu)
 - pokud je uživatelova odpověď gramatikou formuláře rozpoznána, je nutno nastavit hodnotu elementu initial - většinou se používá true
 - filled.
- Dceřiné elementy:
 - výzvy (prompt)
 - ošetření událostí (catch, nomatch, noinput).

Ukázka použití elementu initial

```
<?xml version="1.0" encoding="utf-8"?>
<vxml version="2.0" xmlns="...">
  <form id="main">
    <grammar src="registrace.grxml"
      type="application/srgs+xml"/>
    <block>
      <prompt>
        Vítejte v registraci předmětů na FI
      </prompt>
    </block>
```

Ukázka použití elementu initial

Pokračování

```
<initial name="mixed">
  <prompt>
    Zde můžete zadat, jaké předměty s jakým
    ukončením si chcete zaregistrovat
  </prompt>
  <noinput>
    Řekněte něco jako Předmět PB095 na zkoušku
  </noinput>
  <noinput count="2">
    Dobře zkusíme to postupně.
    <assign name="mixed" expr="true"/>
  </noinput>
  <reprompt/>
</initial>
```

Ukázka použití elementu initial

Pokračování

```
<nomatch count="2">
  Dobře zkusíme to postupně.
  <assign name="mixed" expr="true"/>
  <reprompt/>
</nomatch>
</initial>
<field name="predmet">
  <grammar src="registrace.grxml#predmet"/>
  <prompt>Zadejte kód předmětu</prompt>
  <nomatch>Zatím si lze zaregistrovat předměty
  PB162, PB095, PB125. </nomatch>
```

Ukázka použití elementu initial

Pokračování

```
<nomatch count="3">
  Bohužel se nám zadávání nedaří. Nashledanou.
  <exit />
</nomatch>
<noinput count="3">
  Jelikož jste nic nezadal, tak se s Vámi loučím.
  <exit />
</noinput>
</field>
<field name="ukonceni">
  <grammar src="registrace.grxml#ukonceni"/>
  <prompt>Zadejte požadované ukončení.</prompt>
```

Ukázka použití elementu initial

Pokračování

```
<nomatch>Předměty lze ukončit zkouškou  
nebo zápočtem.
```

```
</nomatch>
```

```
<nomatch count="3">
```

```
Bohužel se Vám zadávání nedaří, zkuste to  
klasicky na adrese is.muni.cz. Nashledanou.
```

```
</nomatch>
```

```
<noinput>Zadejte, zda chcete předmět ukončit  
zkouškou nebo zápočtem.
```

```
</noinput>
```

```
<noinput count="3"> ... </noinput>
```

```
</field>
```


Ukázka použití elementu initial

Dokončení

```
<filled>
  <prompt>
    Provádím registraci předmětu s kódem
    <value expr="predmet"/>
    a ukončením <value expr="ukonceni"/>.
  </prompt>
</filled>
</form>
</vxml>
```

Další informace o VoiceXML

- Popis na W3C Voice Browser Activity
(<http://www.w3.org/TR/voicexml20/> doplnění
změn na <http://www.w3.org/TR/voicexml21/>)
- www.voicexml.org (<http://www.voicexml.org>)
- Šimek, Richard - Tutoriál jazyka VoiceXML (bakalářská
práce FI), 2005
- ...

SRGS

- W3C specifikace jazyka pro zápis bezkontextových gramatik pro podporu rozpoznávání řeči.
- Aktuální verze 1.0 (březen 2004).
- Nahrazuje původně používaný standard JSGF – i nadále používán některými systémy pro rozpoznávání řeči.
- Dvě varianty zápisu gramatiky:
 - XML
 - Augmented Backus-Naur Form (ABNF).
- Liší se pouze zápis nikoliv vyjadřovací síla.
- Možnost použití způsobu zápisu závisí na použité platformě.
 - Větší podpora XML formátu SRGS.

Gramatiky (Teorie Formálních Jazyků)

- Gramatika $G = (N, \Sigma, P, S)$
 - N - množina neterminálních symbolů
 - Σ - abeceda
 - P - množina pravidel
 - S - kořenový neterminál.
- Bezkontextová gramatika
 - gramatika $G = (N, \Sigma, P, S)$
 - pravidla ve tvaru: $N \rightarrow \{N \cup \Sigma\}^*$.

XML Formát Gramatiky

- XML prolog.
- Kořenový element - grammar.
- Atributy:
 - root - pravidlo odpovídající kořenovému neterminálnímu symbolu
 - xml:lang - jazyk gramatiky
 - version - použitá verze SRGS (aktuálně pouze 1.0)
 - mode
 - dtmf
 - voice - implicitní hodnota
 - ...
- Element grammar - obsahuje množinu pravidel (elementů rule).

ABNF Gramatika

- ABNF hlavička

- #ABNF verze [*kódování*]

- ```
#ABNF 1.0 ISO-8859-2
```

- root \$ *jméno pravidla*; - kořenový neterminální symbol

- language *jazyk*;

- mode voice|dtmf;

- ```
#ABNF 1.0 UTF-8 root $pozdrav;
```

- ```
language cs-CZ;
```

- ```
mode voice;
```

Zápis pravidla

- Levá strana pravidla:

- XML formát
- element rule
 - atribut id - jednoznačný identifikátor pravidla
 - obsah - pravá strana pravidla

```
<rule id="pozdrav"> ahoj </rule>
```

- ABNF

- `<id_pravidla>`

```
$pozdrav = ahoj;
```

Zápis pravidla

- Pravá strana pravidla
 - může obsahovat terminální a neterminální symboly:
 - sekvenci
 - varianty
 - XML formát

- tělo elementu rule

```
<rule id="vstup">
  Proved <ruleref uri="#prikazy"/>
  s parametry <ruleref uri="#parametry"/>.
</rule>
```

- ABNF

- $\$ \langle \text{neterminál} \rangle = \langle \text{pravá strana} \rangle$

```
$vstup =
  Proved $<http://server/grammar.gram#prikazy>
  s parametry
  $<http://server/grammar.gram#parametry>
```

nebo

Sekvence

- Posloupnost terminálních a neterminálních symbolů.

- $X \rightarrow YZa$

- Lze ji rozdělit na logické části.

- XML zápis:

- zapsat přímo

```
<rule id="spojeni">  
  Chci jet z <ruleref uri="#misto"/>.  
</rule>
```

- dělení na logické části

- využitelnost

- počet opakování dané části (atribut repeat)
 - sémantická interpretace

Sekvence

Ukázky

- XML Formát:

```
<rule id="spojeni">
  Chci jet
  <item>z <ruleref uri="#misto"/> </item>
  <item> do <ruleref uri="#misto"/> </item>
  <item> <ruleref uri="#druh"/></item>
  <item> <ruleref uri="#datum"/></item>
  <item> v <ruleref uri="#cas"/></item>
</rule>
```

- ABNF zápis:

```
$spojeni =
  Chci jet z $misto do $misto $druh $datum
  v $cas
```

Varianty

- umožňují uživateli zadat jeden z možných vstupů
 - $X \rightarrow Y|Z|a$

- XML zápis:

```
<rule id="barvy">
  <one-of>
    <item>cervena</item>
    <item>zelena</item>
    <item>modra</item>
  </one-of>
</rule>
```

- ABNF zápis

```
$barvy = (cervena|zelena|modra)
```

Opakování

- Umožňuje specifikaci:
 - nepovinných částí promluvy
 - opakovaných částí promluvy
- XML zápis
 - pomocí atributu repeat u elementu item

```
<rule id="adresa">
  www
  <item repeat="1-2">
    tečka <ruleref uri="#castAdresy"/>
  </item>
  tečka <ruleref uri="#tld"/>
</rule>
```

Opakování

- ABNF zápis
 - za prvek uvedeme počet opakování uzavřený do $\langle \rangle$
 $\$adresa = www \$castAdresy \langle 1-2 \rangle \tld
 $\$castAdresy = tecka \$text$
- počet opakování
 - číslo - *číslo* krát
 - číslo1- číslo2 - *číslo1* - *číslo2* krát
 - číslo- - *číslo* - ∞ krát

Zvláštní pravidla

- GARBAGE - odpovídá libovolné promluvě až po následující blíže specifikovanou část
- VOID - pravidlo, které nelze vyslovit (zakázání určité promluvy)
- NULL - pravidlo, které je vždy rozpoznáno (může být i prázdné)
- XML formát: `<ruleref special="pravidlo"/ >`
- ABNF: `$pravidlo`

Ukázky

- XML formát:

```
<rule id="spojeni">  
  <ruleref special="GARBAGE"/>  
  z <ruleref uri="#misto"/>  
  do <ruleref uri="#misto"/>  
  <ruleref uri="#prostredek"/>  
</rule>
```

- ABNF formát:

```
$spojeni = $GARBAGE z $misto  
          do $misto $prostredek
```

Literatura

- Specifikace W3C.
- Příklady použité na přednášce (viz složka data ve studijních materiálech v ISu).

SISR

- Sémantika – přiřazuje význam slovům a promluvám.
- Sémantika v dialogových systémech – přiřazuje význam promluvám a jejich částem
- SISR - standard W3C pro zpracování sémantiky promluvy.
 - aktuální verze 1.0
 - publikován - duben 2007
 - úzce spjat se standardy
 - ECMAScript
 - SRGS
- Umožňuje přiřazení základních interpretací částem promluvy a vytváření odvozených interpretací pro nadřazená tvrzení
 - přiřazení interpretace částem promluvy
 - odvozování interpretace na základech dílčích interpretací
 - přiřazení interpretace vstupním polím dialogu

Přiřazení interpretace části promluvy

- Sémantická interpretace bývá součástí pravidla SRGS.
- K pravidlu přiřazena pomocí elementu/atributu tag.
- XML formát SRGS gramatiky:

- element tag

```
<item>  
  <ruleref uri="souhlas"/>  
  <tag>{out=' ano' }</tag>  
</item>
```

- atribut tag elementu item

```
<item tag="ano">jo</item>
```

- ABNF tvar:

- uveden za interpretovanou část promluvy

- tvar: {interpretace}

```
$souhlas = jo {ano}
```

Odvozování interpretace na základě dílčích interpretací

- Zápis pomocí výrazů v jazyce ECMAScript.
- Přiřazeno k pravidlům pomocí elementu tag.
- Interpretace reprezentována pomocí objektů jazyka ECMAScript.
- Stínové proměnné:
 - pro pravidla - objekt *rules*
 - výstup - objekt *out*

Odvozování interpretace na základě dílčích interpretací

XML formát SRGS gramatiky

```
<rule id="vlastnictvi">
  <item>Mám
    <item repeat="0-1">
      <ruleref uri="#barva"/>
    </item>
    <ruleref uri="prostredek"/>
  <tag>
    {out = rules.barva + ';' + rules.prostredek;}
  </tag>
</item>
</rule>
```

Odvozování interpretace na základě dílčích interpretací

ABNF gramatika

```
$vlastnictvi = mam $barva <0-1> $prostredok  
  {out = rules.barva + ';' + rules.prostredok;};  
$barva = (cervenou {cervena}  
  | cervene{cervena}  
  | zelenou{zelena}  
  | zelene{zelena});  
$prostredok = (auto{auto} | kolobezku{kolobezka});
```

Přiřazení interpretace vstupním polím VoiceXML formuláře

- Využívají se atributy stínového objektu out
- XML formát:

```
<rule id="vlastnictvi">
  <item>
    Mám <item repeat="0-1">
      <ruleref uri="#barva"/>
      <ruleref uri="#prostredek"/>
      <tag>
        { out.barva = rules.barva;
          out.prostredek = rules.prostredek; }
      </tag>
    </item>
  </rule>
```

Přiřazení interpretace vstupním polím VoiceXML formuláře

- ABNF gramatika:

```
$vlastnictvi = mam $barva <0-1> $prostredek
{
  out.barva = rules.barva;
  out.prostredek = rules.prostredek;
};
$barva = (cervenou {cervena}| cervene{cervena}|
          zelenou{zelena}| zelene{zelena});
$prostredek = (auto{auto}
              | kolobezku{kolobezka});
```

Literatura

- Specifikace SISR
- ECMAScript
- Příklady použité na přednášce.

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
    </phonem>
  </lexeme>
</lexicon>
```

Ukázka propojení lexikonu a gramatiky

- XML formát SRGS

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

Ukázka propojení lexikonu a gramatiky

- ABNF formát SRGS

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

Ukázka použití lexikonu v SSML

```
<speech 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 in 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  
</lookup>
```

Další standardy W3C VoiceBrowser Activity

- SSML – viz přednáška o syntéze řeči
- Call Control XML – bude probrán příští týden
- State Chart XML – bude probrán příští týden