Chapter

Speech Standards: Lessons Learnt

Paolo Baggia

Abstract

During the past decades, the landscape of speech and DTMF applications has changed from being based on proprietary platforms to being completely based on speech standards. The W3C Voice Browser Working Group played a primary goal in this change. This chapter describes that change, highlights the standards created by the W3C VBWG, and discusses the benefits that these standards have provided in many other application fields, including multi-modal interfaces.

Keywords: human-computer interaction, speech recognition, speech synthesis, pronunciation lexicons, natural language, spoken dialog, speech grammars, multi-modal interaction

1. Introduction

This chapter is a retrospective of a very special moment for speech and dialog technologies. Since the end of the last century, the use of speech and dialog technologies has been limited by proprietary implementation of platforms with strong legacies, as well as limitations on the rapid adoption of the core technology advances. But at the turn of the century, a sudden change occurred, and speech applications started to be deployed in many commercial applications, becoming ubiquitous in a matter of years. That trend has continued to this day. This transformation was catalyzed by the creation and adoption of a family of standard languages. This evolution was quickly accepted and adopted by the industry, even before these languages were completely defined. In the meantime, research was constantly increasing performance, which also fueled the widespread deployment of speech applications.

In the early 2000s, a sharp increase in the number of speech applications occurred in many areas, including customer care, finance, travel, and many other sectors. This increase is continuing, with the diffusion of virtual assistants, automatic chatbots, speech presence in smartphones, in the car, and at home. Speaking to an appliance is now an everyday activity, while it was a dream limited to sci-fi movies only a few decades earlier. That dream is a reality today and industrial standards have played a significant role in this achievement.

The ecosystem created by speech standards has been active for more than 20 years, and it is in the core of the major players, even though there are recent trends to move from in-house technologies to hosting and to access speech resources by Web APIs. In these new developments, speech standards can still play a role to provide customizations to hosted resources.

The success of this enterprise was made possible by a highly collaborative work among a large group of people, from academia to industries and even individual contributors. The hope is that this example will inspire new developments in the future, and research and industry will be ready to create a new open ecosystem.

2. Why and when?

At the beginning of this century, the time was ready for a change of paradigm in the way speech technologies were deployed.

In the previous decade, research had been constantly improving the accuracy and powerfulness of speech technologies. For instance:

- Automatic speech recognition (ASR) moved from very limited tasks, such as digit recognition, to large vocabulary continuous speech recognition (LVCSR) by the adoption of statistical models (dynamic programming, hidden Markov models, statistical language models, etc.) The accuracy improvement was accelerated by government-sponsored competitions among the leading research labs and companies. These included DARPA funded projects such as the Airline Travel Information System (ATIS) [1–3], a speech understanding challenge focused on data collection of spoken flight requests, and the Wall Street Journal Continuous Speech Recognition Corpus [4], attempting to recognize speech from read WSJ articles.

- Speech synthesis and text-to-speech (TTS) during the 1980s reached the goal of high intelligibility and flexibility with a parametric approach [5], but the automatic voices were still robotic. A new technique, Concatenative Unit Selection [6], was less flexible, but capable of a more natural rendering and it was generally adopted by the industry.

- Spoken dialog systems (SDS) research was initially promoted by EU-funded projects, such as SUNDIAL [7], RAILTEL [8], and its continuation ARISE [9]. The results achieved in those projects were very promising to the point that the Italian Railways company (Ferrovie dello Stato, now Trenitalia) decided to deploy the prototype developed within the ARISE project with the help of Telecom Italia Labs (TILAB). The resulting phone service, known as FS_Informa, enabled customers to request train timetables over the phone. For a review of the state-of-the-art on Human Language Technology at that time, see [10], while for a comprehensive and accessible view of speech technologies, see [11].

Speech technologies were ready for commercial deployments, but there were many obstacles along the way. One major obstacle was that each technology company had its own proprietary APIs, to be integrated in a proprietary IVR platform. This slowed down the delivery of the latest technology advances because of the platform provider resistance to changing their proprietary environments. Also, customers were locked in on individual vendor's proprietary legacies.

Another important factor was the contemporaneous evolution of the Web infrastructure spearheaded by the W3C Consortium, led by Tim Berners-Lee, the Web's inventor. W3C, the World Wide Web Consortium is an international community whose mission is to drive the Web to its full potential by developing protocols and guidelines that ensure its long-term growth. This inspired researchers to consider whether a Web-based architecture could accelerate the evolution of speech applications. This was the idea behind a seminal event, a W3C Workshop held in

Cambridge (MA) on October 13, 1998 [12], promoted by Dave Raggett of the W3C and Dr. James A. Larson of Intel. The workshop was named: "Voice Browsers," as an event to discuss different innovative ideas on how to solve the proprietary issues by adopting the latest advances offered by the Web infrastructure. The workshop catalyzed the interest of research labs, companies, and start-ups, and it culminated in the creation of the W3C Voice Browser Working Group (VBWG) [13] chaired by Jim Larson and Scott McGlashan of PipeBeach (later Hewlett-Packard). Inside the W3C VBWG, a subgroup was devoted to study the expansion of the ideas in a multi-modal environment, and after a few years, it spun off a second group: the W3C Multi-Modal Interaction Working Group (MMIWG) [14], chaired by Dr. Deborah Dahl of Unisys (later Conversational Technologies).

The goal of the VBWG was to create a family of interoperating standards, while the MMIWG had the role to re-use those new standards in multi-modal applications, where other modalities were active in addition to voice for input and output (visual, haptic, etc.).

Figure 1 shows the initial diagram proposed by Jim Larson and named: "Speech Interaction Framework" (see the original diagram in Section 4 of [15], see also [16]). It remained the reference point for the development of all the standard languages created along the years.

The solid boxes are the modules of a reference spoken dialog architecture centered around the dialog manager, which is connected with an external telephony system and the Web. This shows the attempt to align the Web along with the main communication channel of the time. In this framework, there are input modules such as the ASR (automatic speech recognition) engine and a touch-tone (DTMF) recognizer. Additional modules include language understanding and context interpretation, but they were not considered to be priorities at that time. TTS (text-to-speech) engine, pre-recorded audio player, language generation, and media planning are considered output modules. After a considerable work by the W3C VBWG, the modules colored in red became completely driven by W3C Recommendations (the dashed red bordered boxes).

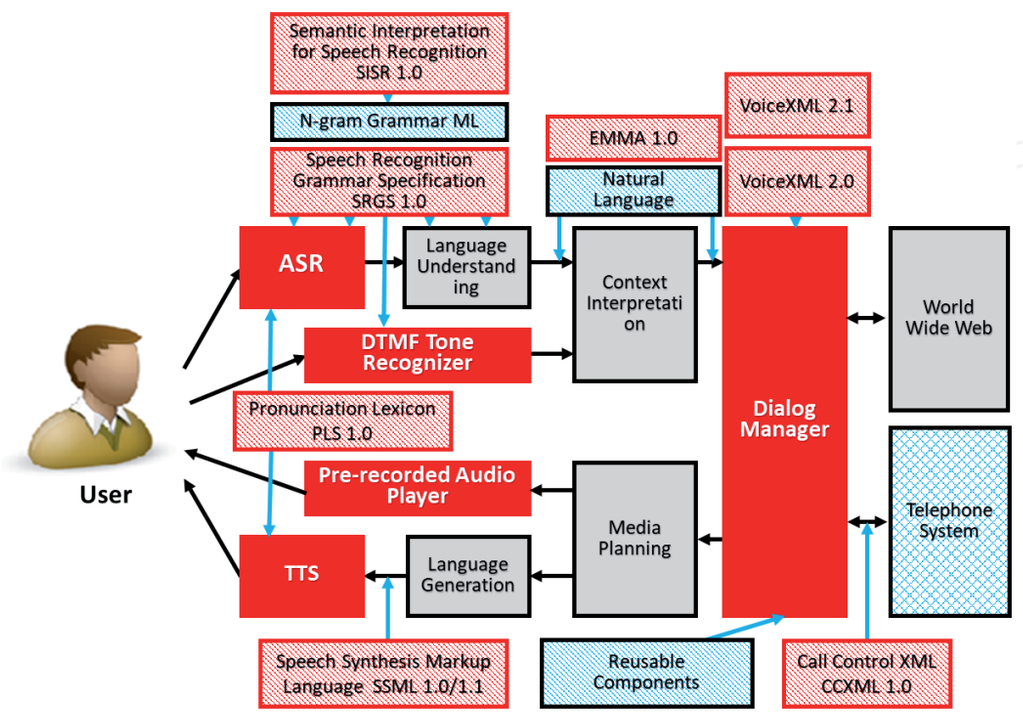

Figure 1.
Speech interaction framework.

From its creation, W3C VBWG started to attract all the companies and labs active in that space. The companies included speech technology providers (at that time L&H, Philips, Nuance, SpeechWorks, Loquendo, and Entropic), research labs (MIT, Rutgers, AT&T Bell Labs, and CSELT/TILAB), large telcos (Lucent, AT&T, BT, Deutsche Telekom, France Telecom, Telecom Italia, Motorola, and Nokia), large players (Microsoft, HP, Intel, IBM, and Unisys), and IVR vendors (Avaya, Genesys, Comverse, and CISCO). In addition, newly created companies such as voice platform providers (PipeBeach, Voxpilot, Vocalocity, VoiceGenie, and Voxeo), voice application host (HeyAnita, BeVocal, and Tellme), and many more joined the effort.

One of the first actions of the W3C VBWG was to acknowledge the contribution of the VoiceXML Forum [17] (founded by AT&T, Lucent, Motorola, and IBM) of a new markup language called VoiceXML 1.0 [18] of their design. From this point on, the W3C VBWG focused on completing VoiceXML with additional features. However, a wise decision was made to create a family of interoperable standards instead of a monolithic language. These standard languages are those described in Section 3. At the same time, the VoiceXML Forum took on a complementary role in the evolution of the VoiceXML ecosystem. It focused on education, evangelization, and support of the adoption of this family of standards. Among the major achievements of the VoiceXML Forum are the following two programs:

- The Platform Certification Program to allow platform developers to thoroughly test and certify that their platforms support all standard features. The first certification program was limited to VoiceXML 2.0 with a large adoption of 26 platforms certified. It was then extended to also certify VoiceXML 2.1, SRGS 1.0, and SSML 1.0 with eight more platforms certified.

- The Developers Certification Program to allow developers to certify their competence in the VoiceXML architecture and in the correlated standards.

All the materials produced are still available in the VoiceXML Forum Web site [17].

3. W3C VBWG standards

The W3C VBWG, supported by the VoiceXML Forum, accelerated a cooperative effort to create the foundations of a new generation of voice applications based on public standards. In a short time, an incredible sequence of Working Drafts was published, demonstrating the energy and creativity underlying the development of the voice standards.

In March 2004, after less than 4 years from the start of VBWG, the first group of complete standards, known as W3C Recommendations, was released. It includes VoiceXML 2.0 [19] for authoring voice applications; SRGS 1.0 [20] for defining the syntax of speech grammars; and SSML 1.0 [21] for controlling speech synthesis (or text-to-speech, TTS). A few years later, in April/June 2007, a second round of W3C Recommendations was released, which includes VoiceXML 2.1 [22], which completes VoiceXML 2.0 with a limited number of new features; and SISR 1.0 [23], which standardizes the creation of a meaning representation from a SRGS 1.0 speech grammar.

The work continued in the following years. SSML 1.0 was revised to version 1.1 [24] to improve the internationalization of speech synthesis in other regions of the world, including India and Eastern Asia, and PLS 1.0 [25], which supports the description of pronunciation lexicons, a shared resource for both SRGS 1.0 and SSML 1.0/1.1 resources. Finally, CCXML 1.0 [26] was released as a real-time

language to implement telephony and VoIP call control in a voice browser platform, while SCXML 1.0 [27] as a general-purpose event-based state machine language that can be used for defining the dialog manager, and other components of a speech system. A comprehensive introduction to SCXML 1.0 is available in [28]. In the rest of this section, these languages will be briefly introduced.

3.1 Dialog management: VoiceXML 2.0/2.1

The Voice Extensible Markup Language (VoiceXML), version 2.0 [19], standard was the center of the innovation. Its key features are as follows:

- It is an XML declarative language.
- It is easy to author, the motto was: "Simple things must be easy and complex things must be possible!"[1]
- It assumes the existence of the Web architecture.

All these features carry clear advantages. An XML language allows a clean syntax checked by DTD/Schema, extensibility by namespaces, and encodings, generally available with any XML processors (user agent). The second feature, simplicity and flexibility, allows to edit VoiceXML 2.0 as text editor then upload it as a static page or generate it dynamically by Web applications (like all of the Web sites today). Finally, to be within the Web architecture means to share an enormous background of tools and techniques and it is part of the mainstream of the current technology evolution.

From a functional point of view, VoiceXML 2.0 allows the creation of speech applications that can replace menu-based, DTMF, and pre-recorded messages by a voice-driven interaction where the messaging is synthesized speech. This was the main reason why all the major IVR platforms quickly adopted VoiceXML, enabling taking advantage of a more powerful application environment. The second reason was the need to open the world of IVR applications to new players, instead of relying on proprietary solutions. Not only does VoiceXML 2.0 allow platforms to take advantage of the latest advances in ASR and TTS engines but also allow them to continue implementing traditional menu-based DTFM applications. Consequently, with VoiceXML, a complete replacement of the previous generation of IVRs became possible. More recently, VoiceXML 2.1 [22] further extended the language with additional features mostly devoted to creating more dynamic applications. This was a general trend in the evolution of the Web, as well as in the evolution in VoiceXML.

Figure 2 shows a simplified VoiceXML 2.0 document that implements a dialog to request departure and arrival airports from a user. The dialog tries first to recognize both the locations in a single utterance; if that fails, it asks them again in sequence. A final confirmation is given before transitioning to another page of the application. This is called a mixed-initiative dialog, where a user has a certain degree of freedom in expressing requests. For a detailed introduction of VoiceXML, see [29–31].

3.2 Speech recognition: SRGS 1.0 and SISR 1.0

Two standards were created by the W3C VBWG to define the knowledge resources for ASR engine: speech grammars and semantic interpretation. The first one is the formal definition of a speech grammar described in the W3C Recommendation "Speech

[1] The original quote is from Alan Key.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
  xml:lang="en-GB">
<form id="dep_arr_airports">
 <grammar src="dep_arr.grxml"
         type="application/srgs+xml"/>
 <initial name="start">
   <prompt>
         What are the arrival and departure airports?
   </prompt>
 </initial>
 <field name="fromcity">
   <prompt>Tell me the departure airport.</prompt>
 </field>
 <field name="tocity">
   <prompt>Tell me the arrival airport.</prompt>
 </field>
 <field name="go_ahead" type="boolean" modal="true">
   <prompt>Do you want to leave from
    <value expr="fromcity"/> and arrive
    to <value expr="tocity"/>?
   </prompt>
   <filled>
     <if cond="go_ahead">
      <submit next="/servlet/dep_date"
             namelist="fromcity tocity"/>
     </if>
     <clear namelist="fromcity tocity go_ahead"/>
   </filled>
 </field>
</form>
</vxml>
```

Figure 2.
A simplified VoiceXML document.

Recognition Grammar Specification Version 1.0" SRGS 1.0 [20]. Speech grammars and statistical language models (SLMs) are the two common ways to provide constraints to the speech recognition process. A grammar is a formal definition of all the sentences that can be spoken. The grammar drives the ASR engine to find the closest match with the acoustic signal. A grammar is a strong constraint for the ASR and is relatively simple to implement. Statistical LMs, typically used in speech-to-text systems where the user in not specifically prompted, are in contrast weaker constraints characterized by the probability of a word to be spoken in the context of the preceding words (known as n-gram probabilities). The W3C VBWG standardization effort focused on the speech grammar only, because it was useful for simpler recognition tasks, but also because the other formats, driven by research, were commonly used for n-grams[2]. A proposal for an SLM standard in the VBWG is described in [32].

SRGS supports the definition of grammars for speech as well as for DTMF inputs. A grammar can be specified in two equivalent formats, an XML document, called GrXML and a more traditional textual format, called ABNF, the acronym for augmented Backus-Naur format (commonly used to describe the syntax of a programming language). The W3C SRGS 1.0 Recommendation very clearly defines those two equivalent formats and offers a great number of examples (see [20]).

[2] For instance, the well-known MIT ARPA LM format, see http://www.seas.ucla.edu/spapl/weichu/htkbook/node243_mn.html

The SRGS 1.0 specification was immediately adopted by all speech recognition engines, allowing them to interoperate within a VoiceXML platform. Of the two formats, GrXML became the predominant one, but it is very easy to transform a grammar from one format to the other.

Figure 3 shows an excerpt of a SRGS 1.0 grammar, in GrXML format, with the goal to recognize utterances like: "from Rome to Paris," where the list of cities might be extended to a longer list.

The part of the grammar devoted to the generation of a meaning representation or semantic interpretation is indicated with blue characters. This is the domain of the second speech grammar standard produced by W3C VBWG "Semantic Interpretation for Speech Recognition Specification Version 1.0" SISR 1.0 [23].

Semantic results are encapsulated in each rule by means of <tag> elements, which contain snippets of the programming language ECMAScript [33], widely known in its Web variety as JavaScript. The W3C SISR 1.0 Recommendation prescribes the use of the Compact Profile ECMA-327, which is a constrained version of ECMAScript. The goal was to gain computational efficiency to enable more compact speech recognition engine processing.

In SISR 1.0, each SRGS 1.0 rule, like "city" in **Figure 3**, contains a predefined variable called "out" whose properties are assigned within the <tag> elements. The content of the "out" variable of the most external rule, called the "root" rule, is returned from the recognition engine to the application environment.

For the input utterance "from Rome to Paris," for example, the SRGS grammar in **Figure 3** will return the ECMAScript object:

{fromcity: "FCO", tocity: "CDG"}

This is the case for simple and focused grammars where the result is just one or a few values. However, SISR supports also conditional logic and algorithms. This would be useful for instance to validate a checksum in a complex numeric (i.e., credit card numbers) or alphanumeric strings (as the personal taxation ID in Italy). That would allow the recognizer to validate and possibly reject a wrong result before returning it to the application and at the same time to increase the confidence of alternative, and possibly correct, recognition result.

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar version="1.0" xml:lang="en-GB"
  xmlns="http://www.w3.org/2001/06/grammar"
  tag-format="semantics/1.0" root="fromto">

<rule id="fromto" scope="public">
   from <ruleref uri="#city"/>
   <tag>out.fromcity=rules.latest();</tag>
    to <ruleref uri="#city"/>
   <tag>out.tocity= rules.latest();</tag>
</rule>

<rule id="city">
     <one-of>
      <item>London<tag>out="LHR"</tag></item>
      <item>Paris<tag>out="CDG"</tag></item>
      <item>Rome<tag>out="FCO"</tag></item>
     </one-of>
</rule>
</grammar>
```

Figure 3.
Simple SRGS grammar with SISR script.

3.3 Speech synthesis: SSML 1.0 and 1.1

Another effort was to define how to control a speech synthesis, or TTS engine. This is to help the engine render the textual prompt in the most accurate way. The XML markup language for this purpose is the Speech Synthesis Markup Language Version 1.0, SSML 1.0 [20], which was released in March 2004.

Figure 4 shows the five major processing steps present in all TTS engines. For each of them, the engine offers a normal behavior, called "non-markup behavior" in the picture. The SSML mark-up instead allows the engine to improve the default rendering by means of elements of the language. Each element is related to one specific processing step, and it is interpreted as a request by the author to perform some specific processing. It is then up to the processor to determine whether and in what way to realize the command.

The SSML example in **Figure 5** shows a prompt for a flight information system structured into a single paragraph (<p > element) and two sentences (<s > elements). Acronyms are substituted (<sub>) into expanded versions, pauses are added (<break>), and a time expression is explicitly labeled (<say-as>) to select the correct way of reading it. Other elements can change additional features, such as prosodic features of speed and rate (<prosody>), and how to change the speaking voice (<voice>).

SSML 1.0 [21] continued to be standardized to promote the use of SSML to more international languages, in particular Asian and Indian languages. Three workshops were held to encourage local companies and universities to propose features to be added to the language:

- Nov 2005 at Beijing (China)
- May 2006 at Crete (Greece)
- Jun 2007 at Hyderabad

A new standard SSML 1.1 [24] was released in September 2010. See Appendix F of [24] for details on the changes. Among them, a < token> element was introduced for languages where the whitespace has peculiar behavior, such as in Mandarin, Japanese, Thai, Vietnamese, and Urdu.

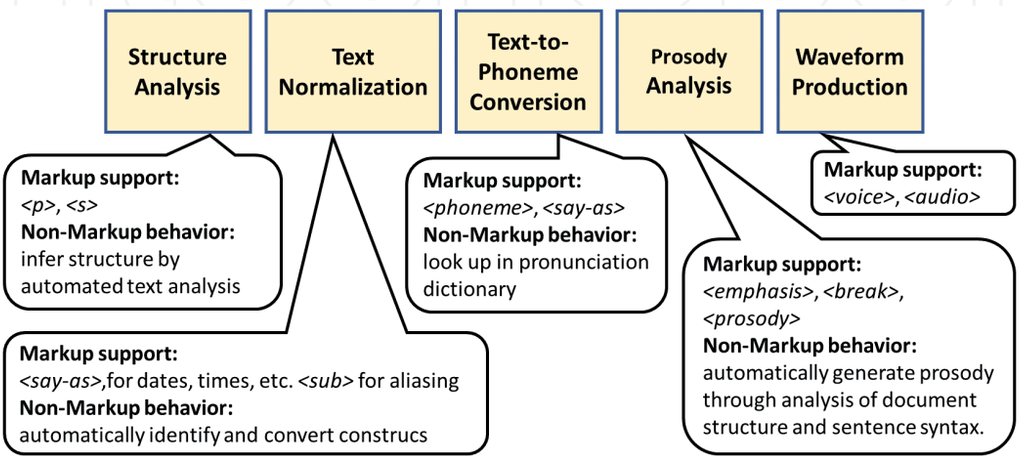

Figure 4.
SSML support for stages of speech synthesis.

```
<?xml version="1.0" encoding="UTF-8"?>
<speak version="1.0"
   xmlns="http://www.w3.org/2001/10/synthesis"
   xml:lang="en-GB">
   <p>The requested flight leaving from
         <s xml:lang="it-IT">
             <sub alias=”Roma Fiumicino”>FCO</sub></s>
         airport
         <emphasis>with destination
               <sub alias=”London Heathrow”>LHR</sub>
          </emphasis>
       are: <break time="1s"/>
       <s>
         <sub alias="British Airways 0 3 0 2">BA0302</sub>
         <break time="0.5s"/>
         leaving at
              <say-as interpret-as=”time”>3:45pm</say-as>
         <break time="0.5s"/>
         from gate number A63.
       </s>
       <!-- Other flight options -->
   </p>
</speak>
```

Figure 5.
A simple SSML document.

3.4 Pronunciation lexicon: PLS 1.0

Both speech grammars and synthesized prompts can require customizing the pronunciation of words in a specific application domain. This is often done by adding a user lexicon. The Pronunciation Lexicon Specification (PLS 1.0 [25]) was created to support the definition of a standard lexicon fully interoperable with SRGS 1.0 and SSML 1.0/1.1. PLS 1.0 became a W3C Recommendation in October 2008.

A PLS document is a container of entries, <lexeme> elements, with a textual part described by the <grapheme> element and with textual replacements provided by <alias> elements or phonetic transcriptions by <phoneme> elements. There can be multiple pronunciations to accommodate different ways to speak a word/token, or for a different spelling for the same pronunciation.

A simple PLS 1.0 document example for a flight application is shown in **Figure 6**. For “Alitalia” and “Lufthansa,” the pronunciations inside the <phoneme> element are given in IPA (International Phonetic Alphabet) [34]—a standard way to express the pronunciations for all spoken human languages. Moreover, the two lexemes have a double pronunciation; the first is the normal English one, while the second is closer to their original language (Italian and German, respectively) as spoken by a native speaker of that language. The prefer attribute indicates which pronunciation has to be selected for TTS rendering. For ASR, all the pronunciations will be taken into account.

3.5 Call control—CCXML 1.0

Another language defined by the W3C VBWG targets programming the call control of a voice browser in an innovative way. An XML markup language was developed to define handlers for telephony events generated by a telephone connection or a VoIP SIP interaction. The Voice Browser Call Control (CCXML 1.0) [26] language was designed to allow a very efficient implementation completely based upon events and handlers to avoid creating any latency that might impact the underlying signaling.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
   xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
   alphabet="ipa" xml:lang="en-GB">
   <lexeme>
     <grapheme>Alitalia</grapheme>
     <phoneme>ˌæl.ɪ.ˈtæi̯.ə</phoneme>
     <phoneme prefer="true">a.li.ˈtaɪ.lja</phoneme>
   </lexeme>
   <lexeme>
     <grapheme>Lufthansa</grapheme>
     <phoneme>ˈlʊft.hænzə</phoneme>
     <phoneme prefer="true">ˈlʊftˌhan.za</phoneme>
   </lexeme>
   <lexeme>
     <grapheme>AF</grapheme>
     <alias>Air France</alias>
   </lexeme>
     <lexeme>
     <grapheme>BA</grapheme>
     <alias>British Airways</alias>
     </lexeme>
</lexicon>
```

Figure 6.
PLS 1.0 document for flight applications.

A CCXML engine is also able to send and receive events through an HTTP/HTTPS connector, which allows for the generation of outbound calls from a Web application and for monitoring calls and conferences via a Web interface.

CCXML 1.0 addresses both simple tasks of call handling (see **Figure 7**), as well as complex ones, such as conditional call handling, conferencing, coaching, etc. Each CCXML document describes transitions to handle specific events. In **Figure 7**, a "connection.alerting" event (incoming call) is accepted by the underlying telephony or VoIP layer, a VoiceXML dialog is started when the "connection.connected" event is received, and then the CCXML processor waits until either the caller disconnects ("connection.disconnect") or the VoiceXML dialog exits ("dialog.exit"). These are simple actions performed during telephony calls, both TDM and VoIP.

While working on the definition of CCXML 1.0, which became a W3C Recommendation in July 2011, the W3C VBWG decided to start another effort to define a state-chart language to generalize the ideas behind CCXML 1.0. This new specification is State Chart XML (SCXML): State Machine Notation for Control Abstraction (SCXML 1.0 [26]), and it can be used as the key component to control a generalized interaction in a multimodal interface. SCXML 1.0 is an XML markup language that provides a generic state-machine-based execution environment inspired by Harel state charts [35].

3.6 IETF protocols: MRCPv1 and v2

The implementation of voice browsing relies on other standards and protocols, the web architecture, with XML documents, namespaces, caching policies to start with, and obviously the HTTP/HTTPS protocols. All these are at the core of the W3C VBWG standards. However, the Internet Task Force Initiative (IETF) [36] was working on needed protocols.

The Media Resource Control Protocol (MRCP), whose initial draft was proposed by CISCO, SpeechWorks, and Nuance, defines the requests, responses, and events to

```
<?xml version="1.0" encoding="UTF-8"?>
<ccxml version="1.0"
   xmlns="http://www.w3.org/2002/09/ccxml">
   <var name="currentState"/>
   <var name="myDialogId"/>
   <var name="myConnId"/>
   <eventprocessor statevariable="currentState">
     <transition event="connection.alerting">
       <assign name="myConnId" expr="event$.connectionid"/>
       <accept connectionid="event$.connectionid"/>
     </transition>
     <transition event="connection.connected">
       <dialogstart
          src="'http://www.example.com/flight.vxml'"
          connectionid="myConnId" dialogid="myDialogId"/>
     </transition>
         <transition event="dialog.started">
         <log expr="'VoiceXML appl is running now'"/>
         </transition>
         <transition event="connection.disconnected">
         <dialogterminate dialogid="myDialogId"/>
         </transition>
         <transition event="dialog.exit">
       <disconnect connectionid="myConnId"/>
         </transition>
         <transition event="*">
         <log expr="'Closing, unexpected:' + event$.name"/>
       <exit/>
     </transition>
   <eventprocessor/>
</ccxml>
```

Figure 7.
Basic handling of incoming calls with CCXML.

control resources of general speech engines, such as ASR and TTS and even speaker verification to enable a distributed and scalable architecture. The initial draft was standardized by IETF as MRCPv1 (RFC 4463 [37]), and it was largely implemented by the industry. The protocol was based on Real-Time Transport Protocol (RTP) for media transport and RTSP (Real Time Streaming Protocol) for controlling speech resources.

In the meantime, standardization continued to MRCPv2, which was instead based on SIP (Session Initiation Protocol) for signaling and SDP (Session Description Protocol) for negotiating and exchanging capabilities. In November 2012, the standardization was completed (RFC 6787 [38]), and it enabled the control of new resources for recording, speaker verification, and identification.

For a complete description of MRCPv2 and its relationship with W3C VBWG standards, see [39].

4. W3C MMIWG standards

The companion working group, Multi-Modal Interaction Working Group (MMIWG), led by Deborah Dahl was attended by almost the same companies attending VBWG. The goal of MMIWG was to extend the scope of standardization beyond the voice or typed input to embrace a much larger set of modalities, such as touch, gesture, emotions, and haptics both as input and output devices for a system.

The major achievements of the W3C MMIWG were the following standards:

- Ink Markup Language, InkML [40], is designed to represent the input of handwriting by a stylus or a finger. In addition to representing traces, InkML offers a rich set of metadata that preserve the appearance of the original input (i.e., color, width, orientation, timing, etc.).

- Extensible multimodal annotation, EMMA [41], is a standard to represent natural language input. It was designed to support annotation from different stages of processing, beginning with the initial results of speech or handwriting recognition and then natural language understanding annotations. EMMA also allows the fusion of different representations across multiple modalities in a multi-modal application, see also [42].

- EMMA was initially inspired by NLSML [43], which is now part of the MRCP protocol, and EMMA 1.0 was then accepted as interpretation result in the MRCPv2 protocol [37].

- Emotion Markup Language, EmotionML [44], is the result of a joint effort of leading researches in the field of emotion and industry. The effort was to create a standard language to annotate emotion in speech, visual, or text corpora, which are not only vital for research but also to represent emotions in recognition engines and to control emotions in TTS rendering. EmotionML became a W3C Recommendation in May 2014. An extended description of EmotionML is available in [45].

Another achievement of the W3C MMIWG was the definition of a multimodal architecture [46]. The multimodal architecture provides an event-based protocol for an interaction manager, possibly implemented in SCXML 1.0 [27], to coordinate an ensemble of modality components, each responsible for processing inputs or producing outputs in specific modalities. The protocol consists of a limited set of standard LifeCycle events—NewContext, Prepare, Start, Pause, Resume, Cancel, Done, ClearContext, Status, and Extension. The standard events include a set of standard fields, for example, fields to record the source and destination of the event, as well as a Data field, which can contain the results of processing an input.

A very detailed and up-to-date description of W3C multimodal standards is contained in [47].

As you see, there was close relationship between these two W3C working groups whose aim was to create a set of interoperable and complementary standards to expand capabilities of state-of-the-art applications.

5. Status and evolutions

This exciting period of an evolution based on standards came to an end after more than 15 years of activity. First, W3C VBWG was declared closed in October 2015 [48] because its mission to support "browsing the Web by voice" was achieved.

Going to the W3C VBWG homepage, there is the list of all the standards created and additional materials (see [13]). The only unfinished work is VoiceXML 3.0 [49] that was the attempt to create an extensible version of VoiceXML where addition of new features would have the benefit of clear interfaces.

When VoiceXML 3.0 effort started, the landscape had changed, greatly due to the success of VoiceXML 2.0 and companion standards. After the adoption of those standards, the industry was in a consolidation process of acquisition of innovative players by larger ones, where the goal was to have those standards at the core of the industry. Therefore, the pressure on innovations was reduced and, as consequence, the process slowed down, and ultimately stopped. One of the last activities was the publication of the first Working Draft of VoiceXML 3.0 [49] before dissolving the working group.

Nevertheless, after more than 20 years, these standards are still firmly at the core of the whole voice application industry, especially for customer care applications and other sectors. The creation of a family of interoperable standards is an advantage, because even new approaches to the development of more advanced speech application, for instance, by hosted APIs [50] or tools, are free to re-use what is already done, such as grammars, TTS controls, lexicons, result formats, and annotations.

A few years later, in February 2017, the W3C MMIWG was also dissolved for similar reasons. The first group of standards that includes InkML 1.0, EMMA 1.0, and EmotionML 1.0 and also the multimodal architecture were completed as W3C Recommendations. Other Working Drafts were also published (see [14]), among them was EMMA 2.0 [51], which was intended to extend EMMA from input results of different modalities to the annotation of output too.

The main lesson learnt is that when times are mature, a neutral and highly collaborative environment, such as W3C working groups, can attract all players that want to innovate to work together for the benefit of a whole industry, or advance new technologies. The shift from proprietary to standard-based technologies was the case described in this paper.

Current human interface platforms are very siloed, using proprietary formats and with little or no concern for interoperability. This means that the kind of inflexible vendor lock-in that we saw 25 years ago with telephony applications is very much with us today. As the underlying technologies continue to evolve, stabilize, and mature, it will become more and more apparent, as it did in the late 1990s, that open standards are a path toward accelerating the ubiquity of voice and multimodal applications and will truly benefit the entire industry.

Acknowledgements

It is thanks to an incredible group of talented people that I wrote this paper. I got to know each of their voices during innumerable conference calls, and their sense of humor in many face2face meetings. First, I would like to thank the chairs, Jim Larson, Dan Burnett, and Debbie Dahl, and not forgetting Scott McGlashan, whom I first met in the early 1990s when he was a PhD student involved in the SUNDIAL project and then later as co-chair with Jim Larson of VBWG. He showed great talent in leading the project's development. His departure in February 2014 was a big loss. I am also indebted to all the people who played such an active role throughout the years. These standards would not have been possible without their efforts. As there are too many to thank individually, I thank them collectively.

The W3C became our home, and from there, I would first like to thank the team contacts who always helped us to understand the W3C's processes and also gave us some very good ideas. Thanks to Dave Raggett, Kazuyuki Ashimura, Max Froumentin, and Matt Womer. Second, thanks go to the other team leads who contributed to broadening the scope of our work, such as Philipp Hoschka, the W3C Domain leader for the Ubiquitous Web, Richard Ishida for Internationalization (I18N), Judy Brewer and Janine Sajka for the Web Accessibility Initiative, and many other great people we met during the W3C Technical Plenary meetings, of course including Tim Berners-Lee.

I also have to mention my second home, the VoiceXML Forum, especially Val Matula and Rob Marchand who sit with me on the Board and, especially, Katie Valenti, our invaluable Program Manager for the ISTO team. Thanks.

I am also very grateful to Debbie Dahl and Roberto Pieraccini who read this paper and contributed so many comments and ideas. Finally, I cannot overlook Simon Parr for his timely assistance.

Author details

Paolo Baggia
Nuance, Torino, Italy

*Address all correspondence to: paolo.baggia@nuance.com

IntechOpen

References

[1] Pallett DS. Session 2: DARPA resource management and ATIS benchmark test poster session. In: Proceedings of the 4th DARPA Workshop on Speech and Natural Language. Pacific Grove (CA); 1991. pp. 49-58

[2] Pallett DS, Dahlgren NL, Fiscus JG, Fisher WM, Garofolo JS, Tjaden BC. DARPA February 1992 ATIS benchmark test results. In: Proceedings of the DARPA Speech and Natural Language Workshop. Harriman (NY); 1992. pp. 201-206

[3] Dahl D, Bates M, Brown M, Fisher W, Hunicke-Smith K, Pallett D, et al. Expanding the scope of the {ATIS} task: The {ATIS}-3 corpus. In: Proceedings of Human Language Technology. Plainsboro (NJ); 1994

[4] Paul DB, Baker JM. The design for the Wall Street journal-based CSR corpus. In: Proceedings of the DARPA Speech and Natural Language Workshop. Harriman (NY); 1992. pp. 357-363

[5] Klatt D. Review of text-to-speech conversion for English. The Journal of the Acoustical Society of America. 1987;**82**(3):737-793

[6] Balestri M, Pacchiotti A, Quazza S, Salza PL, Sandri S. Choose the best to modify the least: A new generation concatenative synthesis system. In: Proceedings of the 6th Conference on Speech Communications and Technology (EUROSPEECH-99); 5-9 September 1999; Budapest, Hungary. Vol. 5. 1999. pp. 2291-2294

[7] Peckham J. A new generation of spoken dialog systems: Results and lessons from SUNDIAL project. In: Proceedings of the 3rd Conference on Speech Communications and Technology (EUROSPEECH-93); 22-25 September 1993; Berlin, Germany. 1993. pp. 33-40

[8] Billi R, Castagneri G, Morena D. Field trial of two different information enquiry systems. Speech Communication. 1997;**23**(1-2):83-93

[9] den Os E, Boves L, Lamel L, Baggia P. Overview of the ARISE project. In: Proceedings of the 6th Conference on Speech Communications and Technology (EUROSPEECH-99); 5-9 September 1999; Budapest, Hungary. 1999. pp. 1527-1530

[10] Cole R, Mariani J, Uszkoreit H, Varile GB, Zaenen A, Zampolli A, et al. Survey of the State of the Art in Human Language Technology. Cambridge, UK: Cambridge University Press; 1997

[11] Pieraccini R. The Voice in the Machine, Building Computers that Understand Speech. Cambridge, MA: The MIT Press; 2012

[12] W3C. Voice Browsers W3C Workshop; 13 June 1998; Cambridge, MA. Available from: https://www.w3.org/Voice/1998/Workshop/ [Accessed: 26 April 2020]

[13] W3C. Voice Browser Working Group. Available from: https://www.w3.org/Voice/ [Accessed: 26 April 2020]

[14] W3C. Multimodal Interaction Working Group. Available from: https://www.w3.org/2002/mmi/ [Accessed: 26 April 2020]

[15] Larson J. Introduction and overview of W3C speech interface framework. W3C Working Draft; 4 December 2000. Available from: https://www.w3.org/TR/voice-intro/ [Accessed: 26 April 2020]

[16] Larson JA. W3C speech interface language: VoiceXML. IEEE Signal Processing Magazine. 2007;**4**(3):126-130

[17] VoiceXML Forum. Available from: http://www.voicexml.org/ [Accessed: 26 April 2020]

[18] VoiceXML Forum. Voice eXtensible Markup Language (VoiceXML)

version 1.0. W3C Note; 05 May 2000. Available from: http://www.w3.org/TR/2000/NOTE-voicexml-20000505 [Accessed: 26 April 2020]

[19] McGlashan S, Burnett DC, Carter J, Danielsen P, Ferrans J, Hunt A, et al. Voice Extensible Markup Language (VoiceXML) version 2.0. W3C Recommendation; 16 March 2004. Available from: https://www.w3.org/TR/voicexml20/ [Accessed: 26 April 2020]

[20] Hunt A, McGlashan S. Speech recognition grammar specification version 1.0. W3C Recommendation; 16 March 2004. Available from: https://www.w3.org/TR/speech-grammar/ [Accessed: 26 April 2020]

[21] Burnett DC, Walker MR, Hunt A. Speech Synthesis Markup Language (SSML) version 1.0. W3C Recommendation; 16 March 2004. Available from: https://www.w3.org/TR/speech-synthesis/ [Accessed: 26 April 2020]

[22] Oshry M, Auburn RJ, Baggia P, Bodell M, Burke D, Burnett DC, et al. Voice Extensible Markup Language (VoiceXML) 2.1. W3C Recommendation; 19 June 2007. Available from: https://www.w3.org/TR/voicexml21/ [Accessed: 26 April 2020]

[23] Van Tichelen L, Burke D. Semantic interpretation for speech recognition (SISR) version 1.0. W3C Recommendation; 05 April 2007. Available from: https://www.w3.org/TR/semantic-interpretation/ [Accessed: 26 April 2020]

[24] Burnett DC, Shuang ZW. Speech Synthesis Markup Language (SSML) version 1.1. W3C Recommendation; 07 September 2010. Available from: https://www.w3.org/TR/speech-synthesis11/ [Accessed: 26 April 2020]

[25] Baggia P. Pronunciation Lexicon Specification (PLS) version 1.0. W3C Recommendation; 14 October 2008. Available from: https://www.w3.org/TR/pronunciation-lexicon/ [Accessed: 26 April 2020]

[26] Auburn RJ. Voice browser call control: CCXML version 1.0. W3C Recommendation; 05 July 2011. Available from: https://www.w3.org/TR/ccxml/ [Accessed: 26 April 2020]

[27] Barnett J, Akolkar R, Auburn RJ, Bodell M, Carter J, McGlashan S, et al. State Chart XML (SCXML): State machine notation for control abstraction. W3C Recommendation; 01 September 2015. Available from: https://www.w3.org/TR/scxml/ [Accessed: 26 April 2020]

[28] Barnett J. Introduction to SCXML. In: Dahl DA, editor. Multimodal Interaction with W3C Standards. Cham, Switzerland: Springer International Publishing; 2017. pp. 81-107

[29] Larson J. VoiceXML: Introduction to Developing Speech Applications. Upper Saddle River, New Jersey: Prentice Hall; 2003

[30] Dahl D. Practical Spoken Dialog Systems. Berlin, Heidelberg: Springer-Verlag; 2005

[31] Jokinen K, McTear M. Spoken Dialogue Systems. Princeton, NJ: Morgan & Claypool; 2009

[32] Brown MK, Kellner A, Raggett D. Stochastic language models (N-Gram) specification. W3C Working Draft; 02 January 2001. Available from: http://www.w3.org/TR/ngram-spec [Accessed: 26 April 2020]

[33] Standard ECMA-327. ECMAScript 3rd Edition Compact Profile; June 2001. Available from: http://www.ecma-international.org/publications/files/ECMA-ST-WITHDRAWN/Ecma-327.pdf [Accessed: 26 April 2020]

[34] IPA. Handbook of the International Phonetic Association. Cambridge, UK: Cambridge University Press; 1999

[35] Harel D. Statecharts: A visual formalism for complex systems. Journal Science of Computer Programming. 1987;**8**(3):231-274

[36] The Internet Engineering Task Force (IETF). Available from: https://www.ietf.org/ [Accessed: 26 April 2020]

[37] Shanmugham S, Monaco P, Eberman B. A Media Resource Control Protocol (MRCP). RFC 4463; Informational; April 2006. Available from: https://tools.ietf.org/html/rfc4463 [Accessed: 26 April 2020]

[38] Burnett D, Shanmugham S. Media Resource Control Protocol Version 2 (MRCPv2); RFC 6787; Internet Standard; November 2012. Available from: https://tools.ietf.org/html/rfc6787 [Accessed: 26 April 2020]

[39] Burke D. Speech Processing for IP Networks: Media Resource Control Protocol (MRCP). Chichester, UK: Wiley; 2007

[40] Watt SM, Underhill T. Ink Markup Language (InkML). W3C Recommendation; 20 September 2011. Available from: http://www.w3.org/TR/InkML [Accessed: 26 April 2020]

[41] Johnston M. EMMA: Extensible MultiModal Annotation markup language. W3C Recommendation; 10 February 2009. Available from: http://www.w3.org/TR/emma/ [Accessed: 26 April 2020]

[42] Johnston M. Extensible multimodal annotation for intelligent interactive systems. In: Dahl DA, editor. Multimodal Interaction with W3C Standards. Cham, Switzerland: Springer International Publishing; 2017. pp. 37-64

[43] Dahl DA. Natural language semantics markup language for the speech interface framework. W3C Working Draft; 20 November 2000. Available from: https://www.w3.org/TR/nl-spec/ [Accessed: 26 April 2020]

[44] Burkhardt F, Schröder M. Emotion Markup Language (EmotionML) 1.0. W3C Recommendation; 22 May 2014. Available from: https://www.w3.org/TR/emotionml/ [Accessed: 26 April 2020]

[45] Burkhardt F, Pelachaud C, Schuller BW, Zovato E. EmotionML. In: Dahl DA, editor. Multimodal Interaction with W3C Standards. Cham, Switzerland: Springer International Publishing; 2017. pp. 65-80

[46] Barnett J, Bodell M, Dahl D, Kliche I, Larson J, Porter B, et al. Multimodal architecture and interfaces. W3C Recommendation; 25 October 2012. Available from: https://www.w3.org/TR/mmi-arch/ [Accessed: 26 April 2020]

[47] Dahl DA, editor. Multimodal Interaction with W3C Standards. Cham, Switzerland: Springer International Publishing; 2017

[48] Burnett DC. ALL: Thoughts and thanks as the VBWG comes to a close. W3C Mailing List Archive; 26 September 2015. Available from: https://lists.w3.org/Archives/Public/www-voice/2015JulSep/0029.html [Accessed: 26 April 2020]

[49] McGlashan S, Burnett D, Akolkar R, Auburn RJ, Baggia P, Barnett J, et al. Voice Extensible Markup Language (VoiceXML) 3.0. W3C Working Draft; 16 December 2010. Available from: http://www.w3.org/TR/voicexml30/ [Accessed: 26 April 2020]

[50] Natal A, Shires G, Cáceres M, Jägenstedt P. Web speech API. Draft Community Group Report; 21 January 2021. Available from: https://wicg.github.io/speech-api/ [Accessed: 26 April 2020]

[51] Johnston M, Dahl DA, Denney T, Kharidi N. EMMA: Extensible MultiModal Annotation markup language version 2.0. W3C Working Draft; 08 September 2015. Available from: https://www.w3.org/TR/2015/WD-emma20-20150908/ [Accessed: 26 April 2020]