Chapter 2
The Role and Importance of Speech Standards

Paolo Baggia, Daniel C. Burnett, Rob Marchand, and Val Matula

Abstract Within only a few years the landscape of speech and DTMF applications changed from being based on proprietary languages to being completely based on speech standards. In that, a role of primary importance was played by W3C Voice Browser Working Group (VBWG). This chapter describes this change, the implications, and highlights the standards created by the W3C VBWG, as well as the benefits that these standards can induce in many other application fields, including multi-modal interfaces.

2.1 Introduction

A strong wind of change was sweeping the stuffy world of Interactive Voice Response (IVR) and speech applications in general. This call for change developed in the very last years of the last century, resulting in a key event—the workshop on "Voice Browsers" held in Cambridge, MA on 13 October 1998 [1]. The workshop was sponsored by the W3C, and it raised huge interest in the standardization of voice application technologies. The direct result was the birth of a W3C Working Group—the Voice Browser Working Group (W3C VBWG [2]), formed to create an interconnected family of standards. This chapter offers a short introduction to most of the W3C VBWG standards and also describes their close relationship with the W3C Multimodal Interaction Working Group (W3C MMI [3]).

P. Baggia (✉)
Department of Enterprise, Nuance Communications, Inc., Torino, Italy
e-mail: paolo.baggia@nuance.com

D.C. Burnett
StandardsPlay, Lilburn, GA, USA

R. Marchand
Genesys, Markham, ON, Canada

V. Matula
Avaya Inc., Santa Clara, CA, USA

© Springer International Publishing Switzerland 2017

D.A. Dahl (ed.), *Multimodal Interaction with W3C Standards*,
DOI 10.1007/978-3-319-42816-1_2

Several factors combined to drive this change; the most relevant ones are

- The development of an IVR application was cumbersome and required the use of proprietary IDEs that were bound to individual vendors. At the time of the formation of the VBWG, IVR technology was proprietary and there was virtually no chance to exchange expertise or application assets between them.
- Speech technologies were very limited in their use; only simple commands, menu options, and sequences of digits were allowed. However, the core speech technologies were rapidly evolving to be more powerful and flexible and to allow a new generation of speech applications.
- Voice interactions were limited to simple menu navigation, with no flexibility to allow more advanced dialog capabilities. Their implementation was clumsy.
- But the most powerful factor was the advent of the Internet era: the HTTP protocol, the HTML language, and the flourishing of web sites. All these advances were based on public standards, while the world of voice applications was missing the opportunity to follow these new trends.

It was this combination of factors that drove the creation of the W3C VBWG, changing forever the world of voice applications. With the Working Group, a large number of companies now had a place to work together on this standardization effort, strongly motivated by a common interest. These companies included: speech technology makers (at that time L&H, Philips, Nuance, SpeechWorks, Loquendo, and Entropic), research laboratories (MIT, Rutgers, AT&T Bell Labs, and CSELT), large telephone companies (Lucent, AT&T, BT, Deutsche Telekom, France Telecom, Telecom Italia, Motorola, and Nokia), large software and hardware companies (Microsoft, HP, Intel, IBM, and Unisys), newly formed Voice Platform companies (PipeBeach, Voxpilot, Vocalocity, VoiceGenie, and Voxeo), hosting and developer studios (HeyAnita, BeVocal, and Tellme), IVR vendors (Avaya, Genesys, Comverse, and CISCO), and many more.

In the meantime, an industry organization named the VoiceXML Forum [4], created by AT&T, Lucent, Motorola, and IBM, proposed a new language called VoiceXML 1.0 [5] and started to evangelize its adoption. The newly created W3C VBWG, led at various times by Jim Larson (Intel), Scott McGlashan (PipeBeach and HP), and Dan Burnett (Voxeo and StandardsPlay), selected VoiceXML 1.0 as the starting candidate to inspire the creation of the standard to come.

With the standardization of VoiceXML and related technologies now firmly in the hands of the W3C, the VoiceXML Forum took on a complementary role in the evolution of VoiceXML. The Forum took on responsibilities including:

- Education: The Forum developed tutorials, sample applications, and produced a monthly e-zine for the developer community [6].
- Evangelism: The Forum marketed VoiceXML as an emerging standard to the IVR community, both on-line and at industry conferences.
- Conformance: Perhaps the most critical role of the Forum was the development of conformance test suites (based on the W3C specification Implementation

Reports[1]) for VoiceXML, SSML, and SRGS. The Forum also provided independent third-party conformance test certifications.
- Developer certification: The Forum created a VoiceXML Developer Certification program, including several test suites and access to third party certification testing. This helped to build a developer community.
- Technology and tools evaluation: The Forum hosted several committees with the task of assessing evolving technologies and tools related to the adoption of VoiceXML. These groups investigated speech technology protocol standardization, security aspects (including biometrics), and other topics.

These areas are outside the scope of W3C standards development, yet are critical in supporting adoption and acceptance of a new standard. The role fulfilled by the VoiceXML Forum helped avoid early standards fragmentation, and fostered adoption of VoiceXML by industry.

This collaboration accelerated the 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 innovation under development. Supported by the broad involvement of stakeholders from different groups, the IVR industry began to implement these drafts as soon as they were delivered. Adoption worries were rapidly left behind in this new ecosystem.

In March 2004, after just 4 years, the first complete standards, i.e., W3C Recommendations, were released: VoiceXML 2.0 [7] for authoring a voice application; SRGS 1.0 [8] for precisely defining the syntax of speech grammars; and SSML 1.0 [9] for controlling speech synthesis (or text-to-speech). A few years later, in April/June 2007, a second round of W3C Recommendations concluded, delivering: VoiceXML 2.1 [10], adding some interesting features on top of VoiceXML 2.0; and SISR 1.0 [11], formalizing the representation of meaning within a speech grammar and complementing SRGS 1.0. The work didn't stop there. SSML 1.0 was revised to version 1.1 [12] to ease the internationalization of speech synthesis in other world regions; PLS 1.0 [13], a language to describe phonetic lexicons supporting interoperability between SRGS 1.0 and SSML 1.0/1.1, was created; and finally CCXML 1.0 [14] was developed as a real time language to implement call control in a voice browser platform. In the rest of this chapter, these languages will be briefly introduced and other aspects of this revolution will be highlighted.

It is worth noting that the entire industry took the advent of the W3C VBWG standards as a change to be immediately adopted. Vendors implemented the standards in their solutions, so that the IVR and telephony application world started to speak the VoiceXML standard language in a matter of a few years. The use of VoiceXML enforced a clean separation between the IVR platforms and the hosting of voice applications accessed by HTTP/HTTPS. Voice applications were at first

[1] All W3C Recommendations include a reference to an Implementation Report document to assess the implementability of the proposed standard. For instance, the VoiceXML 2.0 Implementation Report [44] was very important in showing how to implement a procedure to automate most of the tests.

static "pages" stored on a Web server, and then progressively became dynamic as the rest of the Web evolved. A side effect of this adoption of the web architecture by VoiceXML was that many web-related technologies became available to IVR applications. For example, VoiceXML 2.1 added the `<data>` element to take advantage of emerging AJAX access to web services. Infrastructure elements like web caches and load balancers were immediately useful within IVR deployments. Several books and articles describing VoiceXML were published. A review of the language is presented in [15] from the W3C VBWG Chair Jim Larson. For a discussion of VoiceXML in the broader context of Spoken Dialogue Systems see [16]. Many start-ups sprang up to offer voice browsers, tools, hosting and, as in all industry sectors, the larger companies filled out their offerings by acquiring these start-ups.

Over time this process slowed down, indicating that the revolution had occurred, but it also meant that continued change was becoming more difficult. There was a definite and very ambitious attempt to re-write VoiceXML to be extensible and modular, simplifying the incorporation of future advances, but this radical re-formulation stopped after producing the first Working Draft (VoiceXML 3.0 [17]). The last advance was to lay down and complete SCXML 1.0 [18], described in another chapter of this book. This language offers to both the IVR world and the world of multimodal applications a clean and powerful way to encode the interaction of different components thorough Harel's state-charts [19]. Other activities that were not completed included a standard for speaker verification/identification from voice prints and statistical language models [20].

The W3C VBWG had completed its role, developing the solid foundation of an open and powerful generation of standards, so it closed its activities in October of 2015 [21]. The VoiceXML Forum remains active in sustaining the VoiceXML 2.0/2.1 ecosystem, most notably with work in Conformance and Developer Certification.

As with most technologies, these standards will remain for a long time in the core of these industries, but there may also be opportunities to reuse them in novel ways and to fuel new advances and revolutions. The variegated world of multimodal interfaces and, more generally, of the Internet of Things (IoT), will greatly benefit from the work of the W3C VBWG, especially for a speech modality. Voice continues to prove to be the most powerful means for humans to control and influence the world around us. Section 2.4 describes the role of voice standards in these additional domains, while Sect. 2.2 is devoted to explaining the standards developed by W3C VBWG and Sect. 2.3 complements the discussion with related IETF protocols.

2.2 Quick Tour of W3C VBWG Major Standards

This section presents a quick tour of the W3C VBWG standards. The review will be limited to a brief introduction with some highlights of the major benefits of each standard.

Although VoiceXML is the most visible of the VBWG standards, there are a number of related languages that work together with VoiceXML to provide a complete facility for creating IVR applications. The individual languages have also in some cases been used independently for other purposes as well, such as:

- The grammar language, SRGS 1.0, for processing text input instead of speech.
- The speech synthesis language, SSML 1.0/1.1, for talking books, or assistive applications.
- The pronunciation lexicon language, PLS 1.0, in language training applications.
- The call control language, CCXML 1.0, for managing calls over IP.
- The state-chart language, SCXML 1.0, for generalized interactions in a multimodal interface.

The following is a brief description of these standards developed by W3C VBWG.

2.2.1 VoiceXML 2.0

The Voice Extensible Markup Language (VoiceXML), version 2.0. [7], is the flagship and most relevant standard produced by the W3C VBWG. It is an XML markup language specialized to declaratively describe a dialog interaction between a caller and an automated application. The language leverages all the advantages of the Web: an application is stored in a Web Application server; it might be statically or dynamically generated; a specialized user agent, called a Voice Browser, downloads and interprets a VoiceXML application, together with scripts, audio prompts, and grammars; the syntax is enforced by an XML Schema or a DTD; the application might be in any human language and declares an appropriate encoding; and so on.

The standard was built on top of the initial VoiceXML 1.0 [5] proposal made by the VoiceXML Forum [4]. The original Forum members were from AT&T, IBM, Lucent Technologies, and Motorola. In the W3C VBWG, a much larger number of people from many companies and organizations participated in the joint effort to transform the proposal into a widely accepted standard. This promise was realized in March 2004, when VoiceXML 2.0 was declared a W3C Recommendation, with nine companies[2] presenting an Implementation Report to demonstrate interest in promoting this standard in the industry. The Implementation Report was based on a test suite of over 600 test assertions defined in a special language to facilitate its automation. Based upon this test suite, the VoiceXML Forum delivered a Platform Certification program [22] with at least 27 platforms certified to date.

[2] The companies which submitted an Implementation Report [44] for VoiceXML 2.0 were nine: Comverse, Genesys, Loquendo, Motorola, PublicVoiceXML Consortium, Tellme Networks, Vocalocity, VoiceGenie Technologies, and Voxpilot.

A VoiceXML application is made of dialogs whose building blocks are the <menu> and <form> elements. The former is used for designing simple menu-based IVR applications, while the latter is used for extending the interaction to form filling, where an algorithm called the Form Interpretation Algorithm (FIA, described in Appendix C of the specification [7]) is used. The FIA precisely describes how the filling of different <field>s is performed. VoiceXML 2.0 also supports extending the interaction to a mixed-initiative dialog, where additional flexibility allows a caller to say more complex sentences, like: "I'd like to travel from Boston to Detroit in First Class next Monday around noon." In this modality, several <field>s are filled at the same time to take maximum advantage of the compactness and flexibility of natural language.

A novelty of VoiceXML 2.0 was to delegate[3] the definition of speech grammars and of synthesized prompts to two interoperable standards. These standards are: SRGS 1.0 [8] for grammars and SSML 1.0 [9] for prompts, described below. This choice allowed the standards to be developed in parallel, but more importantly it promoted the reuse of speech recognition and of speech synthesis in other application contexts, such as multimodal interfaces, or appliances. The <field> element may include one or more <prompt>s to solicit the caller to say or type the expected information and several <grammar>s to model callers' sentences, while the <filled> element is triggered if the information items are collected either by voice or DTMF, upon which they are automatically stored in a variable associated with the <field> itself. The filling of field values will restart the collection cycle as described in the FIA algorithm.

Besides the <field> element, other form items are supported by VoiceXML 2.0 to enrich the dialog interaction. For instance:

- <block> element to declare prompts and to perform a block of computations;
- <record> element to record the caller's voice and provide access to the stored audio;
- <transfer> element to transfer a call to another party either by a "bridge" or a "blind" transfer;
- <subdialog> element to pause the current interaction, spawn the processing of another context to complete a task, and then return to the calling environment with the results;
- <object> element to allow for new functionality extensions; this was used for extending the capabilities of a voice browser to allow additional features, for instance, the inclusion of voice biometrics capabilities in a VoiceXML application.

Data are handled and processed by an ECMAScript processor with elements in the language to declare and assign variables (<var> and <assign> elements) or load scripts (<script> element). The variables are organized into different

[3] The XML Schema of VoiceXML 2.0 includes the references to: SRGS 1.0 and SSML 1.0 XML Schemas, see Appendix O of VoiceXML 2.0 specification [7].

scope levels: “application” for the sharing of data across different VoiceXML documents, “document” for variables that need to be visible across a single document, “dialog” for variables active only inside a single <form> or <menu>, and the internal context of an inner element. Above these scopes, there is an additional one called “session” that contains read-only variables related to that specific session. For instance, the session scope provides access to telephony information (e.g., ANI, DNIS, etc.). Finally, each recognition step allows the browser to access information related to the most recently occurring recognition. Some examples include the input modality (either “voice” or “dtmf”), a numeric value for the confidence of the results, the recognized/keyed text and the meaning of that interaction.

The FIA describes the flow of the interaction inside a dialog element. To transition to the next dialog a <goto> element contains a URI attribute that points either to a dialog in the same document or to another VoiceXML document to which to transition. The <submit> element is used to upload data collected during the dialog interaction to the Web Application. The result is a new VoiceXML page for continuing the interaction. Moreover, an event handling mechanism is present to allow firing predefined events: “help,” “repeat,” but also “noinput” and “nomatch” to indicate a missing response from the caller, or that the input was not properly recognized. It is also possible to throw (<throw> element) user defined events that will trigger the execution of a handler defined via the <catch> element. In this way the application can deal with predefined and unexpected behaviors by continuing within the same dialog under the FIA, or by transitioning to another one, or even by closing the interaction. An application can be explicitly terminated by the <exit> element or closed via the <disconnect> element. The latter hangs up the call if necessary.

The VoiceXML specification had a terrific impact on the IVR industry, being widely adopted even before the language was completed. The traditional IVR platform vendors had to change their architectures by including either a home-grown VoiceXML browser or one obtained by acquiring a newly formed start-up company. Other vendors opted for hosting the VoiceXML browser and providing a Web-based development environment to create, test, and deploy voice applications. Some examples include HeyAnita, BeVocal, TellMe, and Voxeo, all since acquired by major players. Other companies specialized in tools or development environments, and they were progressively acquired as well. The VoiceXML Forum provided user groups, newsletters, journals, and events to sustain the VoiceXML ecosystem. A critical contribution from the Forum was the development of two Certification Programs, the previously mentioned Platform Certification and a Developer Certification program, both still active at present.

2.2.2 VoiceXML 2.1

Although the VoiceXML 2.0 specification was immediately implemented and rapidly became the major standard for voice and DTMF applications, a follow-on effort added a limited number of extensions. This key collection of extensions was published in June 2007 as VoiceXML 2.1 [10]. It includes eight additional features, including a means to dynamically reference grammars and scripts, a new <foreach> iteration element for dynamically composing prompts or executing computations on list of objects, and a <data> element to allow a VoiceXML application to dynamically load data from a server using the equivalent of an XML HTTP Request. These extensions were all motivated by the need to reduce the number of (expensive) VoiceXML page transitions. With VoiceXML 2.1 a single running VoiceXML page was able to adapt to external or dynamic conditions. In addition, the <transfer> element was extended with a new "consultation" mode to allow the interaction to be suspended while a transfer was attempted, resuming it if the transfer was not possible. The recording capabilities were extended to be active during the recognition to enable fetching of both the audio and the recognition results. Finally, the <disconnect> element was extended to return a list of results.

The VoiceXML 2.1 extensions were also widely implemented, and the VoiceXML Forum Platform Certification Program was extended to additionally certify both VoiceXML 2.1 and the grammar language described in the next section.

2.2.3 SRGS 1.0

Speech recognition greatly benefits by knowing in advance what a caller might say. A speech grammar is a compact way to declaratively describe the admissible sentences. The W3C VBWG was very successful in clearly defining the syntax of speech grammars. SRGS 1.0, the Speech Recognition Grammar Specification [8], defines two different formats for encoding a grammar: an XML format, called GRXML, and a textual one, called ABNF (cfr Augmented Backus-Naur Form). The two formats are homologous, with very irrelevant differences. A grammar defines sequences of words/phrases or alternatives to be legally accepted by the speech recognizer. The grammar is organized into rules, where only a few are accessible from the outside (declared "public"), while all the others are hidden (declared "private") to enforce modularity and a clean composition among different grammar files.

If words define the admissible sentences, a grammar also provides a way to compose a result to be returned to the application. This is done by the execution of small scripts contained in the <tag> element, with the following permitted as a

return value: numbers "123" when the caller speaks "one hundred twenty three," date and time expressions, telephone numbers, or arbitrary key-value pairs.

The SRGS 1.0 specification immediately became the format supported by all speech recognition engines, allowing them to interoperate within a VoiceXML platform.

2.2.4 SISR 1.0

The production of a result remained undefined in the SRGS 1.0 specification, having been delegated to a subsequent specification, "Semantic Interpretation for Speech Recognition" (SISR 1.0 [11]), which was released as a W3C Recommendation in April 2007. SISR 1.0 formally defines the content of the <tag> element in SRGS 1.0 grammars to be an ECMA-327 [23] script. While ECMA-327 is a constrained version of ECMAScript, the goal was to gain computational efficiency to enable more extensive speech recognition engine processing.

The language defines how rules produce results and how they return them when they are referenced. This process allows the final result of a grammar to be composed progressively. Attention was paid to allow both a sequential and parallel execution of the result composition.

The presence of scripting capabilities inside a grammar helped move application-dependent normalization inside of the grammar and to clearly separate the application needs from the need for a natural way of expressing the caller's expected language.

2.2.5 SSML 1.0 and 1.1

Another effort was to define how to control a speech synthesis, or text-to-speech, engine. The controls help the engine to render the textual prompt in the most accurate way. The XML markup language for this purpose is the Speech Synthesis Markup Language (SSML 1.0 [9]).

SSML 1.0 includes elements that describe the structure of the text to be spoken (<p> element for paragraphs, and <s> element for sentences), text normalization and phonetic input (<sub> element for textual substitutions and <phoneme> for pronunciations), prosodic features such as pauses (<break> element), speed and rate (<prosody> element), and how to change the speaking voice (<voice> element).

An extension of SSML 1.0, SSML 1.1 [12], was a continuation of the standardization effort to promote the use of SSML to more international languages, in particular Asian and Indian languages.

2.2.6 PLS 1.0

Both speech grammars and synthesized prompts might require customizing the pronunciation for a specific application domain. This is often done by adding a reference to a user lexicon. The Pronunciation Lexicon Specification (PLS 1.0 [13]) was created to allow for the definition of a standard lexicon fully interoperable with SRGS 1.0 and SSML 1.0/1.1. The lexicon is a container of entries, `<lexeme>` elements, with a textual part described by the `<grapheme>` element and textual replacements provided by the `<alias>` element or phonetic transcriptions by `<phoneme>` elements.

PLS 1.0 documents support the expansion of abbreviations and acronyms, addressing both multiple orthographies and multiple pronunciations. PLS 1.0 became a W3C Recommendation in October 2008.

2.2.7 CCXML 1.0

Another language defined by the W3C VBWG focused on 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 [14]) 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.

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 the monitoring of calls and conferences via a web interface.

During the definition of CCXML 1.0 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 was State Chart XML (SCXML): State Machine Notation for Control Abstraction (SCXML 1.0 [18]) described in another chapter of this volume, and it can be used as the key component to control a generalized interaction in a multi-modal interface.

2.3 IETF, Companion Protocols

The VoiceXML revolution wouldn't have been possible without the presence of several other standards, especially protocols. For instance, in a Voice Platform the application documents, which might include VoiceXML 2.0/2.1 pages, SRGS 1.0 grammars, audio files, ECMAScript scripts, and PLS 1.0 lexicons, are accessed through HTTP/HTTPS protocols, as in any other Web user agent. Many of the web

browser/web server related protocols apply equally well to voice browsers as well. For example, voice browsers respect content-types, cookies, and cache control directives, as used by web browsers.

A new requirement for IVR platforms is standardization of the communication between the VoiceXML browser platform and the servers providing speech resources. Historically based on proprietary APIs and formats, speech recognition resources require grammars and audio, returning recognition results to the IVR platform. Similarly, text to speech resources require text that is to be rendered, and then return audio to the IVR platform. With the definition of SSML, SRGS, and SISR, the high-level interaction with the speech resources became standardized. A standard network level protocol was then defined for speech resources by the IETF [24]. Media Resource Control Protocol (MRCP), whose initial draft was proposed by CISCO, Nuance and SpeechWorks in April 2006 was standardized as MRCPv1 (RFC 4463 [45]). MRCPv1 is based on Real time Protocol (RTP) for media transport and on Real time Streaming Protocol (RTSP) for controlling speech resources such as speech synthesizers and speech or DTMF recognizers. The MRCP protocol defines the requests, responses, and events to control the processing inside resource servers. For a detailed description of the MRCP protocol in relationship with W3C VBWG standards see [26].

The introduction of MRCPv1 allowed voice platforms to be implemented with a distributed and scalable architecture, and was hence immediately adopted by all the IVR platforms. In the meantime, the standardization process continued with the definition of MRCPv2, becoming an IETF standard in November 2012 (RFC 6787 [25]). MRCPv2 is based on Session Initiation Protocol (SIP) for signaling and Session Description Protocol (SDP) for exchanging and negotiating capabilities. Moreover, MRCPv2 was extended to access new resources for recording and speaker verification and identification, and to support encryption.

2.4 Current Trends and Future Evolutions

Although the W3C VWBG is now closed [21], the influence of its standards is still broadly felt across many sectors. In the IVR/Customer Care world, the presence of VoiceXML and related standards is ubiquitous, and there are no signs this will change in the near future. There is an established industry in place, so drastic changes are very unlikely. In less than 15 years the W3C VBWG standards moved from an idea to mandatory requirements for a whole industry—a remarkable outcome.

Other speech languages developed outside of the W3C VBWG include XHTML + Voice [28] from IBM and Opera Software to allow a direct integration of VoiceXML in an XHTML document, and Speech Application Language Tags (SALT) [29] from a consortium led by Microsoft, to integrate speech into a web application. More recently, Google started a separate effort to develop the Web Speech API in a W3C Community Group to enable web developers to incorporate speech recognition and synthesis into their web pages; the Final Report is available at [30].

There are several innovations and new application domains that might require these standards when ready to integrate a voice or textual interaction. A quick review of the trends and evolutions that are happening are quickly described in the following sections.

2.4.1 IVR in the Multi-Channel World

The IVR world today is experiencing change due to the proliferation of channels available to a customer for seeking support or gathering information. In the recent past the only available way was a phone call, and while these days the phone call is still predominant, the contribution of other channels is increasingly evident. For instance, textual chats may be offered to a user during a web session, often with the presence of dedicated agents, and sometimes including even some degree of automation. Moreover, a web site often provides more than web search inside its content, for instance, a text interaction to intelligently search among FAQs or even a limited capability to provide precise answers to customer requests. Finally, social media is becoming a place for seeking support as well, and can be used to express very polarized—and highly visible—opinions on the company of interest. For example, a high-profile complaint on twitter will often result in a rapid response to a problem, perhaps out of proportion to the original issue.

Given all of these options, users may switch between channels (multi-channel) if they run into challenges, or will often use multiple channels at once (omni-channel). Consequently, most vendors in the customer experience field must support a number of different channels in their solutions. In order to integrate and correlate interactions taking place over multiple channels, possibly over disparate time spans (consider a voice call vs. an SMS exchange), the standards specific to a particular channel (for example, VoiceXML for voice, HTML for web) must be combined with the ability to receive and send events and data, manage state, and coordinate activities over multiple channels. SCXML (more fully described in a separate chapter) fulfills these requirements, providing the ability to coordinate between channels. For example, an inbound voice call can be connected to a VoiceXML session under SCXML control. Once the call is complete, the SCXML session can schedule a follow-up SMS message to the original caller as a reminder (perhaps days later) or as a transaction summary (immediately).

An omni-channel session could, after an inbound voice caller is identified, take advantage of the fact the caller was browsing the company web site when they decided to call. This could lead to a co-browse web session with the caller still on the phone, helping to complete a transaction or solve a problem.

There may be opportunities for the use of VBWG standards in other ways as well. For example, representation of meaning, or extraction of meaning from textual inputs—gathered using chat or SMS channels—might be enabled using SRGS or SISR. In another example, the SSML standard could be used to improve the rendering of text to speech in web-based interfaces. In the areas of multi-channel and omni-channel communications, the W3C VBWG standards can perhaps extend their role beyond voice interactions to be exploited with other channels.

2.4.2 Virtual Assistants

Since the deployment of the Siri virtual assistant on Apple's iPhone and iPad in 2011, interest in Virtual Assistants (VAs) in general has increased. The VAs can take the aspect of avatars, or, like Siri, be just a speaking voice, allowing users to ask open-ended questions, and typically using cloud resources to determine intent and return results. This new kind of voice activated VA is moving beyond mobile phones to cars, and to home appliances—e.g., Amazon Echo and the home robot Jibo. A distinctive characteristic of this kind of VA is the ability to speak and understand user commands, sometimes with amusement and/or irony in the answers and with personality.

In this area there is still a strong need to extend the capabilities of the interaction, but certainly voice is the most natural means with which to interact with these Virtual Assistants, allowing multiple requests to be packed into a single sentence.

The standards produced by the VBWG have less applicability in this realm. VoiceXML is best suited for applications that are system-directed rather than user-directed as is seen in typical VAs. However, the supporting standards may have a role to play. SCXML can be used as described in Sect. 2.4.1 to coordinate multiple channels and interactions, where the VA can be viewed as another channel. SISR and the Extensible Multi-Modal Annotation (EMMA 1.0 [27]) specification can be used to exchange information related to input, intent, and output across different channels. Multimodal interfaces and EMMA are further detailed in the following section.

2.4.3 Multimodal Interfaces

A richer style of interaction can be offered by a multimodal interface, which integrates not only a voice modality, but also gesture, text, haptic, or other kind of input. A multimodal interface is able to integrate requests given by different complementary and supplementary modalities. The W3C MMIWG [3] is responsible for developing standards in this area, but the cross-collaboration with the W3C VBWG was very well maintained. For instance, the main specification developed by the W3C MMIWG is an MMI architecture framework [31] whose key components are an Interaction Manager and one or more Modality Components. Among them, voice input is dealt with by a specific Modality Component that can be directly modeled using W3C VBWG standards. For instance, SRGS 1.0 can be used for speech recognition, SSML 1.0 for speech synthesis, and VoiceXML 2.1 for modular dialogs.

A core aspect of multimodal interfaces is the need to integrate meanings from different modalities. This is made possible by the Extensible Multi-Modal Annotation (EMMA 1.0 [27]) specification. EMMA 1.0 was designed to encode meaning representations produced by SRGS 1.0, where semantic interpretation by SISR 1.0

can produce results in EMMA 1.0 format. This volume describes in a dedicated chapter the recent extensions to the EMMA specification (EMMA 2.0 [32]) to extend its role from representing input only, to also cover a variety of outputs. Another important contribution of W3C VBWG standards is the use of SCXML 1.0 [18] to direct the Interaction Manager inside the MMI Architecture Framework. This direction is proposed by many authors (cf. [33, 34]), and a related workshop has been active since 2014. The "EICS Workshop on Engineering Interactive Computer Systems with SCXML" [35] demonstrates the interest of the research community in this topic.

Finally, the W3C MMIWG produced a standard for describing emotions expressed by face, voice, or other modalities. The EmotionML 1.0 [36] standard is a good candidate to be integrated with SISR 1.0 to encode emotions recognized in human voice. Similarly, EmotionML 1.0 could be used with SSML 1.0 to instruct a speech synthesis engine to express emotions. A detailed description of EmotionML 1.0 is present in another chapter of this volume.

2.4.4 Internet of Things

The Web continues to evolve and expand, now with the theoretical inclusion of all objects interconnected by a network interface, often called Internet of Things (IoT). These objects can be anything in the surrounding environment including, for example, home appliances, cars, hand-held devices, televisions, etc. In literature there are many examples where both the SCXML 1.0 and VoiceXML 2.1 standards are proposed for a variety of IoT applications. These range from the SmartHome [37, 38], Ambient Assisted Living [39], Semantic Sensor Network [40] to more general Pervasive Environments [41, 42], and embedded applications like automotive ones [43]. In those projects, the control of interactions allowed by SCXML 1.0 is fundamental and often inside the MMI Architecture previously described. Moreover, a user can take advantage of a voice interaction within this multifaceted world. The W3C VBWG speech related standards offer the basis for implementing this voice interaction. Examples are voice commands from a mobile device to inquire about the status of home appliances and also give commands to remotely activate these appliances.

2.5 Conclusion

This chapter has described the development and evolution of speech-related standards, and how they have impacted the IVR industry and voice applications. The shift from proprietary to standards-based technologies provided many important benefits, including:

- Conversion of an industry from fragmented and proprietary development to environments that more easily interoperate and support application portability. Application portability prior to standards availability generally meant a complete reimplementation. Now, it may be as simple as completion of testing. An ancillary to this is the education of a workforce that is more portable as well.
- The ability to leverage widely available web-related technologies within speech environments, supporting common techniques of scaling, architecture, and development practices. And although voice user interface design and telephony are important skills, many of the other skills required to implement IVR infrastructure are now more easily available due to the migration to a web-based architecture. This can also mean one less silo in an organization, reducing costs.
- A separation of interface presentation from business logic. It is now common to use the same business-level web services to support web, voice, and other channels.

These benefits, along with advancements in speech technology, have allowed the construction of more powerful voice applications while improving portability, maintainability, and interoperability. Although some of these advancements may have occurred without the development of speech related standards, it is likely that voice application development would have remained as a separate silo within the organization, requiring niche skills across the breadth of an implementation. There are also some useful lessons that may be taken from the W3C VBWG standards development experience:

- The standards themselves are important, but shouldn't be developed in a vacuum. The involvement of industry from the beginning ensured a set of standards that would meet real-world needs. The VoiceXML ecosystem provided important support for the acceptance of the standards developed within the W3C.
- A modular collection of standards can possibly support changes in technology over a longer period of time. While VoiceXML itself is modeled around a particular type of interaction (and is limited by the FIA in this regard), the supporting standards (SCXML, SRGS, and SSML) have provided value for other communication channels and interaction types. However, it is more difficult to advance multiple specifications simultaneously, and to ensure completion of a complete set meeting the original need.

The development of speech-related standards by the W3C, in combination with wide support—both through the W3C and the VoiceXML Forum—led to a transformation of the Interactive Voice Response industry. This transformation remains an important component in overall contact center modernization, and has aided in the advancement of voice application usage and usability.

References

1. W3C (1998). Voice Browsers, W3C Workshop, Cambridge, MA. https://www.w3.org/Voice/1998/Workshop/. Accessed 1 Mar 2016.
2. W3C (2016). Voice Browser Working Group. https://www.w3.org/Voice/. Accessed 1 Mar 2016.
3. W3C (2016). Multimodal Interaction Working Group. https://www.w3.org/2002/mmi/. Accessed 1 Mar 2016.
4. VoiceXML Forum (2016). http://www.voicexml.org/. Accessed 1 Mar 2016.
5. VoiceXML Forum (2000). Voice eXtensible Markup Language (VoiceXML) version 1.0. https://www.w3.org/TR/voicexml/. Accessed 1 Mar 2016.
6. VoiceXML Forum (2016). e-zine. http://www.voicexml.org/voicexml-review-archive/. Accessed 15 Mar 2016.
7. McGlashan, S., Burnett, D. C., Carter, J., Danielsen, P., Ferrans, J., Hunt, A., et al. (2004). Voice Extensible Markup Language (VoiceXML) version 2.0, W3C Recommendation. https://www.w3.org/TR/voicexml20/. Accessed 1 Mar 2016.
8. Hunt, A., & McGlashan, S. (2004). Speech Recognition Grammar Specification Version 1.0, W3C Recommendation. https://www.w3.org/TR/speech-grammar/. Accessed 1 Mar 2016.
9. Burnett, D. C., Walker, M. R., & Hunt, A. (2004). Speech Synthesis Markup Language (SSML) Version 1.0, W3C Recommendation. https://www.w3.org/TR/speech-synthesis/. Accessed 1 Mar 2016.
10. Oshry, M., Auburn, R. J., Baggia, P., Bodell, M., Burke, D., Burnett, D. C., et al. (2007). Voice Extensible Markup Language (VoiceXML) 2.1, W3C Recommendation. https://www.w3.org/TR/voicexml21/. Accessed 1 Mar 2016.
11. van Tichelen, L., & Burke, D. (2007). Semantic Interpretation for Speech Recognition (SISR) Version 1.0, W3C Recommendation. https://www.w3.org/TR/semantic-interpretation/. Accessed 1 Mar 2016.
12. Burnett, D. C., & Shuang, Z. W. (2010). Speech Synthesis Markup Language (SSML) Version 1.1, W3C Recommendation. https://www.w3.org/TR/speech-synthesis11/. Accessed 1 Mar 2016.
13. Baggia, P. (2008). Pronunciation Lexicon Specification (PLS) Version 1.0, W3C Recommendation. https://www.w3.org/TR/pronunciation-lexicon/. Accessed 1 Mar 2016.
14. Auburn, R. J. (2011). Voice Browser Call Control: CCXML Version 1.0, W3C Recommendation. https://www.w3.org/TR/ccxml/. Accessed 1 Mar 2016.
15. Larson, J. A. (2007). W3C speech interface language: VoiceXML. *IEEE Signal Processing Magazine, 4*(3), 126–130.
16. Jokinen, K., & McTear, M. (2009). *Spoken dialogue systems*. Princeton, NJ: Morgan & Claypool.
17. McGlashan, S., Burnett, D. C., Akolkar, R., Auburn, R. J., Baggia, P., Barnett, J., et al. (2010). Voice Extensible Markup Language (VoiceXML) Version 3.0, W3C Working Draft. https://www.w3.org/TR/voicexml30/. Accessed 1 Mar 2016.
18. Barnett, J., Akolkar, R., Auburn, R. J., Bodell, M., Carter, J., McGlashan, S., et al. (2015). State Chart XML (SCXML): State Machine Notation for Control Abstraction, W3C Recommendation. https://www.w3.org/TR/scxml/. Accessed 1 Mar 2016.
19. Harel, D. (1987). StateCharts: A visual formalism for complex systems. *Journal Science of Computer Programming, 8*(3), 231–274.
20. Brown, M. K., Kellner, A., & Raggett, D. (2001). Stochastic Language Models (N-Gram) Specification, W3C Working Draft. https://www.w3.org/TR/ngram-spec/. Accessed 1 Mar 2016.
21. Burnett, D. C. (2015). ALL: Thoughts and thanks as the VBWG comes to a close. W3C Mailing List Archive. https://lists.w3.org/Archives/Public/www-voice/2015JulSep/0029.html. Accessed 1 Mar 2016.

22. VoiceXML Forum (2016). VoiceXML Platform Certification Program. http://www.voicexml.org/certification-programs/voicexml-platform-certification-program/. Accessed 1 Mar 2016.
23. ECMA (2001). ECMAScript 3rd Edition Compact Profile. http://www.ecma-international.org/publications/files/ECMA-ST-WITHDRAWN/Ecma-327.pdf. Accessed 1 Mar 2016.
24. The Internet Engineering Task Force (IETF) (2016). https://www.ietf.org/. Accessed 1 Mar 2016.
25. Burnett, D., & Shanmugham, S. (2012). Media Resource Control Protocol Version 2 (MRCPv2), RFC 6787—Internet Standard. http://www.rfc-base.org/txt/rfc-6787.txt. Accessed 1 Mar 2016.
26. Burke, D. (2007). *Speech processing for ip networks: Media resource control protocol (MRCP)*. New York, NY: Wiley.
27. Johnston, M., Baggia, P., Burnett, D. C., Carter, J., Dahl, D. A., McCobb, G., et al. (2009). EMMA: Extensible MultiModal Annotation markup language, W3C Recommendation. https://www.w3.org/TR/emma/. Accessed 1 Mar 2016.
28. Axelsson, J., Cross, C., Lie, H. W., McCobb, G., Raman, T. V., Wilson, L. (2001). XHTML +Voice Profile 1.0, W3C Note. https://www.w3.org/TR/xhtml+voice/. Accessed 1 Mar 2016.
29. Microsoft Corporation, Speech Application Language Tags (SALT) (2003). Technical article. https://msdn.microsoft.com/en-us/library/ms994629.aspx. Accessed 1 Mar 2013.
30. Shires, G., & Wennborg, H. (2012). Web Speech API Specification, W3C Community Group Final Report. https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html. Accessed 1 Mar 2016.
31. Barnett, J., Bodell, M., Dahl, D., Kliche, I., Larson, J., Porter, B., et al. (2012). Multimodal Architecture and Interfaces, W3C Recommendation. https://www.w3.org/TR/mmi-arch/. Accessed 15 Mar 2016.
32. Johnston, M., Dahl, D., Denney, T., & Kharidi, N. (2015). EMMA: Extensible MultiModal Annotation markup language Version 2.0, W3C Working Draft. https://www.w3.org/TR/emma20/. Accessed 15 Mar 2016.
33. Kistner, G., & Neurenberger, C. (2004). Developing user interfaces using SCXML statecharts. In *Proceedings of the 1st EICS Workshop on Engineering Interactive Computer Systems with SCXML*, pp. 5–11. http://tuprints.ulb.tu-darmstadt.de/4053/.
34. Almeida, N., Silva, S., & Teixeira, A. (2004). Multimodal multi-device application supported by an SCXML state chart machine. In *Proceedings of the 1st EICS Workshop on Engineering Interactive Computer Systems with SCXML*. pp. 12–17. http://tuprints.ulb.tu-darmstadt.de/4053/.
35. Schnelle-Walka, D., Radomski, S., Lager, T., Barnett, J., Dahl, D., Mühlhäuser, M. (Eds.) (2014). *Proceedings of the 1st EICS Workshop on Engineering Interactive Computer Systems with SCXML*. Darmstadt: TU Darmstadt.
36. Burkhardt, F., Schröder, M., Baggia, P., Pelachaud, C., Peter, C., & Zovato, E. (2014). Emotion Markup Language (EmotionML) 1.0, W3C Recommendation. https://www.w3.org/TR/emotionml/. Accessed 15 Mar 2016.
37. Schnelle-Walka, D., Radeck-Arneth, S., & Striebinger, J. (2015). Multimodal dialog management in a smart home context with SCXML. In *Proceedings 2nd Workshop on Engineering Interactive Systems with SCXML*, Duisburg, DE.
38. López, G., Peláez, V., González, R., & Lobato, V. (2011). *Voice control in smart homes using distant microphones: A VoiceXML-based approach, in ambient intelligence*. Lecture Notes in Computer Science (Vol. 7040) (pp. 172–181). Berlin/Heidelberg: Springer.
39. Teixeira, A., Almeida, N., Pereira, C., & Oliveira, M. (2013). *W3C MMI architecture as a basis for enhanced interaction for ambient assisted living*. New York, NY: W3C Workshop on Rich Multimodal Application Development.
40. Sigüenza, A., Blanco, J. L., Bernat, J., & Hernández, L. A. (2010). Using SCXML for semantic sensor networks. In *Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN10)*. Workshop at the 9th International Semantic Web Conference (ISWC2010) - ISWC 2010 Workshops Volume V, Shanghai, China, pp. 33–48. http://ceur-ws.org/Vol-668/.

41. Radomski, S., & Schnelle-Walka, D. (2012). VoiceXML for pervasive environments. *International Journal of Mobile Human Computer Interaction, 4*(2), 18–36.
42. Schnelle-Walka, D., Radomski, S., & Mühlhäuser, M. (2015). Modern standards for VoiceXML in pervasive multimodal applications. In J. Lumsden (Ed.), *Emerging perspectives on the design, use, and evaluation of mobile and handheld devices*. IGI Global: http://www.igi-global.com/book/emerging-perspectives-design-use-evaluation/125520
43. Bühler, D., & Hamerich, S. W. (2005). Towards VoiceXML compilation for portable embedded applications in ubiquitous environments. In *Proceedings of Interspeech 2005*, Lisbon, PT, pp. 3397–3400. http://www.isca-speech.org/archive/interspeech_2005/i05_3397.html; http://www.isca-speech.org/archive/interspeech_2005/index.html.
44. Oshry, M., Adeeb, R., Baggia, P., Blackman, A., Bodell, M., Burke, D., et al. (2004). VoiceXML 2.0 Implementation Report. https://www.w3.org/Voice/2004/vxml-ir/. Accessed 1 Mar 2016.
45. Shanmugham, S., Monaco, P., & Eberman, B. (2006). A Media Resource Control Protocol (MRCP), RFC 4463—Informational. https://tools.ietf.org/html/rfc4463. Accessed 1 Mar 2016.

http://www.springer.com/978-3-319-42814-7

Multimodal Interaction with W3C Standards
Toward Natural User Interfaces to Everything
Dahl, D. (Ed.)
2017, XXIX, 422 p. 555 illus., 430 illus. in color.,
Hardcover
ISBN: 978-3-319-42814-7