How to Converse with a Virtual Agent by Speaking and Listening Using Standard W3C Languages

James A.. Larson

Intel Corporation
16055 SW Walker Rd, #402, Beaverton, OR 97006 USA
jim@larson-tech.com

Abstract

The PC, a successful multimedia device, is being extended to support multimodal input, including speech recognition, pen-based gestures, and video-as-input. The W3C Multimodal Interaction Framework provides development language standards for interactive speech and multimodal applications, enabling PC programmers to create multimodal applications for entertainment, education, and business.

1 Motivation for agents that speak and listen

Without speech a virtual world is a silent, cold place where communication is restricted to keystrokes or mouse movements. Speech makes the following virtual agents more interesting:

- *Tour guide*—A visual tour enables users to learn where things are. A visual tour with verbal commentary explains why things are there and what users can do with each thing. Just as narration adds a new dimension to an art gallery, narration informs the user about things the user sees.

- *Robots*—Verbally instructing a robot to perform an action may be more convenient than typing the instruction. In some computer games, users might use one hand to manipulate a joystick and the other hand to press keys on a keyboard or keypad. Voice provides a kind of "third hand" by enabling the user to speak voice commands. Voice commands also allow the user to escape from the "office position"—sitting in front of a monitor with one or both hands on the keyboard.

- *Language trainer*—Special speech recognition engines listen to how users pronounce words or sing notes. If the sound is not within acceptable bounds, the application presents instructions for improving the word or sound, possibly by speaking the correct word or singing the correct note. Applications include English as a second language, singing training, and public speaking training. One compelling application uses virtual agents to help deaf children learn to speak.[1]

- *Synthetic agents*—People use speech to interact with one another. It is natural to use speech to interact with virtual agents. The agents may lack intelligence, such as an "Elisa" agent that responds to comments by inserting a word from the user's previous comment into one of several arbitrary sentence templates. Or the agent may respond with a predefined answer when it hears a specific word in the question. However, complex discussions about complicated subjects are, in general, beyond the state of the art of natural language processing. Artificial interviews were pioneered by Scott Stevens.[2]

2 Motivation for standard languages

Standards are a double-edged sword. Standard languages hide many of the underlying technological details of how recognition systems work. They save developers' time and effort by reusing existing language processors rather than implementing processors from scratch for new languages. Also, there is the potential to swap modules with other developers and researchers. In general, standard languages promote reusability and portability. On the other hand, if the standard languages do not support the functions you need, they may be difficult to extend. This may limit the flexibility of your prototype or application. Sometimes standard languages stifle creativity.

3 Tour the W3C multimodal framework

To interact with a user, each virtual agent in a virtual environment needs an interaction manager. The interaction manager controls what the virtual agent presents to the user and which actions the user may make in response, as well as the style of interaction between the user and the agent. There are many different approaches to implementing an interaction manager. Popular approaches include a "fill in the form" approach as typified by VoiceXML 2.0,[3] the W3C standard language for telephony applications.

Figure 1 is a high-level representation of the W3C Multimodal Interaction Framework.[4] In addition to an interaction manager, each virtual agent has mechanisms to accept input from the user and present output to the user in addition to its own virtual agent functions. Figures 2 and 3 illustrate expansions to Figure 1. Figure 2 illustrates some of the many input devices and modules which may be available to users. We will discuss the related W3C languages below.

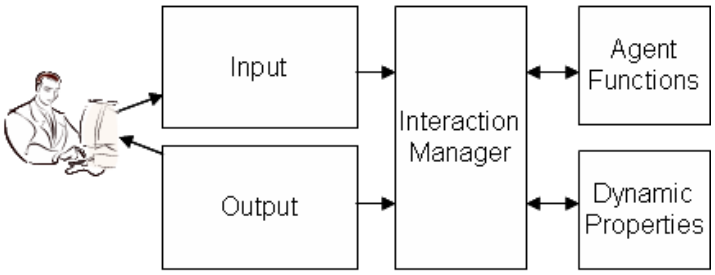

Figure 1. W3C Multimodal Interaction Framework

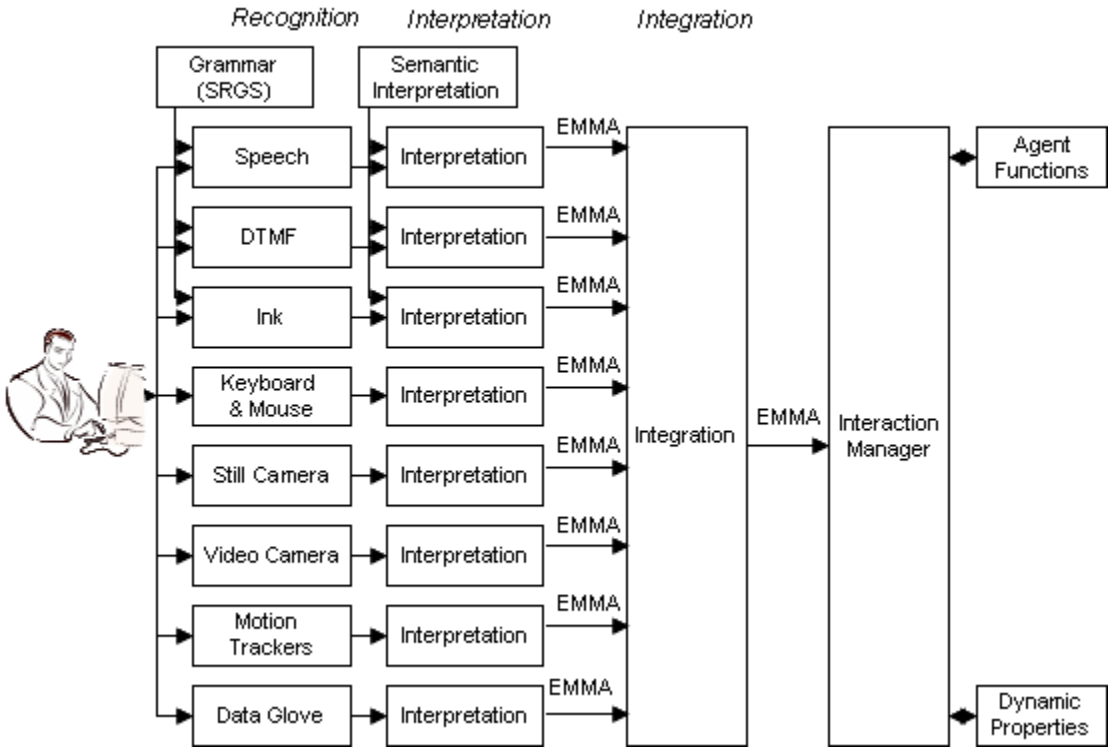

Figure 2. Input Devices and Modules

Accepting and processing speech from a user is complicated because speech recognition systems occasionally make mistakes. In the best situations, speech recognition systems will make recognition errors 3−5 percent of the time. A virtual agent listening to user speech needs to compensate for speech recognition errors by (1) prompting the user with specific questions and (2) listening only for prespecified words that may be spoken by the user. The collection of words recognized by a speech recognition system is called a *grammar*. Using grammars to guide the speech recognition system is beneficial because grammars greatly improve the accuracy of the speech recognition systems. Grammars enable the speech recognition system to work faster, which minimizes response delays. The W3C Speech Recognition Grammar Specification (SRGS)[5] is used widely in speech recognition applications and is a leading candidate for use by other recognition techniques.

Even with grammars, recognition systems still have errors such as: (1) the user fails to respond to the prompt before a predefined timeout period, (2) the user responds to the prompt by speaking words not in the grammar, or (3) the user asks for help. Virtual agent developers create error handlers for each of these errors to present additional prompts to the user to encourage the user to respond appropriately.

While each recognition system uses grammars to produce text based upon the user's input, the text from various recognizers may not be in a format to be integrated into a unified input representation. The W3C has established a standard language for representing input—Extended MultiModal Annotation (EMMA).[6] Like SRGS, EMMA is an XML language containing a textual representation of user input, as well as various attributes and qualifiers that assist the integration module to combine inputs from multiple sources into a unified input for the interaction manager. Developers may also use the W3C Semantic Interpretation Language[7] to inform the various recognition modules how to convert raw text from the recognition module to the appropriate EMMA format. The Semantic Interpretation Language is based on ECMAScript.

4 Example of an invisible virtual agent

To illustrate these languages, let's suppose that a virtual agent asks the user what to do next. For this example, we will assume that the virtual agent is not visible to the user; the user speaks and listens to a disembodied voice that assists the user to perform tasks. The user points with a stylus or pen to the location on a map on the screen while saying "zoom in here" to the virtual agent. The pen module records the time and point on the map to which the user points using EMMA notation such as:

```
<location start="10879959696666" end="1087995969999">
    <point> 42, 158 </point>
</location>
```

where the start and stop attribute represent the time when the user pointed to the map.

The grammar used by the speech recognition system contains the phrase "zoom in" and "here." The speech recognition system recognizes the words in the user phrase "zoom in here" and generates the EMMA notation such as:

```
<command>
    <actionstart="1087995961111 end="1087995964444">zoom-in</action>
    <location start="108799595555 end="1087995968888">here </location>
</command>
```

Many different techniques, including unification, may be used to integrate the EMMA <*location*> notation from the pen system with the EMMA notation for <*location*> from the speech recognition system. In this example, a simple time matching algorithm determines that the time the user pointed to the location matches the time that the user spoke "here," and merges the pen and speech EMMA information into a single EMMA notation:

```
<command>
    <action>zoom-in</action>
     <location>
```

```
        <point> 42, 158 </point>
      </location>
</command>
```

The EMMA command is transferred to the interaction manager, which determines that an enlarge command should be sent to the generation component. The generation component figures out how to enlarge the screen to display the enlarged area and sends the appropriate HTML commands to the HTML browser, which displays the revised map. Figure 3 illustrates some possible rendering components. The generation component also creates a command to the speech synthesis system, which produces audio that is presented to the user: "Zoom in complete; ready for next request."

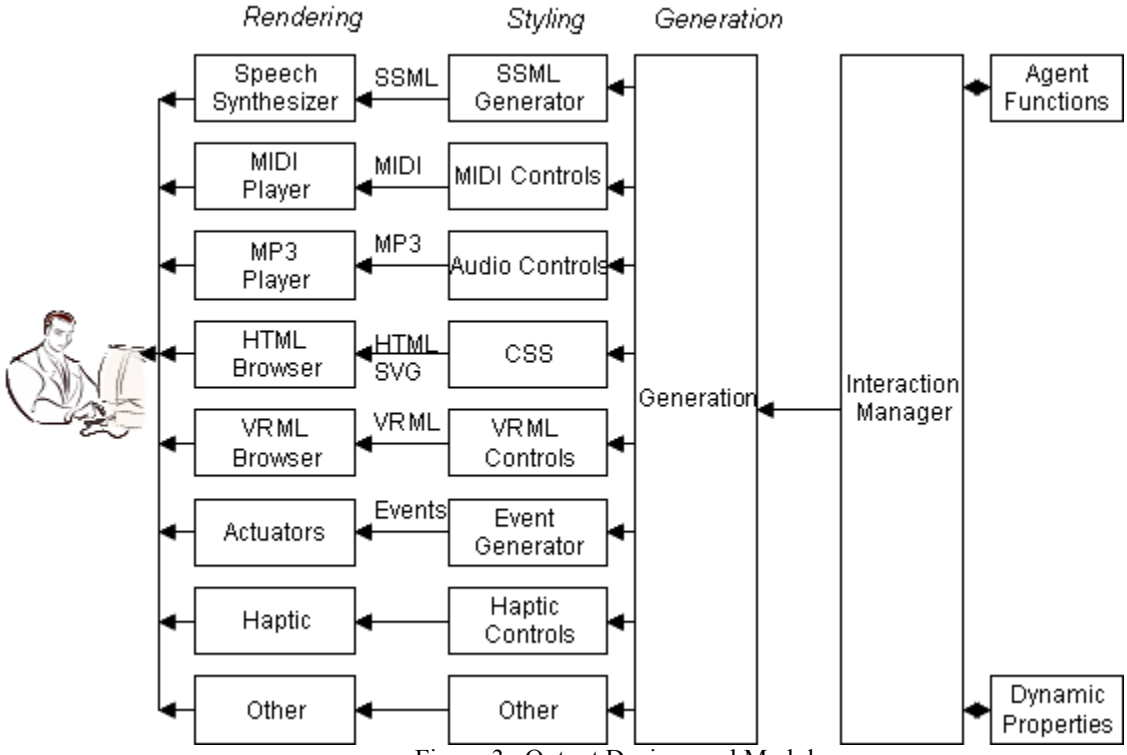

Figure 3. Output Devices and Modules

5 Example of a visible virtual agent

Some applications are more realistic if the user can see the virtual agent as well as converse with it. The visible virtual agent is represented by an avatar which reacts to user requests. For example, the user uses a pointing device to select a position and instructs the avatar to "Move here." In this example, capturing and integrating the input from two devices proceeds as in the invisible virtual agent above. The pointing module records the time and location to which the user points using EMMA notation such as:

```
<location start="10879959696666" end="1087995969999">
    <point> 42, 158 </point>
</location>
```

The speech recognition system recognizes the words in the user phrase "move here" and generates the EMMA notation such as:

```
<command>
    <action start="1087995961111 end="1087995964444">move</action>
```

```
    <location start="108799595555 end="1087995968888">here </location>
</command>
```

The integration module combined the two inputs and generates integrated input such as:

```
<command>
    <action>move</action>
     <location>
       <point> 42, 158 </point>
     </location>
</command>
```

Rather than just presenting a verbal message to the user, the generation module performs two actions: (1) reposition the avatar to the new location, which may involve animation showing the avatar walking to the new location, and (2) inform the user that the command is completed by presenting a verbal message to the user. To increase the realism of the avatar, the avatar's mouth should move synchronously with the verbal message. To achieve this, insert XML tags into the text to be synthesized. The synthesized text will instruct the avatar rendering subsystem to open the avatar's mouth as each word is spoken and close its mouth after each spoken word.

6 Available languages

If you plan to develop a virtual agent with which a user converses with one or more virtual agents, consider using the following W3C language specifications:

- *Speech Recognition Grammar Language (SRGS)*—an XML language for specifying words the speech recognition system listens for during a point in the dialog between the user and the virtual agent
- *Semantic Interpretation*—a JavaScript-like language for extracting and translating semantaics from the text produced by speech recognition engines, handwriting recognition engines, vision systems, etc.
- *InkML*—an XML language for specifying input from a pen, stylus, or pointing device
- *Extended MultiModal Annotation (EMMA)*—an XML language for representing the semantics of information recognized by recognition systems such as speech recognition, handwriting recognition, vision, etc.
- *Extended Hypertext Markup Language (XHTML)*[8]—an XML version of HTML for presenting visual information on screens
- *Speech Synthesis Markup Language (SSML)*[9]—an XML-based language used to render text as speech
- *Scalar Vector Graphics 1.2 (SVG)*[10]—an XML-based language for writing two-dimensional vector and mixed vector/raster graphics
- *Synchronized Multimedia Integration Language 2.0 (SMIL)*[11]—an XML-based language for writing interactive multimedia presentations

7 Using standard W3C languages

W3C standard languages enable the implementation of virtual agents that talk and listen, making your application more interesting and more effective.

The W3C Voice Browser and Multimodal Interaction Working Groups want to learn about your experiences using the W3C standardized languages. Developer input will impact the final form of the languages that have not yet reached the recommendation (W3C terminology for standard) and influence the design of future languages for developing multimodal applications. Send comments to the W3C public mailing lists (www-voice@w3.org or www.mmi@w3.org).

---

[1] http://www.oraldeafed.org/schools/tmos/news-050698.html
[2] http://www-2.cs.cmu.edu/~sms/
[3] http://www.w3.org/TR/2004/REC-voicexml20-20040316/

[4] http://www.w3.org/TR/mmi-framework/
[5] http://www.w3.org/TR/2004/REC-speech-grammar-20040316/
[6] http://www.w3.org/TR/emma/
[7] http://www.w3.org/TR/semantic-interpretation/
[8] http://www.w3.org/TR/2004/WD-xhtml2-20040722/
[9] http://www.w3.org/TR/2004/REC-speech-synthesis-20040907/
[10] http://www.w3.org/TR/SVG12/
[11] http://www.w3.org/TR/2005/REC-SMIL2-20050107/