# Multimodal Dialogmanagement in a Smart Home Context with SCXML

**Dirk Schnelle-Walka**
S1NN Gmbh & Co KG
Germany
dirk.schnelle-walka@s1nn.de

**Stephan Radeck-Arneth**
TU Darmstadt
Germany
stephan.radeck-arneth@cs.tu-darmstadt.de

**Jürgen Striebinger**
Cibek GmbH
Germany
juergen.striebinger@cibek.de

## ABSTRACT
The W3C MMI architecture is a recommendation for a common conceptualization for multimodal interaction focusing on the components involved and the messages passed between them. However, the standard does not cover integration of multimodal fusion and fission as addressed in the multitude of prototypical implementations, frameworks and applications prior to this standard. In this paper we describe an integration of current multimodal fusion and fission into this standard with an SCXML dialog manager in the context of smart homes.

## Author Keywords
multimodal architecture; dialog management; SCXML; MMI, EMMA, smart homes

## ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

## INTRODUCTION
Our homes are becoming smarter. Controlling devices in these smart homes by multiple modalities is already a reality. A typical architecture of such a smart home is shown in figure 1.

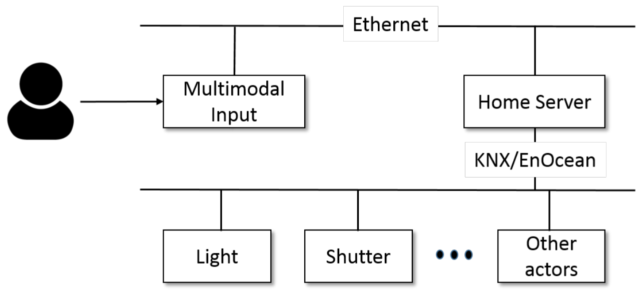

**Figure 1. Typical architecture of a Smart Home**

Multimodal input into these system should follow the concepts shown in figure 2.

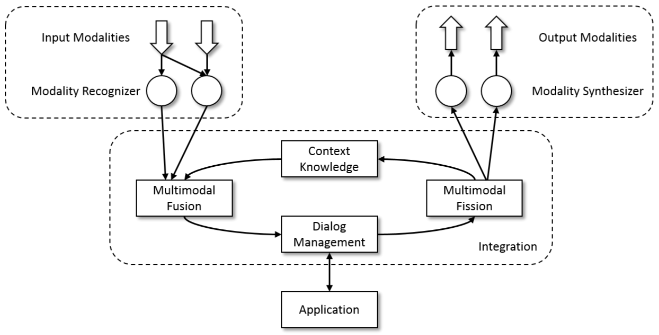

**Figure 2. High-level multimodal architecture, adapted from [5]**

Researchers as well as industry developed a plethora of deviations thereof for decades [5]. This makes it harder to reuse established knowledge and best-practices. With the advent of the *W3C Multimodal Architecture and Interfaces* recommendation [3] a promising candidate to standardize multimodal systems is available. A first analysis of the *nuts and bolts* is provided by [20]. However, the actual approach on how input coming from multiple sources is fused into a coherent meaning (multimodal fusion [12]) as well as state-of-the-art concepts on how to deliver information using more than a single available modality (multimodal fission [12]) is vaguely specified in the respective standards. Some first thoughts are described by Schnelle-Walka et al.in [21] and Ruf et al. [19]. This paper builds on [19]. While the latter deals on the aspects of implementing multimodal fusion and fission within the W3C architecture, this paper focuses on dialog management in the same setting. The W3C suggests the use of SCXML [2] as the dialog manager which has been proven to be suitable to decouple the control flow and presentation layer in dialog management [25]. It has been used in several applications to express dialog states [4] or to easily incorporate external information [22].

## RELATED WORK
A similar multimodal architecture was proposed by Fernandez et al. in [9] for the smart home. They combined the model-view-presenter pattern (MVP) with a service oriented paradigm. MVP is derived from the Model-view-controller pattern (MVC) and synchronizes different views with a presenter component. The fusion and fission functionality are

integrated in the presenter. The communication between presenter and modalities is realized via an event communication channel within the OSGi infrastructure. The user preferences, available entities and execution context are integrated in a multimodal interface instance. The prototype supports visual, haptic, gesture and voice. The platform uses OSGi service and is implemented as an OSGi service factory. In comparison to our work, the components of the resulting architecture are coupled and some components are not open-source.

In [6] Cohen et al. describe Sketch-Thru-Plan (STP) as another, but more recent closed-source multimodal system. It combines gesture, handwriting, touch and speech. For robust command & control speech recognition, a grammar based speech recognition was selected. STP enables collaborative planning between several users through the multimodal interface. For speech recognition the Microsoft Speech Engine (SAPI) of Windows 7/8 is used. Push-to-talk (PTT) is correlated with the interaction of the screen by touch events to reduce the effects of conversational speech. Cohen et al. consider statistical language models (SLM) for future extensions, but the authors consider grammar-based recognition to be sufficient, since the users are trained to use a specific vocabulary. However, STP is not designed for reusability.

Another WWHT-based [18] multimodal fission contribution were introduced by Costa et al. [7]. The developed GUIDE system supports elderly users to communicate through employing appropriate modalities. The system uses video, audio and haptic interfaces and is integrated with the television. The multimodal fission selects the interface best suited for the communication for several users individually. However the system focus on multimodal fission. The integration in a multimodal architecture including fusion is out of scope for the work described in this paper.

As an alternate approach to WWHT, Pitsikalis et al. [14] trained Hidden Markov Models for multimodal fission. HMMs also proofed to be useful when fusing input from multiple modalities. Potamianos et al. [15] rely on HMM for audiovisual ASR, i.e. multimodal fusion. We consider HMMs for a later stage of the project. In order to actually train the models sufficient data is required which may be obtained by the rule based approach described later on.

## MULTIMODAL ARCHITECTURE WITH THE W3C

The recommended architecture by the W3C decomposes a multimodal application into an interconnected structure of *interaction managers* (IM) for dialog control and *modality components* (MC) for in- and output. An implementation is formulated as a set of control documents reflected e.g. in SCXML for the interaction managers and a set of presentation documents with modality-specific markup for the modality components [21]. A topmost root controller document articulates the global dialog and instantiates modality components as required. Each modality component can, in turn, again be an interaction manager, managing more fine granular concerns of dialog control, such as error correction or even sensor fusion/fission.

### Multimodal Fusion

In multimodal systems users are capable to express their dialog move by more than a unique modality. Multimodal fusion synthesizes the input arriving from the varying modalities into a unified semantic interpretation that declares the user's interaction intent [1]. Multimodal fusion mainly serves two purposes (i) provide an abstraction that enables usage of the provided information regardless of the used modality and (ii) derive a meaning thereof. Hence, the fusion engine needs an application independent representation of the current application context to infer meaning [8] which is available with EMMA[1]. Following the MMI W3C architectural pattern, the MCs send their input as EMMA events to the upper IM for further refinement.

For the fusion engine, Bui [5] infers in his survey paper a high-level view onto multi-level fusion as shown in Figure 3. According to Bui, this type of fusion consists of two main lev-

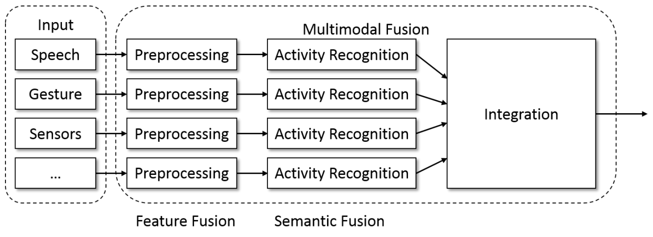

**Figure 3. Multi-Level fusion, adapted from [5]**

els of fusion: (i) feature-level fusion and (ii) semantic-level fusion. In the feature-level fusion the features as an output of the modality component are transformed into a modality-independent format. The subsequent semantic fusion decides, in a frame-based fusion manner [24], if the provided information is already sufficient to execute an action. In case it is insufficient, it is stored for later integration. The last integration step fuses the received semantic information into a coherent meaning. If required information is not received within a predefined time span, the available information is also forwarded to the upper IM, e.g. to initiate actions to ask explicitly for the missing piece.

### Multimodal Fission

Multimodal systems provide for the consolidated or alternative application of separate input modalities and to choose output modalities most suitable for a given context. Alongside the benefits nominated by Oviatt and Cohen [13], particularly that the system increases stability and is fewer prone to errors, a suitable selection and combining of output modalities has the opportunity to facilitate or allow communication. One concept for fusion was established as WWHT by Rousseau et al. [17] (see Figure 4). (i) **W**hat is the information to render, (ii) **W**hich modalities should we utilize to present this information, (iii) **H**ow to roll out the information applying these modalities and (iv) and **T**hen, how to manage the progress of the deriving presentation [18]. These questions also shape the processing stages during fission and are described in [21] as follows:

[1]`http://www.w3.org/TR/EMMA/`

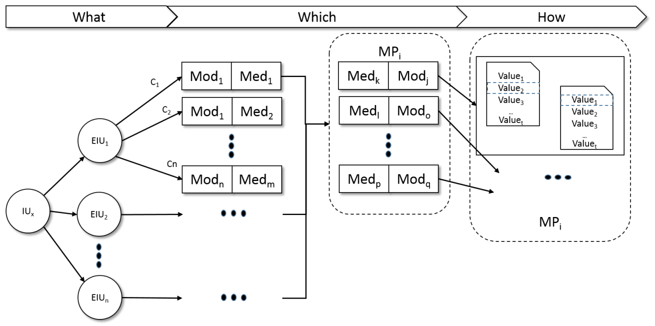

**Figure 4. Multi-Level Fission with WWHT, derived from [17]**

In the *What* level, a message is split into basic information unit.

The *Which* level offers for the choice of suitable modalities founded on rules and the affordances of the unique modalities.

In the *How* level, the output is actually rendered. This is achieved by the specified MCs.

### Dialogmanagement with SCXML

Throughout the W3C MMI architecture recommendation, SCXML is stated as the preferred option for controller documents. At its origin, SCXML is a specification of finite state machine with parallel and nested states, as defined by Harel et al [11]. An SCXML implementation can, e.g., initiate actions when states are entered or transitions are taken. These actions include updating an internal data model or submitting events to other components. Transitions can be guarded by conditions, their evaluation is elicited via internal or external events.

In essence, SCXML is able to design dialog behavior even in the uppermost IM as the current application or nested in MCs e.g. for form filling or local error correction.

## IMPLEMENTATION

In this section, we describe our implementation and some short-comings as a proof-of concept of the theoretical approaches mentioned above. We rely on OSGi as it used in the smart home controller by Cibek[2]. OSGi has proven to provide sufficient flexibility to address the different settings in actual deployments of smart homes. Multimodal fusion and fission are already described in [19] but are translated into English.

### Multimodal Fusion

The nature of the WWHT approach to multimodal fusion is rule-based. A suitable framework for such rule-based systems is JBoss Drools[3]. It can be triggered by an OSGi bundle with an HTTP server to receive incoming MMI POST requests. The OSGi bundle injects the request into the knowledge base. Usually these MMI events will feature EMMA content that are extracted and injected as objects into the knowledge base. In the following, the request can be processed within the multimodal fusion via specialized rules. We used a separate file per modality to keep an overview. Consecutive processing includes further refinement of the received object and execution of Java code. At this stage of development we offered support for touch, speech, gesture and sensor input.

We had serious problems to carefully design the rules in order to avoid endless loops. Special care has to be taken to remove objects from the knowledge base when they are no longer needed.

### Multimodal Fission

After an action has been executed an additional output may be requested or the application has to query for additional information. Note that an action is already one type of output so that an output is not needed in all cases.

We considered the modalities: text based output on a TV screen or a wall-mounted monitor, text-to-speech and avatar. Following the concept of WWHT the election in the *Which* stage first considered all available modalities. Based on the user's abilities, e.g. visual or aural impairment which are not uncommon in AAL settings, all modality-medium pairs with the corresponding inaccessible medium were removed. Then, all pairs that did not match the current context, were filtered. For instance, a sensor notified that the user started the interaction at a wall-mounted display and moved to the TV set. In this case, all subsequent output would be displayed using the TV screen.

In order to render the output in the *How* stage, corresponding MMI messages carry modality-specific markup, e.g. VoiceXML [?] for spoken interaction or BML [23] for the Avatar, in the data element.

### Dialogmanagement with SCXML

Following the suggestion of the W3C MMI architectural pattern, the topmost interaction manager deals with high level tasks while fine-granular concerns of dialog control are handled by modality components acting as interaction managers. We employed uSCXML[4] as the SCXML interpreter. For a smart home control scenario the main task of the interpreter is on adapting to the current context, represented as states in an SCXML document (see figure 5). These states feature

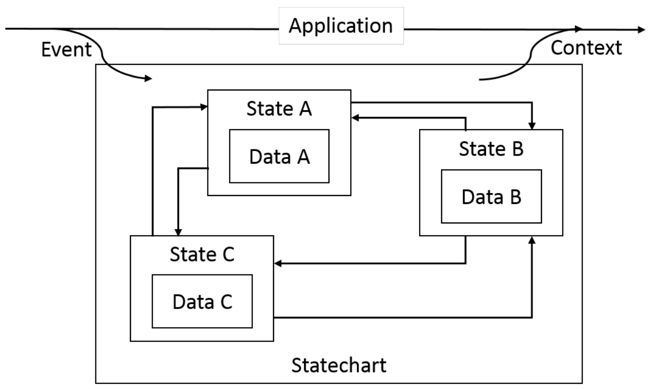

**Figure 5. State-based context changer**

[2]`http://www.cibek.de`

[3]`http://www.drools.org/`

[4]`http://github.com/tklab-tud/uscxml`

information relevant for the current situation. For example, consider grammars for a speech recognizer. Usually, speech input is recognized with higher accuracy if grammars are used [16, 10]. Incoming data from the fusion engine may cause a context switch, thus updating the application with new contextual information. The following SCXML snippet listing 1 shows how grammar can be injected into a speech recognizer using the `<send>` tag when the state is entered.

**Listing 1. SCXML snippet**

```
<scxml version="1.0">
 <state id="State_A">
  <onentry>
   <send type="application">
    <content>
     <output>[...]</output>
     <grammar>
      <rule>[...]</rule>
      <rule>[...]</rule>
     </grammar>
     [...]
    </content>
   </send>
  </onentry>
 <transition event="event.B" target="State_B"/>
</state>

<state id="state_B">
 [...]
</state>

[...]
</scxml>
```

The voice modality is handled by VoiceXML. Here we employed JVoiceXML [20] since it already features MMI communication capabilities. VoiceXML is a dialog manager on its own. For the topmost interaction SCXML-based interaction manager VoiceXML serves as a modality component that is capable of handling synthesized spoken output and speech input. This is achieved by sending corresponding MMI events to the voice browser with VoiceXML snippets in their data element. The grammars mentioned above are part of it. As a dialog manager, the voice browser itself is another interaction manager that independently handles the spoken dialog. This includes error handling within the modality and is in line with the principle of the *Russian Doll* as mentioned in the W3C MMI standard. Local error management however violates the concepts of multimodal error correction according to the architecture described in the high-level multimodal architecture (ref. to figure 2). Following the general idea that is described here, error management should be another iteration through the complete processing pipeline to allow for error correction by other modalities. We solved this by issuing MMI extension notifications to the multimodal fission in case of detected error correction within VoiceXML as shown in figure 6. This way, it is possible to exploit VoiceXML's

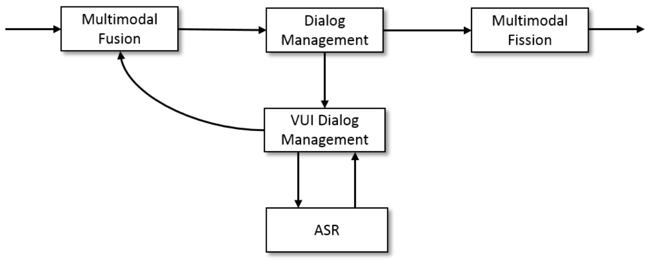

**Figure 6. Use of multiple interaction managers**

built-in capabilities to handle errors while being able, e.g., to show a puzzled avatar on a screen.

## WALKTHROUGH

The following exemplifies the usage of the system. Imagine the following a scenario. In the morning, Alex enters the kitchen for the first time. The wall-mounted monitor shows how an avatar enters the scene to greet him: "Good Morning Alex. How may I help you?" As it is still dark inside the kitchen, Alex asks top open the shutter. While they are opening, the Avatar asks: "OK. Anything else I can help you with?"

This is reflected by our system as follows: Alex enters the kitchen and is recognized by a motion sensor. We employed an OPUS greenNet motion sensor for this purpose. This is fed into the fusion engine as an MMI new context request to initialize the dialog. The data attribute contains the location *kitchen* encoded in an EMMA document. A drools rule extracts the location data and updates the knowledge source. Additionally, it issues a start request to the upper IM, the context changer. The context changer is defined in a separate OSGi bundle with its own SCXML document *default.scxml* (see Listing 2).

**Listing 2. Excerpt from default.scxml with information for the context changer**

```
<state id="default−state">
 <onentry>
  <send type="drools">
   <content>
    <type>VXML</type>
    <currentState>default−state</currentState>
    ...
    <transitions>
     <contextChanges>
      <contextChange>cooking</contextChange>
      <contextChange>select−recipe</contextChange>
      <contextChange>morning</contextChange>
      ...
     </contextChanges>
     ...
    </transitions>
     </content>
  </send>
 </onentry>
 <transition event="event.cooking"
  target="cooking" />
 <transition event="event.select−recipe"
```

```
  target=”cooking” />
 <transition event=”event.morning”
  target=”morning” />
 ...
</state>
```

We employed a special sort of IO processors that are used within the OSGi system. The context changer loads the corresponding *morning.scxml* file into uSCXML for further processing. First, the SCXML is validated to ensure that all transitions are valid. In addition it guarantees that interpretation of the SCXML document is only executed if all employed IO processors are available. This needs to be done to account for the modular nature of OSGi and to ensure that it can communicate with the environment.

The dedicated *morning* process then issues a start request to the multimodal fission bundle to let the avatar appear on the display and greet the user. The multimodal fission knows about the available modalities, avatar and speech output and sends a corresponding BML document to the avatar and a VoiceXML document to JVoiceXML. Knowledge about the available modalities and user preferences is configured as rules in a similar fashion as the multimodal fusion. The avatar is configured as a TTS engine in JVoiceXML which enables parallel use of this modality from within VoiceXML and by the multimodal fission engine.

The avatar movements of lower granularity follow a state-based approach to control the movements at a higher level in the context changer. This includes, appearance on the screen, leaving and some more high-level activities like putting on a cooking hat to adapt to the current context. Controlling the avatar at a lower level, like looking puzzled if a user input could not be matched or TTS output is triggered by dedicated SCXML scripts per dialog. Additionally, the avatar is used as output from JVoiceXML, requiring synchronization of lip movements and TTS output. We rely on the capabilities of the avatar's BML interpreter to merge this into smooth movements.

As a result the avatar appears on the display (controlled by uSCXML) and greets the user “Good morning.” (controlled by JVoiceXML) as shown in figure 7. The grammars con-

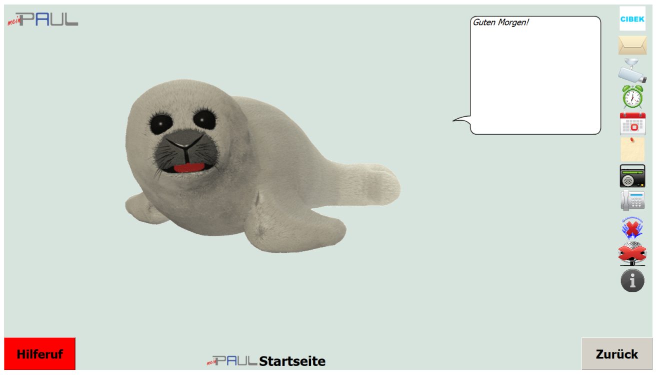

**Figure 7. The avatar appears on the screen and greets the user**

tained in the SCXML, as shown in listing 1, to continue the dialog are sent to JVoiceXML in the same start request. The VoiceXML document contains a single field with a prompt “How may I help you” and expects input for the received grammars. A dialog turn has one JVoiceXML session at maximum. Additional turns require processing by the complete chain from fusion to fission. Alternatively, it may be canceled by a cancel request if voice input is no longer required.

Once the user utters “Open the shutters” the command field gets filled. The dialog terminates and sends out a done notification containing the semantic interpretation as is is obtained from the grammar encoded as EMMA in the data tag. Here, we violate the concepts of MMI and do not send it to the invoking IM but to the multimodal fusion engine for further processing. This way, it is possible to keep the multimodal processing chain. Again the fusion engine fuses this command with the location information that is in the knowledge base and forwards the result to the dialog manager to actually execute the command. In this case, the SCXML triggers another OSGi bundle to send a corresponding command over KNX to the shutter (see figure 1). In addition, the avatar looks up and states “OK” following the same pipeline as described above.

## SUMMARY AND CONCLUSION

In this paper we described a multimodal system to control smart homes employing open source software components. Some of the concepts have been described in previous publications which we now integrated into a fully functional prototype. It follows the W3C MMI architectural recommendation and integrates proven theoretical concepts of multimodal fusion and fission. SCXML turned out to be a good candidate for the topmost interaction manager. However, we had to violate the principle of a tree structure as suggested in the W3C MMI standard to enable multimodal error correction. Our final architecture is shown in Figure 8.

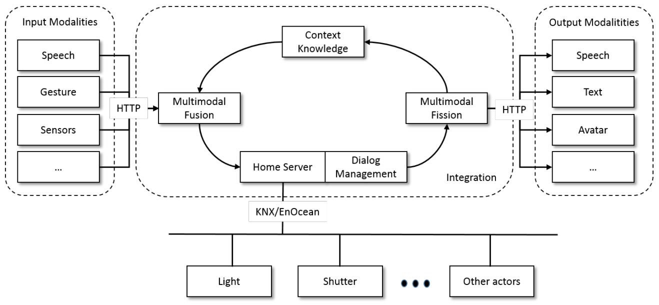

**Figure 8. Implemented architecture**

It is our hope that the results presented here stimulate the discussion around the W3C MMI standard to integrate multimodal fusion and fission as well as the interplay between multiple dialog managers within an architecture following the standard.

Currently, we are about to prepare a user study to evaluate the interaction in an ongoing project.

## Acknowledgments

This work was partly supported by the Bundesministerium für Bildung und Forschung (BMBF), Germany under the programme "KMU-innovativ: Mensch-Technik-Interaktion für den demografischen Wandel".

## REFERENCES

1. Atrey, P. K., Hossain, M. A., Saddik, A. E., and Kankanhalli, M. S. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems 16*, 6 (2010), 345–379.
2. Barnett, J., Akolkar, R., Auburn, R., Bodell, M., Burnett, D. C., Carter, J., McGlashan, S., Lager, T., Helbing, M., Hosn, R., Raman, T., Reifenrath, K., Rosenthal, N., and Roxendal, J. State chart XML (SCXML): State machine notation for control abstraction. W3C working draft, W3C, May 2014. http://www.w3.org/TR/2014/WD-scxml-20140529/.
3. Bodell, M., Dahl, D., Kliche, I., Larson, J., Porter, B., Raggett, D., Raman, T., Rodriguez, B. H., Selvari, M., Tumuluri, R., Wahbe, A., Wiechno, P., and Yudkowsky, M. Multimodal Architecture and Interfaces. W3C recommendation, W3C, Oct. 2012. http://www.w3.org/TR/mmi-arch/.
4. Brusk, J., Lager, T., Hjalmarsson, A., and Wik, P. DEAL: dialogue management in SCXML for believable game characters. In *Proceedings of the 2007 conference on Future Play*, ACM (2007), 137–144.
5. Bui, T. Multimodal Dialogue Management - State of the art. Tech. Rep. TR-CTI, Enschede, Jan. 2006.
6. Cohen, P. R., Kaiser, E. C., Buchanan, M. C., Lind, S., Corrigan, M. J., and Wesson, R. M. Sketch-Thru-Plan: A Multimodal Interface for Command and Control. In *Communications of the ACM*, vol. 58 (Apr. 2015), 56–65.
7. Costa, D., and Duarte, C. Adapting Multimodal Fission to Users Abilities. In *Universal Access in Human-Computer Interaction. Design for All and eInclusion, 6th International Conference, UAHCI*, Springer (2011).
8. Dourlens, S., Ramdane-Cherif, A., and Monacelli, E. Multi levels semantic architecture for multimodal interaction. *Applied Intelligence* (2013), 1–14.
9. Fernandez, M., Pelaez, V., Lopez, G., Carus, J., and Lobato, V. Multimodal Interfaces for the Smart Home: Findings in the Process from Architectural Design to User Evaluation. *Ubiquitous Computing and Ambient Intelligence* (2012), 173–180.
10. Gorrell, G., Lewin, I., and Rayner, M. Adding intelligent help to mixed-initiative spoken dialogue systems. In *ACL-02 Companion Volume to the Proceedings of the Conference* (2002).
11. Harel, D., and Politi, M. *Modeling Reactive Systems with Statecharts: The Statemate Approach.* McGraw-Hill, Inc., Aug. 1998.
12. Landragin, F. Physical, semantic and pragmatic levels for multimodal fusion and fission. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7)* (2007), 346–350.
13. Oviatt, S. L., and Cohen, P. R. Multimodal Interfaces That Process What Comes Naturally. *Communications of the ACM 43*, 3 (2000), 45–53.
14. Pitsikalis, V., Katsamanis, A., and Papandreou, G. Adaptive multimodal fusion by uncertainty compensation. In *IEEE Transactions on Audio, Speech, and Language Processing* (2009).
15. Potamianos, G., Huang, J., and Marcheret, E. e. a. Far-field multimodal speech processing and conversational interaction in smart spaces. *2008 Hands-free Speech Communication and Microphone Arrays, Proceedings* (2008), 119–123.
16. Rayner, E., Bouillon, P., Chatzichrisafis, N., Hockey, B. A., Santaholma, M. E., Starlander, M., Isahara, H., Kanzaki, K., and Nakao, Y. A methodology for comparing grammar-based and robust approaches to speech understanding. *Proceedings of Eurospeech-Interspeech, 9th European Conference on Speech Communication and Technology* (2005), 1103–1107.
17. Rousseau, C., Bellik, Y., and Vernier, F. WWHT: Un modèle conceptuel pour la présentation multimodale d'information. In *Proceedings of the 17th international conference on Francophone sur l'Interaction Homme-Machine*, ACM (2005), 59–66.
18. Rousseau, C., Bellik, Y., Vernier, F., and Bazalgette, D. A Framework for the Intelligent Multimodal Presentation of Information. *Signal Processing 86*, 12 (2006), 3696–3713.
19. Ruf, C., Striebinger, J., and Schnelle-Walka, D. Sprach- und Gestensteuerung fr das Smart Home. *JavaSPEKTRUM* (Mar. 2015).
20. Schnelle-Walka, D., Radomski, S., and Mühlhäuser, M. JVoiceXML as a Modality Component in the W3C Multimodal Architecture. *Journal on Multimodal User Interfaces* (Apr. 2013).
21. Schnelle-Walka, D., Radomski, S., and Mühlhäuser, M. Multimodal Fusion and Fission within W3C Standards for Nonverbal Communication with Blind Persons. In *Computers Helping People with Special Needs, 14th International Conference on Computers Helping People with Special Needs*, Springer (July 2014), 209–213.
22. Sigüenza Izquierdo, Á., Blanco Murillo, J. L., Bernat Vercher, J., and Hernández Gómez, L. A. Using SCXML to integrate semantic sensor information into context-aware user interfaces. In *International Workshop on Semantic Sensor Web, In conjunction with IC3K 2010*, Telecomunicacion (2011).

23. van Welbergen et al., H. BML 1.0 Standard. Standard, SAIBA, Apr. 2014. http://www.mindmakers.org/projects/bml-1-0/wiki#BML-10-Standard.

24. Vo, M. T., and Wood, C. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (1996).

25. Wilcock, G. SCXML and voice interfaces. In *3rd Baltic Conference on Human Language Technologies, Kaunas, Lithuania* (2007).