AsapRealizer 2.0: The Next Steps in Fluent Behavior Realization for ECAs

Herwin van Welbergen, Ramin Yaghoubzadeh, and Stefan Kopp *

Sociable Agents Group, CITEC, Fac. of Technology, Bielefeld University

Abstract. Natural human interaction is highly dynamic and responsive: interlocutors produce utterances incrementally, smoothly switch speaking turns with virtually no delay, make use of on-the-fly adaptation and (self) interruptions, execute movement in tight synchrony, etc. We present the conglomeration of our research efforts in enabling the realization of such fluent interactions for Embodied Conversational Agents in the behavior realizer 'AsapRealizer 2.0' and show how it provides fluent realization capabilities that go beyond the state-of-the-art.

Keywords: AsapRealizer, Fluent Behavior Realization, BML 1.0, BMLA

1 Introduction

Human conversations are highly dynamic, responsive interactions. In such interactions, utterances are produced incrementally [1,2], subject to on-the-fly adaptation (e.g. speaking louder to keep a challenged turn) and (self) interruptions. While listening, plans for next speaking contributions are constructed, allowing very rapid turn transitions [3]. Furthermore, conversations are characterized by interpersonal synchrony [4], including the alignment of movement rhythm (e.g. alignment of walking rhythm, exercise motion, postural sway or even breathing patterns) and smooth meshing/intertwining of behavior between the interlocutors (e.g., smooth turn-taking and backchannel feedback).

To enable such fluent interaction in Embodied Conversational Agents (ECAs) we must steer away from the traditional turn-based non-incremental interaction paradigm in which the ECA first fully analyzes user contributions and subsequently fully plans its contribution, which is then executed entirely ballistically (providing no adaptation in nor interruption of ongoing behavior). Although fluent interaction has recently become a hot research topic (see [5] for an overview), others have focused their research efforts mainly at incremental dialogue understanding and behavior planning (for example: [6]). Truly fluent interaction in ECAs *additionally* requires highly flexible behavior realization. This is a topic we have given a great deal of attention in the development of ECAs and dialogue systems within our two research groups (the Sociable Agents

* This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication. The development of AsapRealizer 1.0 was conducted in collaboration with the Human Media Interaction group at the University of Twente.

Group at Bielefeld University and the Human Media Interaction Group at the University of Twente) in the last 10 years.

We present the conglomeration of our research in AsapRealizer, a BML 1.0 behavior realizer that unifies the fluent behavior realization capabilities that were required in several projects tackled within our research groups. AsapRealizer builds on two existing realizers from both our groups, that have focused on either incremental multimodal utterance construction [7] or interactional coordination [8] as isolated problems. In earlier work [9] we show that by combining the realization capabilities for incremental multimodal behavior construction and interactional coordination in a single realizer, we can enable interaction scenarios that go beyond the capabilities of these individual realizers.

While the preliminary version of AsapRealizer (1.0) [9] provided the specification of and an architecture framework for fluent behavior realization, it provided its implementation mainly for gesture. AsapRealizer 2.0 –discussed in this paper– generalizes AsapRealizer 1.0's capabilities by additionally implementing fluent realization capabilities for speech, gaze and facial expression. This required us to implement a novel adaptive gaze model and to embed a recently developed incremental and adaptive Text-To-Speech system [10].

In this paper we summarize and motivate AsapRealizer fluent realization capabilities and provide a more detailed description of the implementation of its novel capabilities. Additionally, we contribute a thorough comparison of AsapRealizer with the state-of-the-art in behavior realization in terms of incremental realization plan construction and on the fly adaptation of behavior.

2 Motivation

Over the course of more than 10 years, our research groups have developed several research applications in which fluent behavior realization played a key role (see Fig. 1 for some examples).

Fig. 1: Fluent behavior realization applications. Left: the virtual construction instructor Max, middle: the virtual conductor (photo: Henk Postma, Stenden Hogeschool), right: the daily assistant Billie

One of our first applications requiring fluent interaction was the virtual construction instructor Max [11]. Max cooperated with users in a construction task in Virtual Reality. While the interaction with Max was mostly turn-based, he still required incremental

behavior realization from a fluent stream of multiple utterances, in which gestures were to be fluently connected. This required its realizer to adjust the timing and shape of ongoing behavior.

At the University of Twente, we have been interested in investigating and implementing ECA behavior for interactions in which there is simultaneous expressive behavior by a human interlocutor and an ECA. Prototypical applications for such behavior are a virtual dancer that aligns her movement both to movement of a human dance partner and to live music, a virtual orchestra conductor that conducts a real orchestra, and a virtual trainer that exercises together with a human trainee (see [12] for an overview). In all these applications the ECA exhibits tight coordination of movement timing with the interlocutor, which required both synchronization based on *predictions* from the ECA's own performance and that of the interlocutor and very flexible behavior realization in which the timing and shape (e.g. amplitude of movement) of behavior can be adapted on the fly. In Bielefeld, similar adaptivity in behavior realization was required in robotic speech-gesture synchronization [13]: the unreliable timing of robotic movement required an adaptable execution process in which the timing of speech is modified on-the-fly.

In more traditional dialog-based interfaces, both of our research groups have looked at going beyond turn-based interaction paradigms and providing attentive speakers that adapt their ongoing behavior based on interlocutor feedback and active listeners that continuously signal their understanding to a human speaker [14,15,16]. The implementation of behavior realization for attentive speakers is especially challenging. It requires graceful interruption of utterances (in speech, gesture, gaze, facial expression), the modification of ongoing behavior (e.g. speaking louder to keep the turn), partial rephrasing of ongoing behavior realizations (while keeping the rest of the realization plan intact) and the employment of several strategies to provide (the illusion of) very reactive behavior. The latter include the use of fillers (e.g. uhm) to keep the turn and gain some time for behavior planning, and the preplanning of (multiple alternative) utterances for instant later execution.

Beyond application in laboratory settings, we are currently employing AsapRealizer in a cooperation project (VASA/Verstanden) with a health care provider. The project aims to ascertain the feasibility (and acceptability) of spoken-language controlled daily assistants for people with cognitive impairments. Starting from the first Wizard-of-Oz prestudies, interruptible generation was employed, which in combination with the low latency led to easy recovery in situations with disputed floor. Quick and seamless generation and incremental processing are of paramount importance for the system, since participants understand small sequential chunks of information best [17].

In summary, incremental behavior plan construction, graceful interruption and adaptation of ongoing behavior have been essential capabilities for fluent behavior realization with AsapRealizer (see also Table 1). Adaptations of behavior may be steered: 1) by the behavior planner (top-down adaptations), for example when requesting the ECA to speak louder, 2) by the AsapRealizer itself (bottom-up adaptations), for example to achieve co-articulation between gestures on the fly and 3) by external constraints from the environment, for example to align the ECAs exercise movement to that of a user.

Capability	Motivating Project(s)
Incremental plan construction	[11,12,13,14,15,16,17]
Graceful interruption	[14,15,16,17]
Top-down adaptation of ongoing behavior	[14,15,16]
Bottom up adaptation of ongoing behavior	[11,13]
Adaptations of ongoing behavior to the changing environment	[12]

Table 1: Motivation for AsapRealizer's fluent behavior realization capabilities.

3 Fluent Behavior Realization Capabilities

We have developed the BML extension BMLA [5] to allow the specification of AsapRealizer's fluent behavior realization capabilities. Within AsapRealizer, these capabilities are implemented both with a flexible architecture (see Fig. 2) to manage the behavior plan and with the implementation of modality specific flexibility for the behaviors themselves.

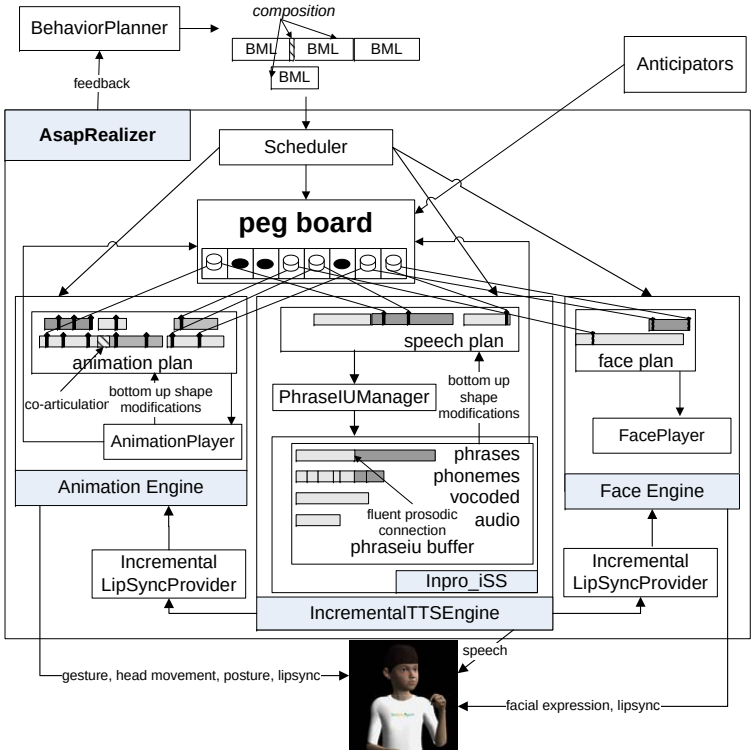

Fig. 2: AsapRealizer 2.0's architecture

3.1 Incremental Plan Construction

Constructing a plan out of small increments allows AsapRealizer to start realizing behavior early and to do part of its plan construction while it is executing previous increments, thus making the ECA more reactive. Such incrementality has a biological basis: psycholinguistics has identified incrementality as an important property of fluent human language production (e.g. [2]). While speech can be seen as a concatenation of increments that are smaller than a sentence [1], the increments connect smoothly, and as a result prosodic properties that are *suprasegmental* (e.g. rhythm, sentence intonation) are observed in speech. A similar incrementality has been proposed in gesture research. According to McNeill's segmentation hypothesis [18], speech and gesture are produced in successive increments that each contain one prosodic phrase in speech and one co-expressive gesture phrase. Gesture movement between the strokes of two successive gestures (in two successive increments) depends on their relative timing, and may range from retracting the arm to an in-between rest position, to a smooth direct transition movement to the next gesture.

Incremental plan construction in AsapRealizer therefore not only requires specifying sequential relations between the increments, but also the implementation of smooth connections between the increments. AsapRealizer supports incremental plan construction from increments specified in BML blocks. Since the occurrence of smooth connection between increments (or the lack thereof) can well have a communicative function (e.g. marking information boundaries [19]) we allow the behavior planner to have control over whether or not it should occur. To this end, we provide specification mechanisms in BMLA, that allow a detailed specification of how BML blocks are to be composed (see [5] for examples and syntax). AsapRealizer's automatic gesture coarticulation is implemented using functionality first developed for ACE [7], which was generalized to BML (see [9] for implementation details). Fluent prosodic connection is achieved by embedding the Inpro_iSS [10] incremental Text-To-Speech (TTS) system. Unlike current mainstream TTS systems that requires a the full utterance in advance to generate its intonation, Inpro_iSS can incrementally construct utterances with appropriate intonation from small increments. Fluent speech realization is achieved by the implementation of an IncrementalSpeechEngine that embeds Inpro_iSS. The IncrementalSpeechEngine executes a speech plan containing IncrementalSpeechUnits, which are constructed on the basis of the BML specification of the desired speech. In Inpro_iSS, the current utterance plan is represented by a buffer of PhraseIUs. PhraseIUs are Incremental Units that typically represent part of a sentence. This PhraseIU buffer forms a stretch of continuous speech to be uttered. When devising a phrase's prosody, the whole buffer is used as context. For each IncrementalSpeechUnit, a corresponding PhraseIU is created in Inpro_iSS. AsapRealizer's PhraseIUManager fills (and empties) the buffer based on the (predicted) timing of the IncrementalSpeechUnits in the plan. Its goal is to keep the buffer as full as possible, allowing the maximum quality of prosody, given the currently known speech plan. The timing of the ongoing utterance is subject to change (e.g. subject to timing changes caused by prosody enhancements or parameter changes). These timing changes are automatically communicated to behavior in aligned other modalities (e.g. lipsync, gesture) using AsapRealizer's PegBoard (see also Section 3.3).

In addition to incremental production, humans employ fillers (e.g. uhm) to keep or take a turn without having a plan at hand [3]. Bauman and Schlangen show that a dialog system that uses such fillers is preferred by users over one that waits with speaking until all information is available [20]. However, in certain situations the usage of those fillers, may communicate unintended communicative functions and should be avoided. We therefore allow the behavior planner to specify both the occurrence of fillers and whether or not they are to be automatically skipped if a new increment concatenated after the filler is scheduled on time. The latter bears resemblance to the skipping of the retraction phase of gestures and is implemented in a similar fashion. Automatic

BML Example 1 Incremental speech construction.

```
<bml id="bml1">
  <speech bmlis:generatefiller="true" id="s1">
    <text>The car goes around the corner and</text>
  </speech>
</bml>
<bml id="bml2" bmla:chunkAfter="bml1">
  <speech id="s1"><text>turns right.</text></speech>
</bml>
```

filler insertion is illustrated in BML Example. 1: first the `bmlis:generatefiller` attribute specifies that a filler may be generated in the first speech behavior; secondly, the `bmla:chunkAfter` attribute specifies that `bml2` is chunked directly after `bml1`, allowing filler-skipping. If `bml2` is not scheduled in time, the sentence "The car goes around the corner and uhm.. turns right." is produced, otherwise the filler is omitted and and a fluent prosodic connection between `bml1` and `bml2` is established. We have implemented automatic filler insertion using Inpro_iSS's HesitationIUs. HesitationIUs are realized as "uhm" if they are last in the buffer, and skipped otherwise. The PhraseIUManager adds both a PhraseIU and a HesitationIU to the buffer for each IncrementalSpeechUnit that is constructed from a speech behavior with the `generatefiller` attribute. For these IncrementalSpeechUnits the relax phase occurs during the filler. New BML blocks can be chunked after them, skipping their relax phase (and thus the filler).

To achieve even higher reactivity, AsapRealizer provides the capability to preplan BML blocks that can be instantly activated at later time (see [5] for examples and syntax). This allows very reactive behavior realization, e.g. in contexts where only few ECA responses are valid; each response can then be pre-planned while a user is speaking and then the 'appropiate' one can be activated without scheduling delay once user input is analyzed and the ECA has the floor.

3.2 Graceful Interruption

A behavior planner may interrupt ongoing behavior using BMLA interrupt behaviors (see [5] for the syntax). Interrupting behavior in a natural manner entails more than

simple halting e.g. speech or gesture. AsapRealizer makes use of the BML 1.0 notion of *ground state* for postures and gaze targets and generally implements an interruption by gracefully restoring the ECA's ground state. Gestures are interrupted by automatic insertion of a transition motion that guides the ECA's arms to the posture ground state; gaze is interrupted by the insertion of a new gaze motion to the gaze target ground state; facial expressions are interrupted by inserting a transition motion to a neutral facial expression. Speech may be interrupted instantly, or at phoneme or word boundaries. The ground states themselves are changed using postureShift and gazeShift behaviors respectively.

3.3 Adaptation of Ongoing Behavior

AsapRealizer realizes a stream of BML blocks, each of which specifies the timing (e.g. sync points X of behavior A and Y of behavior B should occur at the same time) and shape (e.g. behavior A should be performed with the left hand) of the desired behaviors. Generally, BML blocks are under-specified and leave realizers freedom in their actual realization. Realizers can make use of this to achieve natural looking motor behavior, e.g. by setting a biologically plausible duration of a gesture preparation. AsapRealizer employs a flexible behavior plan representation –the PegBoard. The PegBoard maintains a set of TimePegs that symbolically link to the synchronization points of behaviors that are constrained to be at the same time. The timing of these TimePegs may be updated, which moves the timing of the associated synchronization points, but maintains the time constraints specified upon them. The PegBoard thus allows one to do timing modifications of the behavior plan as it is being executed, but in such a way that BML constraints remain satisfied and no expensive re-scheduling is needed (see [21] for implementation details).

Additionally, adaptation of ongoing behavior requires e.g. animation and speech realizations in which certain parameters of ongoing behavior can be changed on the fly. AsapRealizer provides a facial animation system in which the intensity of ongoing morph based, MPEG-4 based, Action Unit based or emotional animation can be changed [22]. We provide a procedural animation system which allows arbitrary mathematical formulas and parameter sets to be used for motion specification and allows the parameters of ongoing animation to be changed (see [8] for implementation details). This design is more flexible than traditional procedural animation models that define motion in terms of splines or other predefined motion formulas and that use fixed parameter sets (e.g. [23,24]). To demonstrate that our procedural motion captures such models as a subset, we have semi-automatically converted several motion units from Greta [23] as procedural animations in our system and provide on-the-fly adaptation of their spatial extent, fluidity and power parameters. Ongoing speech is adjusted through Inpro_iSS, which currently supports on-the-fly adaptations of speech pitch, loudness and speaking rate.

Top-down Adaptation of Ongoing Behavior Our BMLA `parametervaluechange` behavior allows the behavior planner to adapt ongoing behavior elements (see [5] for syntax). Such adaptations allow the behavior planner to, for example, raise the loudness of ongoing speech to keep a challenged turn (see BML Example 2).

BML Example 2 Change the volume of the bml1:speech1 from its current value to 90, over a linear trajectory.

```
<bmla:parametervaluechange target="bml1:speech1" paramId="volume"
    start="bml1:speech1:s1" end="bml1:speech1:s1+1">
  <bmla:trajectory type="linear" targetValue="90"/>
</bmla:parametervaluechange>
```

Bottom-up Adaptation of Ongoing Behavior Bottom up adaptation of gesture is required since the preparation and retraction phases need to be constructed on the fly because the start position of the preparation, the hand position at the start/end of the stroke and the posture ground state at the end of the gesture are all subject to change during gesture execution. The initial hand position may vary by previously executed motion and/or posture changes, the hand position at the start/end of the stroke may vary by top-down parameter adjustments in the gesture, and the rest posture state may change as a result of posture shifts. We have implemented an adaptive timing process for the preparation and retraction of gestures, which is described in detail in [9]. Similarly, the timing of preparation and retraction of gaze is subject to changes in the position of the gaze target, the position of e.g. the head and eyes at the start of the gaze, and the gaze ground state. To realize adaptive, full body gaze in which the timing of the preparation and retraction of gaze is automatically determined, AsapRealizer provides a novel biologically motivated gaze model and the implementation of the gaze 'ground state', which keeps track of the current gaze target and automatically creates motions to it whenever the selected body parts (e.g. eyes, neck, spine) for the ground state are not occupied with other movement that has higher priority. The BML 1.0 gazeShift behavior has been implemented to change the desired gaze ground state. AsapRealizer's gaze model is based on [25], and introduces some improvements to it: the eyes reach the target first and then lock on to it, that is, they overshoot their end rotation and then move back while remaining locked on the target (implementing [26]), the maximum speed of the eye and its velocity profile are biologically motivated (implementing [27]) and the eyes adhere to their biological rotation limits (obtained from [28]).

Adaptation to a Changing World AsapRealizer's flexible plan representation is also used to allow tight synchronization with interlocutor behavior. On the specification side this is achieved by allowing the synchronization of behaviors to time events provided by *anticipators* (see BML Example. 3 for an example). An anticipator manages TimePegs

BML Example 3 Aligning the start of a speech behavior to occur slightly before the predicted end of interlocutor speech.

```
<speech start="anticipators:turnStopAnticipator:turnStop-0.1">..</speech>
```

within AsapRealizer's PegBoard that may be used to make adjustments to the timing

of behavior that is synchronized to them. Anticipators use perceptions of the real world to continually update the timing of these TimePegs, by extrapolating these perceptions into predictions of the timing of future events (e.g. the end of an interlocutor turn, the next beat in music) that correspond to the managed TimePegs. We have implemented an anticipator to predict tempo events in real-time music for a virtual dancer or conductor [29], an anticipator that predicts the timing of fitness exercises of a user that exercises together with a virtual trainer [30], and several anticipators for wizard of Oz experiments that provide time events on button presses.

4 Comparison with Other Realizers

Table 2 provides an summary of AsapRealizer 2.0's fluent behavior realization capabilities in comparison with other Realizers. The listed capabilities are a selection of the capabilities offered by the different realizers for 1) incrementally constructing a plan out of multiple BML blocks and 2) adapting ongoing behavior (in e.g. timing and shape), as gathered from the papers describing them, their manuals and personal communication with their authors.

The first version of AsapRealizer [9] combined the incremental behavior realization capabilities of ACE [7] with Elckerlyc's capabilities for interactional coordination [8]. AsapRealizer 2.0 enhances AsapRealizer 1.0's capabilities by implementing the capabilities that were only available for gesture in AsapRealizer 1.0 also for speech, gaze and facial expression. It also provides incremental plan construction capabilities for prepending (rather than just appending), which is useful for the insertion of short delays and rephrases, while keeping the original behavior plan mostly intact.

The capabilities for the incremental construction of the behavior out of multiple BML blocks in the state-of-the-art Realizers SmartBody [31], EMBOT realizer [32], and the BML realizer of the Greta ECA [23] is limited to the specification and realization of sequential relations between BML blocks (but not fluently connecting them) and instantly merging BML blocks with ongoing behavior.

Unlike AsapRealizer, ACE, and Elckerlyc, each of these realizer has a rigid underlying behavior plan representation. The SmartBody scheduling algorithm resolves a BML block to a fixed plan in which all behaviors are assigned absolute timestamps ([31], p154) and animation is represented by specialized controllers. EMBOT Realizer represents its ongoing behavior in EMBRScript, an animation layer in which all elements have absolute time stamps and only absolute space descriptions may be used ([24], p513). In Greta, increments (with one gesture as their smallest granularity) of ongoing behavior are sent to a dedicated MPEG-4 player, which does not allow the adaptation of its ongoing movement ([23], p5). Because of their rigid plan representations, these realizers lack the ability to adapt their ongoing behavior and do not provide interruption capabilities beyond cutting of their speech. This is true to a lesser extent in SmartBody, which allows changes in shape, but not timing of ongoing behavior. SmartBody employs this flexibility mostly in allowing a rich behavior repertoire for interaction with the environment. Most realizers provide some autonomous behavior (e.g. blinking, breathing) and allow top-down control over some of its parameters (e.g.

	EMBOT realizer	Greta	SmartBody	ACE	Elckerlyc	AsapRealizer 1.0	AsapRealizer 2.0
Incremental plan construction							
Merging increments	✓	✓	✓	✓	✓	✓	✓
Immediate/high priority increments	✓	✗	✗	✗	✗	✗	✗
Appending increments	?	?	✓	✓	✓	✓	✓
Prepending increments	✗	✗	✗	✗	✗	✗	✓
Preplanning and activation	✗	✗	✗	✗	✓	✓	✓
Chunking increments	✗	✗	✗	✓	✗	✓	✓
On-the-fly gesture co-articulation	✗	✗	✗	✓	✗	✓	✓
On-the-fly incremental intonation	✗	✗	✗	✗	✗	✗	✓
Automatic speech fillers	✗	✗	✗	✗	✗	✗	✓
Graceful interruption							
Speech	?	?	✓	✗	✓	✓	✓
Gesture	✗	✗	✗	✗	✗	✓	✓
Gaze	✗	✗	✗	✗	✗	✗	✓
Facial expression	✗	✗	✗	✗	✗	✗	✓
Top-down adaptation							
Speech	✗	✗	✗	✗	✗	✗	✓
Gesture	✗	✗	✗	✗	✓	✓	✓
Facial expression	✗	✗	✗	✗	✓	✓	✓
Breathing	✓	?	✓	✓	✓	✓	✓
Blinking	✓	?	✓	✓	✓	✓	✓
Bottom-up adaptation							
Speech	✗	✗	✗	✓	✗	✗	✓
Gesture	✗	✗	✗	✓	✗	✓	✓
Facial expression	✗	✗	✗	✗	✗	✗	✗
Interacting with a changing world							
Gaze at moving targets	?	✗	✓	✓	✓	✓	✓
Manipulate moving targets	✗	✗	✓	✗	✗	✗	✗
Point at moving targets	✗	✗	✓	✓	✓	✓	✓
Follow/walk to moving targets	✗	✗	✓	✗	✗	✗	✗
Maintain endeffector constraints	✗	✗	✓	✗	✗	✗	✗
Emit (synchronized) events to indicate world changes	✗	✗	✓	✗	✗	✗	✗
On-the-fly synchronization to predicted (external) time events	✗	✗	✗	✗	✓	✓	✓

Table 2: Fluent behavior realization capabilities of different Realizers

frequency), which can, for example, be used by the behavior planner to express some emotion of the ECA.

Beyond the capabilities offered by AsapRealizer, SmartBody offers a wider set of behaviors to interact with the world, including 'events' to communicate information to the outside world in tight synchrony to ongoing behavior. This functionality is typically used for inter-ECA communication, but has applications beyond that (for example, communicating that an ECA pressed the light switch to the environment it is in). EMBOT realizer offers the unique capability to specify that an increment requires immediate, high priority execution. These increments are performed as soon as possible, overriding existing elements.

5 The Bigger Picture: The Articulated Social Agents Platform

AsapRealizer is the behavior realization component of the Articulated Social Agents Platform(Asap)[5], a platform specifically designed for the development of ECAs that allow fluent interaction with their human interlocutors. Asap provides a collection of software for social robots and virtual humans jointly developed by our two research groups. In addition to a collection of tools, we also provide the means (through middleware and architecture concepts (see Fig. 3)) to compose virtual human or robot applications in which the tools are embedded. Asap embeds the SAIBA architecture for

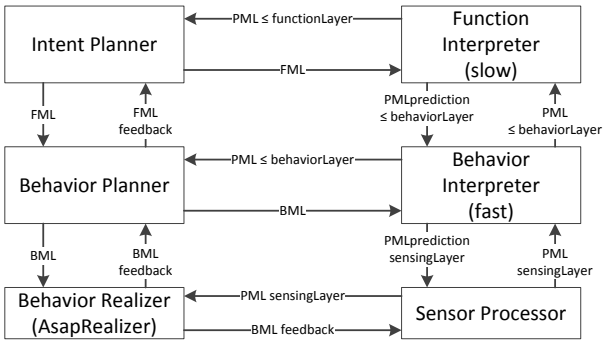

Fig. 3: The Asap Architecture

behavior realization ([33], left side of Fig. 1) and enhances it with two essential features for fast and fluent virtual human behavior: a close bi-directional coordination between input processing and output generation, and incremental processing of both input and output. In this paper we have show how these two features play a part in the behavior realization within AsapRealizer. In [5] and on the Asap website [1] we illustrate in several scenarios how AsapRealizer's realization capabilities play out in concert with fluent intent and behavior planning and input processing.

[1] `http://www.asap-project.org`

6 Discussion

We have presented AsapRealizer 2.0, a BML behavior realizer that has several fluent behavior realization capabilities that, in most aspects, go beyond the state of the art. AsapRealizer is eminently suitable for behavior realization in very dynamic contexts. Fluent behavior realization is however not the only important aspect of a realizer, and other realizers may have an edge over AsapRealizer in e.g. the realism of their behavior repertoire, which might make them more suitable in contexts where less flexible behavior suffices.

While we model how ECA behavior is changed through changes in the environment, we currently provide no means to model how an ECA steered by AsapRealizer changes the environment. We are planning to implement a mechanism inspired by SmartBody's event system to achieve this. In our applications, we plan to use such events e.g. to adapt a traditional user interface in reaction to the behavior of an embedded ECA or to make changes in the calendar used by the daily assistant Billie.

Our flexible plan representation using the PegBoard has recently been adapted in the Thalamus robotic framework [34], where it provides flexible (input) event based control of interactive robots. Achieving speech-gesture synchronization is challenging in current robotic systems that do not allow changes in ongoing behavior, since the exact timing of robotic gesture can typically not be predicted very precisely beforehand by standard robot software [35]. We propose that AsapRealizer's capability to adjust the timing of ongoing behavior can potentially be used to achieve speech-gesture synchrony for robots on the fly (see [35] for details).

AsapRealizer's fluent behavior realization capabilities open up many exciting possibilities for behavior and intent planning. In particular, we are currently interested in fluent turn-taking. Using AsapRealizer, we can model turn-taking that goes beyond the state of the art in 1) allowing the ECA to interrupt the user (e.g when something urgent comes up, to instantly correct a mistake made by the user or to indicate that the ECA no longer understands the user at an early moment rather than providing such feedback after the user is completely finished speaking), 2) to keep a challenged turn (e.g. to provide a last bit of important information to the user) and 3) to let itself be interrupted in a graceful manner. Each of these use cases can be executed in widely varying ways. For example, one can grab the turn by speaking loud, with a high pitch, fast and/or by gazing at the interlocutor (see e.g. [36,37]). The exact selection of these surface features modulates e.g. the perceived conversational skill, friendliness, dominance and naturalness (perceived realism) of the ECA, the perceived urgency of the message she/he has to deliver, and the effectiveness of the behavior (see e.g. [38,39]). In future work we aim to model the relations between such social and communicative parameters and the surface behavior required to achieve them.

References

1. Howes, C., Purver, M., Healey, P.G.T., Mills, G., Gregoromichelaki, E.: On incrementality in dialogue: Evidence from compound contributions. Dialogue & Discourse **2**(1) (2011) 297–311
2. Levelt, W.J.M.: Speaking: From Intention to Articulation. The MIT Press (1989)

3. Clark, H.H.: Using language. Cambridge University Press (1996)
4. Bernieri, F.J., Rosenthal, R.: Interpersonal coordination: Behavior matching and interactional synchrony. In Feldman, R.S., Rimé, B., eds.: Fundamentals of Nonverbal Behavior. Studies in Emotional and Social Interaction. Cambridge University Press (1991)
5. Kopp, S., van Welbergen, H., Yaghoubzadeh, R., Buschmeier, H.: An architecture for fluid real-time conversational agents: integrating incremental output generation and input processing. Journal on Multimodal User Interfaces (2013) Online First Article.
6. Traum, D., DeVault, D., Lee, J., Wang, Z., Marsella, S.: Incremental dialogue understanding and feedback for multiparty, multimodal conversation. In: Intelligent Virtual Agents. Volume 7502 of LNCS., Springer (2012) 275–288
7. Kopp, S., Wachsmuth, I.: Synthesizing multimodal utterances for conversational agents. Computer Animation and Virtual Worlds **15**(1) (2004) 39– 52
8. van Welbergen, H., Reidsma, D., Ruttkay, Z.M., Zwiers, J.: Elckerlyc: A BML realizer for continuous, multimodal interaction with a virtual human. Journal on Multimodal User Interfaces **3**(4) (2010) 271– 284
9. van Welbergen, H., Reidsma, D., Kopp, S.: An incremental multimodal realizer for behavior co-articulation and coordination. In: Intelligent Virtual Agents. Volume 7502 of LNCS., Springer (2012) 175–188
10. Baumann, T., Schlangen, D.: Inpro_iss: A component for just-in-time incremental speech synthesis. In: ACL System Demonstrations, ACL (2012) 103–108
11. Kopp, S., Jung, B., Leßmann, N., Wachsmuth, I.: Max - a multimodal assistant in virtual reality construction. Künstliche Intelligenz **17**(4) (2003) 11–17
12. Nijholt, A., Reidsma, D., van Welbergen, H., op den Akker, H., Ruttkay, Z.M.: Mutually coordinated anticipatory multimodal interaction. In: Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction. Volume 5042 of LNCS., Springer (2008) 70– 89
13. Salem, M., Kopp, S., Joublin, F.: Generating finely synchronized gesture and speech for humanoid robots: a closed-loop approach. In: HRI. (2013) 219–220
14. Reidsma, D., de Kok, I., Neiberg, D., Pammi, S., van Straalen, B., Truong, K.P., van Welbergen, H.: Continuous interaction with a virtual human. Journal on Multimodal User Interfaces **4**(2) (2011) 97– 118
15. Buschmeier, H., Kopp, S.: Towards conversational agents that attend to and adapt to communicative user feedback. In: Intelligent Virtual Agents. Volume 6895 of LNCS., Springer (2011) 169–182
16. Buschmeier, H., Baumann, T., Dosch, B., Schlangen, D., Kopp, S.: Combining incremental language generation and incremental speech synthesis for adaptive information presentation. In: SIGdial. (2012) 295–303
17. Yaghoubzadeh, R., Kramer, M., Pitsch, K., Kopp, S.: Virtual agents as daily assistants for elderly or cognitively impaired people. In: Intelligent Virtual Agents. Volume 8108 of LNCS., Springer (2013) 79–91
18. McNeill, D.: Hand and Mind: What Gestures Reveal about Thought. University of Chicago Press (1995)
19. Kendon, A.: Gesticulation and speech: Two aspects of the process of utterance. In Key, M.R., ed.: The relation of verbal and nonverbal communication. Mouton (1980) 207– 227
20. Baumann, T., Schlangen, D.: Interactional adequacy as a factor in the perception of synthesized speech. In: ISCA Speech Synthesis Workshop. (2013)
21. van Welbergen, H., Reidsma, D., Zwiers, J.: Multimodal plan representation for adaptable BML scheduling. AAMAS **2** (2013) 305–327
22. Paul, R.: Realization and high level specification of facial expressions for embodied agents. Master's thesis, University of Twente (2010)

23. Anh, L.Q., Huang, J., Pelachaud, C.: A common gesture and speech production framework for virtual and physical agents. In: ICMI Workshop on Speech and Gesture Production. (2012)
24. Heloir, A., Kipp, M.: Real-time animation of interactive agents: Specification and realization. Applied Artificial Intelligence **24**(6) (2010) 510–529
25. Grillon, H., Thalmann, D.: Simulating gaze attention behaviors for crowds. Computer Animation and Virtual Worlds **20 2-3** (2009) 111– 119
26. Radua, P., Tweed, D., Vilis, T.: Three-dimensional eye, head, and chest orientations after large gaze shifts and the underlying neural strategies. Journal of Neurophysiology **72**(6) (1994) 2840– 2852
27. Carpenter, R.H.S.: Movements of the Eyes. Second edn. Pion Ltd (1988)
28. Tweed, D.: Three-dimensional model of the human eye-head saccadic system. Journal of Neurophysiology **77**(2) (1997) 654– 666
29. Reidsma, D., Nijholt, A., Bos, P.: Temporal interaction between an artificial orchestra conductor and human musicians. Computers in Entertainment **6**(4) (2008) 1– 22
30. Reidsma, D., Dehling, E., van Welbergen, H., Zwiers, J., Nijholt, A.: Leading and following with a virtual trainer. In: International Workshop on Whole Body Interaction, University of Liverpool (2011)
31. Thiebaux, M., Marshall, A.N., Marsella, S., Kallmann, M.: Smartbody: Behavior realization for embodied conversational agents. In: AAMAS. (2008) 151– 158
32. Kipp, M., Heloir, A., Schröder, M., Gebhard, P.: Realizing multimodal behavior: Closing the gap between behavior planning and embodied agent presentation. In: Intelligent Virtual Agents. Volume 6356 of LNCS., Springer (2010) 57–63
33. Kopp, S., Krenn, B., Marsella, S., Marshall, A.N., Pelachaud, C., Pirker, H., Thórisson, K.R., Vilhjálmsson, H.H.: Towards a common framework for multimodal generation: The behavior markup language. In: Intelligent Virtual Agents. Volume 4133 of LNCS., Springer (2006) 205– 217
34. Ribeiro, T., Vala, M., Paiva, A.: Thalamus: Closing the mind-body loop in interactive embodied characters. In: Intelligent Virtual Agents. Volume 7502 of LNCS., Springer (2012) 189–195
35. Lohse, M., van Welbergen, H.: Designing appropriate feedback for virtual agents and robots. In: Position paper at RO-MAN 2012 Workshop "Robot Feedback in Human-Robot Interaction: How to Make a Robot "Readable" for a Human Interaction Partner". (2012)
36. Kendon, A.: Some functions of gaze direction in social interaction. Acta Psychologica **26** (1967) 22– 63
37. Esposito, R., Yang, L.c.: Acoustic correlates of interruptions in spoken dialogue. In: ESCA Workshop on Interactive Dialogue in Multi-Modal Systems. (1999)
38. Goldberg, J.A.: Interrupting the discourse on interruptions : An analysis in terms of relationally neutral, power- and rapport-oriented acts. Journal of Pragmatics **14**(6) (1990) 883 – 903
39. ter Maat, M., Truong, K.P., Heylen, D.: How turn-taking strategies influence users' impressions of an agent. In: Intelligent Virtual Agents. Volume 6356 of LNCS., Springer (2010) 441– 453