Continuous Interaction within the SAIBA Framework

Job Zwiers, Herwin van Welbergen, and Dennis Reidsma

University of Twente, Human Media Interaction
{j.zwiers,h.vanwelbergen,d.reidsma}@utwente.nl

Abstract. We propose extensions to the SAIBA BML language, aiming in particular to specify reactive behavior and continuous interaction. Examples of interaction scenarios are provided that illustrate the usefulness of these extensions. The impact on the SAIBA architecture is discussed.

1 Introduction

Multimodal generation of behavior for Virtual Humans is becoming more sophisticated, aiming to eventually capture the richness of human-human interaction. Not surprisingly, the specification of virtual human behavior and the structure of architectures and frameworks for controlling virtual humans have also become more sophisticated[11,8,3,6,10]. An important initiative has been the proposal of a common reference architecture in the form of the SAIBA framework. (see Figure 1). Several stages in multimodal behavior planning have been identified as well as clearly defined interfaces in the form of XML based languages BML and FML [5,4,12].

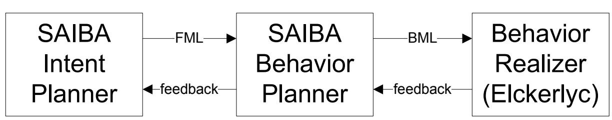

Fig. 1. The abstract SAIBA architecture

There is a sound theory behind this modular setup: the agent's communicative intents are first translated into rather abstract communicative signals, and only in a second stage these abstract signals are converted into concrete speech signals and concrete body and face animations. Although this modularization as such is a good idea, the "pipelined" processing of signals is less convincing. It adheres to the well known Sense-Think-Act cycle (STA) of traditional AI. As argued in [1,13] human behavior does not always conform to this pattern, in that not all human behavior stems from intent planning. In particular, bodily behavior is often an immediate and quick reaction on events, either from the environment, or from the behavior of other communicative partners. Such behavior is called *reactive* in [1]. Examples are mimicry and mirroring, where one

H. Högni Vilhjálmsson et al. (Eds.): IVA 2011, LNAI 6895, pp. 324–330, 2011.

human copies behavioral elements, like body pose or gaze direction, from other, observed partners.

Another interesting situation is where one human *predicts* certain events, like in a handshake, where both partners must somehow ensure that their hands arrive at the right place at the right time. This usually requires last minute adaptation to the timing of the appropriate body animation. We refer to this more demanding interaction paradigm as *continuous interaction*.

So we observe that a "pipelined" architecture as indicated in figure 1 is incomplete: in addition to the indicated feedback loops, there are often other interaction/feedback loops involved. As seen in many systems [11,6,1], behavior planning must deal with a reactive behavior loop that bypasses the intent planning stage.

We argue that for continuous interaction one needs an even tighter feedback loop than for reactive behavior: For continuous interaction a Behavior Realizer has to adapt its plans "on the fly" when it must react almost instantly to bodily behavior like head nods, pose shifts, or facial expressions, either observed or predicted for other (virtual or real) humans. The instant reaction times for real humans, as well as the fact that such reactions are largely unconscious, indicate that it is plausible that processing of this type of signals differs from more conscious thinking and planning. We think it is appropriate to maintain an analogous separation of concerns within the agent platform.

Another example of continuous interaction is shown in figure 2 where a feedback loop for sensor date(like tempo and heart rate) is shown.

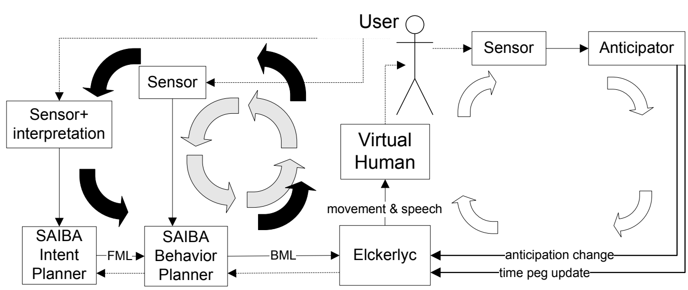

Fig. 2. Continuous Interaction

It reacts on data but also anticipates such data for the near future. What is relevant here is that, again, it bypasses all stages from the SAIBA pipeline and rather affects current, ongoing bodily behavior of the virtual human in a direct way. The idea is rather simple and clear enough, but it is somewhat problematic to deal with within current BML, since the virtual human behavior for making movements in accordance with the tempo has already been planned somewhere in the past. We propose that such alternative interaction mechanisms should be reflected, not just inside the implementation of planners and realizers, but also explicitly at the behavior specification level. Consequently, BML should be extended with appropriate language elements.

We also would like to draw the attention to the SAIBA feedback loops as indicated in Figure 1. Such feedback has more potential than just reporting "errors and problems". In fact, it should be used to communicate important state information from the Behavior Realizer back to Behavior Planner and Intent Planner. Such feedback concerning realizer state is essential for *adaptation* of behavior that is already in the stage of being "executed" or "realized". Typical examples are situations where one communicative partner interrupts another partner. The interrupted agent should not simply continue its current behavior but rather adapt to the situation, for instance by terminating it in a graceful way, or, alternatively, by modifying timing and volume with the aim to "keep the turn". We note that interruption is already part of several BML Realizers, including [10,13]. For the sake of modularity the feedback streams should employ similar mechanisms as the FML and BML streams.

In Figure 2 we see part of the Elckerlyc platform, where several feedback loops are shown, including a loop feeding the Behavior Planner directly from sensor information, for reactive behaviors, and a similar feedback loop feeding the Realizer directly from Anticipators that make predictions directly from sensor information, bypassing Intent Planner and Behavior Planner.

2 Impact on the SAIBA Framework

The continuous interaction paradigm has a profound impact on the SAIBA framework, both on a conceptual level as well as on an implementation level. For instance, when behavior interruption and "on the fly" modification of behavior parameters, including timing, are required then this puts certain demands on the capabilities and implementation of the behavior realizer. But also the "pipelined" nature of the SAIBA framework is challenged. For instance, when a behavior realizer interrupts or even cancels some behavior, then other components such as the behavior planner should be informed. Conceptually one should think in terms of a *behavior plan* that is being constructed "on the fly" and that serves as the *joint* information structure shared by behavior planners [11], reactive behavior planners, behavior realizers, and behavior adapters for continuous interaction.

At first one might think that a behavior plan is merely an implementation aspect, internal to SAIBA components like the Behavior realizer. This is in line with the idea that languages like FML and BML are declarative languages, so a stream of BML behaviors would carry no explicit state information. But for certain continuous interaction mechanisms this in an untenable position. An important example of why this is the case is that of interruption of speech: a behavior planner sending a speech behavior for a single sentence to the behavior realizer cannot prevent the environment, including humans, to interrupt that sentence at a later time. When that happens, continuous interaction demands that there is an instantaneous reaction, for instance by means of sending an interrupt behavior that partially cancels the speech behavior and removes it from the behavior plan, possibly replacing it with some alternate behavior.

The important observation here is that all this implies that behavior planners, and ultimately also intent planners, must be aware of the state (in the form of a behavior plan or otherwise) of the behavior realizer. An intent planner, for instance, can use feedback concerning behavior progress and interrupt information to decide whether some message can be considered to have actually "arrived" at the receiver, or not, and then act accordingly. Based on similar information the dialogue management functionality inside intent planner and behavior planner can make inferences about who has the floor at any moment. (Although "owning the floor" is clearly a higher level concept, it is affected nevertheless by lower level events like interruption, gaze behaviors, etcetera).

It appears that for modeling reactive behavior and continuous interaction, it is better to view major components like various planners and realizers as parallel, loosely coupled, processes (PP), communicating with each other, but not necessarily in a strict pipelined model. For in such architectures it is more natural to include components like action selectors, or components that deal with feedback in a more explicit way. Architecture along these lines have been proposed in [1,13].

2.1 Environment State and Virtual Human State

Another important aspect of the behavior plan is the representation of the state of situated virtual humans themselves. This becomes important whenever the cumulative effect of BML behaviors has a lasting effect on, for instance, the VH body pose or facial expression. State becomes also important when behaviors have an impact on the VH environment, if only by walking from one place to a different physical location. All planners and realizers will have to take this state into account. For example, we must know whether our virtual human is currently looking at some person, or whether his location and body pose are such that some gesture for grabbing some particular object is feasible.

State aspects of bodily behavior cannot be explicitly *specified* in core BML. One possibility to introduce state aspects is to use so called *persistent behaviors* in BML, for instance, to keep a VH in a certain body pose. This is a partial solution, although not without problems. For instance, in [13] there are slightly complicated priority rules for BML behaviors with the effect that persistent behavior can be overruled by other non-persistent behaviors, or that define the cumulative effect of several persistent behaviors.

2.2 The Behavior Plan State

State aspects within the SAIBA framework do not only apply to virtual human state or environment state, but even more so to "Behavior Plan state", shared by the various planners and realizers. As a motivating example consider a classical turntaking scenario, where the Intent Planner has been informed that the human user would like to get the turn. The Intent Planner decides that the virtual human should yield its turn, and seeks to do this in a way that is sensible from a continuous interaction point of view. By this we mean that the behavior realizer should not continue with its currently gesturing behavior and speech behavior, but rather that each of these behaviors could be interrupted or modified in an appropriate way. This requires that the SAIBA Behavior Planner keeps track of the behavior sent to the Realizer and monitors its execution progress. For instance, the gesturing behavior should, depending on the current realization state, be canceled (when it is not yet executing), be ignored (when it is already in a retraction phase, or when it has finished), or be replaced by a "gracefull ending" (when it has started and has not yet entered its retraction phase). In SAIBA based systems that fully implement the feedback mechanism such state-awareness can be achieved using feedback messages sent to the Behavior Planner by the Realizer.

It appears that a shared Behavior Plan is a vital component for specifying continuous interaction. Several existing systems, for instance the Gandalf system [11], do rely on such a shared plan.

2.3 Incremental Scheduling

The Max system [6] is the First ECA-system that simulates the mutual adaptations between the timing of gesture and speech that humans employ to achieve synchrony within *chunks* [7] between the coexpressive elements in those two modalities. Such "chunks" map naturally onto BML blocks.

Gesture movement in the Max system between the successive strokes of two gestures in two *successive* chunks depends on their timing. What is of interest here is that *inter-chunk synchronization* would be specified in the most natural way by introducing synchronization constraints for behaviors from *different* BML blocks. Such inter-block synchronization is cumbersome in core BML, using the start and end of BML blocks as somewhat artificial intermediate synchronization points. It would be more natural to admit inter-block synchronization constraints to be specified *directly*.

2.4 Interactional Synchrony

The current (core) SAIBA framework focusses on behavior for a *single* virtual human. Interaction with the environment, including interaction with other humans, virtual or real, does not affect BML behavior specifications. Such interaction can influence behavior only via coordination on the level of intent planners or possibly on the level of behavior planners. As argued above, this pattern does not suffice for continuous interaction, From the literature there are many examples of interactional synchrony, where observing and predicting external events, possibly stemming from the behavior of other agents, is required. According to Clark [2], joint actions (such as dialogue) can be coordinated because they divide into phases. For example, in a handshake phases include extending the hands, shake, and withdraw. Synchrony requires the coordination of entry and exit times of each phase. In many tasks, interlocutors are unaware of their mutual synchronization [9]. This suggest, again, that such synchronization does not result from conscious intent planning, but rather from more primitive processing that affects behaviors in a direct way. Synchronization, anticipation and prediction of mutual behavior of several (real or virtual) humans is not easily done in current BML. We propose to correct this situation by adding suitable synchronization mechanisms.

3 Recommendations for the SAIBA Framework

We have argued that the current (core) BML language is lacking in various aspects; we therefore propose to augment the language:

First of all there is a need for BML behavior that *interrupts, modifies, or replaces current, ongoing, behavior.*

Second, it will be necessary to *keep track* of the behavior plan state in other modules than just a behavior realizer. To achieve this goal, it is necessary to add communication from Behavior Realizer towards those other modules, for instance in the form of XML based feedback from realizer to planners.

Third, we propose to augment the SAIBA framework with mechanisms for synchronization with, and prediction of, external events, caused by the environment or by "other" humans than the virtual human under control.

Fourth, and finally, it is necessary to define carefully the *semantics* of BML behaviors. For instance, what will be the cumulative effect of various inputs, from behavior planner, reactive behavior insertion, or continuous interaction. The current specification of BML is quite precise when dealing with behavior synchronization inside a BML block. But the relation of behaviors from several independent blocks, or the effect of external agents on behavior is to some extent under exposed in the current BML description.

The current version of our platform [13] addresses several of these points, by offering $\mathbf{BML}^{\mathbf{T}}$ language extensions, and by offering a modular setup where a Behavior Plan is effectively a shared structure.

4 Conclusions and Future Work

We discussed various proposals for extensions to the BML language. The continuous interaction paradigm and its repercussions on the BML language and the BML Realizer enable a lot of interesting use cases, where more advanced social signals play a role than in more classical turn-based dialogue. A challenge is to extend these ideas to multi party interaction, where mutual interaction between members of (small) groups is the goal. An interesting question here is how to deal with synchronization between behavior Plans and BML streams for *different* virtual humans.

Acknowledgements. This research has been supported by the GATE project, funded by the Dutch Organization for Scientific Research (NWO).

References

1. Bevacqua, E., Prepin, E., de Sevin, R., Niewiadomski, R., Pelachaud, C.: Reactive behaviors in saiba architecture. In: Workshop Towards a Standard Markup Language for Embodied Dialogue Acts, held in conjunction with AAMAS 2009 (2009)
2. Clark, H.H.: Using language. Cambridge University Press, Cambridge (1996)
3. Gratch, J., Okhmatovskaia, A., Lamothe, F., Marsella, S.C., Morales, M., van der Werf, R.J., Morency, L.-P.: Virtual rapport. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 14–27. Springer, Heidelberg (2006)
4. Heylen, D., Kopp, S., Marsella, S.C., Pelachaud, C., Vilhjálmsson, H.H.: The next step towards a function markup language. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) IVA 2008. LNCS (LNAI), vol. 5208, pp. 270–280. Springer, Heidelberg (2008)
5. Kopp, S., Krenn, B., Marsella, S.C., 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: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 205–217. Springer, Heidelberg (2006)
6. Kopp, S., Stocksmeier, T., Gibbon, D.: Incremental multimodal feedback for conversational agents. In: Pelachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 139–146. Springer, Heidelberg (2007)
7. McNeill, D.: Hand and Mind: What Gestures Reveal about Thought. University of Chicago Press, Chicago (1995)
8. Perlin, K., Goldberg, A.: Improv: a system for scripting interactive actors in virtual worlds. In: SIGGRAPH, pp. 205–216. ACM, New York (1996)
9. Schmidt, R.C., Richardson, M.: Dynamics of interpersonal coordination. In: Fuchs, A., Jirsa, V. (eds.) Coordination: Neural, Behavioral and Social Dynamics. Understanding Complex Systems, vol. 17, pp. 281–308. Springer, Heidelberg (2008)
10. Thiebaux, M., Marshall, A.N., Marsella, S.C., Kallmann, M.: Smartbody: Behavior realization for embodied conversational agents. In: Autonomous Agents and Multiagent Systems, pp. 151–158 (2008)
11. Thórisson, K.R.: Gandalf: an embodied humanoid capable of real-time multimodal dialogue with people. In: Autonomous Sgents, pp. 536–537. ACM, New York (1997)
12. Thórisson, K.R., Vilhjálmsson, H.H.: Functional description of multimodal acts: A proposal. In: Function Markup Language Workshop at AAMAS 2009 (2009)
13. 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), 271–284 (2010)