Discussion Summary

H. Buschmeier, A. Cafaro, M. Chollet, P. De Loor, I. Gris, A. Hartholt, M. Jégou, S. Moon, E. Nouri, D. Novick, F. Pecune, F. Popineau, B. Ravenet, A. Robb, H. H. Vilhjálmsson, H. van Welbergen, A. A. Zadeh, and R. Zhao

The discussion groups focused on the following three topics:

1. Planning and realization of multimodal behavior;
2. Sharing assets, resources and knowledge;
3. Describing contextual information.

1 Planning and realization of multimodal behavior

In the first discussion group we focused on the increasing demand in novel IVA applications for continuous rather than discrete multimodal behavior planning and realization. In the SAIBA framework the Behavior Markup Language (BML) supports the specification of discrete behavior (e.g. gestures, speech, posture shifts) and their synchronization. Currently, however, it is impossible to specify behavior in BML that in nature is exhibited in a continuous fashion (e.g. maintaining interpersonal distance, walking in a crowded environment). Related to this, incremental approaches have emerged that compose reactive behavior from small chunks of BML – which can be regarded as partially realizing such continuous behavior. A definition for incremental and continuous behavior is needed and the distinction between the two should be clarified. Then, at the design level, it remains a question if and how BML should support the specification of continuous behavior.

During the discussion some ideas came up for such a specification. Firstly, two layers seem to be needed. A BML representation layer that enables the description of discrete behavior (extending the current BML standard), and an additional layer dealing with continuous behavior representation. This approach may lead to conflicts between the two layers that need to be handled. For example, in a group interaction there could be a BML chunk continuously updating interpersonal space, i.e. locomotion behavior, and another chunk that requires a different movement, i.e. movement towards a given destination. Conceptually, realization capabilities can be seen as shared resources that need to be allocated among competing BML chunks that require them, and some mechanism needs to be designed to determine which chunks are allocated which resources.

Secondly, an overall control logic is required to enable, disable and modify behaviors in the continuous behavior layer itself and to select behaviors, for example, according to priorities. To this end, a scripting language embedding BML descriptors could be introduced which supports these enabling/disabling/modification operations and can additionally chain together behavior using loops in a similar fashion as in imperative programming languages (e.g. for repeating gaze behavior).

Inspiration for the specification of such a two-level specification language and the corresponding realizer implementation may be drawn from design patterns in software engineering or in other fields (e.g. crowd simulation). Since continuous realization requires adaptation of ongoing motion and speech, we established that real-time animation and incremental text-to-speech experts are key figures that need to be involved in the design and implementation process.

We scheduled an agenda for the short and long term. In the short term we plan to organize a collaborative meeting where participants will model a given scenario featuring an interaction that involves both continuous and discrete behavior (e.g. similarly to the Cheeseburger scenario discussed in the 2010 FML workshop at ICT in USA). The goal will be to describe the interaction using the current BML specification, and propose new descriptors when needed, taking into account incremental and continuous behavior. Starting from this exercise, the gaps discussed above, will be formally defined, and a sketch design for the control logic (i.e. logic dealing with the control of discrete and continuous behavior) will be defined. This will be the basis for a preliminary new version of BML and a shared library of control logic patterns (i.e. scripts).

In the long term, all the gaps in the BML standard should be implemented in a new BML 2.0 version. Challenges for this new BML version consist of specifying continuous behavior and providing the specification means required for resolving conflicts between behaviors and allocating their required resources.

2 Sharing assets, resources and knowledge

In the second discussion group we focused on how to best facilitate the sharing of a variety of resources, including capabilities, assets, data and best practices.

First, this requires a common understanding of terms, and to a certain extent technologies and methodologies, in order to make sharing possible. This then allows for the decision of the scale of what's being shared, from individual assets to tools to modules. In the long term, the definition of an overall common theoretical and technical framework based on common standards would act as an organizing principle for available elements. It was felt that the most usable elements to share would be animations, software modules (e.g. a dialogue manager), and experimental data sets and results.

Second, there needs to be a common portal for these shared elements. In its most basic form, this could be an overview of available resources, both academic and commercial. This would provide the additional benefit of acting as an archive that can preserve current views and capabilities for future reference. In a more elaborate form, actual elements created by the research community can be uploaded and shared. Each element would require a minimum set of meta data, including developer info, ease of use, technical readiness level, category, used standards, relevant papers, etc. Software will need additional information, including a defined API and documentation and tutorials. Social elements should be available, providing the ability to comment on and rank elements as well as answer questions and facilitate discussion.

For the short term, this requires a more formal set of requirements, prioritized with a focus on a baseline functionality that can grow over time. Organizers and project leads will have to be identified, as well as possible sources of funding. Explicit attention needs to be paid on defining how to be successful; it is difficult and time consuming to initiate, grow and sustain a community. A central location and forum will have to be set up to facilitate initial discussions. These points should be formalized within a white paper and circulated and updated amongst a small group of interested parties acquired through the IVA emailing list. The finalized white paper can be the starting point for engaging funding sources.

In the long term, an ontology and related formal descriptions can be drafted related to categories of shared resources, target audiences, the overall landscape and associated topology. Meta data will have to be specified and templates set up for lessons learned, tutorials, documentation, etc. If the community grows to a critical mass, the hope would be to be able to connect researchers with other audiences, including application developers and the general public, resulting in potential study participants, end-user feedback, industry collaborations and funding.

3 Describing contextual information

In the third discussion group, we discussed about memory systems and representation of contextual information. In particular, in long term user-agent interactions, it becomes important to keep information about the context of the interactions and build agents with memory capabilities (e.g. relational agents). Indeed, communicating with agents that do not remember who is the user or what he/she said earlier might break the illusion, and lead to an unpleasant interaction. However, there is yet no standardized system allowing virtual agents to keep records of the interactions and manage them. Many knowledge base systems already exist, but each of them use their own process to store information, and to retrieve it.

Then, for the short term, we agreed to design standards regarding these two questions: (1) How do we store the information? We should first try to focus on how to abstract information according to a particular context. Each interaction should be summarized in different chunks of information and store in a knowledge base. Each chunk of information should also be linked to a set of contextual tags, so it can be retrieved later. (2) How do we infer which information should be used during the interaction? The process of retrieving and using chunks of information according to the interaction should be standardized as well.

For the long-term, we should try to focus on a third question, about forgetting the information: How long should we keep the information in the system? Since agents that remember everything might seem creepy, we considered two situations: If the human has to be immersed in a virtual world, the agents he encountered should be granted an accurate but short-termed memory. If the agents are introduced in the real world, their memory should last longer, but should be less accurate.