OpenBMLParser: An Open Source BML Parser/Analyzer

Herwin van Welbergen[1,2,⋆]

[1] Sociable Agents Group, CITEC, Fac. of Technology, Bielefeld University
[2] Human Media Interaction, University of Twente

1 Introduction

The Behavior Markup Language (BML) [1] has become the de facto standard for multimodal behavior specification for virtual agents. A BML block describes the behaviors (e.g. gaze, speech, gesture) a virtual agents should realize, the synchronization constraints between them, and how the block is to be composed in the ongoing behavior plan. Besides the synchronization constraints that are explicitly defined in BML, several implicit constraints act upon a BML block [2]. Parsing BML is not a trivial endeavor, since the language is highly redundant and extensions can be hooked up at several points. Properties that are useful for both BML scheduling and queuing can be obtained by analysis of a BML block. Our open source Java BML parser/analyzer OpenBMLParser provides authors of new and existing BML Realizers with a building block that can handle BML parsing and BML block analysis and allows them to easily hook up their own BML extensions.

2 Parsing, Analyzing and Extending BML

To schedule a BML block, the realizer needs a list of behaviors in it, the set of its 'at' constraints and the set of its 'after' [1] constraints[2]. After parsing a BML block, OpenBMLParser provides exactly that information. We have also implemented the BML cluster property predicates defined in [2]. These predicates can be used by the realizer to resolve or maintain the block's implicit constraints. Both BML behaviors and constraints may be wrapped inside `required` elements, to indicate that the BML block should fail as a whole if required behaviors or constraints fail. OpenBMLParser captures this information by listing the requiredness for each behavior and constraint.

The BML specification requires that BML blocks are scheduled in the order they are submitted to a realizer. BML blocks that are waiting to be scheduled

⋆ An early version of what is now OpenBMLParser was implemented by Ronald C. Paul. 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 author is responsible for the contents of this publication.

[1] Before constraints can be rephrased as after constraints.

form a scheduling queue. More reactive behavior realization can be achieved by allowing a realizer to schedule multiple blocks in parallel. The scheduling of a block may start as soon as no dependent blocks are before it in the scheduling queue. OpenBMLParser supports such multi-threaded scheduling by providing the realizer with the dependency information of a BML block. A BML block is dependent on another block if 1) its constraints refer to that block 2) it has behaviors referring to that block or 3) the `bml` element of the block defines dependencies on other blocks.

OpenBMLParser allows users to register their own classes that parse behaviors or description levels, and allows the registration of custom attributes on existing behaviors. It also provides users with functionality to register one or more handlers for the parsing of their own custom attributes and attributes values within the `bml` element itself, e.g. to parse custom block composition values or to specify that a block should be preplanned for later execution.

3 Using OpenBMLParser

OpenBMLParser[2] can help the authors of new Java[3] realizers, realizers that do not support BML 1.0 yet, or other applications that make use of BML to focus on the development of new scientific or business value, rather than on the tedious parsing and analysis of BML blocks. OpenBMLParser is fully BML 1.0 compliant: it parsers and analyzes all BML 1.0 Core behaviors, Core Extensions, constraint definitions and BML 1.0 feedback. It is currently used as a BML parser/analyzer in AsapRealizer [3], allowing its extensions for BML block composition and preplanning, and the parsing of its over 20 custom behaviors and description levels; and as a BML feedback parser in RealizerTester [4] –an integration testing tool employed by multiple realizers.

References

1. 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)
2. van Welbergen, H., Reidsma, D., Zwiers, J.: Multimodal plan representation for adaptable BML scheduling. In: Autonomous Agents and Multi-Agent Systems, pp. 1–23 (2013)
3. van Welbergen, H., Reidsma, D., Kopp, S.: An incremental multimodal realizer for behavior co-articulation and coordination. In: Nakano, Y., Neff, M., Paiva, A., Walker, M. (eds.) IVA 2012. LNCS, vol. 7502, pp. 175–188. Springer, Heidelberg (2012)
4. van Welbergen, H., Xu, Y., Thiebaux, M., Feng, W.-W., Fu, J., Reidsma, D., Shapiro, A.: Demonstrating and testing the BML compliance of BML realizers. In: Vilhjálmsson, H.H., Kopp, S., Marsella, S., Thórisson, K.R. (eds.) IVA 2011. LNCS, vol. 6895, pp. 269–281. Springer, Heidelberg (2011)

[2] Available at `https://github.com/saiba/OpenBMLParser`

[3] Or any other JVM-based language.