Ontology-based Business Activity Monitoring Agent (extended abstract)[1]

Duco N. Ferro[a] Mark Hoogendoorn[b] Catholijn M. Jonker[c]

[a] *Almende B.V., Westerstraat 50, 3016 DJ Rotterdam*
[b] *Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam*
[c] *Delft University of Technology, Mekelweg 4, 2628 CD Delft*

1 Introduction

Business activity monitoring (BAM) is software that supports the monitoring of those activities that are implemented in computer systems. It is intended to provide real-time summaries of business activities to operations managers and upper management, and to detect and warn of impending problems. This technology is a competitive differentiator for companies. Since the emphasis is on analysis, analysis tools are of the utmost importance [6]. The need for more complex forms of analysis on the huge relational data sets companies collect is growing along with the technology to support the analytical process. These more complex forms of analysis can be eloquently represented in constraints, and extended forms of first order logic. However, in current practice, solutions to the problem are implemented in ad-hoc and difficult to maintain procedural code that accesses the data through embedded SQL programming. Lohfert et al., [5] propose to use more elegant solutions that involve the use of declarative languages that integrate constraint modeling with database access in transparent ways.

2 Formulating Properties

The core of OBAMA consists of the specification and monitoring of key properties within the business processes of the company. In order to formulate these, two approaches are used.

The first language used to specify properties to be verified upon an organization is TTL (for Temporal Trace Language, cf. [1]) that features an automated checker. This predicate logical temporal language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. Specifying such behavior in TTL is not a trivial matter, often documents describing the goals of the company as well as procedural specifications can be used as a basis for such a behavioral description. However, such a specification typically lacks sufficient detail to obtain a complete behavioral description. Therefore, a three step process is specified to formalize these properties:

1) Informal behavior descriptions are translated into a semi-formal format.
2) Defining an *ontology* suitable for this particular organization which is based upon the semi-formal rules that have been distinguished, and the terms that occur in such rules, see [4].
3) Translate the semi-formal rules into formal ones using the ontology which has been created.

The second language used for specifying properties to be monitored at an organization is the structured query language (SQL), a widely accepted standard programming language for querying and manipulating databases often used by users with no or little formal training in informatics [3]. Its applications range from simple retrieval in web-interfaces to complex functions that aggregate the stored data into useful information. In business settings many data storage solutions, including the storage of logging data, are based on SQL. To enable, as an alternative, formulating properties using TTL on data

[1] The full version of this paper appeared in: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2008)

stored on a SQL database server, some sort of pre-processing is required. In order to verify properties using TTL upon the information stored within the company databases, pre-processing is proposed.

3 OBAMA Agent Design

The languages for the specification of properties to be monitored and validated form the basis of the component-based design of the OBAMA agent. Note that the design approach followed is the DESIRE approach (cf. [2]). The agent consists of three main components:

- The first component concerns the setting of the properties that ought to be monitored.
- After the properties and the required information accompanying these properties are known, the component *monitoring of properties* starts to reason.
- The final component within the agent is the component for informing the user.

4 Experiments & Discussion

In order to test the agent designed, we have conducted an extensive case study within the private security domain. After having identified the properties, these were inserted into the OBAMA agent that started the monitoring process. All properties were set to regular report properties for which a weekly report should be generated. The following results were found by OBAMA:

P1: Average response time (SQL): The average response time was 23.6 minutes

P2: Patroller leaving object before contact person (TTL): With a minimum task time of 5 minutes, 7.9% of the cases the security guards left the alarm early.

P3: Status check response time (TTL): The average duration of status check responses was found to be 41.6 minutes.

The simple, but effective design of the OBAMA agent makes it easy to maintain, and extend with future functionality. Its ontology-based nature enabled the support of the user in formulating properties by a incremental refinement method, as presented in [4]. This approach has been fully integrated in the TTL parts of OBAMA.

Finally, the rich tools for SQL proved essential not only for the more standard properties to be monitored, but also to do the necessary preprocessing for the more complex properties for which the user needs to use the TTL components of OBAMA.

References

[1] Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J. (2008). Specification and Verification of Dynamics in Agent Models. *International Journal of Cooperative Information Systems*. In press, 2008.

[2] Brazier, F.M.T., Dunin-Keplicz, B.M., Jennings, N.R. and Treur, J.: 1997, Formal Specification of Multi-Agent Systems: a Real World Case, *in* Lesser, V. (ed.), *Proceedings First International Conference on Multi-Agent Systems, ICMAS'95*, MIT Press pp. 25-32; extended version *in*: M. Huhns and M. Singh (eds.), *International Journal of Co-operative Information Systems*, *IJCIS* vol. **6**(1), 67-94, (1997), special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems.

[3] Ferro, D.N., and Jonker, C.M., (2008). Filtering Algorithm for Agent-Based Incident Communication Support in Mobile Human Surveillance. In: Proceedings of MATES 2008, to appear.

[4] Herlea Damian, D.E., Jonker, C.M., Treur, J., and Wijngaards, N.J.E., Integration of Behavioural Requirements Specification within Compositional Knowledge Engineering. *Knowledge-Based Systems Journal*. Vol. 18, 2005, pp. 353 – 365. Kochar, H., (2005/12/25), Business Activity Monitoring and Business Intelligence. In: ebiz, The Insider's Guide to Business and IT Agility.

[5] Lohfert, R., Lu, J.J., and Zhao, D, (2008). Solving SQL Constraints by Incremental Translation to SAT. In: N.T. Nguyen et al. (Eds.): proceedings of IEA/AIE 2008, LNAI 5027, pp. 669–676.

[6] Negash, S., and Gray, P., (2008). Business Intelligence. In: Burstein, F., and Holsapple, C.W. (eds.) Handbook on Decision Support Systems 2. Springer Berlin Heidelberg, pp. 175—193.