

The Timeless Way of Building REA Enterprise Systems

Guido L. Geerts

Department of Accounting and MIS, University of Delaware
Purnell Hall 224, Newark, DE 19716, USA
geertsg@lerner.udel.edu

Abstract. In a continuously changing business environment, the need for more flexible enterprise systems has been recognized by many. Different solutions are being suggested to improve the adaptability of enterprise systems, including model-driven architectures and reflective systems. A model-driven architecture starts from an enterprise schema that formally defines the business activities of a company. This schema is then progressively transformed into more implementation-specific models until source code is generated [1, 2]. Changes in business practices are accommodated by enterprise schema updates that are then automatically propagated into software code. Reflective systems record enterprise schemas explicitly so that they can be manipulated at run time [3, 4]. A reflective system is more flexible since the definition of the enterprise schema and its integrity rules can be configured external to the execution of the program [5]. The expected benefits of reflective systems include improved adaptability, semantics, and reusability. On the other hand, their design is substantially more complex, and the execution of applications built with such systems is slower since the definition of the enterprise schema needs to be interpreted. A timeless REA enterprise system integrates ontological specifications as part of its reflective architecture. The Resource-Event-Agent (REA) enterprise ontology [6, 7] is strongly rooted in accounting and economic theory and addresses the issue of what phenomena should be captured in an enterprise system. In addition, it provides structuring guidelines on how economic phenomena should be assembled into business process and value chain specifications [8]. This paper explores the reflective architecture, design, and operation of timeless REA enterprise systems. We first describe the integration of ontological specifications as part of reflective enterprise systems. Next, we use claim management and cost calculation examples to illustrate how the ontological specifications can be employed for the design of reusable, ontology-driven applications.

Please use the following format when citing this chapter:

Geerts, G., L., 2006, in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.359-360.

References

1. D.S. Frankel, *Model Driven Architecture: Applying MDA to Enterprise Computing* (Wiley, 2003).
2. A. Kleppe, J. Warmer, and W. Bast, *MDA Explained: The Model Driven Architecture – Practice and Promise* (Addison-Wesley, 2003).
3. G.L. Geerts and W.E. McCarthy, Augmented Intensional Reasoning in Knowledge-Based Accounting Systems, *Journal of Information Systems* **14**(2), 127-150 (2000).
4. G.L. Geerts, An XML Architecture for Operational Enterprise Ontologies, *Journal of Emerging Technology Accounting* **1**, 73-90 (2004).
5. J.W. Yoder and R. Johnson, The Adaptive Object Model Architectural Style, Proceedings of The Working IEEE/IFIP Conference on Software Architecture, Montreal, Canada (2002).
6. W.E. McCarthy, The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment, *Accounting Review* **57**(3), 554-578 (1982).
7. G.L. Geerts and W.E. McCarthy, The Ontological Foundation of REA Enterprise Systems, <http://www.msu.edu/~mccarth4/> (2000).
8. G.L. Geerts and W.E. McCarthy, Using Object Templates from the REA Accounting Model to Engineer Business Processes and Tasks, *Review of Business Information Systems* **5**(4), 89-108 (2001).