Journal of Information Systems
Vol. 25, No. 1
Spring 2011
pp. 213–215

American Accounting Association
DOI: 10.2308/jis.2011.25.1.213

Book Reviews

Robert Pinsker

PAVEL HRUBY, *Model-Driven Design Using Business Patterns* (Berlin and Heidelberg, Germany; New York, NY: Springer, 2006, ISBN: 10 3-540-305154-2, 368, $24.95).

INTRODUCTION

Model-Driven Design Using Business Patterns describes the functional requirements necessary to develop and implement the Resource-Event-Agent (REA) model as a meta-model for specific business applications. When teaching the undergraduate Accounting Information Systems (AIS) course, the book provides useful supplemental material in many areas, most notably in value modeling. Most undergraduate-level AIS textbooks do not provide a framework for value modeling of business process applications; the Hruby book does. Value modeling is important in that it clarifies the meaning of process activities and business applications involving the value chain. Given the current financial crisis, companies have to pay attention to each value contribution in their business processes. Thus, *Model-Driven Design Using Business Patterns* can be useful when teaching value chain concepts, business process activities, or even internal controls and documentation.

Model-Driven Design Using Business Patterns uses the REA model as the basis to describe the business processes with a focus on actors and resources. Hruby contends that REA provides a greater semantic meaning than double-entry bookkeeping applications, because it stores the primary data about resources at the lowest level to ensure consistency of all financial reports. Thus, process applications are derived from the value models.

This book also can be valuable for applied research purposes. By discussing the impact of strengths and limitations of value models in business process applications, *Model-Driven Design Using Business Patterns* provides a guideline to compare the presented applications with current non-REA applications in Enterprise Resource Planning (ERP) systems within the context of value-oriented semantics. Practically speaking, the book is created to assist both business experts and information system developers through its functional and technical examples and implications.

SYNOPSIS

In Chapters 1 and 2, Hruby introduces the readers to the explanation of REA entities and relationships and their use in business processes. The goal is to get an understanding of the diverse axioms of REA. Hruby defines REA entities and relationships in a detailed way to describe application models governing what events should or should not do under certain conditions. For this reason, the exchanges as well as the conversion process are described as "patterns," which describe the purpose of each entity and relationship on an aggregated level. The different processes are linked by the value chain pattern. Hruby uses a case study of a pizzeria to introduce and explain the exchange and conversion process so that readers are easily able to understand the patterns. The reader gets introduced to how to schedule events within a commitment and how to design the relationship to represent the exchange and conversion of resources with agents. These chapters provide significantly more in-depth details than most AIS textbooks. Thus, they could serve as complementary material for professors desiring a more detailed introduction to basic REA concepts.

Chapter 3 breaks down the components of an REA-based application. Hruby begins by discussing the first component, the REA meta-model. Then, he moves on to describe the domain model, web pages that deal with the graphical interfaces, and the other components, until eventually reaching the Online Analytical Processing (OLAP) layer (which deals with the definition of reports). The explanation of the different components is important to understand the future chapters of the book. The defined components reflect the goal of the book to address both accountants and information system developers creating a business application.

Chapter 4 explains how to implement an REA-based application using the aspect-oriented software development (AOSD), which emphasizes the identification of crosscutting concerns (Rosenhainer 2004). The use of crosscutting enables the disjunction of the logical and physical semantic from the technical detail. Thus, from an applied research perspective, crosscutting provides artifacts researchers can use when creating their own applications. AOSD is chosen because it enables the modeling of crosscutting concerns, as well as manipulating them as modular features. Chapter 5 describes 13 different patterns that are used in business applications. All of the patterns are modeled with the AOSD and divided into two different levels: the aspect level (where business logic is developed) and the application level (which contains specific attributes). In this context, aspects represent the REA entities and relationships. Unfortunately, the drawbacks using AOSD, especially in large enterprise systems, are not discussed in detail.

These two chapters are also helpful for AIS professors who wish to concentrate more on systems analysis and design (SAD). They can use material from Chapters 4 and 5 in addition to the limited SAD examples typically found in undergraduate AIS textbooks (they are limited due to space constraints). For example, Hruby uses several figures, especially UML class diagrams, to facilitate the understanding. He defines patterns that ensure the REA consistency and describes how to record the history of events, store the events in accounts, aggregate financial information in reports, and to set payment terms.

In Chapter 6, Hruby presents two programming alternatives: C# and Extensible Markup Language (XML). XML enables a more understandable view to the programming code, because it is easier to read compared to C#. From an AIS perspective, the discussion about storing aspect data is less relevant when presenting the object diagrams and designing the aspects and applications.

Chapters 7 and 8 present the main exchange and conversion processes at the operational level that is derived from the aspect categories. Hruby uses object diagrams that show the application objects. The objects are deduced from the patterns explained in the previous chapters. The graphics help ensure that the readers understand the REA process.

Finally, Chapter 9 explains additional events within the conversion of resources, such as purchase order, labor acquisition, people management, shipment, and tax. The definition is complemented in Chapter 10 by describing processes with contracts (e.g., purchase order or transport). Once again, Hruby's in-depth examples complement the material typically found in most undergraduate AIS textbooks.

IMPLICATIONS AND CONCLUSION

In summary, *Model-Driven Design Using Business Patterns* presents in-depth material helpful in discussing or even designing and implementing REA-based applications. Hruby drills down into a detailed task level in order to attract the attention of the reader. Furthermore, Hruby describes the processes and patterns by using numerous diagrams, which facilitates understanding.

The SAD portions are particularly helpful when considering expanding the material found in most AIS textbooks.

From an applied research perspective, whereas most of the relevant literature is narrowly focused on REA as an ontology for enterprise systems (Gailly and Poels 2005, 2006; Geerts and McCarthy 2000) or REA implementation (Murthy and Wiggins 2004), this book provides both a business and a technical perspective of REA. It includes the basic literature about business modeling and REA, but its purpose is to discuss an REA application as a value model in ERP applications and compare it with current financial application solutions that are based on double-entry bookkeeping. REA is also used at the meta-level to define the aspects; thus, the book provides implications to compare REA with other value models like the e^3 value (Gordijn et al. 1999) at the conceptual level.

Even though Hruby refers to the advantages of an REA-based solution, he renounces the perspective of costs and benefits of the presented application. The drawbacks of using the AOSD are not discussed. Moreover, the semantics of the module explained in the book make AOSD more complex. For these reasons, the book is most appropriate for an undergraduate-level AIS class on how to present the distinction of possession and ownership of economic exchanges with regard to value-added processes. Furthermore, it has a strong focus on information systems, which ultimately leads to a practical discussion about costs and benefits of an REA-based application.

REFERENCES

Gailly, F., and G. Poels. 2005. Development of a formal REA-ontology representation. *Proceedings of the Open Interop Workshop on Enterprise Modeling and Ontologies for Interoperability* (EMOI), June 13–14, Porto, Portugal.

Gailly, F., and G. Poels. 2006. Towards an operational REA ontology using web ontology languages. *Proceedings of the 3rd International Conference on Enterprise Systems and Accounting*, June 26–27, Santorini, Greece.

Geerts, G. L., and W. E. McCarthy. 2000. The ontological foundation of REA enterprise information systems. *Proceedings of the American Accounting Association Conference*, August 13–16, Philadelphia, PA.

Gordijn, J., H. De Bruin, and H. Akkermans. 1999. Integral design of e-commerce systems: Aligning the business with software architecture through scenarios. *Proceedings of the First ICT-Architecture in the Benelux*, November 18–19, Amsterdam, NED.

Murthy, U., and C. Wiggins. 2004. OOREA: An object-oriented resources, events, agents model for enterprise systems design. *Proceedings of the 25th ICIS*, December 12–14, New York, NY.

Rosenhainer, L. 2004. Identifying crosscutting concerns in requirements specifications. Workshop in Aspect-Oriented Requirements Engineering and Architecture Design, October 25, Vancouver, Canada.

JOHANNES BUDER
Technische Universität Freiberg