



BROOD: Business Rules-driven Object Oriented Design

Pericles Loucopoulos, Loughborough University, UK

Wan M.N. Wan Kadir, Universiti Teknologi Malaysia, Malaysia

ABSTRACT

A critical success factor for information systems is their ability to evolve as their environment changes. There is compelling evidence that the management of change in business policy can have a profound effect on an information system's ability to evolve effectively and efficiently. For this to be successful, there is a need to represent business rules from the early requirements stage, expressed in user-understandable terms, to downstream system design components and maintain these throughout the lifecycle of the system. Any user-oriented changes could then be traced and if necessary propagated from requirements to design specifications and evaluated by both end-users and developers about their impact on the system. The BROOD approach, discussed in this article, aims to provide seamless traceability between requirements and system designs through the modelling of business rules and the successive transformations, using UML as the modelling framework.

Keywords: business rules; object-oriented design; software architecture; software evolution

INTRODUCTION

The ubiquitous nature of information systems and the increasing dependency of organizations, government and society on such systems highlight the importance of ensuring robustness in their operation. At the same time rapid changes in the environment of information systems places an increasing emphasis on the ability of these systems to evolve according to emerging requirements. A large proportion of a total systems' lifecycle cost is devoted to introducing new requirements, and removing or

changing existing system functionality (Grubb & Takang, 2003). Software evolution therefore is considered as a key challenge in the development and maintenance of information systems (Erlikh, 2000).

In recent years there has been an increasing interest of the IS community in business rules, which has resulted in dedicated rule-centric modeling frameworks and methodologies (Ross & Lam, 1999; Zaniolo et al., 1997), international initiatives for the investigation of business rules' role in the context of knowledge management

(Hay & Healy, 1997), conferences, workshops and tutorials (Mens, Wuyts, Bontridder, & Grijseels, 1998), and rule-centric rule management tools and application development support environments (e.g., Blaze Advisor Builder, BRS RuleTrack, Business Rule Studio, Haley Technologies, ILOG Rules, Platinum Aion, Usoft Developer and Visual Rule Studio). Whilst these efforts make significant contributions in their own right, a key challenge remains unanswered namely the *linking of business rules specifications to software designs*.

The aim of the BROOD (business rules-driven object oriented design) approach is to address the issue of software evolution from both *requirements* and *design* perspectives. This confluence should provide a seamless and traceable facility that arguably should bring about a more effective way of dealing with software evolution, by aligning changes of the information system to changes in its environment. BROOD adopts as its methodological paradigm that of object orientation with UML as its underlying graphical language. It augments UML by explicitly considering business rules as an integral part of an object-oriented development effort. To this end BROOD aims:

- i. To explicitly model business rules in a manner understandable to end-user stakeholders.
- ii. To map these to formal descriptions amenable to automation and analysis.
- iii. To provide guidelines on the deployment of business rules in the development process.
- iv. To provide guidelines on the evolution of requirements and related design specifications.

The article is organized as follows. Section 2 discusses the background to business rules modeling. Section 3 introduces the motivation for BROOD. Section 4 introduces the BROOD metamodel as the foundation for modeling business rules. Section 5 discusses the manner in which business rules are linked to design components via the concept of ‘rule phrase.’

The BROOD process is detailed in section 6. The BROOD approach is supported by an automated tool and this is briefly discussed in Section 7. The article concludes with an overview of BROOD, observations on its use on a large application and comparisons with traditional approaches.

The language details for business rules definition are given in appendix A. The BROOD approach is demonstrated through an industrial application which is described in appendix B. This application had originally been developed using a traditional approach. Therefore, it proved useful not only as a means of providing a practical grounding on BROOD but also on comparing and contrasting the use of BROOD with a traditional development effort.

BUSINESS RULES MODELLING

The motivation of BROOD is to provide a development environment whereby the business analysis and system design domains are supported by business rules modeling with the specific aim to facilitating more effective software evolution.

The term “business rule” has been used by different authors in different ways. For example, in (Rosca, Greenspan, Feblowitz, & Wild, 1997), business rules are:

statements of goals, policies, or constraints on an enterprise's way of doing business.

In (Herbst, 1996a), they are defined as:

statements about how the business is done, i.e. about guidelines and restrictions with respect to states and processes in an organization.

Krammer considers them as “programmatically implemented policies and practices of a business organization” (Krammer, 1997) whilst Halle states that:

depending on whom you ask, business rules may encompass some or all relationship verbs, mathematical calculations, inference rules,

31 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/brood-business-rules-driven-object/3381

Related Content

Experimental Study of a Self-Tuning Algorithm for DBMS Buffer Pools

Patrick Martin, Wendy Powley, Min Zheng and Keri Romanufa (2005). *Journal of Database Management* (pp. 1-20).

www.irma-international.org/article/experimental-study-self-tuning-algorithm/3329

Database Cooperation: Classification and Middleware Tools

Paolo Atzeni, Luca Cabibbo and Giansalvatore Mecca (2000). *Journal of Database Management* (pp. 3-11).

www.irma-international.org/article/database-cooperation-classification-middleware-tools/3248

Blockchain Technology and Future Banking: Opportunities and Challenges

Derya Üçolcu (2022). *Applications, Challenges, and Opportunities of Blockchain Technology in Banking and Insurance* (pp. 43-68).

www.irma-international.org/chapter/blockchain-technology-and-future-banking/306454

A Distributed Spatial Index With High Update Efficiency for Location-Based Real-Time Services

Junhua Fang and Zonglei Zhang (2023). *Journal of Database Management* (pp. 1-28).

www.irma-international.org/article/a-distributed-spatial-index-with-high-update-efficiency-for-location-based-real-time-services/318454

Benchmarking Data Mining Algorithms

Balaji Rajagopalan and Ravi Krovi (2002). *Journal of Database Management* (pp. 25-35).

www.irma-international.org/article/benchmarking-data-mining-algorithms/3274