

Chapter 62

A Systematic Approach for Designing Educational Recommender Systems

Patrick H. S. Brito

Federal University of Alagoas, Brazil

Evandro Costa

Federal University of Alagoas, Brazil

Ig Ibert Bittencourt

Federal University of Alagoas, Brazil

Olavo Holanda

Federal University of Alagoas, Brazil

Aydano Pamponet Machado

Federal University of Alagoas, Brazil

Rafael Ferreira

Federal University of Pernambuco, Brazil

Thiago Ribeiro

Federal University of Alagoas, Brazil

ABSTRACT

The construction of Educational Recommender System (ERS) demands the incorporation of quality attributes at the software design, such as availability for preventing the service to be unavailable for a long time, and scalability for preventing the system from going offline due to a large number of simultaneous requests. The incorporation of such characteristics makes ERS more complex and expensive, but existing strategies for designing ERS do not consider quality attributes in an explicit way. This chapter presents an architecture-centered solution, which is partially supported by tools and considers quality attributes as early as possible in the software development process in a systematic way, from requirements to the source code. The feasibility of the proposed process is showed in terms of a case study executed in a “step-by-step” fashion, presenting how the software architecture can be designed and gradually refined until it achieves the level of object-oriented classes generated based on design patterns.

INTRODUCTION

Recommender Systems are systems that typically compare a user profile to some reference characteristics, aiming to rate the items according to the particular needs of the user.

In the context of education, Educational Recommender Systems (ERS) are used for recommending content to learners according to their needs and knowledge levels. ERS are particularly important to the learning process since they allow teachers and learners to find content in a more appropriate way, according to their educational profile and needs. The importance of ERS is even

DOI: 10.4018/978-1-4666-4301-7.ch062

greater in the context of distance education, in which the average number of learners per teacher is usually higher than in traditional face-to-face education. In a scenario of a high number of learners, the recommendation of personalized content is impracticable for a human teacher, so there is a natural tendency to generalize the recommended educational content, such as exercises and reading material, according to the average profile of the class. The recommendation of general learning objects may discourage learners that need special attention thus increasing the dropout rate.

Due to its importance for distance education, ERS should be integrated to Virtual Learning Environment (VLE), in order to allow learners to access specific learning content, according to their profiles at the time that suits them. Knowing the importance of ERS and the fact that the VLEs should be online in a 24x7, ERS can be considered a critical system with which we must take special care throughout their life cycle, especially during its design and development.

Despite its importance to the learning process, the construction of ERS is considered a complex task, which demands the incorporation of quality attributes (Sommerville, 1995) usually related to critical software systems, such as scalability and security (privacy-protecting). Scalability property means that the system should be able of recommending content to a large number of users at the same time; while the security property aims to protect the personal data into the system profile from unauthorized access.

Although the aforementioned characteristics are very important in the context of ERS, their incorporation causes software to be more complex, and consequently makes the development process more error prone and the final software more expensive. To cope with the inherent complexity related to quality attributes, it has been claimed that quality requirements should be systematically incorporated as early as possible in the software development process, especially during the requirements engineering and the architectural design (Rubira, de-Lemos,

Ferreira, & Castor-Filho, 2005), (Brito, Rocha, Castor-Filho, Martins, & Rubira, 2005). Software architectures explicitly represent the structure of systems, and it is one of the earliest artifacts that permits the analysis of system quality attributes, such as, adaptability, performance and reliability (Bass, Clements, & Kazman, 2003).

Although the addition of quality attributes in complex software systems has received considerable attention (Bass, Clements, & Kazman, 2003), (Brito, de-Lemos, Rubira, & Martins, 2009), (Bradbury, 2004), in the context of ERS it is not considered in an explicit way. Regarding the design of ERS, some authors have proposed multi-agent architectures that can provide personalized services for learners and teachers (Zhu, Ip, & Cao, 2008), (Casili, Gerling, Deco, & Bender, 2010). In such solutions, software agents coordinate work for offering recommendation services including personalized recommendation. (Wang, Tsai, Lee, & Chiu, 2007) propose an adaptive personalized model for learning objects recommendation. They propose to combination of two algorithms for recommending leaning objects: preference-based and correlation-based. (Lu, 2004) presents a framework for recommending learning content by using a multi-attribute evaluation method combined with a fuzzy matching method. A common characteristic of these approaches concerns the fact that in terms of quality attributes, the ERS are developed in an ad-hoc manner, by focusing only on the system functionality (e.g., recommendation algorithm, learner profile management, and domain management), but they do not consider different design decisions according to quality attributes derived from non-functional requirements as part of the development process. This ad-hoc way of dealing with quality attributes can impair the recommendation accuracy or even derail its use in a real environment with, for example, a critical requirement of scalability.

Regarding the quality of ERS, Intelligent Tutoring Systems (ITS) can be seen as an important ally for improving the quality of the ERS, as it

23 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/chapter/systematic-approach-designing-educational-recommender/77757

Related Content

Software Design for Passing Sarbanes-Oxley in Cloud Computing

Solomon Lasluisa, Ivan Rodero and Manish Parashar (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 1659-1674).

www.irma-international.org/chapter/software-design-passing-sarbanes-oxley/77775

A Consensus of Thought in Applying Change Management to Information System Environments

Jeffrey S. Zanzig, Guillermo A. Francia III and Xavier P. Francia (2015). *International Journal of Information System Modeling and Design* (pp. 24-41).

www.irma-international.org/article/a-consensus-of-thought-in-applying-change-management-to-information-system-environments/142514

Formalization of Ontology Conceptualizations Using Model Transformation

Malika Boudia and Mustapha Bourahla (2022). *International Journal of Information System Modeling and Design* (pp. 1-21).

www.irma-international.org/article/formalization-of-ontology-conceptualizations-using-model-transformation/305229

Hierarchical Agent-Based Monitoring Systems for Dynamic Reconfiguration in NoC Platforms: A Formal Approach

Sergey Ostroumov, Leonidas Tsiopoulos, Marina Waldén and Juha Plosila (2014). *Advancing Embedded Systems and Real-Time Communications with Emerging Technologies* (pp. 302-333).

www.irma-international.org/chapter/hierarchical-agent-based-monitoring-systems-for-dynamic-reconfiguration-in-noc-platforms/108449

Software Modernization of Legacy Systems for Web Services Interoperability

Chia-Chu Chiang (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 380-388).

www.irma-international.org/chapter/software-modernization-legacy-systems-web/29398