Chapter XXI
Frameworks for Model-Based Design of Enterprise Information Systems

Mara Nikolaidou
Harokopio University of Athens, Greece

Nancy Alexopoulou
University of Athens, Greece

ABSTRACT

System design is an important phase of system engineering, determining system architecture to satisfy specific requirements. System design focuses on analyzing performance requirements, system modeling and prototyping, defining and optimizing system architecture, and studying system design tradeoffs and risks. Modern enterprise information systems (EIS) are distributed systems usually built on multitiered client server architectures, which can be modeled using well-known frameworks, such as Zachman enterprise architecture or open distributed processing reference model (RM-ODP). Both frameworks identify different system models, named views, corresponding to discrete stakeholder's perspectives, specific viewpoints, and could serve as a basis for model-based system design. The main focus of this chapter is to explore the potential of model-based design for enterprise information systems (EIS). To this end, the basic requirements for model-based EIS design are identified, while three alternative approaches are discussed based on the above requirements, namely, rational unified process for systems engineering (RUP SE), UML4ODP and EIS design framework.

INTRODUCTION

Many different stakeholders are usually involved in the process of constructing an enterprise information system. Each of them focuses on certain concerns and considers these concerns at a certain level of detail. Therefore, various methodologies and frameworks have been developed aiming at the consistent construction of enterprise information systems. Most of them have adopted the notion of *separating concerns* by establishing different *viewpoints*, each depicting the concerns of a specific stakeholder (e.g., user, designer, implementer, etc.). Following, we focus on the *system designer* viewpoint, exploring issues related to the analysis and design of enterprise information systems (EIS).

System design is the process of analyzing system requirements, designing the desired architecture of a system, and exploring performance requirements, ensuring, thus, that all system components are identified and properly allocated and that system resources can provide the desired performance. System design is a phase of *system engineering*, defined by the International Council on System Engineering (INCOSE) as "the interdisciplinary approach and means to enable the realization of successful systems" (INCOSE, 1998). System engineering covers different phases, as the definition of customer needs and required functionality early in the development cycle, the documentation of requirements, the design synthesis, and system validation, while the overall system life cycle, performance, and even maintenance is also considered (Oliver, Kellher, & Keegan, 1997).

System design corresponds to the system designer viewpoint. Although, vendors (as IBM or Oracle) actively promote information system development based on multitiered architectures, the proposed solutions, although expensive, often fail to provide the desired performance. This is due to the fact that system designers often neglect engineering issues contributing to the overall system performance. In practice, discrete issues, as network architecture description or resource allocation, are supported by autonomous automated or semiautomated tools, each of which adopts its own metamodel for system representation. Thus, no interaction between them is supported. To effectively explore EIS design, heterogeneous tools and system models should be integrated. This integration could be accomplished by adopting *model-based system design* (MBSD). MBSD provides a central system model (tool-independent) that captures system requirements and design decisions that fulfill them at different levels of abstraction. It enables integration of system models supported by autonomous design tools and interoperability between them without interfering with their internal implementation. When applying model-based system design, a multilevel, technology-neutral model for EIS representation should be defined, taking into account different aspects of the system, such as network architecture, resource allocation, application execution requirements, and so forth, involved in system design.

Existing well-known frameworks may be used for system modeling. The *open distributed processing reference model* (RM-ODP) is such a framework, dealing with aspects related to the distribution, interoperation, and portability of distributed information systems. Another widely referenced framework is the *enterprise architecture framework* defined by Zachman, which specifies the development process of enterprise information systems, starting from the identification of the enterprise's business objectives and resulting in a detailed system implementation. Independently of the framework used, the different system views defined from each viewpoint should be represented by graphical models enhancing the designer perspective of the system. Graphical models may be expressed using various modeling languages. However, the most popular and widely adopted modeling language is the unified modeling language (UML). System designers commonly use the extension mechanisms provided by UML to create profiles (i.e., specializations of UML diagrams) to better serve their modeling purposes.

The main focus of this chapter is to explore the potential of model-based design for enterprise information systems. To this end, the basic requirements for model-based EIS design are identified, while three alternative approaches are discussed based on the above requirements. Although they are not the only ones, they were chosen since they focus on system design, provide formal EIS models, and adopt UML as the modeling language for EIS representation. These are: a) the rational unified process (RUP) system engineering approach, b) the UML for open distributed processing (UML4ODP) proposed standard with emphasis on engineering viewpoint, and c) the EIS engineering framework proposed by the authors.

Related Content

Hybrid Representation to Locate Vulnerable Lines of Code
Mohammed Zagane, Mamdouh Aleneziand Mustapha Kamel Abdi (2022). *International Journal of Software Innovation (pp. 1-19).*
www.irma-international.org/article/hybrid-representation-to-locate-vulnerable-lines-of-code/292020

Dynamic Analysis and Profiling of Multithreaded Systems
Daniel G. Waddington, Nilabja Royand Douglas C. Schmidt (2010). *Advanced Operating Systems and Kernel Applications: Techniques and Technologies (pp. 156-199).*
www.irma-international.org/chapter/dynamic-analysis-profiling-multithreaded-systems/37949

Agile Software Methods: State-of-the-Art
E. Mnkandlaand B. Dwolatzky (2007). *Agile Software Development Quality Assurance (pp. 1-22).*
www.irma-international.org/chapter/agile-software-methods/5066

A Genetic Algorithm-Based QoS Analysis Tool for Reconfigurable Service-Oriented Systems
I-Ling Yen, Tong Gaoand Hui Ma (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications (pp. 3118-3141).*
www.irma-international.org/chapter/genetic-algorithm-based-qos-analysis/29553

Introduction
Neal G. Shaw (2001). *Strategies for Managing Computer Software Upgrades (pp. 1-2).*
www.irma-international.org/chapter/introduction/98484