A Metamodel for Automatic Generation of Enterprise Information Systems

Sofia Larissa da Costa[1,*], **Valdemar Vicente Graciano Neto**[1], **Luiz Fernando Batista Loja**[1], **Juliano Lopes de Oliveira**[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
P.O. Box 131 – Campus II – CEP 74001-970 – Goiânia – GO – Brazil

{sofia, valdemar, luizloja, juliano}@inf.ufg.br

Abstract. *Building Information Systems (IS) is an expensive and complex activity involving (i) domain modeling; (ii) user tasks modeling; (iii) modeling of appearance and behavioral aspects of the interaction; and (iv) coding these aspects in a programming language. This article presents an Enterprise Information System (EIS) Metamodel that integrates business data and User Interface (UI) description, allowing the automatic generation of the code in a Model-Driven approach. This facilitates not only the construction but also the maintenance, evolution, and reuse of EIS.*

Keywords: *User Interface metamodel, Information System metamodel, Business Domain metamodel.*

1. Introduction

Information Systems (IS) construction involves organization and control of complex business information. The Model-Driven Development (MDD) paradigm makes models a first class citizen of the software development processes [Mraidha et al. 2008]. This paradigm automates the transformations from source models, representing the different aspects of IS, to target models [Mellor et al. 2003]. The latter models are, in general, software code in some programming language, and each source model is based on a Domain Specific Language (DSL) or metamodel [Vale and Hammoudi 2009].

The framework described in [De Almeida et al. 2009] follows this MDD approach to generate specialized Enterprise Information Systems (EIS) based on a metamodel that conceptually represents the structural aspects of EIS domain, such as business concepts, entities, relations, and static constraints. The framework metamodel is an object-oriented variation of the classic Entity-Relationship (ER) Model and can be considered a DSL for EIS specification. This metamodel allows the automatic generation of user interfaces for EIS [Da Silva and de Oliveira 2009].

In spite of the increasing productivity in software development, the application of this framework to build complex EIS has indicated some limitations: 1) user interface and business domain features are highly coupled in the metamodel; 2) the generated user interface supports only a single interaction model and it is not customizable; and 3) the framework lacks business processes support.

*CAPES provides financial assistance for this project.

Intending to overcome these limitations, and based on the lessons learned from the construction and utilization of the framework, we propose that MDD of EIS be based on three main aspects: the Business Domain; the Business Processes; and the User Interface. Business Domain comprises business data and associated rules that will be modified and applied by the IS. Business Processes organize actions and tasks to generate and to modify business data, improving the understanding of the activities and their interrelationships [Wesk 2007]. User Interaction defines the way business stakeholders interact with the IS, specifying actions that are converted in stimulus to perform a task in the underlying IS [Valverde et al. 2007].

According to this proposal, we are currently refactoring the framework to decouple Interaction aspects from Domain aspects, and to provide support to Business Processes. In this sense, a new User Interface Metamodel was introduced [da Costa and de Oliveira 2010], focusing on the support of varying appearance and behavioral styles, in order to enable customizable user interfaces. We are also developing a Business Process Metamodel and a Process Execution Engine to assist business processes specification of generated EIS.

This paper discusses the integration of the Business Domain and User Interface aspects in the refactored EIS MDD framework. This discussion is based on the two involved metamodels: the Business Domain Metamodel and the HCI (Human-Computer Interaction) Metamodel. Due to space limitations, the Business Process aspect of the new framework is not discussed in this paper. Section 2 briefly introduces our theoretical background and related works. Section 3 explains the integration of the metamodels in the EIS framework. Section 4 presents an example of the application of the integrated models. Section 5 summarizes the key points of our proposal and indicates directions for future works.

2. MDD of Information Systems

In a conventional software development environment, workflow, model-to-model and model-to-code transformations are manual, error-prone, difficult and expensive tasks. The focus of modern development methodologies has shifted from implementation technology concerns to modeling the semantics of complex domains. One of the goals of modeling technologies is to map modeling concepts directly to domain-specific concepts [Wada et al. 2005]. In order to ensure that the involved models are understandable by computing tools, the MDD approach for IS construction requires specific modeling languages supported by appropriate tools [Bittar et al. 2009].

Considering the User Interface (UI) aspect of IS, recent research has focused in high level modeling and in automatic generation of UI code [Falb et al. 2007]. The MDD approach to automatic transformation of models (in a high level abstraction) to executable code has been a successful strategy in this context [Moreira and Pimenta 2006]. However, no consensus has been reached as to which models are the best for describing the UI and no standard method has emerged from the MDD IS generation initiatives for representing the interaction between users and system [Ahmed and Ashraf 2007, Vanderdonckt 2008, Valverde et al. 2007, Memmel and Reiterer 2008].

Several UI design initiatives based on the MDD approach deal with IS modeling [Ahmed and Ashraf 2007, Vanderdonckt 2008, Calvary et al. 2003]. Nevertheless,

no current method joins UI and Software Engineering aspects, and existent technologies and tools are not appropriate for this purpose [Memmel and Reiterer 2008]. The gap between the involved disciplines becomes larger when Business Process Modeling is included in the IS context. Thus, an approach that models IS from the perspectives of these three disciplines is necessary, since current IS UI approaches are not prepared to bridge this gap.

The Abstract Interaction Model [Valverde et al. 2007], for instance, does not provide a structured definition of concepts. Moreover, there is no concern with either the appearance of UI elements or the related business logic. Alternative approaches also have limitations: the Conceptual Discourse Metamodel lacks reusability [Popp et al. 2009] , while the Interface Metamodel [Da Silva and de Oliveira 2009] only generates UI for CRUD (Create, Retrieve, Update and Delete) operations.

Considering the Business Domain aspect of IS, UML is the most used modeling language for design of systems in MDD approaches, although BPMN is gaining increasing attention. UML is a general-purpose modeling language providing concepts to ensure behavioral completeness of models [Mraidha et al. 2008], but it has some drawbacks such as the limited support to temporal aspects and the lack of detailed, formal semantics needed for formal analysis and code generation.

Domain-Specific Metamodels (DSL), on the other hand, are designed to be useful for a limited set of tasks and provide a good basis for domain-specific formal analysis and fully-automated tool support [Brucker and Doser 2007, Jouault and Bézivin 2006]. However, to the best of our knowledge there are neither DSL designed to EIS modeling, nor a metamodel that integrates Business Domain, Business Processes and UI aspects. The next section introduces Domain and UI Metamodels to overcome these limitations.

3. Information Systems Metamodels

This section describes the integration of Business Domain and UI metamodels to provide adequate support to IS modeling and code generation. Because of space, we omit the discussion of the third metamodel of our framework, the Business Process Metamodel, which is described in a forthcoming paper.

3.1. The Human-Computer Interaction Metamodel

The Human-Computer Interaction (HCI) Metamodel describes appearance and behavioral aspects of the IS interaction. The proposed metamodel is based on the *UI Stereotype* concept, introduced in [da Costa and de Oliveira 2010], which identifies, in a high level of abstraction, features of UI appearance and behavior independently of the underlying IS application.

Using stereotypes to define interfaces promotes consistency, standardization and reuse of both UI appearance and behaviour. From simple UI stereotypes (tables and button panels, for example), it is possible to generate complex compositions of UI stereotypes that are important to many IS, such as query pages, spreadsheets, web portals and master-detail reports.

We created a taxonomy of UI stereotypes, defining UI elements, and a strategy to specify UI behaviour and to integrate this behaviour with the business logic in the

subjacent IS. Each UI stereotype defines the interaction objects that must be created and associates these objects with a behaviour specified via interface rules (implemented by the framework) and application rules (obtained from the underlying IS). Application rules provide the integration of the UI with the IS via the framework, as described in [De Almeida et al. 2009].

Figure 1 presents the HIC metamodel using UML notation. Elements represented in white boxes describe UI appearance (the UI Metamodel), while gray blocks describe behavioural aspects of the HCI. The external package represents the underlying IS generated by the framework.

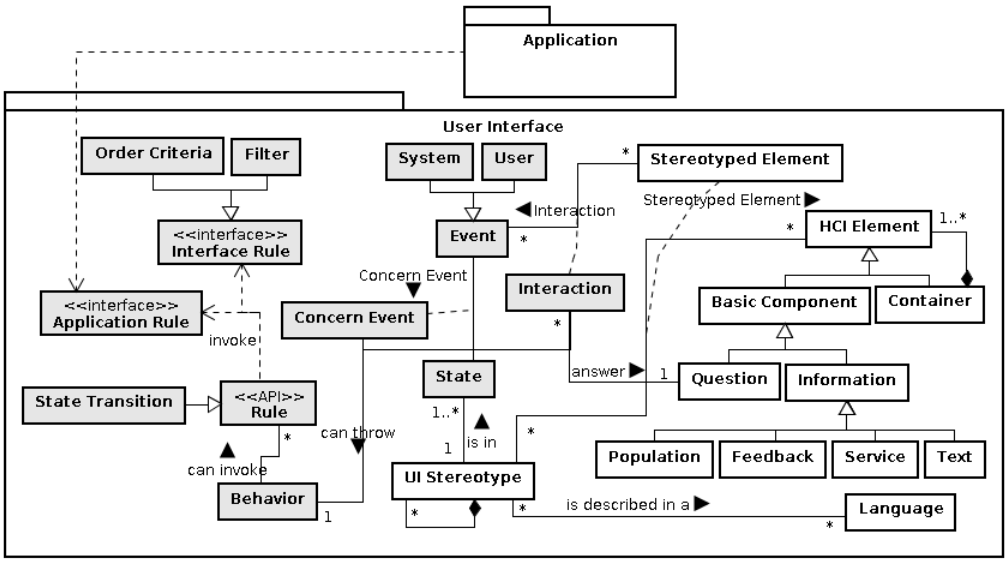


Figure 1. HCI Metamodel

3.2. The Business Domain Metamodel

The Business Domain (BD) Metamodel is described in Figure 2. To generate an IS application it is necessary to instantiate the IS business concepts using the BD metamodel modeling primitives. An IS application is described as a set of Business metadata which can be classified as Entities, Weak Entities (dependent concepts) or Relationships. Business Rules can be defined on these metadata.

This metamodel solves a classic modeling problem which arises when a given entity type participates more than once in a relationship type with different roles. For example, the SUPERVISION relationship type relates an employee to a supervisor, where both employee and supervisor are members of the same EMPLOYEE entity type. By defining a Relationship specialization, the metamodel supports this restriction in a natural way instead of transfering this kind of concern to the application logic (even though this logic is automatic generated).

The BD Metamodel also describes a taxonomy for Business Data Attributes, and the concept of Enumerated Values, which supports application internationalization.

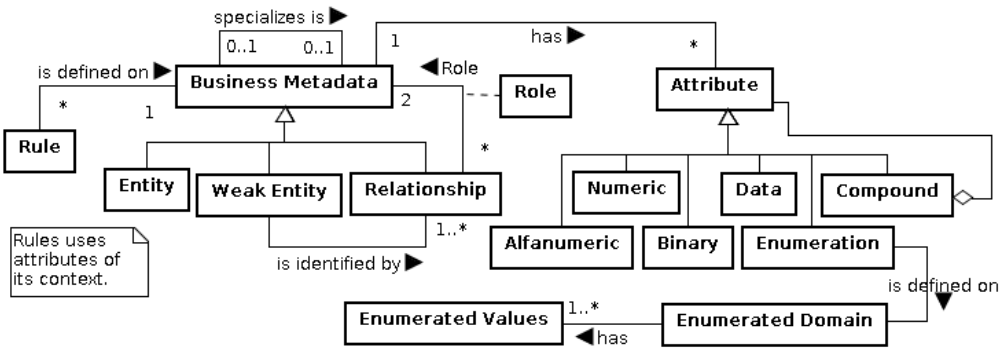


Figure 2. Business Domain Metamodel

3.3. Metamodels Integration

The HCI and BD Metamodels integration is described by Figure 3. Every behaviour triggered by the UI is treated by either a business rule or an application functionality generated to connect to one or more UI rules.

The EIS framework provides some functionalities that assist data transfering within the IS architecture. The framework allows to register and manage business rules using OCL and generates stored procedures in the underlying database management system that implements the modeled business rules. These rules are shared by the UI and other components of the IS.

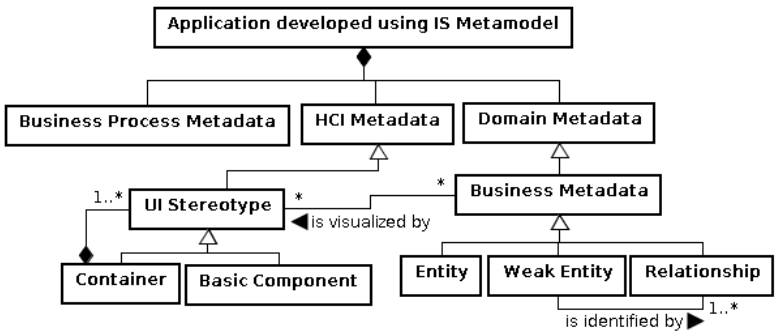


Figure 3. Metamodels Integration Overview

4. Metamodel Application: a simple example

To demonstrate a practical application of the integrated metamodels within the EIS framework, we describe the generation of a simple IS function that integrates UI aspects as well as the underlying IS through business domain rules. The example is the same used in [De Almeida et al. 2009], ilustrating a Person registration form (Figure 4).

Using the metamodel we first create the UI stereotypes. The Application stereotype constitutes the main part of the UI, containing menus, toolbars and exhibiting opened windows; the CRUD stereotype includes edition, query, messages and UI services; and the Form stereotype contains the edition area. The CRUD stereotype contains the Form

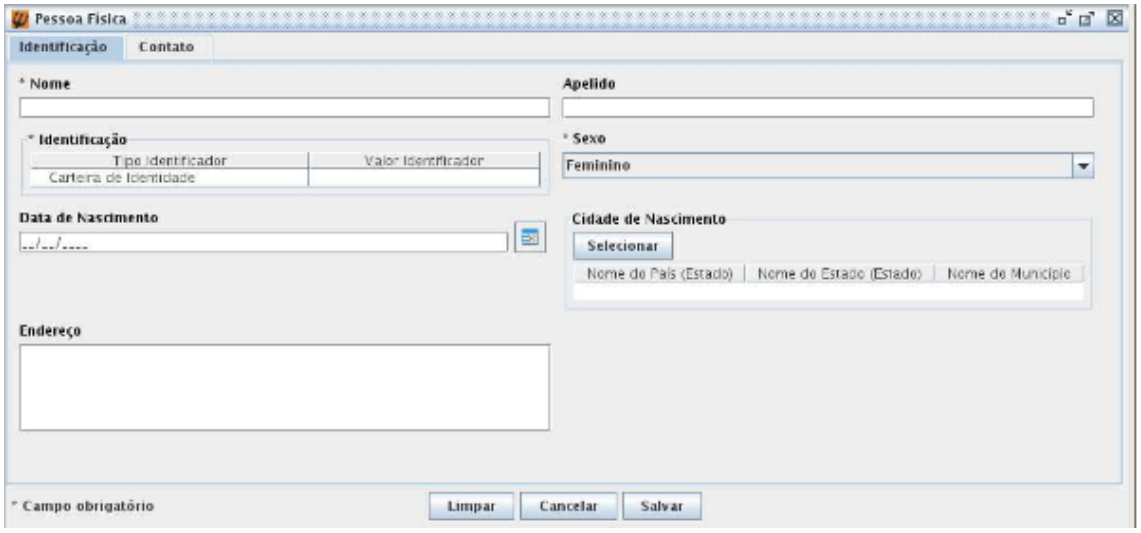


Figure 4. EIS Application Example

stereotype and is part of the Application stereotype. The instantiation of these stereotypes depends on the Business Data to be manipulated (the Person entity, in this case).

When a user clicks in the Save ("Salvar") button, the behaviour triggers the rule Validate Person (that is an IS business rule). If this rule returns true (valid data), the rule Insert Person (pertinent to the business logic) is executed. Finally, the interface rule "Exhibits Message" is called to display a message that informs the user about the result of the selected UI action.

5. Conclusions

Although many initiatives to construct Information Systems (IS) applications in a model-based approach have emerged, they are not capable of supporting features of both UI and Business Domain logic in an integrated way. This article presents such an integrated approach, identifying, colecting and grouping modeling primitives presented in distinct proposals in a single framework. Thus, this work contributes to integrate concepts of HCI patterns with business data aspects, defining a Metamodel to construct IS, and also promoting the maintainability of these aspects.

The HCI metamodel describes UI considering appearance and behaviour aspects. It uses Interaction Patterns [Valverde et al. 2007] to describe appearance and maps interface behaviour to behavioral rules associated with those patterns. This allows the UI behaviour to be described in a structured way, and also enables the integration of the UI Metamodel with the Business Domain Metamodel via business rules triggered by the user interaction.

The metamodels described in this paper are being implemented in a refactoring effort that will produce the second version of the EIS framework described in [De Almeida et al. 2009]. This proposal presents benefits in reuse, productivity, consistency and standardization of IS due to automatic code generation based on the described metamodels. Besides that, the metamodels promote an integration between the UI behaviour and the application logic in the underlying IS.

References

Ahmed, S. and Ashraf, G. (2007). Model-based user interface engineering with design patterns. *Journal of Systems and Software, Vol. 80, n. 8*, pages 1408–1422.

Bittar, T. J., Fortes, R. P. d. M., Lobato, L. L., and Watanabe, W. M. (2009). Web communication and interaction modeling using model-driven development. In *Proc. of ACM international Conference on Design Communication*.

Brucker, A. D. and Doser, J. (2007). Metamodel-based uml notations for domain-specific languages. In Favre, J. M., Gasevic, D., Lämmel, R., and Winter, A., editors, *4th International Workshop on Software Language Engineering*.

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Jean, V. (2003). A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15:289–308.

da Costa, S. L. and de Oliveira, J. L. (2010). Construção e manutenção baseadas em modelos de interfaces para usuário em sistemas de informação. In *Anais do III Workshop de Teses e Dissertações de Sistemas de Informação do VI Simpósio Brasileiro de Sistemas de Informação*.

Da Silva, W. C. and de Oliveira, J. L. (2009). Gerência de interface homem-computador para sistemas de informação empresariais: uma abordagem baseada em modelos. *iSys - Revista Brasileira de Sistemas de Informação, Vol. 2, 2009*.

De Almeida, A. C., Boff, G., and de Oliveira, J. L. (2009). A framework for modeling, building and maintaining enterprise information systems software. In *Proc. of the 23th Brazilian Symposium on Software Engineering*, pages 115 – 125.

Falb, J., Popp, R., Röck, T., Jelinek, H., Arnautovic, E., and Kaindl, H. (2007). Fully-automatic generation of user interface for multiple devices from a high-level model based on communicative acts. In *Proc. of the 4th Annual Hawaii International Conference on System Sciences*, page 10.

Jouault, F. and Bézivin, J. (2006). Km3: a dsl for metamodel specification. In *Formal Methods for Open Object-Based Distributed Systems*. Springer Berlin / Heidelberg.

Mellor, S. J., Clark, A. N., and Futagami, T. (2003). Model-driven development. *IEEE Software, Vol. 20, No. 05, September/October 2003*, pages 14 – 18.

Memmel, T. and Reiterer, H. (2008). Inspector - interactive ui specification tool. In *Proc. of the 7th International Conference On Computer Aided Design of User Interfaces (CADUI)*, pages 161 – 174. Springer.

Moreira, A. and Pimenta, M. S. (2006). Reuso de interfaces através de padrões concretos de interação. In *Proc. of VIII Brasiliam Symposium on Human Factors in Computing Systems*, pages 41 – 44.

Mraidha, C., Tanguy, Y., Jouvray, C., Terrier, F., and Gérard, S. (2008). An execution framework for marte-based models. In *Proc. of 13th IEEE International Conference on Engineering of Complex Computer Systems*.

Popp, R., Falb, J., Arnautovic, E., Kaindl, H., Kavaldjian, S., Ertl, D., Horacek, H., and Bogda, C. (2009). Automatic generation of the behavior of a user interface from a high-

level discourse model. In *Proc. of the 42nd Annual Hawaii International Conference on System Sciences (HICSS-42)*. IEEE Computer Society Press.

Vale, S. and Hammoudi, S. (2009). Comode: A framework for the development of context-aware applications in the context of mde. In *Proc. of Fourth International Conference on Internet and Web Applications and Services*.

Valverde, F., Panach, I., and Pastor, O. (2007). An abstract interaction model for a mda software production method. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual Modeling*.

Vanderdonckt, J. (2008). Model-driven engineering of user interfaces: Promises, successes, and failures. pages 1 – 10.

Wada, H., Suzuki, J., Takada, S., and Doi, N. (2005). A model transformation framework for domain specific languages: An approach using uml and attribute-oriented programming. In *Proc. of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics*.

Wesk, M. (2007). *Business Process Management: concepts, languages, architectures*. Springer.