A User Interface Stereotype to build Web Portals

Sofia Larissa da Costa
Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos - Brazil
sofialc@icmc.usp.br

Valdemar Vicente Graciano Neto
Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos - Brazil
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiania - Brazil
valdemarneto@icmc.usp.br

Juliano Lopes de Oliveira
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiânia - Brazil
juliano@inf.ufg.br

***Abstract*—Software Engineering for Web Systems domain is a complex process where reuse and productivity are desirable attributes. It involves, among other aspects, modeling user interface (UI) software and its binding to underlying applications business logic and process. Despite recent advances, UI engineering for this domain is still expensive, laborious, and error-prone. On the other hand, Model-Driven Development (MDD) has emerged as a paradigm to bridge reuse and productivity gaps by means of abstract models and automatic software generation through model transformations. However, model-based UI engineering (or MDD for UI) is still an emerging discipline. This paper applies a recent MDD for UI research advance, namely the UI Stereotype, to the UI engineering in the Web Systems domain. The UI Stereotype captures UI specificities, modeling recurrent UI presentation and behavior, abstracting users' interactions and tasks. We apply this concept to describe a Web Portal UI Stereotype as a recurrent interaction pattern that enables the automatic generation of many Web Portal UI components based on model-driven practices. This UI generation approach is compliant with recent advances in UI construction, such as Interaction Flow Modeling Language (IFML), a UI modeling language recently standardised by OMG. Moreover, our approach reduces Web Portals UI software development time-to-market, efforts and costs, contributing to both quality and productivity, and improving maintainability of Web applications.**

***Keywords*-User Interface; Stereotype; Web Portals; Model-Based User Interface; Model-Driven Development.**

I. Introduction

Web Information Systems construction involves organization and control of complex business information [1]. Web applications range from simple information display and conventional Create, Retrieve, Update and Delete (CRUD) forms to complex and highly interactive social networks, search engines and Web portals. Thus, designing and implementing User Interface (UI) software for Web applications can be an expensive and error-prone process [2].

Engineering UI software involves multidisciplinary skills and deals with challenges that start in modelling the mental profile of Web users and continues through designing attractive layouts and dealing with a plethora of languages, frameworks and related UI technologies [3].

On the other hand, Model-Based UI (MBUI) development is one of the mainstream UI Engineering methods [4]. It is defined as the process of creating and refining high level models for the automatic generation of UI. The use of abstract models enhances the user comprehension of how the requirements are transformed into UI software and facilitates the reuse of UI concepts [5], [6], [7]. Thus, MBUI approaches can hide complexities, such as mapping several models, or using different implementation languages, improving the UI construction process by focusing directly on a conceptual representation of the interactive features [8].

In this sense, MBUI approach for automatic building of Information Systems UI presented in [9] takes into account the behavioral and presentation aspects of that software application domain and overcomes important limitations of current methods, namely the specification of applications behavior in response to user interactions, and the integration of UI and the underlying software.

The main concept of this approach, **UI Stereotype**, captures UI similarities of some specific software application, and models UI presentation and behavior recurrent features. It abstracts users' interactions and tasks and defines the way information is presented and manipulated for each particular task in the UI [9]. In this way, UI Stereotype contributes to reuse and standardization since to model an UI using a known UI Stereotype consists in mapping the Domain Concepts to the UI Stereotype elements.

This paper defines a UI Stereotype for a recurrent way of interaction in Web Systems domain: Web Portals, a gateway to information and services on the Web [10]. Modeling and building Web Portals based on a UI Stereotype is important because of the continuously evolving of Web Systems applications, which must be frequently updated to aggregate information and make it available for several user profiles. The use of the Web Portal UI Stereotype is illustrated with real examples to demonstrate its capability

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.8

of improving Web Portals building with the application of the MBUI approach of [9].

The remainder of this paper is organized as follows. Section II presents the main concepts and summarizes the proposed approach to build UI. Section III describes the Web Portal UI Stereotype and shows some UI synthesis from it. Section IV discusses the applied approach to develop the Web Portal Stereotype and its correlation to IFML OMG standard. Finally, Section V concludes the paper and points to future works.

II. Model-based Approach to build User Interfaces

Model-Driven Development (MDD) is a software development life cycle [11] where models are the first-class citizens. It addresses system complexity by the intense use of models [12]. Under this paradigm, software development is treated as a set of model transformations that are conducted from the earliest steps of the software development process to the latest stages. After subsequent transformations, the process delivers a deployable, complete and tested software product [13], [14]. It promotes knowledge reuse through models, reduces time-to-market, and increases productivity by means of software automatic generations from model transformations.

In MDD, models must conform to their respective metamodels [15] and transformations are mappings between a source metamodel and a target metamodel [11]. A lot of acronyms have emerged in literature to describe model-driven practices along the last years. Prominent among the MDD initiatives is OMG's Model-Driven Architecture (MDA) [16]. MDA standard defines different model categories as follows [13]:

- Computation Independent Model (CIM), representing the problem domain;
- Platform Independent Model (PIM), representing the solution domain without platform specific details;
- Platform Specific Model (PSM), representing the solution domain with platform specific details.

MBUI development is a MDD approach that uses metamodels to specify essential characteristics for the interaction design through a high-level UI description [17], [18]. The CAMELEON reference framework establishes the basis for this discipline and it is a MDA-compliant approach [3], since this framework classifies the involved models in a similar way. This framework decomposes UI development in four steps: (1) Modeling Domain Tasks and Concepts, considered as CIM; (2) Definition of Abstract UI containers and components, interpreted as PIM; (3) Concretization of UI components, considered as PSM; (4) and Final UI generation, that corresponds to the executable code [4], [3].

The MBUI approach for automatic building of UI applied in this paper [9] uses this framework, disconsidering task modeling, which is a well established technique [19]. Three metamodels are used to describe ISUI [9]:

1) Domain Metamodel, that corresponds to domain concepts;
2) HCI Metamodel, used to define the abstract UI;
3) Presentation Metamodel, that represents the UI in a concrete level.

The first step in this approach, *Modeling Domain Concepts*, is done by building an instance of the Metamodel Domain. It is based on the UML Class Diagram and defines the main Web Information System (WIS) domain concepts and it is used to model domain concepts of business data and associated rules to be applied by WIS, as reported in [20].

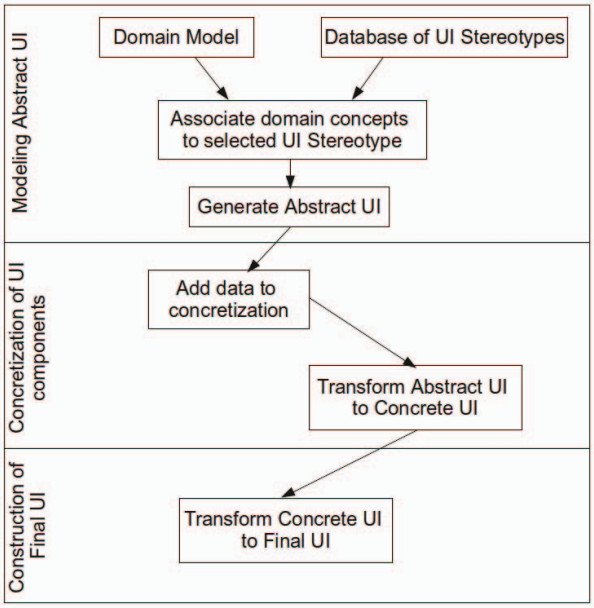

Figure 1. Process to model-based generation of UI

The other steps of the reference framework – defining *Abstract UI*, *Concrete UI* and *Final UI* – involve UI modeling and construction. Figure 1 summarizes this steps of the approach to build a WIS User Interaction (UI). For this, we use the UI Stereotype concept [20], [9], that should be a dominant element in UI which is an abstraction of organization and type of UI elements as well as the proper behavior of UI which perform the same tasks and present the same type of information and interaction.

The second step is modeling abstract UI, as seen in Figure 1. In this sense, the HCI Metamodel aims to describe presentation and behavioral characteristics of WIS UI in abstract level. These UI make intensive use of information stored in databases and organize tasks related to business processes. The key concept of HCI Metamodel is the User

Interface Stereotype, or simply **UI Stereotype**, which consists of an abstraction of the UI intention, independent of the application or underlying software. The UI Stereotype enables to abstract interactions and tasks of the user as well as the form of presentation of information for a particular task. Thus, each stereotype is a UI dominant element, which organizes the presentation of the subordinate elements and its behaviors [9]. An instance of UI Stereotype contains at least one UI Element, according to the taxonomy of UI Elements [9]. Also, the UI actions are predefined by the UI Stereotype. Thus, all instances of a UI Stereotype execute the same behavior. Actions which depend on the application or other external system are specified as external actions and are associated to an instance of UI Stereotype in this step.

In this way, to obtain an abstract UI we select an existent UI Stereotype to associate concepts of business element in the Model Domain which should be in the UI. If the given task do not have a UI Stereotype, we should model it in this step. Using a UI Stereotype, software engineers only configure each HCI element and the actions according to the selected stereotype. In other words, since we select an existent UI Stereotype, we should associate each domain concept to an UI element in the UI Stereotype. Also, the external actions are associated to the correspondent application.

The third step is to obtain the concrete UI, through Presentation Metamodel, which enables UI description in a concrete level, i.e. the target computational platform. It is worth noting that the Presentation Package of the HCI Metamodel deals with abstract definitions of UI appearance, while the Presentation Metamodel aims on mapping both abstract appearance and behavior to concrete UI components to allow code generation [9]. In this way, the obtained model in this step contains data regards to the target platform. Similar to the previous step, if a UI Stereotype is selected, there must be a template in the target computational platform that represents the UI Stereotype selected in a concrete level.

Finally, the fourth step is to generate the final UI in runtime, which is done by a tool that generates the final UI code using the obtained models. However, it is important to obtain a set of UI Stereotypes in order to use this approach. [20] presents the CRUD Stereotype and [9] presents the Survey Stereotype. In this sense, this paper presents another UI Stereotype: Web Portal. We focus on the first three steps. Next section details building of Web Portal Stereotype.

III. Web Portal UI Stereotype

A Web Portal is a special Internet site designed to act mainly as a gateway to give access to other sites [10]. It offers centered access to relevant content and applications for a given interest, allowing the users to access and interact with business elements, applications and processes. Also named as Enterprise Information Portal, a Web Portal can provide access to a wide variety of information about an organization [21], [22], helping users to find products, services and information from this organization in the Web.

The first step to define a UI Stereotype as Web Portal is to model the domain concepts. Figure 2 summarizes the main domain concepts used in a Web Portal. The main business element in this domain is the Web Portal, which brings together information, services and applications regarding a specific organization. Some data of the organization is required, such as name, address, contacts and logo (that should be a graphical element). A Web Portal presents subsites and services to users. There is a rule that defines how to execute a service.

The second step to generate a UI Stereotype is to obtain the HCI Model, that presents the UI Elements required and the organization of them in the proposed UI Stereotype. Figure 3 presents an abstract view of Web Portal Stereotype, adapted from [22]. In this view, a Web Portal is divided into:

1) Top: identification of organization;
2) Horizontal Navigation: it allows to access subsites and subsystems;
3) Vertical Navigation: it allows to execute specific tasks in the subsystems.
4) Content: presents the content of the system or adds other stereotypes, such as Survey [9];
5) Footer: informative data about the developed system.

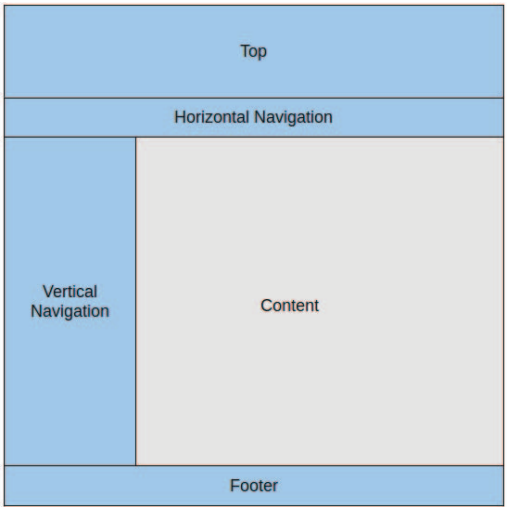

Figure 3. Abstract View of Web Portal Stereotype

In addition, we note that in recent years the initial page of Web Portals has a different view, ommiting the vertical navigation and inserting in central part the main tasks and tasks of the system. Figure 4 shows the abstract view of initial page.

In this way, we modelled the Web Portal as an instance of HCI Metamodel. Figure 5 presents the HCI elements for a Web Portal Stereotype. Moreover, this model presents the

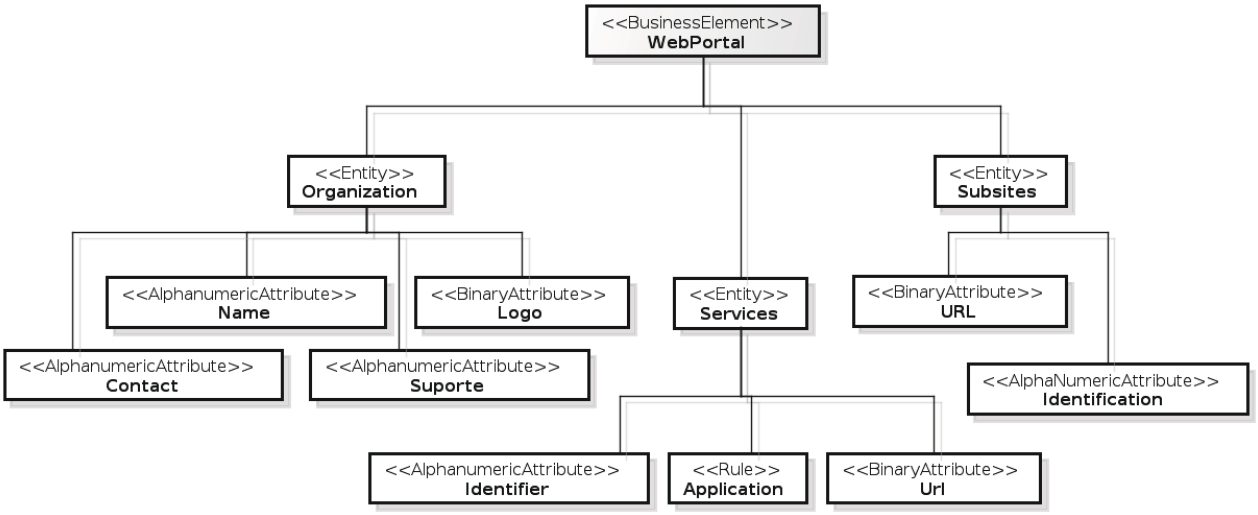

Figure 2. Domain Model of Web Portal

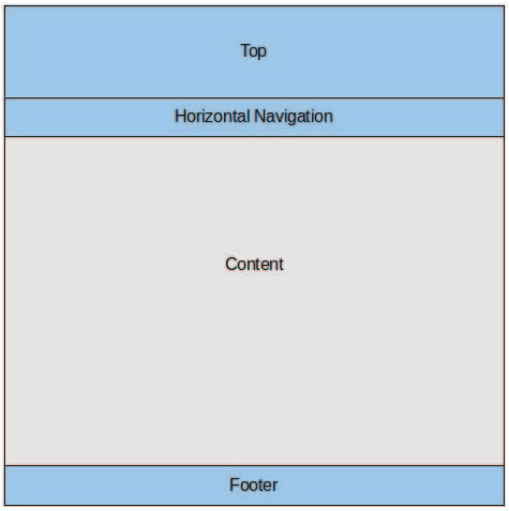

Figure 4. Abstract View of Initial Page of Web Portal Stereotype

type of action that should be executed in this stereotype, i.e. Interface Actions since these actions allows the navigation between pages of each system.

The next step is generating the Presentation Model of Web Portal. This step consists of a mapping between the HCI model and Presentation Model. Figure 6 presents the Presentation Model for Web Portal. This figure represents the concrete classes that will be generated to execute a Web Portal in a web browser. Layout, that is related to UI panels position, is described in a JSF file. Moreover, each panel points to a CSS (*Cascading Style Sheets*) class, that describes the appearance format. The class WebPortal should have a method that allows the navigation between pages, through the Interface Rule executed in menu items.

A. Example

Figure 7 presents an example of Web Portal for Instituto de Informática of Universidade de Goiás[1], in Brazil. Figures are in Portuguese, but we analyze the structure and organization of them. In Figure 7 we note that the structure and organization is similar to the abstract view shown in Figure 3. The identification of the institute is on top of the INF Web Portal. Follow the top, we note a horizontal navigation with some subsites. On the left of page, there is a vertical navigation bar categorized according to interest information.

However, this portal is a subsystem of the Universidade de Goiás[2], and its initial page is seen in Figure 8. This page is similar to abstract view of a initial page of a Web Portal, as seen in Figure 4, since no exists the vertical navigation on left.

Figure 9 shows the HCI model for INF Web Portal. This figure summarizes the abstract HCI elements of the INF Web Portal. This elements can be mapped to the Presentation Model and them to the application that is presented in Figure 7.

IV. Discussion

Web Portals are ubiquitous Web applications. This paper shows that the UI of this kind of application has the same intention and can be modelled as a single UI Stereotype, promoting reuse of presentation and behavioral aspects. [9] shows how UI Stereotype is distinct from other MBUI development proposals for Web Systems generation.

[1]http://www.inf.ufg.br/
[2]http://www.ufg.br/

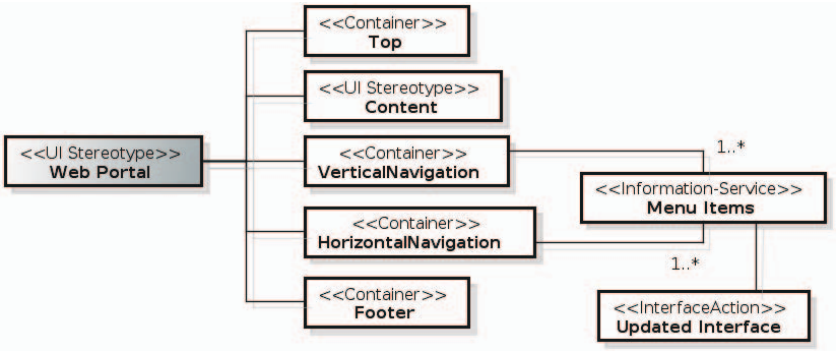

Figure 5. HCI Model for Web Portal Stereotype

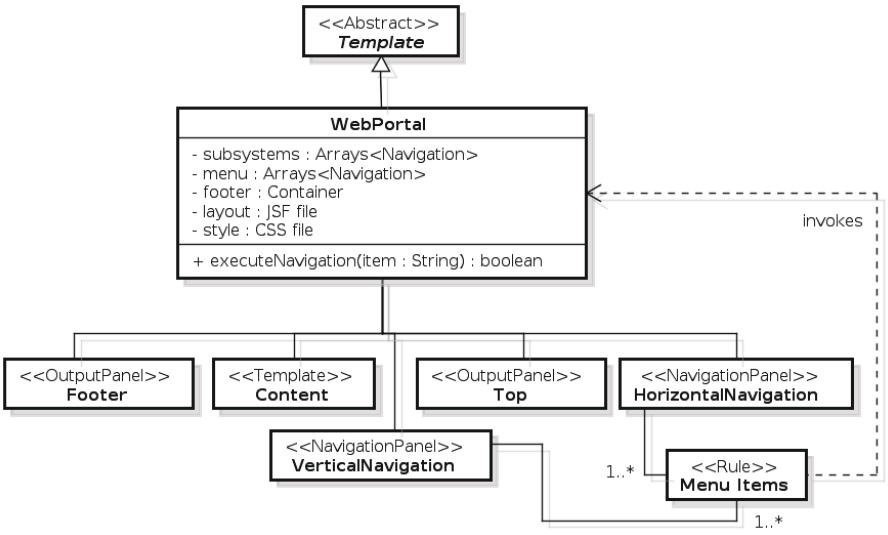

Figure 6. Presentation Model for Web Portal Stereotype

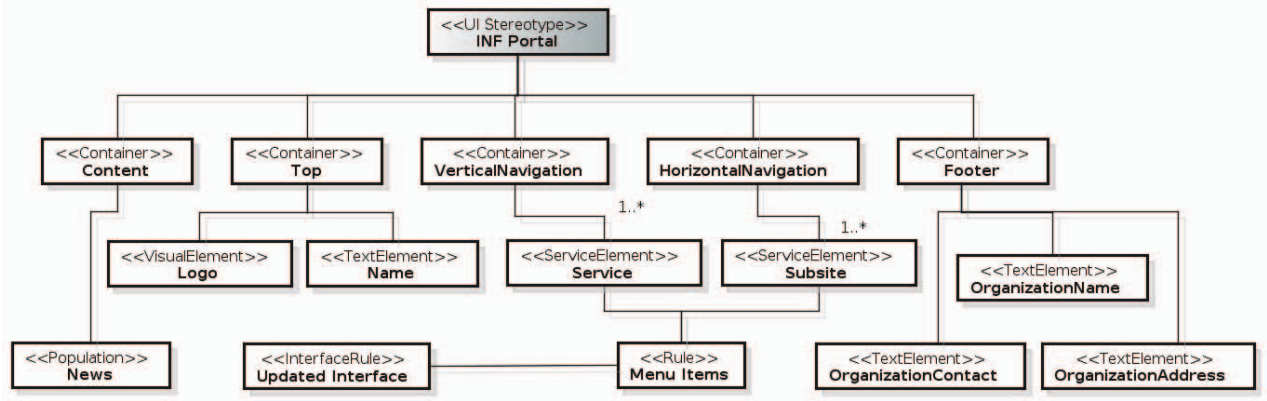

Figure 9. HCI Model for INF Web Portal

Meanwhile, Interaction Flow Modeling Language (IFML) has emerged as an OMG standard [3] and consists of a model-

[3]http://www.omg.org/

Figure 7. Example of Web Portal

Figure 8. Example of Initial Page of Web Portal

driven approach to express the content, user interaction and control behavior of the front-end in (Web) software applications [23]. Therefore, IFML has concerns that also appear in our approach, but the UI Stereotype is not a substitute for IFML. Table I presents a comparison between IFML and the MBUI Stereotype approach we propose in [9]. In fact, we could adapt our metamodels to cope with IFML since the UI Stereotype is an abstraction that can be constructed over an IFML model, taking advantage of tools supporting this language, such as WebRatio [4].

IFML is divided in three packages: the Core package, the Extension package and the DataTypes package. The Core package contains the concepts that build up the interaction infrastructure of the language in terms of InteractionFlowElements, InteractionFlows and Parameters. Core package concepts are extended by concrete concepts in the Extension package with more complex behaviors. The DataTypes package contains the custom data types defined by IFML [23].

UI Stereotype is an abstraction that captures UI similarities in presentation and behavior aspects to model recurrent front-end of software applications. Thus, UI Stereotype is closer to a *pattern for HCI*, since it documents a prescription for a recurrent problem (UI modeling), a context and a systematic repeatable solution. IFML focuses in the structure and behavior of the application as perceived by the end user [23]. The MBUI approach using UI Stereotype can be seen as a concrete initiative in conformance to IFML principles which offers a concrete collection of materialized recurrent visual interaction patterns. A UI Stereotype, such as Web Portal, offers a complete set of presentation appearance and expected behavior, as recommended by the IFML specification.

IFML specification classifies itself as a PIM [23]. The MBUI approach using UI Stereotype could be seen as a PSM for IFML that meets the classic features of a standard user interaction: the visual structure, the navigation model,

[4]http://www.webratio.com/

Table I
COMPLIANCE ANALYSIS

Quality Criteria	MBUI approach to build ISUI (PSM) [9]	IFML [23] (PIM)
Beautification	It supports via View Container.	CSS Insertion in Presentation Model with linking to a concrete View Container element.
Layout	Modeled as a diagram of View Containers.	Template in JSF
Separability	Supported by different Action Concepts	Use of different models
Intention	Implemented in concrete level	A set of UI Stereotypes
Decomposition		Taxonomy of UI elements in HCI Metamodel
Standardization	Supported by a grouping of View Containers	UI Stereotypes
Clarity	Supports once UI Stereotype is in accordance with IFML	Mapping rules to different models
Flexibility	MDD transformation technologies enables that and IFML public metamodel allows it to be done.	Information addition in concrete level
Direction	Mapping between IFML and our model encompasses that	Mapping rules
Generality	Totally supported by IFML once it abstracts UI structure, navigation, and business association	HCI Metamodel
Structure	UI concepts are documented in IFML metamodel	Metamodels described in UML
Contextualization	Implemented in concrete level	Mapping Domain Metamodel to HCI Metamodel
Correlability	Totally supported	Integration of the three models

and the underlying software behavior through business rules and actions. To elucidate similarities, an excerpt of IFML metamodel is described in Figure 10 (the View Elements). It presents the View Containers that we use to argue, in Table I, that we can accomplish our respective concrete implementation conforming to the IFML metamodel.

Table I shows an adaptation of a comparison made between the UI Stereotype concept and other user interaction representation initiatives presented in [24]. This table takes into account a set of quality criteria organized in a taxonomy in accordance with presented challenges for MBUI approaches in [24]. Thus, a MBUI approach should satisfy these criteria to be considered a suitable MBUI approach. Considering IFML as a PIM and our approach under a PSM perspective, Table I presents how IFML supports those features which a MBUI approach should satisfy, and how the UI Stereotype approach implements the respective features in an implementation.

The requirements were divided into a taxonomy of five classes related to MBUI approaches (such as IFML) established in [24]: Organization, Decomposition, Mapping, Abstraction and Compliance.

Organization brings the requirements associated with the organization of UI building, and its features are Beautification, Layout and Separability. Beautification regards to embellished and refinement of UI appearance support throughout its life cycle. It involves details such as the type and font size, background color, icon images, placement of components, among other aesthetic details. Layout specifies elements grouped and organized in the layout and its arrangement. Separability regards to separation of concerns in the approach, although application domain concepts involved. This separation promotes maintenance of business concepts and UI in isolation.

Requirements related to Decomposition classify the subdivision of UI components: Intention, Decomposition and Standardization. There are UI for different purposes, and the Intention feature should bring together a set of UI application intentions that can assist in building productivity UI. One recurrent user interaction intention being used in WIS is the Web Portal. The concept of dominant element, the one that makes the grouping of simple elements to construct more complex elements, can solve recurrent problems in UI, and is abstracted in Decomposition feature.Isolated elements do not have a purpose for use. However, together they become reusable in similar contexts, fostering productivity in UI building and promoting the Standardization of the appearance and behavior of the UI.

Mapping requirements are related to the need of transformation of models for automated UI construction. Clarity, Flexibility and Direction are features of Mapping. Clarity relates to the efficient use of transformation rules between the abstract and concrete metamodel that must be clear and precisely specified. In addition, some components can be more abstracts than one mapping for concrete elements, and Flexibility must be allow to configure this mapping. The Direction is observed when the mappings are done in a proper way, from a more abstract to a more concrete representation.

Abstraction presents features that support an independent technology approach. Generality is related to maintaining a high level abstraction and generality in the models used for specifying UI, making independent technology specification. Structure is related to providing a set of precisely defined terms on IHC, covering concepts of presentation and behavior of UI.

The Compliance requirements are related to building UI according to the application need. For this, Contextualization and Correlability characteristics are expected. Correlability involves the association between features and business con-

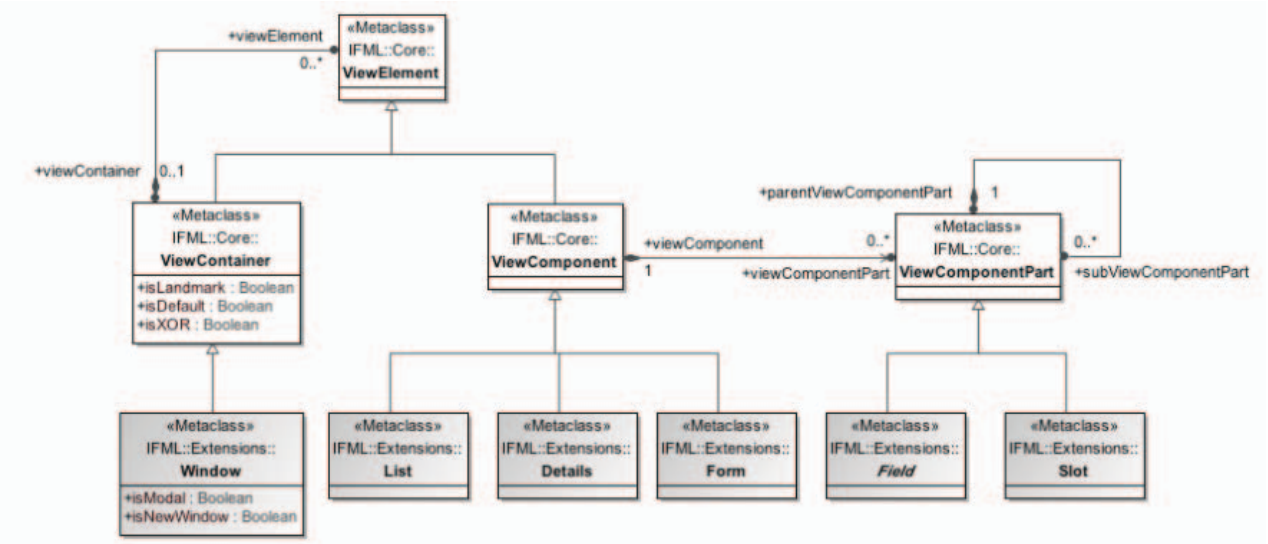

Figure 10. IFML View Elements - Excerpt of IFML Metamodel.

cepts with the interface specification, allowing to understand which business concepts and features must be present in the UI [25]. The Contextualization is concerned with the appropriate presentation of the UI according to the business domain information types and relationships.

Thus, our approach complies with IFML standard and is suitable to address Web Systems modeling and automatic UI generation issues. Furthermore, in a comparative analysis, existing alternative approaches do not support all requirements presented in Table I [9]. Other recent approaches, such as CIAT [26] and Malai [27], do not use a dominant element that standardizes UI which have the same type of information and the same purpose of interaction. In spite of the advantages of use UI Stereotype, we do not control how UI Stereotype is modeled. Moreover, modeling a UI Stereotype which is rarely used can be onerous to the UI development process.

V. Conclusions

This paper describes a UI Stereotype to improve the construction of Web Portals using a model-based approach to build the UI for Web Systems domain. The proposed Web Portal UI Stereotype can generate different Web portals with the same intention and from a common set of metamodels, fostering reuse in User Interaction (UI) Engineering. This approach allows programmers only configure each HCI element in a UI Stereotype instance. The stereotyped behavior should be aggregated to the tool (as WebRatio) so programmers should need to implement only specific functions of an IS. Thus, the use of a UI Stereotype that describe Web Portals appearance and behavior promotes the standardization of this kind of application.

As future directions, we will migrate the concept of UI Stereotype for IFML standard, taking advantage of the underlying tool support, fostering reuse through the abstraction of a UI Stereotype over the IFML model. Also, we will apply the proposed UI Stereotype in the development process of Web Portals to observe how to improve this approach and how programmers react to the use of this method.

We also intend to investigate the description of UI Stereotypes as Design Patterns [28], [29] for UI Engineering, since the collection of UI Stereotypes a(such as Web Portal, Survey, and CRUD) can be seen as recurrent interaction UI challenges for Web applications. The solution for these challenges can be structured and documented as a tuple composed by a Recurrent Problem (UI Modelling and Construction), a Context (Web application UI development), and a Repeatable Solution (automatic UI synthesis via Model-Driven Development based on UI Stereotypes).

References

[1] J. L. de Oliveira, L. Loja, S. Costa, and V. Graciano Neto, “A component for business processes management in information systems (in portuguese),” in *SBSI 2011*, Salvador, Brazil, May 2011.

[2] C.-M. Lee, “User Interface Prototype Generation Technique Supporting Usage-Centered Design,” *Intern. Journ. of Software Engg and Knowledge Engg*, vol. 19, no. 1, pp. 23 – 46, 2009, world Scientific Publish. Co.

[3] J. Vanderdonckt, “A MDA-compliant environment for developing user interfaces of information systems,” in *CAiSE 2005*, Porto, Portugal, June 13-17 2005, pp. 16 – 31, springer.

[4] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, "A unifying reference framework for multi-target user interfaces," *Interacting with Computers*, vol. 15, no. 3, pp. 289 – 308, 2003, computer-Aided Design of User Interface.

[5] S. Ahmed and G. Ashraf, "Model-Based User Interface Engineering with Design Patterns," *Journal of Systems and Software*, vol. 80, pp. 1408 – 1422, August 2007, elsevier Science Inc.

[6] T. Memmel and H. Reiterer, "Model-Based and Prototyping-Driven User Interface Specification to Support Collaboration and Creativity," *Intern. Journal of Universal Computer Science - Special Issue on New Trends in HCI*, vol. 14, no. 19, pp. 3217–3235, March 2009.

[7] J. Falb, R. Popp, T. Rock, H. Jelinek, E. Arnautovic, and H. Kaindl, "Fully-automatic generation of user interfaces for multiple devices from a high-level model based on communicative acts," in *Hawaii ICSS 2007*. Washington, DC, USA: IEEE Comp. Society, 2007, pp. 26 – 35.

[8] T. J. Bittar, R. P. d. M. Fortes, L. L. Lobato, and W. M. Watanabe, "Web Communication and Interaction Modeling using Model-Driven Development," in *DC 2009*, Bloomington, Indiana, USA, October 5-7 2009, pp. 193 – 198.

[9] S. L. da Costa, V. V. Graciano Neto, B. dos Reis Calçado, and J. L. de Oliveira, "User interface stereotypes: A model-based approach for information systems user interfaces," in *SBSI 2014 (to appear)*, Londrina-PR, May 2014.

[10] A. Tatnall, *Web Portals: The New Gateways to Internet Information and Services*, H. Arthur Tatnall, Ed. Idea Group Publishing, March 22 2005.

[11] V. V. Graciano Neto and J. L. de Oliveira, "Evolution of an application framework architecture for information systems with model-driven development (in portuguese)," in *SBSI 2013*, 2013, pp. 1–12.

[12] W. A. D. Santos, B. runo B. . F. . Leonor, and S. tephan Stephany, "A Knowledge-Based and Model-Driven Requirements Engineering Approach to Conceptual Satellite design," in *ER 2009, Proc.*, vol. 5829, 2009, pp. 487–500.

[13] M. B. Nakićenović, "An agile driven architecture modernization to a model-driven development solution - an industrial experience report," *Intern. Journ. on Advances in Softw.*, vol. 5, no. 3, pp. 308 – 322, 2012.

[14] A. van Deursen, E. Visser, and J. Warmer, "Model-driven software evolution: A research agenda," in *MoDSE 2007*, D. Tamzalit, Ed., Amsterdam, The Netherlands, March 2007, pp. 41–49.

[15] J. Canovas and J. Molina, "An architecture-driven modernization tool for calculating metrics," *IEEE Software*, vol. 27, no. 4, pp. 37–43, 2010.

[16] "Object Management Group (OMG), Model-driven Architecture (MDA)," http://www.omg.org/mda/, 2003.

[17] J. Vanderdonckt and A. R. Puerta, "Introduction to Computer-Aided Design of User Interfaces," in *CADUI 1999*, Louvain-la-Neuve, Belgium, Oct 21-23 1999, pp. 1–6.

[18] A. Puerta and J. Eisenstein, "Towards a General Computational Framework for Model-Based Interface Development Systems," in *Proc. of the Intern. Conference on Intelligent User Interfaces*. ACM Press, 1999, pp. 171 – 178.

[19] Q. Limbourg and J. Vanderdonckt, *The Handbook of Task Analysis for Human-Computer Interaction*. Mahwah: CRC Press, 2003, ch. 6, lawrence Erlbaum Ass.

[20] S. L. da Costa, V. V. Graciano Neto, L. F. B. Loja, and J. L. de Oliveira, "A Metamodel for Automatic Generation of Enterprise Information Systems," in *BW-MDD 2010*, vol. 8. Salvador, Brazil: UFBA, Setembro 2010, pp. 45 – 52.

[21] M. White, "Enterprise information portals," *The Electronic Library*, vol. 18, no. 5, pp. 354–362, 2000.

[22] M. van Welie, "Interaction Design Pattern Library," available in: http://www.welie.com/patterns/index.php. Last access: February 1 2011.

[23] OMG, *Interaction Flow Modeling Language (IFML)*, Object Management Group (OMG) Std. ptc/2013-03-08, March 2013. [Online]. Available: http://www.omg.org/spec/IFML/1.0

[24] S. L. da Costa, "A model-based approach for automatic construction of user interfaces for information systems (in portuguese)," Master's Thesis, UFG, 2011.

[25] J. Vanderdonckt, "Model-Driven Engineering of User Interfaces: Promises, Successes, Failures, and Challenges," in *ROCHI 2008*. Bucarest, Romania: Matrix-ROM, September 2008, pp. 1–10.

[26] A. I. Molina, W. J. Giraldo, J. Gallardo, M. A. Redondo, M. Ortega, and G. GarcÃa, "Ciat-gui: A MDE-compliant environment for developing graphical user interfaces of information systems," *Advances in Engineering Software*, vol. 52, no. 0, pp. 10 – 29, 2012.

[27] A. Blouin and O. Beaudoux, "Improving modularity and usability of interactive systems with Malai," in *SEICS 2010*. Berlin, Germany: ACM, June 19-23 2010, pp. 115 – 124.

[28] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns*. Boston, MA, USA: Addison-Wesley Longman Pub. Co., Inc., 1995.

[29] T. Neil, "Rich Internet Application Screen Design," *UX Magazine*, no. 483, February 11 2010.