

# Integrate Model-driven UI with Business Transformations: Shifting Focus of Model-driven UI

Noi Sukaviriya, Santhosh Kumaran, Prabir Nandi, Terry Heath  
IBM T.J. Watson Research Center  
19 Skyline Drive  
Hawthorne, NY 10532

{noi, sbk, prabir, theath}@us.ibm.com

## ABSTRACT

This paper is not a technical paper that presents a new model-based UI technique. It is about a project that revisits model-driven UI design techniques in the context of business process modeling and business transformation. We have embarked on model-driven business transformation research and are beginning to tackle the model-driven UI design portion of it. For this aspect, the approach we are taking is drawn based on our extensive experience as user interface designers, as well as our experience within the past few years with business transformation engagements with real customers. The paper describes 4 different perspectives in our approach that shift the focus of model-based UI tools from just producing the end results to better supporting UI designers with a faster, model-driven UI design process.

## Categories and Subject Descriptors

H1.2 [Models and Principles]: User/Machine Systems – *Software psychology*. H5.2 [Information Interfaces and Presentations]: User Interfaces – *Graphical user interfaces, methodology, user-centered design*. D.2.2 [Software Engineering]: Design Tools and Techniques – *User interfaces*.

## General Terms

Design, Human Factors.

## Keywords

Model-based UI; Business Process Integration; UI Design Process

## 1. INTRODUCTION

Model-driven architecture [1, 2] addresses the need for the industry to add agility to technology solutions. By componentizing IT solutions into reusable components, IT solutions can supposedly be reconfigured to meet the changing demand of the world. On-going research on model-driven business transformations [3, 4] embraces model-driven architecture; it attempts to use business processes to automate IT

solutions. It proposes to capitalize on formal business process modeling, turning formal specifications of process inputs and outputs into transition-based business objects. These objects are used to automate code generation and service mapping deemed necessary to retrieve and save information for such processes. By tying business solutions close to business process descriptions, business process models become a kind of high-level programming tool.

Model-driven UI has always been about explicitly representing components that makes up UIs from user tasks to interaction techniques. According to Myers, Hudson & Pausch [5], the motivation behind model-based UI design tools was to enable programmers who, without user interface design experience, could implement only the functionality and let user interfaces be automatically generated. This attempt to automate from functionality specification is parallel to model-driven business transformation research. The UI models and the automation, one could say, entitle model-based UI as “the” approach to add agility to building UI solutions. But we still have more work to do to turn this concept into reality.

In the industry, UI design and development still lacks models and automation. Based on our experience for the past few years, the traditional UI design and development process typically takes 65-80% of the overall development time in a model-driven business transformation project. While model-based UI has the potentials to provide an impact in large IT projects, its approach in past research focusing UI design and development as a stand-alone process (ranging from task modeling to data modeling all the way to UI implementation and application action binding) is not quite appropriate. The UI design and development process has to be an integral part of and contribute appropriately at various places in the development cycle. This paper addresses the approaches we are taking to establish the model-driven UI process in this direction.

## 2. APPROACHES

As mentioned earlier, we have been involved with and have delivered business transformation projects. These projects start off with modeling customer’s business processes, both the processes as they exist today and the new processes which we help design to address customers’ problems. The projects then proceed to designing, building, and delivering IT solutions according to the new business design. This process also includes designing and implementing user interfaces. Our basis for competition in the marketplace is speed and quality of services of this kind of business engagements. While model-driven business transformations [3, 4] tools are being developed to speed up the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MDDAUI’05, October 2, 2005, Montego Bay, Jamaica.  
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

process, not much is happening to support the UI design side. The UI design process has become relatively time-consuming, especially when other parts of the solutions can be automated through models. Our goal is to reduce the time and increase the agility to the UI design process. Our approach is to underlay the traditional UI design process with UI models and interweave the model-driven UI process with the business transformation practice. Notice the significance of the term “underlay” we use as it implies that we do not intend to change the UI design practice; we’d rather enhance it.

In this section, we will discuss in more detail about our approaches as following. First we explain why we believe that model-based UI still needs to address traditional GUIs. Then we discuss how model-driven UI design and business process modeling have much in common, hence should leveraged off each other. Then we follow on to address how and where some of the known model-based UI techniques would be applicable. Finally we discuss the importance of understanding and focusing on the users of the model-based UI design tools and close the paper with a hint on how model-based UI would be a probable approach for reusable UI patterns.

### 2.1 Traditional GUIs is still in High Demand

While earlier model-based UI design tools focused on automatic generation of user interface [6, 7], we have seen the shift in later model-based UI research to address limited automatic generation, emphasizing user interface transformations among small devices [8]. This shift grew out of the popularity in PDAs and cellular phones hence increased the need to quickly create similar UI on several platforms. These research prototypes centered on generating multiple small platform interfaces off of common UI models [9, 10, 11, 12]. Though this is mostly GUIs, the smaller interfaces make automatic generation more manageable, we believe. On one hand, it is essential that this MDDAUI workshop pushes model-based UI agenda further into multi-modal and more

complex interactions such as games and multimedia. However, we believe the work on traditional GUIs is not done yet. As a matter of fact, model-based UI has not started as a common practice in the industry as a whole.

Most of the IT solutions our company delivers are still desktop applications, more likely web-based applications in recent years. The good news is that the design of web-based UIs, particularly in business applications, has somewhat converged making the scope of interface designs rather well-defined. Business applications deal with lots of data entering and display, giving model-based UI an opportunity to do what it can do best - automation. Please see Section 2.3 for more detail on this subject matter.

Interestingly, there is a drive towards role-specific UIs from two different sources. First, the rise of portal-based style of design is amplifying the requirements to build user-role specific interfaces. Secondly, when modeling business process, user roles are specifically defined for each business task. This makes it easy to understand and separate out user roles – who are conducting which tasks in the process – hence making it rather easy to define the scope of role-specific user interfaces. In a project in which multiple user roles must be specially designed for, it is very tedious and time-consuming design process – a pain point for user interface designers. Managing the design is more complex. While the principal of the user interface design has not changed over the years; the design process has become richer with more depth. Designers often have to juggle to maintain consistency among user roles while they can, at the same time attempt to pan off each design to appropriately support the unique set of functionality required by each user role. Tweaking the design in this case is extremely cumbersome. This is one area where having a model of user roles and user tasks with some automation can help provide a semantic structure to support the design process.

**Table 1. Diagram Showing Parallel Activities between Business Analysts and UI Designers**

<b>Business Analysts</b>	<b>UI Designers</b>
Conduct business analysis and model business process & tasks	Conduct user research and model user tasks
	Document user roles quite extensively including background, motivations, and role descriptions
Have interests in task inputs & outputs as business transactions	Have interests in what information people need to perform tasks and what they need to perform a task and to complete a goal
Identify and model business artifacts	Identify business artifacts and personal artifacts
	Pay attention to overall user experience as well as task interaction experience
Demonstrable by use case scenarios	Demonstrable through low-fidelity mock-ups and storyboard, and high-fidelity mock-ups
Outputs validated by domain experts and business stakeholders	Outputs validated by domain experts, and end-users
End results must meet business objectives – business objectives vary from solution to solution	End results must meet business objectives and enable users to perform tasks with better effectiveness, efficiency and experience

## 2.2 Work with Business Process Modeling

Business transformation projects are becoming common in the IT industry. Explicitly capturing business process models has become a common practice to capture and analyze parameters that drive strategic analyses behind business transformations. As mentioned earlier, model-driven business transformation research [3, 4] uses business processes to help automate IT solution composition. Our experience with business transformation projects showed that business process modeling has many common components with the user interface design process. Table 1 compares business process modeling and the UI design process – the elements in the same row are those components that are very similar. As we can see, these common elements in fact complement each other. As a matter of fact, our study of UI designers showed that the task model UI designers often draw before starting the design look very similar to business processes and vice versa. Business processes can indeed serve as an almost perfect starting point for the UI design process.

While business processes provide a framework for how business tasks are sequenced and who perform them, they are also considered part of the business requirements – this is how business strategist and analysts believe the organization should be structured. Very importantly, UI designs are required to follow these process models. Significantly, the UI design process comes in to enhance the process model with human aspects necessary to create richer user experience design. It can also enhance the data aspect of the business process to cover all necessary data items deemed necessary for the end-users to carry out business tasks. Collaboration from the business and user-centered perspectives in general enhances the quality of the solution being designed. Being able to understand the relationship among UI designs and business processes is crucial in keeping IT design align with business requirements.

Some model-based UI research systems use the reverse-engineering approach to extract semantic model of existing UIs then map them to new designs [13, 14, 15]. Business process models can be very similar to the semantic model of these systems. It consists of user tasks, task inputs and outputs, input and output attributes and the role players. Techniques for mapping from common semantic/task models to new UI design can be useful in bootstrapping the UI design from business process models.

One caution though – it is not always the case that business process models will end up being defined at the same level as the semantic model in model-based UIs. Our experience showed that in some cases business processes are stated at a much higher level and may need to be decomposed to be comparable to those of user tasks in model-driven UI.

## 2.3 Focus on Current Lack of Support for UI Designers

An earlier model-based UI design tool was very ambitious attempting to intelligently map UI components from (enhanced) data type definitions and automatically laying out the entire screen [6]. While the attempt was noble, it is not likely to be practical today. There has been a surge in user interface design talents within the past decade. Designing screen layouts, while

can be systematically complex, is an easily achievable task by human designers with proper visual design training. There is no point in creating such a complex system that doesn't do a good job for skills that are not lacking. As a matter of fact from the author's 10-year experience as a UI designer, this is a rather fun part of the work that should not be taken away from the designers. So much creativity can be used in this part of the design process making automatic generation of large UI layouts not a probable goal.

Tools commonly used to deliver UI design today such as Adobe Illustrator, Photoshop, Macromedia Director, Flash, etc. are visual layout tools and focus their support for designing high definition visual UI designs. Our current study of the UI design process shows that designers tend to go to these tools when their hand sketches start to get messy or they start to lose sense of the screen boundary. Since these tools are visual design tools, they tend to distract the designers with visual details but yet their precise layout support is valuable. The designers in the study commonly agreed that the level of visual detail was not necessary at this stage of the design. They also voiced in unison that there was lack of support for the UI design in the early phase. The UI mocking up process in general is still rather cumbersome today.

This is where we believe model-based UI techniques can come in handy. Automatically mapping data components to UI components at times can be very useful if designers can use it piecemeal as part of the design process. Such a system was built many years back [7] that would be a straightforward integration into a design tool. Design transformations such as those for small devices [8, 9, 10, 11, 12] can be used to transform a small portion of a design at a time. Regeneration of UI from common UI models such as those in [9, 10, 11, 12] can be used to help designers cranking up design alternatives. However, these techniques have to be an integral part of the design process that is driven by the designers. Model-based UI research tends to address the design process as a black box. That is, some information is given to model-based systems on one end; the automatic generations then produce the UIs on the other end and the process is done. This monolithic approach needs to be teased apart and brought in to support the design process at various points when they are really needed. It should be up to the designers to call on an appropriate technique when they need it, and for them to call out when a design is "done."

## 2.4 Differentiate and Support the Role Players in the Tools

Model-based UI design and development spans from integrating high-level information such as user tasks, to recording UI design choices for interaction techniques, screen layouts, to generating code and system elements for supporting the design at run-time. That is quite a wide coverage. We often look at the model-based UI systems as single systems. They need certain inputs, we give the right information to them, and they produce UIs. Model-based UI researchers are often designers, software engineers, and sometimes programmers too. The interfaces to these model-based UI research prototypes reflect this improbable continuum of skills in the real world.

We are taking the approach that there are different role players in making IT solutions, from business analysts, to UCD practitioners, UI designers, solution architects, application developers, deployment and testers, etc. This approach is not a revolutionary thought, but it is a thought that is often neglected. Exercising this thought throughout the process of designing model-based UI tools are very critical to general acceptance of the tools by practitioners. Currently, we are very cautious of every interaction step we call upon in our model-based tool design. We attempt to be clear who the role players for these steps are. This fundamental grounding will allow us to choreograph the UI design process appropriately to our user roles. For example, we only expose design steps for the UI designers and back end steps to solution architects. We can hide unnecessary technical information from UI designers and at the same time expose the needed information about the UI design to solution architects who will finish up the design and implementation for the overall IT solution.

## 2.5 Model-based UI as an Infrastructure for Reusable Components

Finally, we are seeing a shift in the industry towards reusing assets to increase competition through competency. Software engineering is a crucial part in driving this effort. Similarly to how model-driven UI can be thought of as the approach to enable agility in the UI design process, UI models are also the appropriate step towards capturing reusable UI components and patterns. It is essential that the models that we design today will support reusable UI design patterns in the future.

## 3. SUMMARY

Our motivation to make the model-driven UI design process work stems from the need to reduce the time and the need to increase the speed and quality of the IT design services in business transformation engagements. Model-based UI research has been around for almost two decades; its research potentials have been demonstrated through numerous publications. We have an opportunity to bring model-driven UI into model-driven environments for creating IT solutions. However, more work needs to be done.

We draw upon our experience as user interface designers to revisit model-based UI techniques as well as to consider future benefits model-based UI has to offer. As we are designing a model-driven UI design process and integrating it with the business transformation practice, we draw out essential considerations that we believe will make this effort work. These considerations on 1) dealing with complexity and breadth of designing graphical UIs today, 2) integration with business process modeling, 3) picking the right spots to support in the design process, and 4) clarity of users of the model-driven UI tools, are drawn from our past UI design experience, our recent business transformation experience, as well as our initial study to understand the design process.

## REFERENCES

[1] Miller, J., Mukerji, J. (ed.): MDA Guide Version 1.0.1. Object Management Group (2003) <http://www.omg.org/docs/omg/03-06-01.pdf>

[2] Kleppe, A., Warmer, J., Bast, W. MDA Explained: The Model Driven Architecture – Practice and Promise. Addison-Wesley, Reading, MA (2003)

[3] Kumaran, S. Model-Driven Enterprise. Proceedings of the Global EAI (Enterprise Architecture Integration) Summit. (2004) 166-180

[4] Kumaran, S., Nandi, P.; Adaptive Business Objects: A New Component Model for Business Integration. Proceedings of ICEIS 2005: 7<sup>th</sup> International Conference on Enterprise Information Systems (2005)

[5] Myers, B.; Hudson, S.E., Pausch, R. Past, Present, and Future of User Interface Software Tools. TOCHI : ACM Transactions on Computer-Human Interaction. Volume 7 (2000) 3-28.

[6] Kim, W.C., Foley, J.D.; Providing High-level Control and Expert Assistance in the User Interface Presentation Design. Proceedings of CHI 93: ACM Conference on Human Factors in Computing Systems (1993) 430-437

[7] de Baar, D., Foley, J.D., Mullet, K.E.; Coupling Application Design and User Interface Design. Proceedings of CHI 92: ACM Conference on Human Factors in Computing Systems (1992) 259-266

[8] Nichols, J., Myers, B.A., Litwack, K.; Improving Automatic Generation with Smart Templates. Proceedings of ACM Conference on Intelligent User Interfaces (2004) 286-288

[9] Banavar, G., Bergman, L.D., Gaeremynck, Y., Soroker, D., Sussman, J.: Tooling and System Support for Multi-device Applications. Journal of Systems and Software, Vol. 69 Issue 3. (2004) 227-242

[10] Mori, G., Paterno, F., Santoro, C.; Tool Support for Designing Nomadic Applications. Proceedings of ACM Conference of Intelligent User Interfaces (2003) 141-148

[11] Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., Pignol, M.; Generating Remote Control Interfaces for Complex Appliances. Proceedings of ACM Conference on user Interface Software and Technology (2002) 161-170

[12] Eisenstein, J., Vanderdonck, J., Puerta, A.; Applying Model-based Techniques to the Development of UIs for Mobile Computers. Proceedings of ACM Conference of Intelligent User Interfaces (2001) 69-76

[13] Bouillon, L., Vanderdonck, J., Chow, K.C.; Flexible Re-engineering of Web Sties. Proceedings of ACM Conference on Intelligent Interfaces (2004) 132-139

[14] Bandelloni, R., Paterno, F.; Flexible Interface Migration. Proceedings of ACM Conference of Intelligent User Interfaces (2004) 148-155

[15] Gaeremynck, Y., Bergman, L.D., Lau, T.; MORE for Less: Model Recovery from Visual Interfaces for Multi-device Application Design. Proceedings of ACM Conference of Intelligent User Interfaces (2003) 69-76

[16] Puerta, A., Cheng, E., Ou, T., Min, J.; MOBILE : User-centered Interface Building. Proceedings of CHI 99: ACM Conference on Human Factors in Computing Systems (1999) 426-433

[17] Paterno, F., Mancini, C. Model-based Design of Interactive Applications. ACM Intelligence Magazine Winter (2000) 26-37.

