

Model-driven business process integration and management: A case study with the Bank SinoPac regional service platform

J. Zhu
Z. Tian
T. Li
W. Sun
S. Ye
W. Ding
C. C. Wang
G. Wu
L. Weng
S. Huang
B. Liu
D. Chou

Business process integration and management (BPIM) is a critical element in enterprise business transformation. Small and medium-sized businesses have their own requirements for BPIM solutions: The engagement methodology should be fast and efficient; a reusable and robust framework is required to reduce cost; and the whole platform should be lightweight so that one can easily revise, develop, and execute solutions. We believe that model-driven technologies are the key to solving all of the challenges mentioned above. Model Blue, a set of model-driven business integration and management methods, frameworks, supporting tools, and a runtime environment, was developed by the IBM China Research Laboratory (CRL) in Beijing to study the efficacy of model-driven BPIM. To verify the technology and methodology, Model Blue was deployed with Bank SinoPac, a mid-sized bank headquartered in Taiwan. A lightweight BPIM solution platform was delivered for Bank SinoPac to design, develop, and deploy its business logic and processes. During the eight-month life span of the project, IBM teams developed four major solutions for Bank SinoPac, which also developed one solution independently. In spite of the remote working environment and the outbreak of the Severe Acute Respiratory Syndrome illness, the project was completed successfully on schedule and within budget, with up to 30% efficiency improvement compared with similar projects. Bank SinoPac was satisfied with the technology and methodology, and awarded IBM other projects. In this paper, we illustrate how each key business process integration and solution development phase was carried out and guided by business process modeling, together with major experiences gained. The following technical aspects are discussed in detail: a two-dimensional business process modeling view to integrate flow modeling and data modeling; a lightweight processing logic automation environment with tooling support; and the end-to-end BPIM methodology, with models and documents successfully integrated as part of (or replacement for) the deliverables defined in the existing servicing methodologies and software engineering approaches.

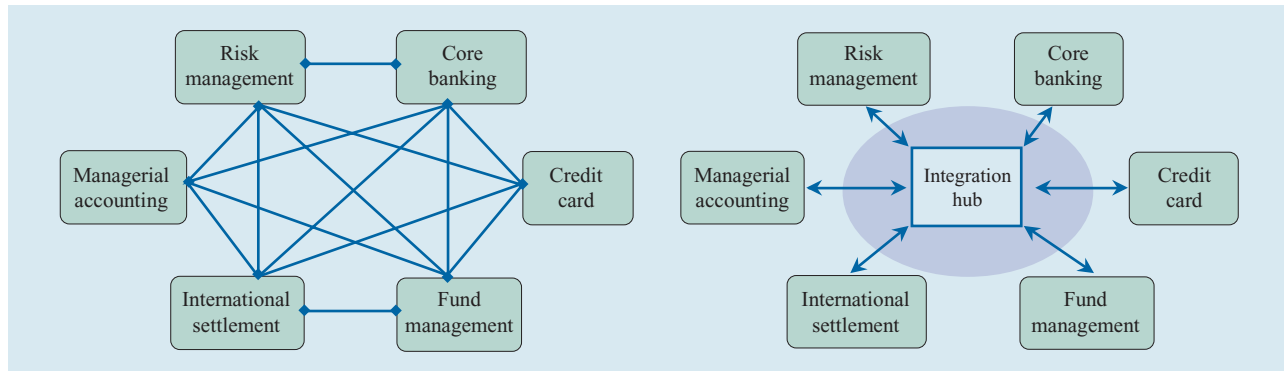


Figure 1

Traditional Enterprise Application Integration (EAI) compared with an integration hub.

1. Model-driven business integration

Business Process Integration and Management (BPIM) is important for transforming today's business enterprises. Enterprises require end-to-end solutions in order to effectively link internal and external business applications, systems, and staff so that they can respond with flexibility and speed to changing business conditions. The technologies for business integration have evolved over the last twenty years.

Most early integration solutions were focused on connecting systems. Vendors have provided various Enterprise Application Integration (EAI) adapters (e.g., iWay [1]) to help create linkages among different applications (e.g., SAP*, SQL Server, and DB2*). Such solutions tend to establish relationships among systems in an *ad hoc* point-to-point manner. This is illustrated in **Figure 1** with applications in a typical bank as an example. Usually it leads to a complex interaction network among systems and business units and results in problems in areas such as performance, maintenance, lack of scalability and extensibility, and stability.

The concept of an integration hub (also shown in Figure 1) has been widely adopted by today's leading business application integration solutions, such as IBM WebSphere* Business Integration (WBI) [2], Microsoft BizTalk Server** [3], and BEA Weblogic** Integrator [4]. These solutions connect different systems through a centralized integration engine or hub, where the integration logic resides and executes. As a result, the creation, maintenance, and changing of integration logic can be managed in a more flexible and efficient way.

The most challenging part that remains unsolved, however, is how to plan, build, maintain, and utilize the integration hub for real business cases. Many solutions are still focused on the information technology (IT) aspects of

system integration, which, it is asserted, is becoming the commodity portion of enterprise integration solutions [5]. Historically, there have been attempts and practices to improve enterprise businesses through IT advances, but according to [6], these have not proven to be very successful. The differentiation of today's business integration solutions is elevated to the design and analysis of high-level business strategy and processes, and the way in which IT systems can be refined or created to support them. From this perspective, an integration hub is insufficient. In fact, an integration platform is required to guide the end-to-end solution lifecycle. Such a solution lifecycle typically includes phases such as requirement collection, macro and micro design, implementation, test, and deployment.

Model-driven business integration (MDBI) is an emerging approach for building such an integration platform. The distinctive feature of this approach is that models are employed as the key elements to drive the end-to-end integration solutions from business requirements down to IT implementation. For each major phase of a solution lifecycle, guidelines are given to specify the input models, output models, and model operations (e.g., creation, transformation, and modification). According to the layered modeling concept in Model-Driven Architecture (MDA) [7], the separation of business logic from IT technology can be achieved by using separate models at the computation-independent model (CIM) layer, the platform-independent model (PIM) layer, and the platform-specific model (PSM) layer. Model transformations among layers will help to fill the so-called business-IT gap.

As reasonable as it sounds, most work in MDBI is still in the research stage and has not been verified by studies through real business cases. The fundamental challenge for this approach is not each single modeling method and

tool itself, but how to build a methodology with tools to facilitate its execution. The success of this methodology can be attributed to the careful crafting of the tools to the different roles of the methodology, and the integration of the tools via the transformation of artifacts. This is actually something that cannot be fully verified without field practices. Therefore, it has been a goal of this research to couple and apply the model-driven approach to real business practices and gain insight on how practitioners, rather than just researchers and technology developers, will use these approaches.

A set of model-driven business integration and management methods, frameworks, supporting tools, and runtime environment were created in 2002 by the IBM China Research Laboratory (CRL) to conduct this study. For convenience, this set of technologies and methodologies is called Model Blue in this paper. Model Blue forms a platform with an end-to-end methodology to help enterprises meet high-level business goals through effectively connecting internal and external business applications and systems. As illustrated in **Figure 2**, Model Blue includes technologies for modeling business processes in two views—the Model Blue Business View and the Model Blue IT View. The Model Blue Business View provides an easy and intuitive method of modeling integrated business processes. The Model Blue IT View offers a lightweight processing logic modeling method with tooling and runtime support. The IT View model can also be transformed to other formats (i.e., WBI Collaboration, Flow Definition Language (FDL) [8], and Business Process Execution Language (BPEL) [9]) and integrated with related commercial tools and runtimes to form the Model Blue solution suite. The Business View and IT View are seamlessly connected with model transformers under the guidance of the Model Blue solution methodology. In the remainder of this paper, the term *Model Blue* is used to represent the whole integrated solution suite and methodology; the terms *Business View* and *IT View*, if not specifically noted, are used to represent the Model Blue Business View and the Model Blue IT View, respectively.

Bank SinoPac, an Asia Pacific regional bank headquartered in Taiwan and known for its business and technical innovation, has the need to support its business strategy with a business integration platform, on which it can design, develop, and deploy business processes for its cross-Pacific (including mainland China, Hong Kong, Taiwan, and the U.S.) financial products. Bank SinoPac has its specific requirements for BPIM: The project fulfillment methodology should be fast and efficient; a reusable and robust framework is required to reduce cost; the whole platform should be intuitive and easy to use so that Bank SinoPac can revise, design, and execute solutions independently. These requirements are still a

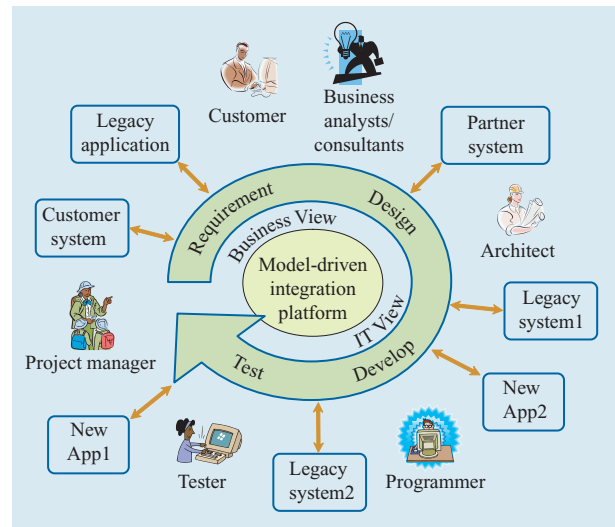


Figure 2

Model-driven integration platform.

challenge for today's business integration solutions. The IBM China Research Laboratory, in collaboration with IBM Business Consulting Services (BCS), took this opportunity to apply the Model Blue technologies to build the integration platform for Bank SinoPac. One mission of the engagement was to verify the anticipated benefits of a model-driven approach in solving real BPIM problems.

One distinctive advantage of a model-driven approach is the representation of business and IT issues with formal models, which make it possible to apply formal analytical methods to derive valuable insight, for example, regarding process correctness and effectiveness. Basic business process model analytical capabilities have been developed and shipped with Model Blue (i.e., to facilitate the detection of interprocess conflicts).

The rest of the paper is organized as follows: Part 2 gives an overview of Model Blue, including its solution suite, Business View, and IT View. In Part 3, the case study is presented in detail, including the profile and requirements of Bank SinoPac, challenges for BPIM, and the execution lifecycle, highlighting major experiences gained. Best practices and lessons learned are presented in Part 4. Part 5 concludes the paper with user feedback and technical highlights. Part 6 briefly indicates some future research challenges.

2. Overview of Model Blue

In this paper, we use the term Model Blue to represent a set of model-driven business integration and management

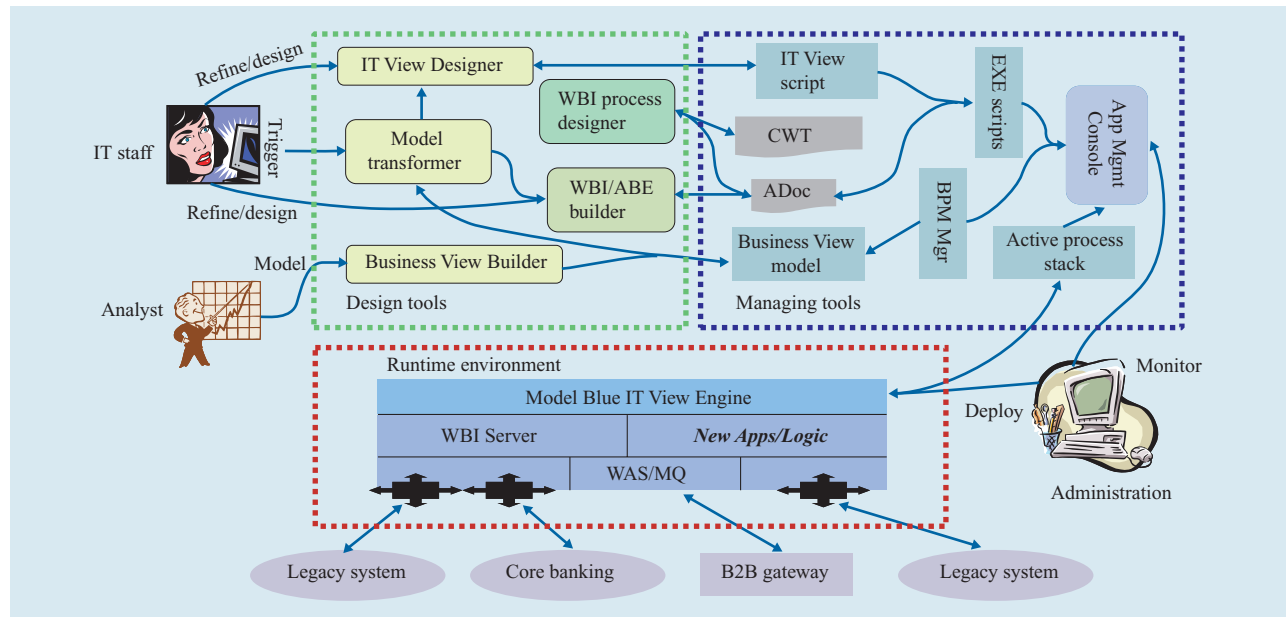


Figure 3

Model Blue suite.

methods, frameworks, supporting tools, and runtime environment. Model Blue Business View provides the modeling methods and tools to help consultants and analysts collect, understand, formulate, discuss, and analyze enterprise business processes. The Model Blue IT View offers a lightweight processing logic automation environment with tooling to help analysts, architects, and designers communicate and implement the business requirements in IT systems. With the help of a model transformation engine, the Business View can be automatically transformed into initial IT View models to help bridge the business–IT gap. The IT View model can be executed on the Model Blue IT View Engine or transformed to industry product format and integrated with related commercial tools and runtimes to form the Model Blue solution suite. All of the above parts constitute the technical base for Model Blue solution methodology, which provides guidelines to seamlessly integrate them across an end-to-end solution lifecycle. From the model transformation point of view, Model Blue starts with an integrated Business View combining flow and data requirements from business consultants, and then generates two sorts of models for IT architects—one focused on flow activities (Model Blue IT View), and one focused on data elements (i.e., XML schema). Different tools are then used to further develop these models across the remaining solution phases such as development, testing, and deployment.

Model Blue suite

Figure 3 illustrates the whole suite of a Model Blue solution in three parts: design tools, managing tools, and runtime environment.

Design tools

The two major tools within the solution, Business View Builder (targeted for business consultants) and IT View Designer (targeted for IT staff) are both built on top of the Eclipse [10] platform and are linked together by the model transformer. The two tools can also work together with other modeling and developing tools, such as the WBI process designer and the WBI/ABE (Adaptive Business Entity, a state-driven broker of multiple process flows) builder.

Managing tools

The business process models created by the design tools, including the Business View model, IT View model, and WBI Collaboration model, are managed by the Model Blue Asset Manager [11]. An application management console is also provided to deploy, manage, and monitor the executable flow models.

Runtime environment

The Model Blue IT View Engine is mainly responsible for the execution of process flows by routing through various business applications and/or transactions. Existing

application connectors in WBI are leveraged here to connect to back-end legacy systems. They are linked together with the IBM WebSphere Application Server (WAS) and WebSphere MQ Series [8] middleware to form the runtime environment. The IT View Engine can also access legacy systems directly with specifically designed actions to support new applications and logic.

Business View

Model Blue Business View is targeted for business analysts and consultants and is designed specifically to meet their needs. A global view of the business processes is important for analysts and consultants to understand the requirements, yet it is often very hard to gather all related parties together to create such a global view. One unique capability of Business View is to provide localized modeling capability so that business analysts or consultants can use it to interview individuals and outline their understanding of the business process, by asking questions such as “What are the steps for doing this job?”, “How do you make decisions to route through the steps?”, and “How do you collaborate with others in these steps?”. Each interviewee contributes partial business process requirements from an individual perspective. Process and data elements can be modeled in isolation and then assembled in a hybrid picture. Business View then provides methods and tools to combine all of these subparts into a global model, which can then be used for analysis or as guidance for future implementation. Since analysts and consultants also wish to be able to show different levels of details under different usage contexts, Business View must have flexible support in model navigation, expanding, and collapsing.

Basically speaking, formalized representation of business processes must include key modeling elements such as activities, execution logic, data definition, control flow, information flow, and external entities. Model Blue Business View provides a two-dimensional usage model to abstract and organize these elements in one picture. In the vertical dimension, the internal execution logic within a business entity (e.g., business organization, service provider, and person or role) is described with common flow diagram constructs. In the horizontal dimension, interaction among business entities is represented by information exchange across entity boundaries. As shown in **Figure 4**, on the left side, execution logic for a business entity (*Clearing bank*) is described inside an *activity lane* (container); on the right side, external entities (*Sender bank* and *Receiver bank*) are represented by *collaborative entities* (person-like icons); across the border of the container, cross-entity interactions are captured as *information artifacts* (round-cornered rectangle icons).

Several Business View models can be put together to form an assembled model on which the interaction among

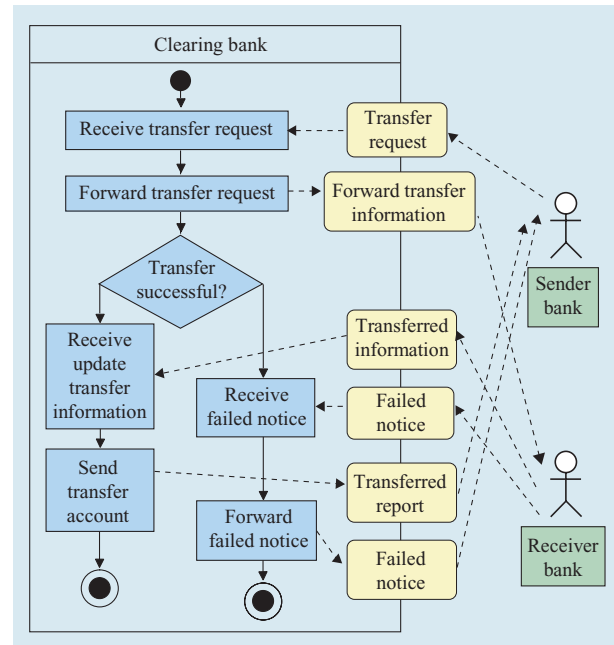


Figure 4

Model Blue Business View.

different business entities can be defined and visualized.

Figure 5(a) gives an example of two *activity lanes* being connected with a *data lane* between to describe information merging and mapping relationships. Two *information artifacts* from different *activity lanes* may have different data structure definitions. It is not necessary for two information artifacts to have exactly the same data definition in order to match. Meta-level structure mapping can be specified on the *information flow* as a data format “transformer” between them in order to establish mapping.

The uniqueness of the Model Blue Business View model and tooling lies in the specific design to help business consultants capture customer requirements. The introduction of *information artifacts* and *collaborative entities* in Model Blue Business View enables business consultants to separately capture business process requirements with individual departments or persons with no requirement for global knowledge. All of these models can then be assembled together with the *data lane* to form a global process picture for verification and analysis.

Figure 5(b) also demonstrates the mapping relationship from the usage diagram model to an equivalent Petri Net [12]. The key value of the Petri Net theoretical foundation is that a formal method can be leveraged to analyze the Business View model for valuable insight. The traditional workflow net [13] analysis method is extended here to test the compatibility among business processes and pinpoint

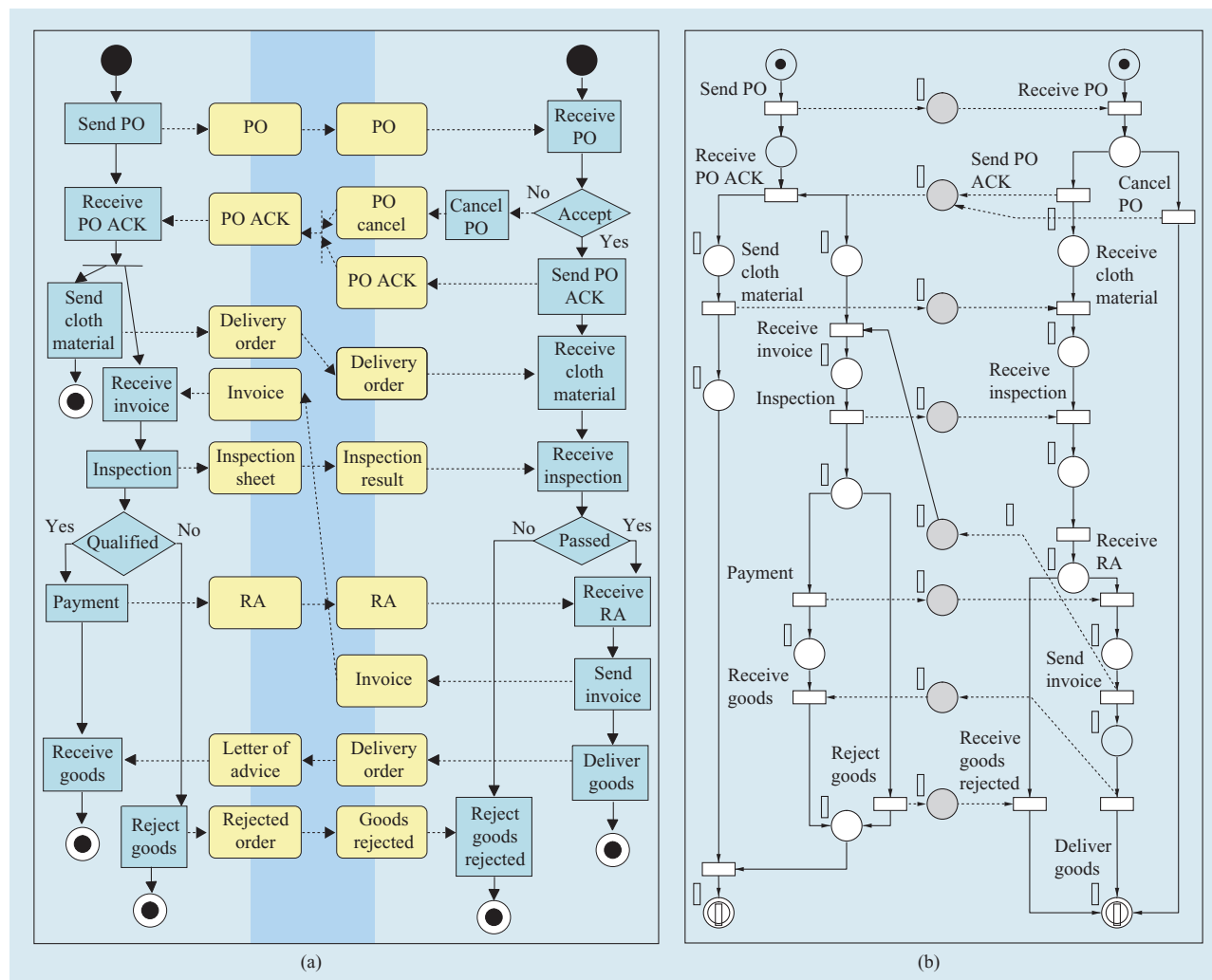


Figure 5

(a) Assembled Model Blue Business View; (b) Petri Net equivalent. (PO: purchase order; ACK: acknowledgment; RA: remit agreement.)

potential interprocess conflicts such as deadlock and infinite waiting [14]. **Figure 6** illustrates the relationship of each business process validation problem to its Petri Net representation. Currently, the effect of using the Petri Net theoretical model analysis is not satisfying enough: Only limited kinds of conflict can be identified, and the result of analysis appears to be trivial for simple processes. Further work is expected in this field to study the applicability of formal model analysis to more valuable business problems.

IT View

Serving as the link between business process and IT implementation, Model Blue IT View provides a set of layered target-oriented modeling methods to lead business requirements toward IT implementation architecture

through predefined guidelines and processes in which IT implementation details (e.g., auditing, performance, and logging) can be introduced gradually. In contrast to prevalent workflow-like representations [15], the IT View model describes business process execution using an annotated tree structure, which comes from the nature of solving complex problems with a “divide-and-conquer” approach. As a result, it provides a common base for business analysts, IT architects, and designers to communicate requirements and design systems. The annotated tree contains three kinds of nodes: The *action node* is a leaf node to represent invocation of an executable software artifact (such as a Java** program, Enterprise JavaBeans** application, Web service, or Remote Method Invocation application); the *sublogic*

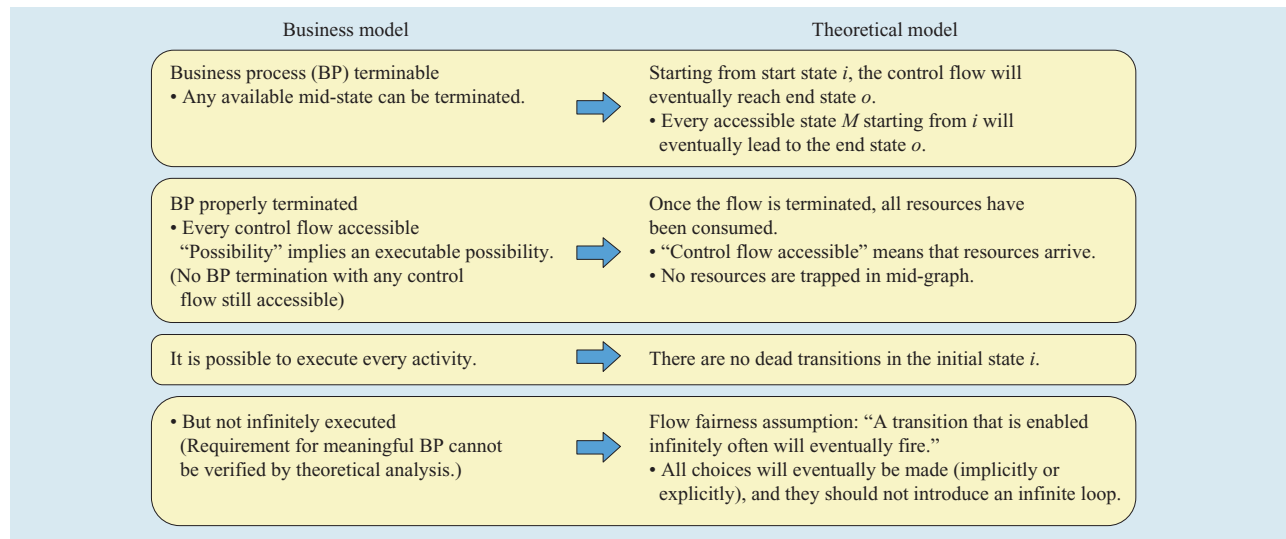


Figure 6

Mapping from business problem validation to Petri Net properties.

node is a leaf node to represent invocation of another annotated tree; the *control node* is a non-leaf node to represent the execution logic (such as sequence, parallel, and choice) of its subnodes. The key value of the annotations tree is that it enables dynamic setting capabilities to describe a complex processing logic with high flexibility and expressiveness, as verified by the workflow pattern defined by [16].

A transformation engine is provided to transform the Business View model into an initial skeleton of the IT View model (as illustrated in **Figure 7**), to which detailed implementation requirements can be added later by IT architects and designers. Starting from the root, the tree progressively shows how the process is decomposed into more manageable or understandable pieces, which can be *sublogic nodes* that represent factoring of the model or *action nodes* that represent the implementation of various facets of the model. Different levels and aspects of the tree presentation enable the business analyst and application developer to access and update the corresponding information they care about. The information about program interfaces, input/output parameters, skills required, and major processing logic can be exported as a development task guideline document and then delivered to developers or vendors as a clear description of the development tasks.

The business process flow composition logic captured in the tree structure can easily be adapted. The leaf nodes can easily be re-bound to another tree or executable software artifact, as illustrated in **Figure 8**. An Action Registry is implemented to maintain and reuse the

interface definition of executable software artifacts. With the help of integrated tooling, one can easily perform interface definition import/export with Java source code. The structured logic representation also makes it easy to extract part of the function and implement it with another mechanism. For example, we can extract a subtree and transform it into another flow representation (i.e., WBI Collaboration); the change is localized. The IT View also supports the debugging and simulation of the business process flow application. This capability can verify the function of an application more easily, shortening the design-to-deployment cycle.

3. Case study with Bank SinoPac

Bank SinoPac profile

Bank SinoPac is a Taiwan-based commercial bank that was founded in 1992 with the corporate mission of becoming a trusted financial institution of the Pacific Rim. In the past ten years, it has been able to achieve the best asset quality among commercial banks in Taiwan and a financial network spanning the Pacific. Despite its moderate size, Bank SinoPac has received awards such as “Best Corporate Governance Company in Taiwan” from *Euromoney* [17], “2003 Bank of the Year in Taiwan” from *The Banker* [18], and “Best Corporate/Institutional Internet Bank in Taiwan” from *Global Finance* [19]. In 2002, it was merged with National Security Corporation to form SinoPac Holding, which now owns a client base of more than 1,000,000.

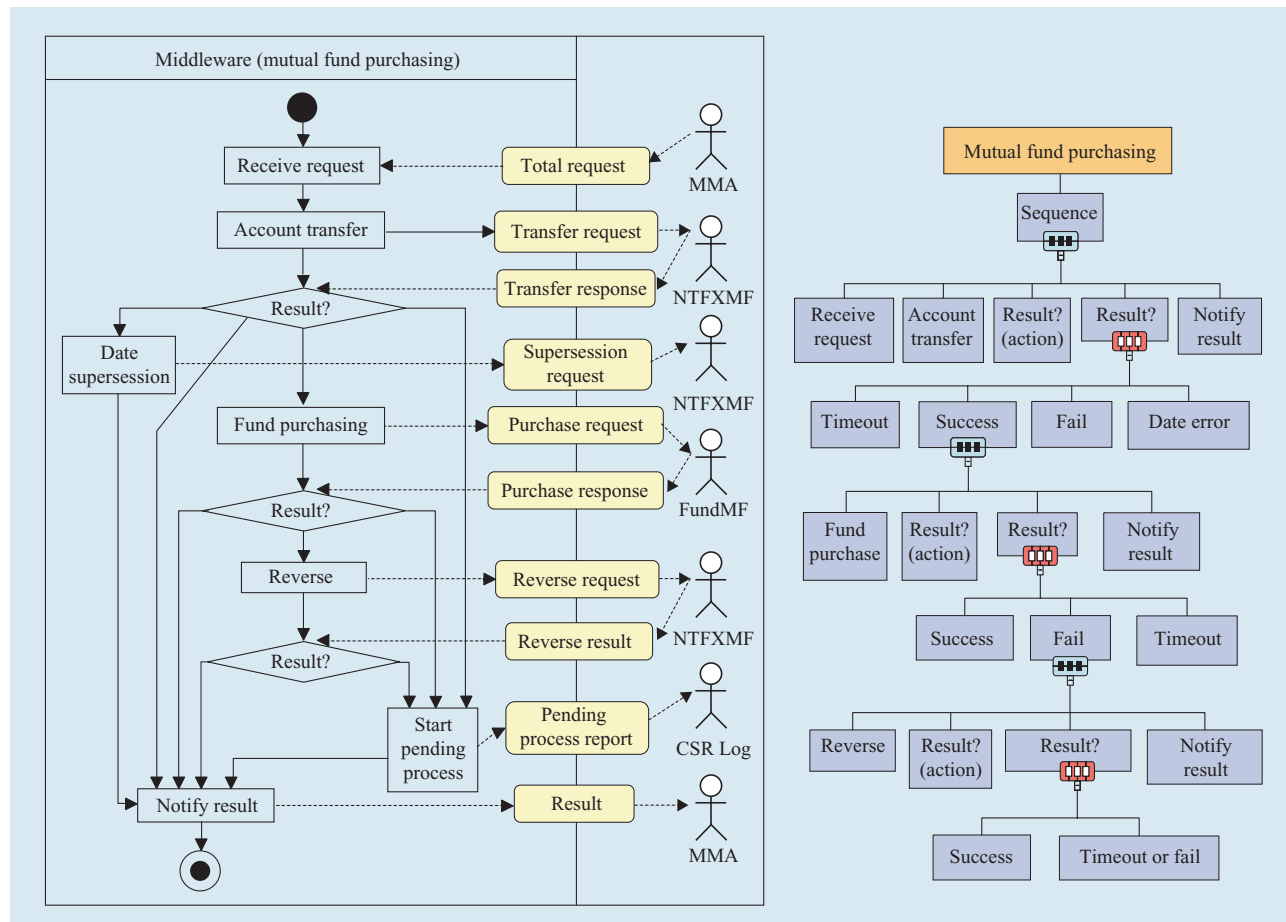


Figure 7

Model Blue IT View skeleton generated from Business View. (MMA: money management account; NTFXMF: new Taiwan and foreign exchange mainframe; FundMF: fund mainframe; CSR: client service representative.)

Bank SinoPac's current operation includes 19 business units at the company headquarters, 44 branch offices throughout Taiwan, an offshore banking unit, one branch in Los Angeles and one in Hong Kong, one liaison office in Vietnam, and an affiliated bank in the United States, Far East National Bank, with a fifteen-branch network and a Beijing representative office. To provide an integrated financial service solution, Bank SinoPac has invested heavily in financial peripheral business, including a finance company, a leasing company, and financial consultation.

Bank SinoPac requirements

To succeed in the highly competitive financial marketplace, Bank SinoPac has made two critical strategic decisions: One is to position itself as a cross-Pacific bank serving the greater China community; the other is to

increase its capabilities through alliance with other financial institutions. Both strategies require a powerful application middleware system that can connect to various legacy systems with flexible business logic processing capabilities.

The legacy application middleware of Bank SinoPac was built primarily on the Enhanced Messaging and Queuing (EMQ) [20] technology, which had only limited business logic processing capabilities. Its scalability, reliability, and extensibility were also issues. Bank SinoPac had reviewed alternative solutions but excluded them because none of these solutions could provide full cycle management from the Business Level to the IT Level. Bank SinoPac intended to rebuild its application middleware to acquire the advanced capability of business logic processing. Bank SinoPac wished to design, develop, and deploy its own business logic on the middleware platform. The

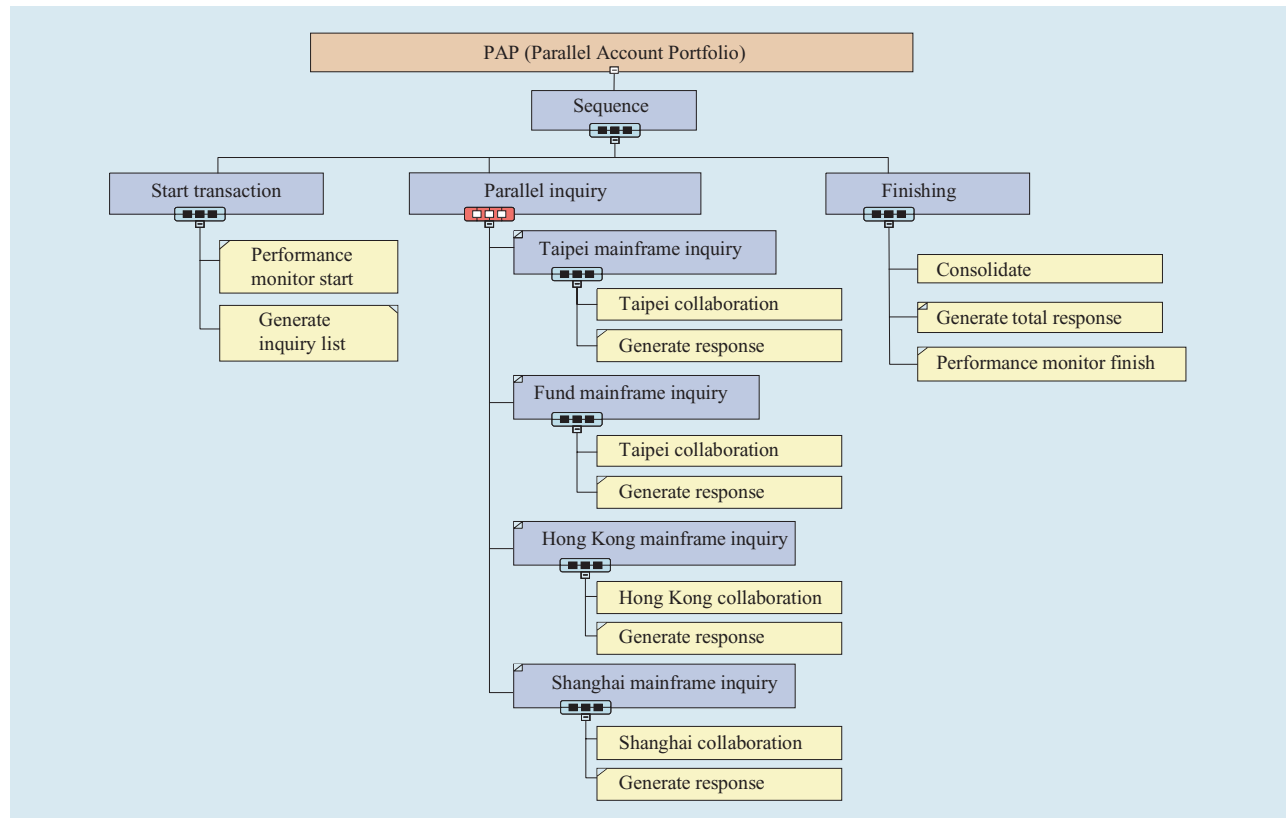


Figure 8

Model Blue IT View with IT details.

application middleware would connect to various back-end and front-end systems through multiple channels, for inter/intra-business integration.

As a specific trial of model-driven business integration technologies, the scope of the project included the following:

- A regional service platform (RSP) for Bank SinoPac to design, develop, and deploy its financial products based on business process modeling, design, and execution. RSP will provide full lifecycle management from the Business Level to the IT Level.
- Key integration solutions for a strategic financial product of Bank SinoPac called Cross Pacific Account (CPA), through which the bank can provide businessmen from Taiwan, mainland China, Hong Kong, and the U.S. with immediate and convenient fund transfer services. CPA includes the following eight sets of business processes, grouped by five patterns for integrating various front-ends, back-end systems, and business units:

- Two-phase transaction (process that contains two consecutive transactions with logical dependency): *Mutual Fund Purchasing* and *Combined Inquiry and Update*.
- Parallel transaction (process that contains several transactions issued simultaneously and then merged): *Account Portfolio Inquiry* and *Credit Portfolio Management*.
- Cross-regional transaction (process that involves transactions to distributed mainframes): *Cross-Regional Money Transfer* and *Account Settlement*.
- Front-end-triggered transaction (process triggered by front-end kiosks): *MMA (Money Management Account) Combo Card*.¹
- Back-end-triggered transaction (process triggered by back-end mainframes): *Business Date Super-Session (daily refreshing mainframe date settings)*.

¹ Money Management Account is a capital management service of Bank SinoPac to help company and individual clients to effectively manage investment through integrated multiple bank accounts.

Challenges and success criteria

The following points were identified by Bank SinoPac and IBM as the key project challenges as well as criteria by which to verify the Model Blue solution, based not only on formerly anticipated effects of the model-driven approach, but also on more specific requirements from the real situation:

- *Lightweight integration solution* For a mid-sized company like Bank SinoPac, a lightweight integration solution is not just a feature that is nice to have. The term “lightweight” here has multiple meanings. First, the whole set of service methodology should be practicable for a small project team with limited external technical support. Second, the methods and tools should be easy to use so that they can be learned and applied by people with a normal business/IT background under limited training. Third, to keep a smaller footprint, most mid-sized companies wish to have a lightweight integration runtime solely on top of a standard application server; on the other hand, the flexibility and performance of the integration runtime should also be considered. In this specific case, for instance, Bank SinoPac wished to reuse existing commercially available application connectors.
- *Cost-effective testing and debugging* The testing of an integration solution usually involves linkage to multiple legacy systems, of which most are critical online parts of the daily operation system. It is far more complex than ordinary software testing, which normally includes a limited number of components and dependencies and can be executed in a staging environment. Testing has become a major challenge for traditional black box debugging technologies [21]. Therefore, an integrated debugging capability to easily step from process logic to connector code (and vice versa) is highly preferable. To take all dependencies into consideration, testers usually face the challenge that even thousands of test cases fail to cover all possible scenarios of a simple integration solution; thus, the capability to automatically generate test cases with acceptable function coverage becomes another critical issue for the efficiency of testing.
- *Easy-to-use communication tools* Many business process modeling tools today are based on either IT-Level models or tools (e.g., UML [22] used in IBM Rational Rose* [23]) or diagram-drawing environments (e.g., Microsoft Visio**, iGrafx FlowCharter** [24], and SmartDraw** [25]). For business consultants and analysts, additional training is required. Moreover, most of the tools are not designed for the working style of business consultants and analysts (for example, how to formulate a global system picture through the results of separate interviews with individuals). Therefore, more effective communication tools are needed to help

establish common understanding on requirements among Bank SinoPac staff (from both business and IT units), and within the IBM project team.

- *Resolution of Business–IT gap* One challenge for today’s software lifecycle engineering approaches is how to bridge the gap between business consultants and IT specialists, due primarily to differences in background, skills, and vocabulary. Some work has been done in the field of enterprise architecture (e.g., Zachman Framework [26] and Popkin Enterprise Architecture [27]), but this work is not focused on solution development. Designing an IT implementation that really conforms to business requirements is essential for the success of the project.
- *Asset reusability* Reusability is a well-studied topic in the literature [28]. Design documents and implementation codes are intellectual assets which, if made reusable, will bring considerable value to current and future solutions. One criterion for software reuse efforts is the level of reuse [29]. It is easier to reuse high-level models than to reuse software code. Therefore, the integrated set of models across the phases of the solution lifecycle promotes more effective asset reuse; this is an important criterion for the project.
- *Distributed working team management* Given the fact that there are different project teams residing in Beijing, Taipei, Hong Kong, and Nanjing, helping geographically distributed teams to understand the complete working environment for their job assignment is a great challenge, and many lessons have been learned over the years [30]. A widely distributed collection of individuals working separately can easily lose track of the others’ work and progress [31]. As a result, another challenge is allocation of work items to outside contract programmers and testers to clarify job specification, job context, and quality assurance criteria, while avoiding disclosure of significant details of Bank SinoPac’s business processes.
- *Effective business/IT change management* For an integration project, there are complex dependencies among system components. One single requirement change usually propagates across many other components of the systems, especially in the latter phases of the project. To guarantee on-time and on-budget delivery, changes should be managed at the enterprise level and analyzed by the architect for their potential impact before being accepted and implemented. Management of changes means managing the process of changes as well as managing all artifacts of an evolving software system [32]. Both Bank SinoPac and the IBM project manager wish to have the capability to demonstrate the possible effects of the change requests, from the perspective of both function and implementation impact, so that they can make the correct decisions.

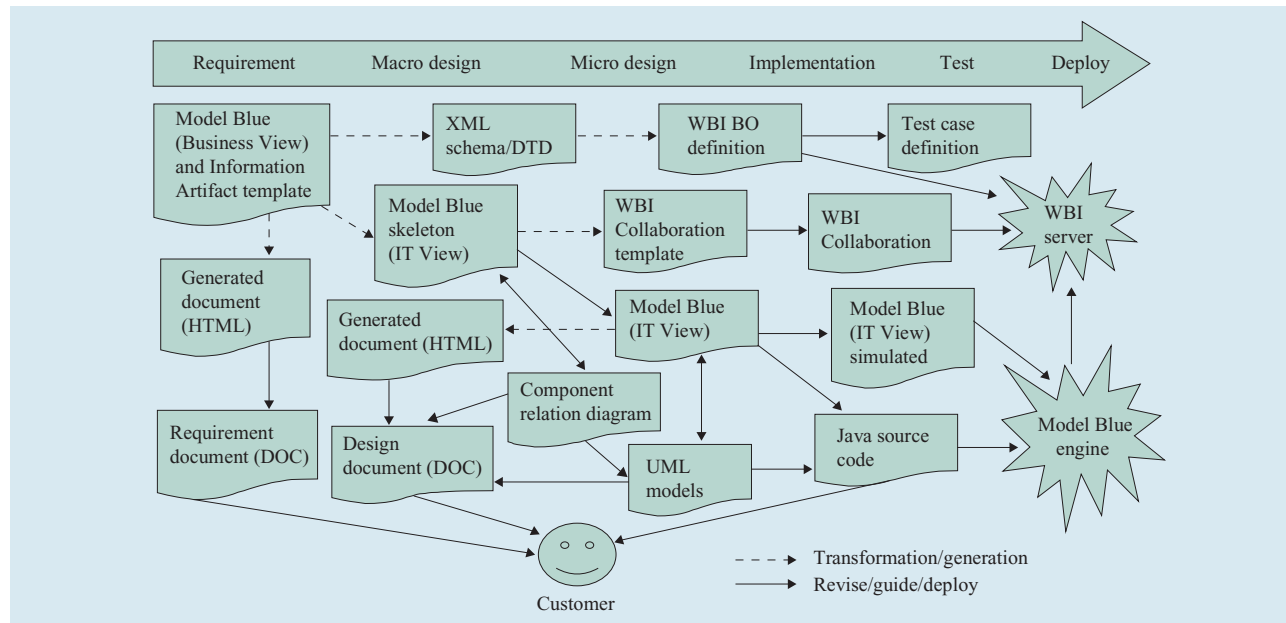


Figure 9

Model Blue engagement lifecycle and output.

- *Alignment with existing service/software methodologies*
Expecting workers to pick up a totally different way of doing a job is unrealistic, especially for service teams that have already endorsed a well-defined software methodology, such as the Rational Unified Processes (RUP) [33]. Providing seamless mapping and alignment to existing paradigms thus becomes a critical issue for acceptance of a new methodology, especially when the project timeline is too tight to allow for much contingency for training time.
- *End-to-end solution* Each challenge above, in isolation, is not a major issue, and we may already have answers for them. There are still unresolved problems in addressing all of these concerns in a consistent and coherent way so that the project can be guided smoothly from the beginning to the end. Seamless connection of various phases of a solution lifecycle with some level of automation is a critical issue for most service and software methodologies. Most of today's integration projects require significant investment in money and time. Projects suffering delay caused by unexpected events are not uncommon. An end-to-end methodology to mitigate these risks is an imperative for success.

Engagement lifecycle and outcome

Figure 9 shows an engagement lifecycle driven by models, more specifically Model Blue Business View and IT View

models. It is not mandatory that all solutions strictly follow the listed phases (i.e., requirement, macro design, micro design, implementation, test, and deploy). In a specific project, recursion and repetition of phases are allowed. From the macro level, however, these phases can be regarded as generic for most business solutions. On the basis of this common lifecycle, Model Blue methodology can easily be mapped and integrated using existing solution methodologies and software engineering approach (e.g., Rational Unified Processes). In this specific case, Model Blue is integrated with the *e-business Custom* engagement model in the IBM IGS Method, which is a widely adopted services methodology within IBM. The models built by practicing Model Blue (and documents generated) are successfully integrated as a part of (or a replacement for) the work-product deliverables defined in the IBM IGS Method. Since these models are built in collaboration with Bank SinoPac as part of the specification and design process, the semi-auto-generated work product (requirements specification, macro design, micro design, test-case specification, etc.) is easily accepted by the corresponding owners of the receiving organizations.

In the remainder of this section, we describe the methodology in detail, including the major models created, transformed, updated, and utilized in each phase. Key experiences received from the engagement are also highlighted.

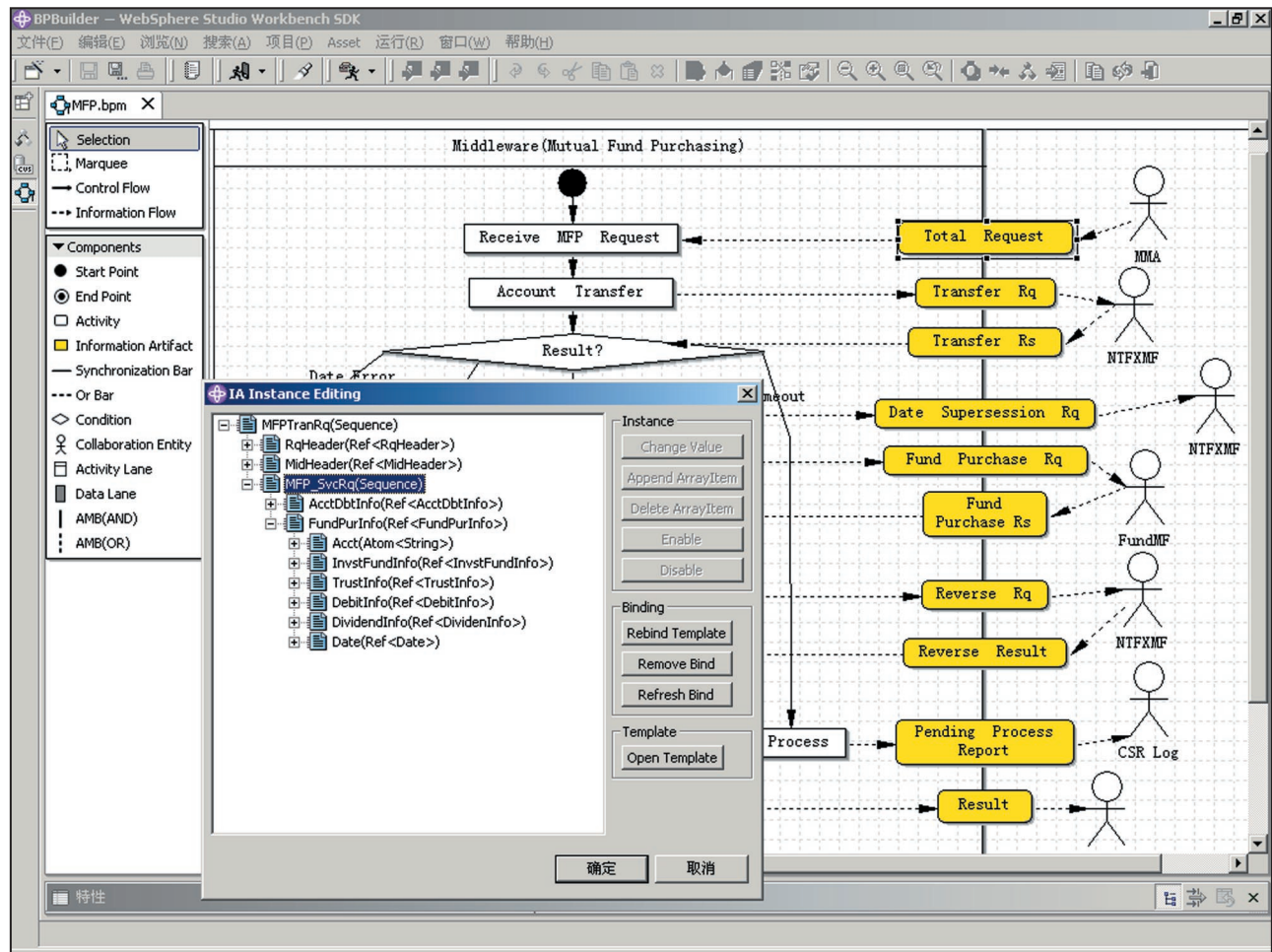


Figure 10

Business View builder: Information Artifact definition.

Requirement specification

In the requirement specification phase, business consultants interview the Bank SinoPac staff to write down their “as-is” and “to-be” business process models. The objective is to discuss, capture, and document the internal process logic as well as interprocess interaction relationships clearly, so that all related parties can fully understand the business process requirements.

Deliverables

- **Business View models** Eclipse-based tooling is used by consultants to create the two-dimensional Business View models. The tool provides an integrated modeling environment with capabilities of diagram drawing, importing and exporting, assembling, verifying, printing, etc. Some add-on capabilities, such as detecting conflict

among assembled business processes, are also provided. The generated model is an XML file with both syntactic (XML schema) and semantic (well-formed rules with tooling support) correctness definitions.

- **Information Artifact templates** The Information Artifact (IA) template is the meta-level definition of data elements that are exchanged across different business processes and entities. One example of an IA template is the account transfer request and response messages that are exchanged with the core banking system. The data structure is defined in a tree-view editor and saved as an XML file, which is then bound with the Information Artifact construct in the Business View model, as illustrated in **Figure 10**.
- **HTML format requirement documents** HTML format requirement documents are automatically generated from the above models as a recordable representation

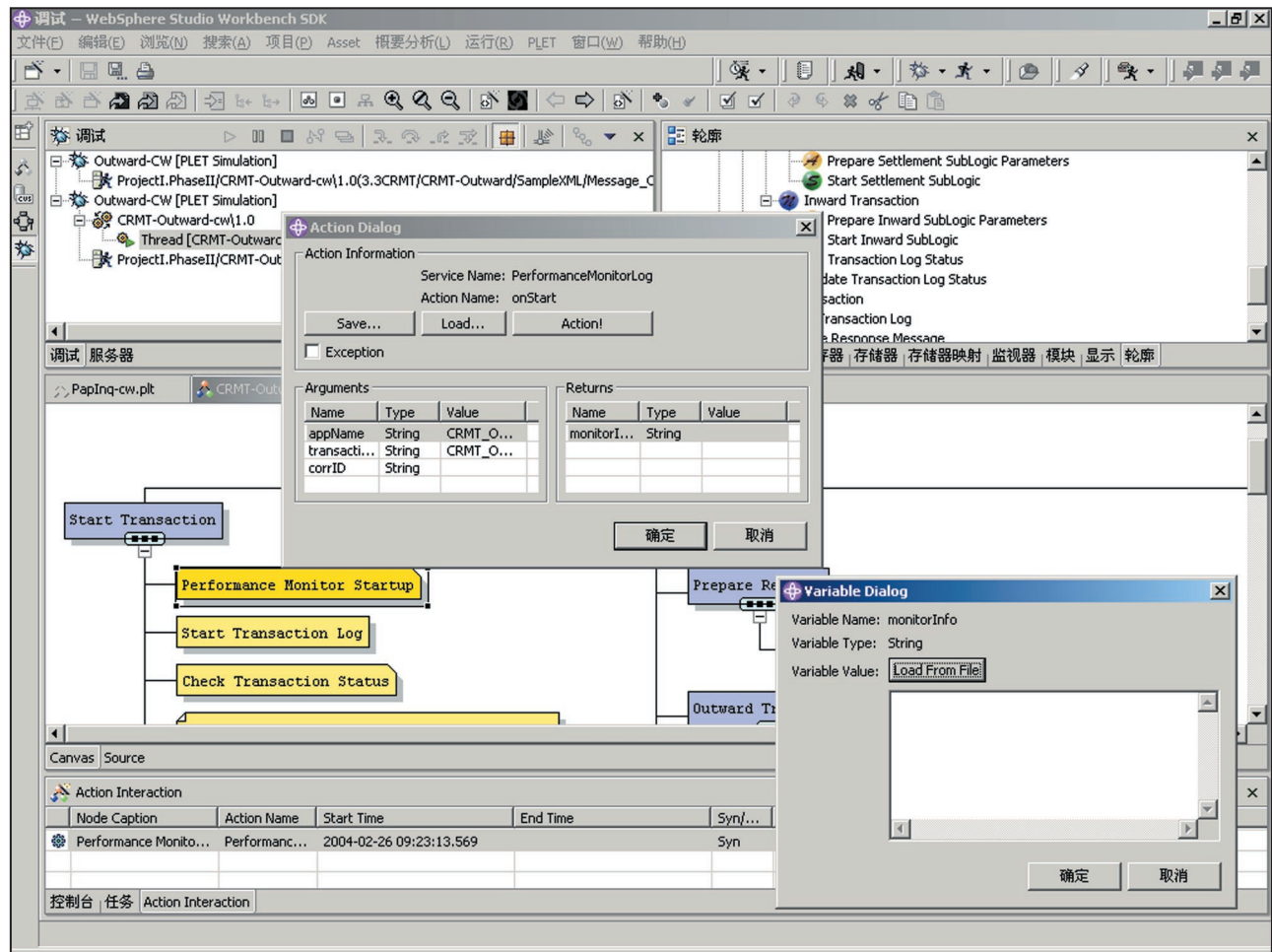


Figure 11

IT View designer: Model Blue IT View model.

of the process description. They also contain information regarding many nonvisualized yet valuable inputs during the interview (e.g., some business rules definitions and text-formatted comments). The HTML documents are then merged into the requirement document.

Key experiences gained

- For the Bank SinoPac staff, an intuitive and simple modeling method with tooling to document or record their requirements (focused on data and process elements) is a major advantage.
- One challenge which remained unsolved was how to effectively manage and handle all of the business terminologies used in the models and design documents. The introduction of ontology or vertical standards to regulate the vocabulary usage seems to be the solution,

yet most existing ontology specifications today (i.e., Resource Description Framework [34, 35] and Web Ontology Language [36]) are too heavyweight to be put into practical usage.

Macro design

In the macro design phase, business consultants work together with solution architects to produce an overall architecture of the target system. The objectives of this phase include designing the execution process and identifying the high-level component architecture.

Deliverables

- *IT View models (skeleton)* An initial IT View model is automatically generated from the Business View model. The flow execution logic embedded in the original

diagram is analyzed, extracted, and then transformed into an equivalent tree-structured representation as the starting point for IT execution flow design. Further process design then follows (e.g., adding some action nodes to describe implementation details such as logging and data transformation, which are not the major concern for business consultants). The Business View designer, transformation engine, and IT View designer are seamlessly integrated to provide unified end-to-end user experience.

- *Component Relationship Diagram (CRD)* A high-level architecture for all key function components within the system is created in the macro design phase. Typical components may include the integration hub, back-end systems, database and directory server, and utility services. A component relationship diagram is then created to show all of the components as well as the interaction relationships among them. There is no need to produce a detailed design for each component in this phase.
- *Mapping from IT View model to components* Mappings from the IT View model to related components are identified in this phase by introducing new components if they are not yet included in the CRD. For example, if one action node in the IT View stands for the invocation of a performance logging that is not included in the CRD, a new performance logging component is added and a mapping is established from the action node to the component.
- *Adapters/Collaboration list* For an integration project, one important element is the connection to various legacy systems. In the macro design phase, all key adapters required are identified. For this specific engagement, the application connection layer is implemented with IBM WebSphere Business Integration (WBI), which provides many standard connectors to legacy systems (e.g., IBM WebSphere MQ Series, Unisys, and ecSolution).

Key experiences gained

- With a few days of training and practicing, the IT View model and tools can be understood and used by the IT staff. However, the business staff needs more time to understand the structured representation of process flows.
- With IT View models, reusable components are easily identified and organized, and as a result the development phase is greatly accelerated.

Micro design

In the micro design phase, solution architects and process designers continue their work on the macro design to

make it complete and executable. The objective of this phase is to create a design specification with enough information for programmers to develop the solution.

Deliverables

- *Model Blue IT View models (simulated version)* With the help of the Eclipse-based IT View designer (**Figure 11**), initial versions of IT View models are elaborated in the micro design phase toward executable scripts, which can then be deployed on the engine to drive business execution. Simulation support is provided for the Model Blue IT View model (Figure 11) so that one can verify the correctness of the flow structure before real coding is started. The simulation engine is built with the same kernel of the Model Blue IT View Engine that actually executes the IT View model. The major difference lies in that input and output from legacy systems are redirected to local file systems so that one can easily emulate back-end systems to produce various scenarios.
- *Finalized adapter (Collaboration) and action list* On the basis of the results of the finalized version of the IT View models, a table is generated to list all of the adapters (WBI Collaboration) and service actions that have to be implemented. It will be used as a basis to create job specifications for contract programmers and testers.
- *Information schema specification* A transformation engine is provided to automatically generate a standard XML schema definition from the associated Information Artifact template definition in the Business View model. The XML schema is then used as the standard data schema specification for the whole project team.
- *WBI² Collaboration template and Business Object (BO) definition* For this project, one specific requirement from Bank SinoPac was to fully leverage existing WBI connectors; therefore, some parts of the functions related to back-end system connection (usually in the form of a subtree of the IT View) are implemented in the form of WBI Collaboration. At runtime, the Model Blue IT View Engine is responsible for executing most flow control logic (such as branch and parallel) and invoking certain functions on the WBI server to access back-end mainframes. To drive the implementation phase smoothly, we developed a transformation engine

² WebSphere Business Integration (WBI) is a collection of IBM software products for business integration which includes the following core capabilities: Model, Integrate, Connect, and Monitor. Because of issues related to customer acceptance and product lifecycle, two major components of WBI were used in this engagement: WebSphere Business Integration Collaboration, which provides pre-built process templates for common business practices, and WebSphere Business Integration Server, which provides a runtime to execute WBI collaboration.

to automatically turn a subtree of the Model Blue IT View into an equivalent representation of the WBI Collaboration template.

For the information specification part, the generated XML schema is transformed into XML data type definition (DTD) and then converted by WBI tools into the definition of the Business Object in WBI.

- *Detailed component specification* On the basis of the component relation diagram, detailed designs in the form of UML models (i.e., class diagram, sequence diagram, and state diagram) are created for those components that must be created during the implementation phase.
- *HTML format design documentation* As in the requirement-collecting phase, HTML format design documents are automatically generated from Model Blue IT View models to create a recordable representation of the whole system design.

Key experiences gained

- Since it is an interface common to different sections of the system, the information schema definition proves to be the critical point in project lifecycle management.
- The Eclipse-based tooling environment is welcomed by business consultants, IT architects, process designers, and programmers. The integrated tooling environment and workspace increase the efficiency of knowledge sharing and discussion among them.

Implementation

In the implementation phase, the software code for the system is developed. The objective of this phase is to create the development document, software code, and test cases.

Deliverables

- *Development task guide document* To let developers understand clearly the specific tasks assigned to them, a document is provided for each task in the developer base of the Model Blue IT View model. This document includes the guide for the program interface, major processing logic, input/output parameters, Java source code skeleton (all of the above items generated from the Model Blue IT View model), skills needed, etc.
- *Test cases* In this phase, test cases for the test phase are prepared. The IT View models are analyzed to create the testing scripts for business process models. The tree traversal structure helps clarify all possible process execution scenarios. As a result, it reduces the time needed to prepare the test cases and enhances the coverage of them.

Key experiences gained

- XML is universally used for the data exchange from front end to integration hub and then to back-end connectors. One problem is the performance overhead caused by XML tree parsing. We recommend that future integration solutions include native XML support (for example, including XML as an internal data type so that unnecessary transformations between strings and the XML tree can be avoided).
- Outsourcing control is a major issue in a relatively large-scale development project. We have combined multiple models into a development task guide document; the result is very encouraging in that most contractors can understand their jobs as described by the document quite well, while the document contains no additional information that they do not need.

Test

In the test phase, architects, testers, developers, and Bank SinoPac staff work together to verify the target system against its requirements. The test phase is actually divided into four stages: Unit Test, Function Verification Test (FVT), System Integration Test (SIT), and User Acceptance Test (UAT).

Deliverables

- *Testing document* A summary of the testing work is presented in the delivered documents. It includes the purpose and principle for testing, a description of each test case (scenario, input, expected output, owner, etc.), results for each test case, and a description of each variance (severity, scenario, input, output, owner, handling result, etc.).
- *Performance test report* Performance is a major concern of a production system. In this project, we used the Rational Team Test suite and Websphere Resource Analyzer as integrated tooling for the performance test. Two major system performance factors, throughput and response time, were generated under different testing scenarios (e.g., different concurrent users).

Key experiences gained

- Testing of an integration project does involve multiple back-end systems. In the FVT stage, when no back-end connectors were available, we had to build additional programs to emulate different back-end mainframe transactions. Even in the SIT stage, when we already had the real adapter code from partner teams, it was still difficult to actually connect to back-end mainframes that were running mission-critical applications.

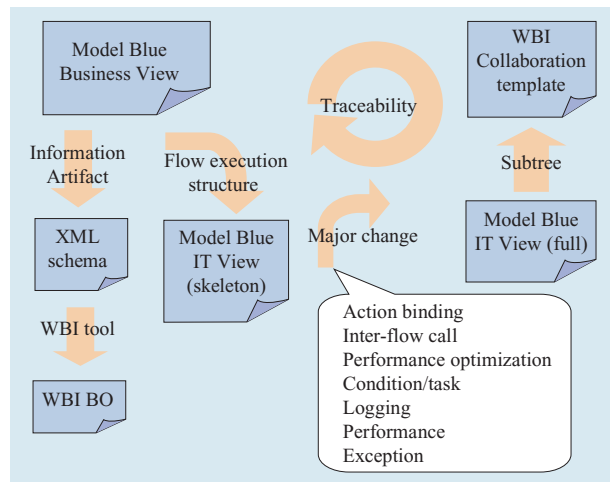


Figure 12

Model transformation in Model Blue.

- IT View provides an integrated visual debugging environment at build time. Integrated debugging can help to detect logic defects between flow and actions before the deployment. This shortened the test and revision cycle and reduced the testing time significantly.
- When performing FVT and SIT of process flows, simply verifying the return XML message is not enough. The system logs of the engine should be inspected to obtain a detailed execution history.
- According to the performance test, XML processing is a memory- and CPU-intensive task. The technique of XSLTC [37] can boost the efficiency of XML transformation. One approximate test (on a PC) revealed that introducing XSLTC with other settings unchanged reduced processing time by at least 60%.

Lifecycle summary

In summary, models are used as the key element to drive the whole BPIM solution. A set of models is taken as input reflecting each phase of the lifecycle. With defined steps of model creation and revision, another set of models (also including the generated documents from models) is generated as output. Model Blue is more focused on the business process model, whether it is in the Business View or IT View, so the *model-driven* approach can be labeled as *process-centric*. The process-centric approach fits well with Bank SinoPac's integration purpose—to support the introduction of new financial services and products by creating business processes to connect various back-end systems, workers, and resources. We recognized the fact that some particular integration scenarios require modeling to be focused on other aspects

[e.g., data modeling supported by the IBM DB2 Information Integrator (DB2 II)]. However, the key challenge for these solutions lies in effectively organizing the data view around the usage context, which is usually the business processes that consume these data.

4. Best practices and lessons learned

Model transformation

Model transformation is critical for the smooth execution of a model-driven approach. It is regarded by some as the heart and soul of model-driven software development [38]. In Model Blue, we have developed different transformation engines serving different purposes, as illustrated in **Figure 12**:

- The “Business → IT” transformation engine, although not full-fledged, has demonstrated its capability as a bridge between the business specification and IT implementation models. We can fill the so-called business-IT gap even better with more transformation capabilities (e.g., how to do round-trip transformation from the IT View to the Business View, and how to transform the data mapping relationship within the Business View into corresponding XSLTC scripts). The most challenging problem, however, lies in propagating changes from the Business View to the IT View when some later changes are required.
- The “Information Artifact template → XML schema → WBI BO” transformation engine is a key step for the handover of data models from a business person's perspective to standard IT representation.
- The “IT View → WBI Collaboration template” transformation engine is intended to simplify the work of collaboration designers with a model transformation approach, yet the results have not been ideal. The collaboration designers were reluctant to learn a new modeling paradigm that did not greatly reduce their workload. This was partly because most collaboration used by this engagement is relatively simple and thus can be easily hand-coded. The value of the IT View model was its capability to define and execute (with the Model Blue IT View Engine) complex process logic (e.g., parallel execution) that was not supported by WBI Collaboration.

In summary, model transformation has played an important role in connections for the whole project lifecycle. However, we should never overestimate the power of model transformation and overlook the complexities it may introduce. One guideline for model transformation is to use this approach only when it really brings value or cannot be replaced by other efficient alternatives.

Asset management

We tried three different methods of managing the vast numbers of assets (models and documents) across the project lifecycle. The first was to use a Lotus Notes* database as a shared repository for model and documents. Outside resources, such as supplemental and contract personnel, needed Notes IDs to access the database. We also had difficulties in synchronizing updates. Notes is currently not well integrated with development environments such as Eclipse and WebSphere Application Developer (WSAD).

The second choice for us was to use the open-source Concurrent Version System (CVS) [39] to share all related models and code. CVS provides a lightweight team document sharing tool which is seamlessly integrated with WSAD and Eclipse. For this project of moderate size (no more than 50 developers and testers), we chose CVS as the collaboration server, mainly because of the efficiency boost.

The third alternative was to use the Model Blue Asset Manager, which is designed to support asset sharing and reuse in an enterprise environment. Asset Manager had all of the capabilities required by a large-scale project environment, including complex categorization and access control capabilities (integration with enterprise directory server); however, it was heavy-weight (requiring WAS, DB2, etc.) and thus hard to install and configure.

The adoption of the CVS server illustrated the fact that most people wish to have easy-to-use, lightweight systems. Packing too many functions into one solution may, contrary to initial expectations, greatly increase the complexity and exclude the solution from practical consideration.

Change management

In an integration project involving many enterprise business applications and processes, requirement changes will inevitably have a significant impact on the whole project. Discipline with respect to change estimates and approvals thus becomes very important. During the project, the IBM project team experienced seven major requirement changes that required Bank SinoPac to sign off officially, in addition to numerous other minor changes that were contained internally by the project team. The Model Blue Business View provided a clear view of the requirement change at the business operation layer. This was used to illustrate and discuss the change requirements with Bank SinoPac to better understand what really had to be changed. The clear mapping relationship from the Business View to the IT View made it possible for the IBM project team to demonstrate various effects of the changes proposed, and at the same time limit the change propagation to a controllable scope. The Model Blue IT View provided a base for calculating additional resource

and time needed to make a change possible. For example, when we had to change the data format of a specific business object, we went through the IT View models to find all actions that utilize the data and thus have to be revised accordingly, and then determined how much workload (in terms of person-days) was required and how much delay was expected.

The ideal method of model-driven change management would be to propagate changes automatically from the Business View to the IT View without major manual work. For Model Blue, this capability is not yet supported, partly because it is still unrealistic for a transformation engine to convert everything from the Business View into the IT View. The support of Model Blue on change management is still focused on the facilitation of requirement illustration, solution discussion, and impact estimation for customer change requests. Even so, the model-driven approach has demonstrated its potential for solving the problem of scope and change control, which are essential factors for the on-time and on-budget delivery of integration projects.

Skill transfer

Transferring skills to the Bank SinoPac staff and enabling them to use and operate the Model Blue methodology was challenging, given that most of them were new to even XML and Java. The IBM team used various alternatives to help them gain the ability to use the tool and methodology. The China Research Laboratory (CRL) hosted two face-to-face workshops with the Bank SinoPac business and IT staff and business consultants in BCS. CRL also delivered a week of lecture-based training at the Bank SinoPac site for about 30 Bank SinoPac staff from different departments. CRL and BCS delivered a week of on-site training in Taipei after the lecture, working with five Bank SinoPac employees from the IT department on a sample integration scenario. Yet another approach used by the BCS team was to invite Bank SinoPac to send their employees to work with them. A BCS project manager then assigned work to the Bank SinoPac employees as if they were IBM team members. Through several months of working together, the Bank SinoPac employees became familiar with most of the details about the project development lifecycle. The skill transfer results were very good.

Since the IT experience, knowledge, and capabilities of a customer's staff are difficult to predict in most engagements, deploying a new methodology can impose major risks in execution and eventual acceptance. In this project, we believe that the lightweight nature of Model Blue played an important role in making it much more accessible and acceptable to new users.

5. Summary

A set of model-driven business integration and management methods, frameworks, supporting tools, and runtime environment was developed and demonstrated. An entire solution was provided, from Business Process Modeling at the operations level, through the execution and implementation levels, all the way to coding and testing. The Model Blue Business View provided a two-dimensional modeling capability that mixed flow modeling and data modeling. This, plus its Eclipse-based tooling, made it a powerful tool for Bank SinoPac to use in defining their business requirements with good structure and detail. Without this, the project team would not have been able to collect requirements and specifications in such a short time. The Model Blue IT View provided a tree-based model to help the Bank SinoPac IT staff work with the IBM project team to identify their IT goals, limitations, and expectations. The project team was also able to explore a number of different designs and implementation approaches before coding. The Model Blue IT View helped the geographically distributed project team to understand the complete working environment for their job assignment. It helped to allocate work items to outside contract programmers and testers in terms of job assignment specification, job context, and quality assurance criteria. The Model Blue IT View also helped to generate testing scripts for the business process, dramatically reducing the time needed to prepare the test cases and enhance test-case coverage. The Model Blue IT View Engine provided a very lightweight WAS-based script execution environment to run the customer's business processes by routing through various business applications and/or transactions. The combination of Model Blue runtime and the WBI Server achieved the customer's goal of high flexibility and high performance.

BPIM methods and models were seamlessly integrated in Model Blue with existing development methodologies, with full lifecycle tooling support. The models and documents built by practicing Model Blue were successfully integrated as a part of (or a replacement for) the work-product deliverables defined in the IBM Global Service Method. Since these models were built in collaboration with the customer as part of the specification and design process, the customer was very happy with the semi-auto-generated work product (requirements specification, macro design, micro design, test-case specification, etc.). Compared with approaches employed on similar projects, the model-driven approach was up to 30% more efficient. The work was done on time, within budget, and with extremely high customer satisfaction, despite remote locations and environmental restrictions which prevented the team from having face-to-face meetings and discussions.

Changes in requirements were contained because of the clear modeling layers in Model Blue and the clear mapping among them. The project team was able to demonstrate various effects for the changes proposed by the customer, in terms of both customer function and implementation impact. Throughout the project, there were seven major requirement changes after design specifications were officially signed off by the customer, as well as numerous other "small or incremental" changes.

6. Opportunities and challenges

Despite all of the positive feedback received from users, the methods and tools described in this paper are far from the degree of perfection which model-driven technologies can achieve. IBM and Bank SinoPac are planning future collaborations to further explore the opportunities and challenges of using a model-driven approach to address more complex business and IT problems. Future topics include simple, flexible, and configurable business process modeling capability; modeling additional existing business processes for Bank SinoPac; strengthening the theoretical foundation to include more analytic features; exploring opportunities for research on business semantics; and applying the model-driven business integration approach to industries other than banking.

Acknowledgments

The authors are grateful to the management teams of Bank SinoPac, BCS and CRL, especially Joan N. Fang, S. B. Hsieh, Cliff Yu, Max Kuo, James Yeh, and Lou Zhou, for their persistent support for the project across the whole project lifecycle. Some of the initial Model Blue ideas came from early work (i.e., Information Function Flow) from the IBM Thomas J. Watson Research Center, and here we would like to express our appreciation to Jen-Yao Chung, Kumar Bhaskaran, Santhosh Kumaran, Ying Huang, Zongwei Luo, Nathan Caswell, and Nigam Anil. We owe thanks to other project participants for their ideas and efforts, including Denise Chen, Wei-Chih Yu, Andy Chen, Winston Huang, Ching-Ting Pan, Nan-Ni Hsu, and Annie Chen from Bank SinoPac; Hank Chen and Jesse Ko from IBM BCS; and Xin Zhang, Guanqun Zhang, Jing Li, Jian Wang, Haiqi Liang, Tony J. Liu, and Walter Yang from IBM CRL. The authors also appreciate the assistance from Dong Liu and Xiulan Yu in reviewing this paper. Finally, special thanks should go to Helen Zhang for her great contributions to the earlier version of Model Blue.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Microsoft Corporation, BEA Systems, Inc., iGrafx (a division of Corel, Inc.), or SmartDraw.com.

References

1. Hurwitz Group, "Connecting the Enterprise: iWay Software"; see <http://www.iwaysoftware.com/products/whitepapers.html>.
2. IBM Corporation, IBM Websphere Business Integration home page; see <http://www.ibm.com/software/info1/websphere/index.jsp?tab=products/businessint>.
3. Microsoft Corporation, Microsoft BizTalk Server home page; see <http://www.microsoft.com/biztalk/>.
4. BEA Systems, Inc., BEA WebLogic Integration home page; see <http://www.bea.com/products/weblogic/integration/index.shtml>.
5. N. G. Carr, "IT Doesn't Matter," *Harvard Business Rev.*, pp. 5–12 (May 2003).
6. K. Nelson and H. Nelson, "The Need for a Strategic Ontology," *Proceedings of the MIS (Management Information Systems) Quarterly Special Issue Workshop on Standard Making: A Critical Research Frontier for Information Systems*, Seattle, December 12–14, 2003, pp. 361–371.
7. Object Management Group, Model Driven Architecture; see <http://www.omg.org/mda/>.
8. IBM Corporation, IBM WebSphere MQ Workflow home page; see <http://www.ibm.com/software/integration/wmqwf/>.
9. Business Process Execution Language for Web Service Specification; see <http://www.ibm.com/developerworks/webservices/library/ws-bpel/>.
10. Eclipse.org, Eclipse White Paper; see <http://www.eclipse.org/whitepapers/eclipse-overview.pdf>.
11. H. Liang, J. Wang, Z. Tian, L. Zhang, J. Zhu, and W. Ding, "Managing Business Process Models: A Private UDDI Application Study," *Proceedings of the 6th World Multi Conference on Systemics, Cybernetics, and Informatics (SCI 2002)*, pp. 408–412.
12. T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proc. IEEE* **77**, No. 4, 541–580 (1989).
13. W. M. P. van der Aalst, "Application and Theory of Petri Nets," in *Verification of Workflow Nets*, P. Azema and G. Balbo, Eds., 1997; published in *Lecture Notes in Computer Science* **1248**, 407–426 (1997).
14. W. Ding, Z. Tian, L. Zhang, J. Wang, J. Zhu, and H. Liang, "Conflicts Analysis for Inter-Enterprise Business Process Model," *Proceedings of the 6th World Multi Conference on Systemics, Cybernetics, and Informatics (SCI 2002)*, pp. 245–250.
15. D. Georgakopoulos, M. Hornick, and A. Sheth, "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure," *J. Dist. & Parallel Databases* **3**, No. 2, 119–153 (1995).
16. W. M. P. van der Aalst and A. H. M. ter Hofstede, "Workflow Patterns: On the Expressive Power of (Petri-Net-Based) Workflow Languages," *Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002)*, Vol. 560 of DAIMI, pp. 1–120; University of Aarhus, Aarhus, Denmark, August 2002.
17. *The Banker* online; see <http://www.thebanker.com/news/fullstory.php/aid/564/Taiwan.html>.
18. *Euromoney* online; see <http://www.euromoney.com/awards2003/>.
19. *Global Finance* online; see http://globalf.vwh.net/content/?article_id=485/.
20. Enhanced Messaging and Queuing home page; see <http://sybooks.sybase.com/onlinebooks/group-emq/eqg0260e/>.
21. J. A. Whittaker and H. H. Thompson, "Black Box Debugging," *ACM Queue* (issue on Distributed Development) **1**, No. 9, 68–74 (December/January 2003/2004).
22. Object Management Group, Unified Modeling Language; see <http://www.omg.org/uml/>.
23. IBM Corporation, IBM Rational Rose; see <http://www.rational.com/>.
24. iGrafx FlowChart home page; see <http://www.igrafx.com/products/flowcharter/>.
25. SmartDraw home page; see <http://www.smartdraw.com/>.
26. J. A. Zachman, "A Framework for Information Systems Architecture," *IBM Syst. J.* **26**, No. 3, 276–292 (1987).
27. Popkin Enterprise Architecture; see http://www.popkin.com/products/system_architect/enterprise_architecture.htm.
28. J. D. Mooney, "Portability and Reusability: Common Issues and Differences," *Proceedings of the ACM 23rd Annual Conference on Computer Science*, Nashville, TN, 1995, pp. 150–156.
29. W. Frakes and C. Terry, "Software Reuse: Metrics and Models," *ACM Computing Surv.* **28**, No. 2, 415–435 (1996).
30. M. Turnlund, "Distributed Development Lessons Learned," *ACM Queue* (issue on Distributed Development) **1**, No. 9, 26–31 (December/January 2003/2004).
31. K. Coar, "The Sun Never Sets on Distributed Development," *Ibid.*, pp. 32–39.
32. G. Joeris, "Change Management Needs Integrated Process and Configuration Management," *Proceedings of the 6th European Conference*, held jointly with the 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering, November 1997, Vol. 22, No. 6, pp. 125–141.
33. IBM Corporation, IBM Rational Unified Process home page; see <http://www-3.ibm.com/software/awdtools/rup/>.
34. RDF (Resource Definition Framework) Primer, W3C Proposed Recommendation, December 2003; see <http://www.w3.org/TR/2003/PR-rdf-primer-20031215/>.
35. S. Decker, P. Mitra, and S. Melnik, "Framework for the Semantic Web: An RDF Tutorial," *IEEE Internet Computing* **4**, No. 6, 68–73 (November/December 2003).
36. OWL Web Ontology Language Overview, W3C Proposed Recommendation, December 2003; see <http://www.w3.org/TR/2003/PR-owl-features-20031215/>.
37. XSLTC home page; see <http://xml.apache.org/xalan-j/xslt/>.
38. S. Sendall and W. Kozaczynski, "Model Transformation: The Heart and Soul of Model-Driven Software Development," *IEEE Software* **20**, No. 5, 42–45 (September/October 2003).
39. Concurrent Versions System home page; see <http://www.cvshome.org/>.

Received December 31, 2003; accepted for publication February 25, 2004; Internet publication September 16, 2004

Jun Zhu IBM Research Division, IBM China Research Laboratory, HaoHai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (zhujun@cn.ibm.com). Mr. Zhu is a Research Staff Member on the Business Integration team at the IBM China Research Laboratory (CRL). He received his B.S. and M.S. degrees in computer science from Shanghai JiaoTong University in Shanghai, China, in 1998 and 2001, respectively. He subsequently joined the IBM Research Division in Beijing, China, where he began working on dynamic e-business technologies and enabling systems. Since 2002, Mr. Zhu has focused his attention on business process modeling and integration technologies, in particular the applicability of model-driven technologies to solve business integration problems. He contributed significantly to the concepts and design of the model-driven business integration platform, Model Blue, specifically Business View modeling and end-to-end solution methodology. Mr. Zhu was also one of the leaders of the CRL technical team to engage Model Blue with a customer—Bank SinoPac—in 2003.

Zhong Tian IBM Research Division, IBM China Research Laboratory, HaoHai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (tianz@cn.ibm.com). Dr. Tian joined the IBM China Research Laboratory after receiving his Ph.D. degree from Fudan University, Shanghai, China, in 1995. Since then, he has worked in a number of areas, including Digital Library, B2B eCommerce, e-Procurement, dynamic e-Business, and model-driven business integration. He was among the first pioneers to advocate Digital Library in China. He was one of the key Research architects of the WCS Marketplace edition. By working with IBM Global Procurement, Dr. Tian and his team helped transform two major IBM production sites into e-Businesses. Most recently, he led a joint team from Research and BCS to successfully deliver a model-driven business integration platform for Bank SinoPac.

Thomas Li IBM Research Division, IBM China Research Laboratory, HaoHai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (thomasli@cn.ibm.com). Dr. Li is the Deputy Director and CTO of the IBM China Research Laboratory. He holds a B.S. degree in electrical engineering, an M.S. degree in computer science, and a Ph.D. degree in management information systems. His business-oriented knowledge and experience have been instrumental in formulating the direction and strategy of numerous organizations in both Asia and the U.S. Dr. Li has more than twenty years of experience in the computer and Internet industry. During the past five years, he has been one of the major drivers in several Internet initiatives in the U.S. and Asia. These multinational projects include the areas of proactive and personalized customer services, a multiregional on-line shopping site, and intelligent multichannel large-volume information delivery. His technical expertise and innovative thinking have led to 39 patents in object technology, digital communication, visualization tools, and database management systems, and he is currently working on better managing and modeling strategic business change. In addition to publications in journals, proceedings, and technical bulletins, he has been one of the key contributors in delivering eight software commercial products, three hardware systems, and a number of architectural designs and technical specifications.

Wei Sun IBM Research Division, IBM China Research Laboratory, HaoHai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (weisun@cn.ibm.com). Dr. Sun is a Research Staff Member on the Business Integration team at the IBM China Research Laboratory (CRL). He received his Ph.D. degree in electronic engineering from the Beijing Institute of Technology in 2001. He subsequently joined the IBM Research Division in Beijing, China, where he began working on B2B integration platform technologies. Data extraction and transformation, processing logic runtime, and design time support have been his research focus. Dr. Sun contributed significantly to the concepts and design of Model Blue with respect to the IT View modeling and execution framework. He was also a leader of the CRL technical team to engage Model Blue with a customer—Bank SinoPac—in 2003.

Sheng Ye IBM Research Division, IBM China Research Laboratory, HaoHai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (yesheng@cn.ibm.com). Mr. Ye is a Research Staff Member on the Business Integration Methodology team at the IBM China Research Laboratory (CRL). He received his B.S. and M.S. degrees in computer science from Xi'an JiaoTong University in Xi'an, China in 1997 and 2000, respectively. He subsequently joined the IBM Research Division in Beijing, China, where he began working on e-procurement and data format transformation technologies. Since 2002, Mr. Ye has focused on lightweight business process runtime engines and tooling. He was one of the major designers of Model Blue for the IT View modeling, including the runtime engine and tooling. Mr. Ye was the major contributor to the integrated debugging environment for the Model Blue IT View. He was also a member of the key CRL technical team to engage Model Blue with a customer—Bank SinoPac—in 2003.

Wei Ding IBM Research Division, IBM China Research Laboratory, HaoHai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (dingw@cn.ibm.com). Dr. Ding is a Research Staff Member in the On Demand Technologies Department at the IBM China Research Laboratory (CRL). She received her Ph.D. degree in management science from Harbin Institute of Technology in 2001, subsequently joining the IBM Research Division in Beijing, China, where she began working on dynamic e-business technologies and enabling systems. From 2001 to 2002, Dr. Ding focused her attention on business process modeling and integration technologies, and in particular, the applicability of model-driven technologies to solve business integration problems. She was the major designer for a conflict-detection algorithm based on formalized mathematical foundations, which is a key function for Model Blue. Dr. Ding was a member of the key CRL technical team to engage Model Blue with a customer—Bank SinoPac—in 2003.

C. C. Wang Information Technology Division, Bank SinoPac, 12F, 66, Lane 9, Shung Shan Road, Taipei 105, Taiwan (c.c.wang@sinopac.com). Mr. Wang is a Vice President in charge of overall e-commerce software system development at Bank SinoPac. He received his B.S. degree in computer science from Tamkang University in Taipei, Taiwan, in 1985. He maintained and developed a banking mainframe (Unisys) online system for the Land Bank of Taiwan from 1986 to 1991. In 1991 he joined the Information Technology

Division of Bank SinoPac in Taipei, Taiwan, where he began working on administering and planning a banking mainframe (Unisys) system; he was also the leader of the banking mainframe software team responsible for online system design, development, and maintenance. Since 2000, Mr. Wang's assignment has been related e-commerce software systems, in particular, the applicability of Internet technologies to fulfill the business requirements of the Bank SinoPac Corporate Banking Division and Retail Banking Division. Mr. Wang managed all of the e-commerce software teams in the Information Technology Division of Bank SinoPac. These teams are responsible for the design, development, and maintenance of an EAI system, a B2B Web site, and a B2C Web site. He was also the Bank SinoPac project manager for the CRL/Bank SinoPac Model Blue engagement in 2003.

Gerland Wu *Information Technology Division, Bank SinoPac, 12F, 66, Lane 9, Shung Shan Road, Taipei 105, Taiwan (cl.wu@sinopac.com)*. Mr. Wu is an Assistant Vice President in charge of EAI software system development at Bank SinoPac. He received his M.S. degree from National Tsing Hua University, Taiwan, in 1990 and his B.S. degree from Fu Jen University, Taiwan, in 1988, both in applied mathematics. He subsequently worked for the Institute for Information Industry in Taiwan from 1990 to 1998 as a software programmer in a financial team and participated in the development of the Transactional Processing Monitor middleware of OnLine Transaction Processing (OLTP) on a UNIX platform. In 1999 he joined the Information Technology Division of Bank SinoPac in Taipei, Taiwan, where he began work maintaining Internet banking and WAP mobile banking. Since 2000, Mr. Wu's assignment has been B2C-related e-commerce system development—in particular, providing EAI middleware to fulfill the Bank SinoPac e-commerce requirements. Mr. Wu leads the EAI application team in the Information Technology Division of Bank SinoPac. His team is responsible for the design, development, and maintenance of EAI middleware to support MMAB2B and MMAB2C Web sites and integrate back-end legacy systems. In addition, he was one of the members who proposed the technical requirements for the CRL/Bank SinoPac Model Blue engagement in 2003.

Leo Weng *Information Technology Division, Bank SinoPac, 10F, 66, Lane 9, Shung Shan Road, Taipei 105, Taiwan (leo.weng@sinopac.com)*. Mr. Weng is an Assistant Vice President in charge of B2B e-commerce software system development at Bank SinoPac. He received his M.B.A. degree in management information systems from Simon Fraser University in Vancouver, Canada, in 1996. From 1992 to 1994 he worked for Unisys Taiwan as a software programmer. He joined the Information Technology Division of Bank SinoPac in Taipei, Taiwan, in 1996 to work on maintaining a banking mainframe batch system. Since 1998, Mr. Weng's assignment has been B2B-related e-commerce system development—in particular, applying Internet technology to fulfill the business requirements of the Bank SinoPac Corporate Banking Division. Mr. Weng was the leader of the B2B application team in the Information Technology Division of Bank SinoPac. His team is responsible for the design, development, and maintenance of the Bank SinoPac MMAB2B Web site. In addition, he also acted as the Bank SinoPac corporate customers' Internet portal for the end product of the CRL/Bank SinoPac Model Blue engagement in 2003.

Simon Huang *Application Innovation, Business Consulting Services, 3F, No. 7, Song-Ren Road, Taipei 110, Taiwan (simonh@tw.ibm.com)*. Mr. Huang is the EAI solution offering manager of IBM Taiwan. He received his B.S. degree in computer science from Soochow University in 1988 and his M.S. degree in computer science from National Tsing-Hua University in 1990. He joined IBM Taiwan in 1996 and has focused on EAI and workflow integration solutions since 1998. During these years, he led the team which successfully implemented more than ten EAI/workflow-related projects in Taiwan, including the Bank SinoPac engagement.

Bill Liu *Application Innovation, Business Consulting Services, 3F, No. 7, Song-Ren Road, Taipei 110, Taiwan (liubill@tw.ibm.com)*. Mr. Liu is the Project Manager of IBM Taiwan, and is a certified Project Management Professional of the Project Management Institute. He joined IBM Taiwan in 1997 and has focused on e-business solutions since 1998. During these years, he led the team which successfully implemented several e-business projects in Taiwan, including the Bank SinoPac engagement.

Daniel Chou *Application Innovation, Business Consulting Services, 3F, No. 7, Song-Ren Road, Taipei 110, Taiwan (danichou@tw.ibm.com)*. Mr. Chou is the EAI solution architect of IBM Taiwan. He received his B.S. and M.S. degrees in computer science from National Chiao Tung University in 1993 and 1995, respectively. He joined IBM Taiwan in 1999 and has focused on EAI and workflow integration solutions since 2001. During these years, he designed and successfully implemented more than five EAI/workflow-related projects in Taiwan, including the Bank SinoPac engagement.