



IBM Software Group

WebSphere Business Modeler V6.2

Model synchronization



@business on demand.

© 2009 IBM Corporation
Updated April 20, 2009

This presentation covers the features in WebSphere Business Modeler that support modeling for deployment. Focus is given to the tools available for keeping the business analysis model synchronized with the deployed implementation model.

Goal

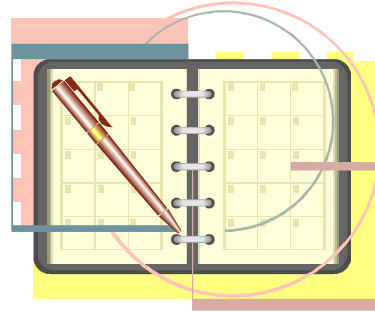
- To introduce you to features available in *WebSphere Business Modeler* and *WebSphere Integration Developer* for managing iterative development when modeling for deployment
 - ▶ Deployment to WebSphere Process Server



When modeling for deployment there are several possible runtime target environments. This presentation focuses on features that are available when the target environment is WebSphere Process Server, using WebSphere Integration Developer as the implementation tool.

Agenda

- Business driven development
- Separation of concerns
- Export
 - ▶ WebSphere Integration Developer
- Import
 - ▶ WebSphere Integration Developer
- Compare and merge
 - ▶ WebSphere Integration Developer



As shown in the agenda, this presentation first discusses what is meant by business driven development and continues with some key concepts and instruction for use.

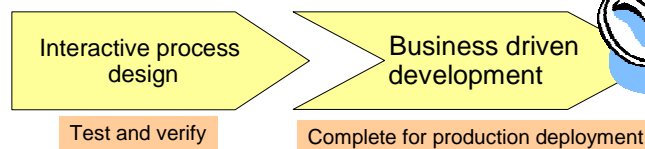
Introduction

▪ New with WebSphere Business Modeler V6.2

▶ Interactive process design

- Business user can deploy and test business model directly
- Best suited for human centric business processes
- Does not preclude the more formal *model driven application development* process presented here. (Business driven development)
- Interactive process design is discussed in a separate presentation

▪ Model synchronization is a facet of business driven development.



With WebSphere Business Modeler V6.2 there is a new feature that enables the business user to deploy and test their business process directly in a managed deployment environment. This new feature, called 'interactive process design', provides a way to quickly test and verify a business model without involving the IT organizations. With the interactive process design the business user can quickly iterate over several possible scenarios before handing the model over to the implementation team.

For iterative development over time, in a production environment, it is still necessary to complete the implementation model in WebSphere Integration Developer and merge in the new changes from the business model. This process is called 'business driven development'.

These two approaches to business modeling are not mutually exclusive. They can be used together, using the interactive design to verify and test the model and then passing it on to the implementation developers to complete the implementation for production.

The focus of this presentation is on business driven development, with emphasis on the model synchronization feature.

Managing business driven development

- Business analyst constructs business model using WebSphere Business Modeler
- Integration developer completes the technical attributes in WebSphere Business Modeler and then exports the model as a project interchange file
 - ▶ to be imported into WebSphere Integration Developer
- In WebSphere Integration Developer the implementation is developed, tested and deployed.
- Meanwhile,
 - ▶ the business analyst makes new changes to the business model and
 - ▶ the integration developer makes changes in the implementation that can impact the business model.
- The business model and the implementation models quickly become out of synch



With business driven development, the business requirements drive the development of the applications to support the needs of the business. Business modeling captures the business requirements and the relationships between the entities within the organization. When a modeling tool such as WebSphere Business Modeler is used, the development process is called model driven business development.

With model driven development, an abstract model of the business processes is created. Then the model is passed on to a tool like WebSphere Integration Developer, which is then used to implement the model for a specific runtime environment. With WebSphere Integration Developer, the target runtime environment is the WebSphere Process Server.

With this approach, there are two models of the business process; one is the business model, created in WebSphere Business Modeler, and the other is the implementation model, created in WebSphere Integration Developer. The implementation model is described in WebSphere Integration Developer using the Business Process Execution Language, or BPEL.

During normal iterative development, keeping the two models from diverging as the separate teams make changes and updates can become a serious problem. Quite often once development of the implementation is started, no attempt is made to keep the models synchronized.

With the release of V6.1 for WebSphere Business Modeler and WebSphere Integration Developer, there are now tools available that will allow the separate development teams to keep their models synchronized.

Three pronged approach

- Separation of concerns
 - ▶ Separate the business logic from the implementation of the services.
 - Achieved by using the 'Recommended Export Option'.
- Model synchronization
 - ▶ Use the model synchronization wizard to **merge** in new changes when importing project interchanges files that originate from WebSphere Business Modeler V6.1 or greater
 - From WebSphere Business Modeler → WebSphere Integration Developer
- Change impact analysis
 - ▶ Provide a change report from WebSphere Integration Developer.
 - ▶ Used by WebSphere Business Modeler to analyze the changes that were made in the implementation model.
 - From WebSphere Integration Developer → WebSphere Business Modeler
 - A *delta* file.

6

Model synchronization

© 2009 IBM Corporation

To facilitate iterative development and model synchronization between the business and implementation models, three areas need to be considered.

First is the concept of separation of concerns. If you are developing the implementation model in WebSphere Integration Developer, it is good to separate the BPEL model from the implementations of the services that are invoked by the BPEL activities. By keeping these in separate projects, the BPEL model can be updated without affecting the service implementations.

Next, when importing subsequent iterations of the model into WebSphere Integration Developer, the new changes can be **merged** into the existing BPEL model without overwriting the existing model elements or their implementations.

Once the models have been synchronized, any changes that were made in WebSphere Integration Developer that might impact the development of the business model must be communicated back to the business analyst in WebSphere Business Modeler. This is achieved by creating a change report that can then be imported into WebSphere Business Modeler and compared against the business model.

All three of these features are fully supported beginning with WebSphere Business Modeler and WebSphere Integration Developer V6.1.

Separation of concerns

- As a best practice the business logic should be separated from the implementation.
 - ▶ Provides a flexible architecture for updating and maintaining the models.
 - ▶ Prevents overwriting of service implementations.
- Supported by the WebSphere Business Modeler export type for exporting to WebSphere Integration Developer



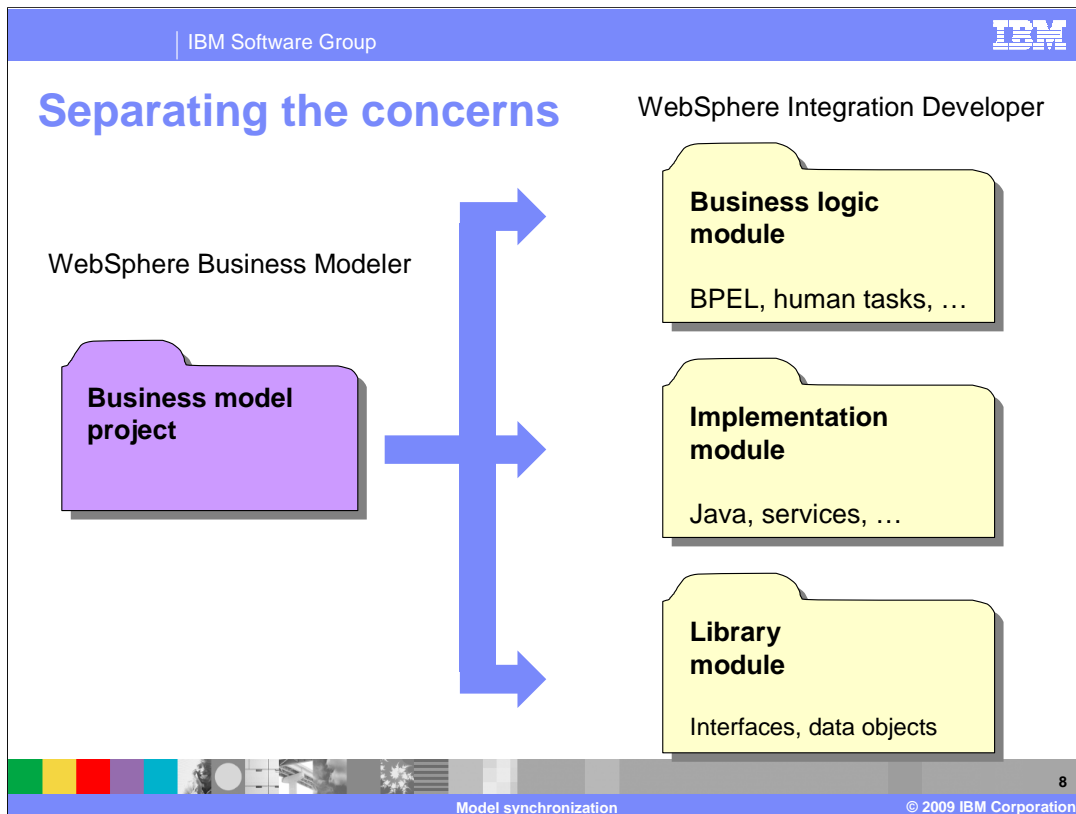
Separating the BPEL implementation model from the implementations of the services it invokes is a key part of the strategy for keeping the business model synchronized with the implementation model.

Having the model and the implementation of the services in separate projects makes it clear where the integration developer should be working, that is, in the implementation project. When there are updates to the business model that need to be merged in, then the compare and merge utility only needs to work with the model project.

Keeping the service implementations in separate projects enables the integration developer to switch to alternate implementations by changing the SCA imports in the assembly diagram of the model project. The service implementations can also be developed and tested independently of the BPEL model.

Separation of concerns is primarily an issue with WebSphere Integration Developer but it begins with the export feature of WebSphere Business Modeler V6.1.

The WebSphere Business Modeler export type as shown in the next few slides supports this practice



Shown here is an illustration of the best way to partition the implementation artifacts based on functionality.

For a given project in WebSphere Business Modeler, create three SCA modules in WebSphere Integration Developer.

The BPEL artifacts representing the business process are placed in the business logic module.

The implementations of the services that are invoked by the BPEL activities are placed in the implementation module.

And the interfaces and the data object definitions that are common to all the modules are placed in the library module.

The terminology around modules and projects can be a little confusing at this point.

WebSphere Business Modeler uses the project construct to contain related artifacts while in WebSphere Integration Developer the project is mapped to an SCA module.

Export to WebSphere Integration Developer

- Export patterns supported
 - ★ ▶ Recommended export option (three modules and projects)
 - ▶ Module project + library project
 - ▶ Library project
 - ▶ none

Select the export option

Recommended Export Option ▼

This option creates three related projects. It separates the business logic from the implementation details. Choose this option for best interoperability when you are working iteratively between WebSphere Business Modeler and WebSphere Integration Developer.

Export using the standard project interchange format for other environments

Project Interchange Name

CATOrderMgmt_Group

Append timestamp to project interchange name

Target Project Names

Modeler Project Name	Business Logic Module Name	Implementation Module Name	Library Name
CATOrderMgmt	CATOrderMgmt	CATOrderMgmt_impl	CATOrderMgmt_lib
Corporate elements	Corporateelements	Corporateelements_impl	Corporateelements_lib

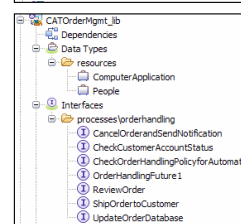
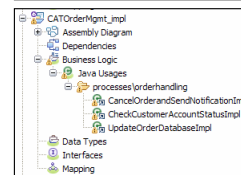
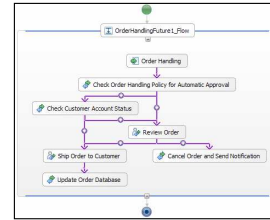
Model synchronization © 2009 IBM Corporation

When exporting a modeling project from WebSphere Business Modeler V6.1, using the WebSphere Integration Developer export type, there are four different options available. The recommended export option creates three related SCA modules. It separates the business logic from the implementation details. Choose this option for the best interoperability when you are working iteratively between WebSphere Business Modeler and WebSphere Integration Developer.

Notice the column headers on the target project names table. They are directly editable fields and you can change the names of the target modules if necessary.

The recommended export option

- Business logic module
 - ▶ MyProject (no suffix)
 - ▶ BPEL, business rules, human tasks, and so on
- Implementation module
 - ▶ MyProject_impl
 - ▶ Artifacts for which the implementations cannot be generated.
 - Services and implementation types are specified in the technical specifications
- Library module
 - ▶ MyProject_lib
 - ▶ Business items and common schema
- The implementations in the implementation module are linked to the business logic model through the SCA imports and exports
 - ▶ SCA imports and exports are automatically created in the assembly diagrams as part of the import process



Using the recommended export option creates three related SCA modules.

The business logic module contains the BPEL model, including elements such as the business rules and human tasks that are specified in WebSphere Business Modeler.

The implementation module holds the implementations for the services that are invoked from the BPEL activities.

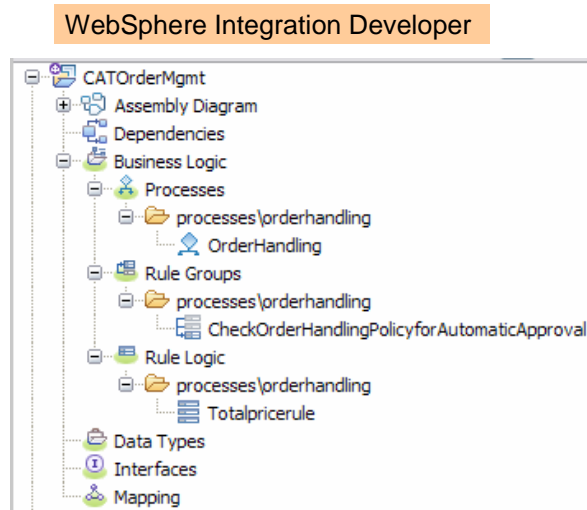
If the implementation type in the technical attributes of the business model in Modeler is set to specific type, then a skeletal SCA component is created in the implementation module.

The library module contains the business object definitions, representing the business items, and the interface definitions.

With WebSphere Process Server, inter-module calls are made through SCA imports and exports, which are defined in the assembly diagram for each module. The exports, imports and the linkages are all defined and connected for you as part of importing the project interchange file.

Business logic module

- All of the artifacts that are part of the business model and can be mapped to a concrete implementation in WebSphere Integration Developer are placed in the *business logic module*.
 - ▶ example:
A human task, local or global
 - ▶ example:
A business rule, local or global
- Imports are defined in the assembly diagram to link the BPEL invokes to the implementations in one or more *implementation modules*.
 - ▶ The default binding is SCA.



The business logic module is the top-level module project and contains the BPEL model, including elements such as the business rules and human tasks. This module should not be edited by the integration developer unless it is absolutely necessary. Changes to the business model should be made using WebSphere Business Modeler and then merged into the business logic module using the model synchronization wizard. In the event that changes must be made to the module project in WebSphere Integration Developer, the change report can be used to communicate those changes back to the business analyst.

By default, the business logic module is created without a suffix for the project name. When using WebSphere Business Modeler and WebSphere Integration Developer in shell sharing mode, there might be a name conflict with the business logic module. This can be fixed by manually adding a suffix to the module project when exporting from WebSphere Business Modeler.

When the module is created, the SCA assembly diagram is also created with the proper SCA imports and exports. The imports and exports are set up to link with the service implementations in the implementation module, using the SCA bindings by default. With this setup, on subsequent iterations the new changes to the business model can be merged into the business logic module without impacting the implementations in the implementation module. At most, the integration developer might have to adjust some of the links, depending on the kind of changes that were made in the business model.

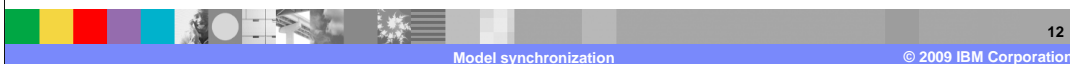
Implementation module

- *moduleName_impl*
- All of the tasks that cannot be mapped to a concrete BPEL activity in the *business logic module*, will go into the *implementation module*.
 - ▶ A skeleton implementation is created in the implementation module with the proper exports
 - Based on the implementation type in the technical attributes
 - ▶ Integration developers can elaborate on the implementations in the implementation module without impacting the business logic model.

Technical attributes in Modeler

Implementation type

Java
State Machine
Rule Group
Java
Process
Import - Web Service binding



The implementation module contains the implementations for the services that are invoked from the BPEL activities.

By default, the implementation module name is appended with “_impl”. This can be changed when exporting it from WebSphere Business Modeler if necessary.

If the implementation type in the technical attributes of the business model in WebSphere Business Modeler is set to a specific type, then a skeletal SCA component is created in the implementation module. Otherwise, except for the assembly diagram and the exports and imports, this module is empty.

This is where the integration developer works, creating the service implementations.

The other export options

- There are several other export options that are available to assist in the case where you are working with existing WebSphere Integration Developer projects.
 - ▶ Module project + library project
 - ▶ Library project
 - ▶ None



There are additional export options available to meet different needs.

The Module Project + Library Project option creates a module project containing the implementations, and a library project containing the business objects and interfaces that can be shared. Choose this option if you do not need to separate business logic from implementation details, for example if all the tasks are invoking existing Web services.

The Library Project option creates a library containing only the business objects and interfaces. Choose this option if you have updated the business objects or interfaces and want to refresh the XSD and WSDL file that were previously exported.

The None option exports everything to BPEL, WSDL, and XSD files in the specified directory. No project is created. Project Interchange is not an option.

Model synchronization through compare and merge

- Allows the integration developer to selectively update the...
 - ▶ business models in the business logic module
 - ▶ assembly diagram
 - ▶ add new skeletons in the implementation module
- The updated version of the business model is *compared and merged* with the existing artifacts in WebSphere Integration Developer
 - ▶ The new model is input as a *project interchange file*.
- Graphical wizard guides the integration developer
- Changes, additions, deletions and conflicts can be reviewed and then either accepted, rejected or resolved.



Some of the features discussed in this presentation are part of WebSphere Business Modeler and some are part of WebSphere Integration Developer.

You start with Modeler and export the model as a project interchange file.

Next, the project interchange file is imported into WebSphere Integration Developer.

Importing a project interchange file will create the business logic, implementation and library modules for you or it will overwrite them if they already exist.

This establishes the baseline for the subsequent iterations.

The business analyst can then continue developing the next iteration of the business model while the integration developer begins development on the implementation of the supporting services.

When the business analyst has completed the next iteration of the business model, it is again exported as a project interchange file.

This time the integration developer does not import the file, but instead uses the project interchange file as input to the model synchronization wizard.

The model synchronization wizard is a graphical utility that will allow the integration developer to inspect all of the new changes and either accept or reject them.

If there are any differences between the business model and the implementation model after the changes have been committed, a change report is generated.

Recommendations for iterative development

▪ In WebSphere Integration Developer

▶ Minimize changes in business logic module

- Changes made to the BPEL in the business logic module cannot be analyzed if there are also changes in the business model during the same iteration.

▶ Minimize changes in library module

- `_lib`
- Changes made here will need to be exported and imported manually to WebSphere Business Modeler



▶ Focus on changes to the implementation module.

- `_impl`

▶ Communicate with the business analyst doing the business modeling.

▶ Use the latest Modeler project interchange file possible.



Keep in mind that this is top down, model driven development.

The business logic module represents the business model. Because the elements in the business logic module map directly to elements in WebSphere Business Modeler, the changes should come from WebSphere Business Modeler. Every effort should be made to minimize structural changes to the business processes in the business logic module.

The library module contains the definitions for the interfaces and data elements. In the WebSphere Integration Developer library module these elements are defined using XML Schema and Web service description language, or WSDL. There is no equivalent in WebSphere Business Modeler.

Structural changes, such as adding inputs or outputs or changing their types, should be done in WebSphere Business Modeler rather than WebSphere Integration Developer. Minimize structural changes to interfaces and business objects in the library module.

When working in WebSphere Integration Developer, focus on changes in the implementation module. The elements in the implementation module do not map directly to the business model.

For example, a task in WebSphere Business Modeler is a black box with no internal structure. When a task is exported to WebSphere Integration Developer, the SCA component that is generated has no concrete implementation. The integration developer in WebSphere Integration Developer must implement the task. The implementation goes into the implementation module.

Because changes in both WebSphere Business Modeler and in WebSphere Integration Developer can affect the business semantics, communication between the business analyst and the integration developer is always important.

Always use the latest project interchange file from Modeler, particularly if you know you made changes in the business logic or library modules. Using the latest export will avoid the situation where there are ambiguities in the Modeler change analysis report that cannot be resolved in Modeler.

Analyze model implementation changes

- The result of the model synchronization session is captured in a change report if the integration developer ...
 - ▶ made changes to the business logic or library modules
 - ▶ rejected changes from the new iteration.
- The change report is then imported into WebSphere Business Modeler
 - ▶ Analyzed by the business analyst to determine
 - if any of the implementation 'changes' impact the business model.
 - ▶ If so, then the business analyst can manually accept and then update the model or
 - ▶ Reject and ignore the changes.



The results of the model synchronization are captured in a change report, but the change report is only generated if there are differences between the two models being compared.

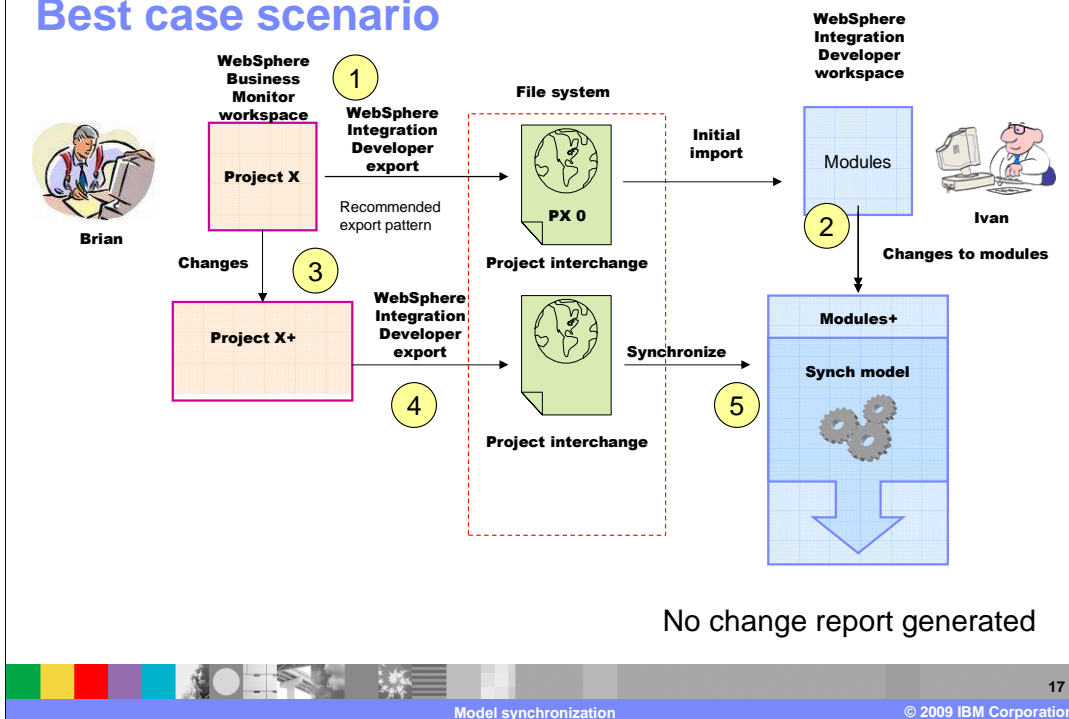
Differences can arise if the integration developer made some change to the business logic or the library module or if they rejected a change coming from the new iteration of the business model.

If there are no changes that need to be communicated back to the business analyst, then no change report is generated.

Changes in the business logic or library modules are the kind of changes that need to be reported back to the business analyst.

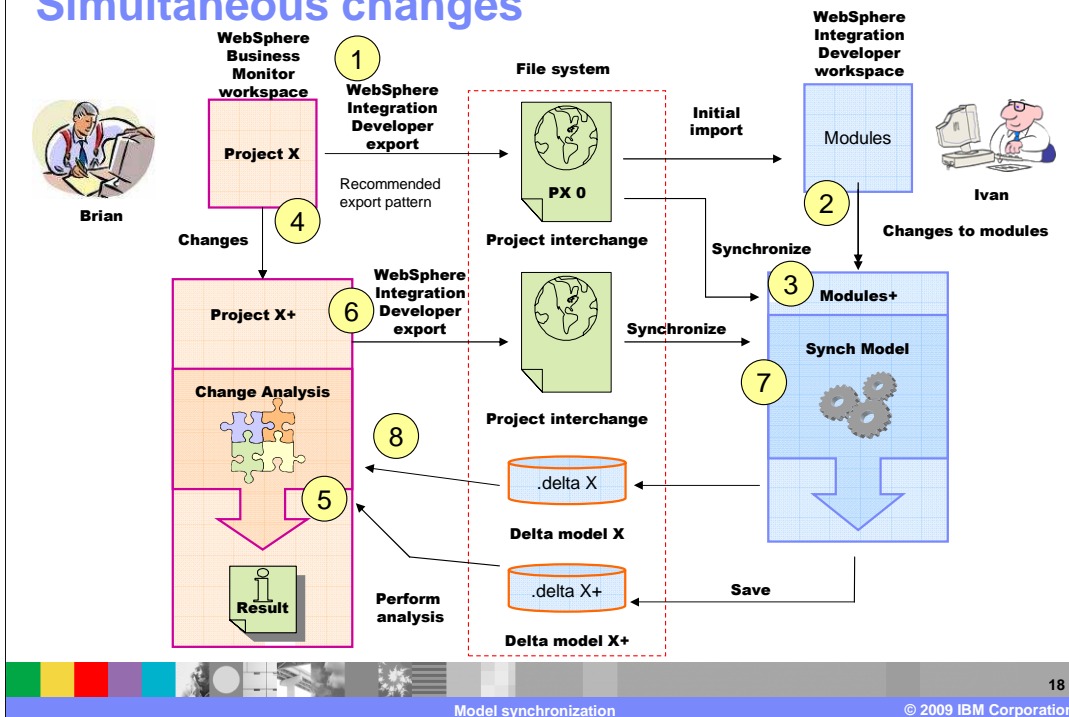
The change report can then be used by the business analyst in WebSphere Business Modeler to identify areas in the business model that might need to be updated.

Best case scenario



1. In this scenario, Brian the business analyst creates the initial baseline model and gives it to Ivan, the integration developer.
2. Ivan starts working on the implementation. He does not make any changes in the business logic and library modules.
3. Meanwhile, Brian has made some changes to the business model.
4. Brian exports the model as a project interchange file.
5. Ivan uses the model synchronization feature to merge in the changes.
6. Because Ivan is not making changes to the business logic module or the library module, there is no change report generated. The models are synchronized and everyone is happy.

Simultaneous changes

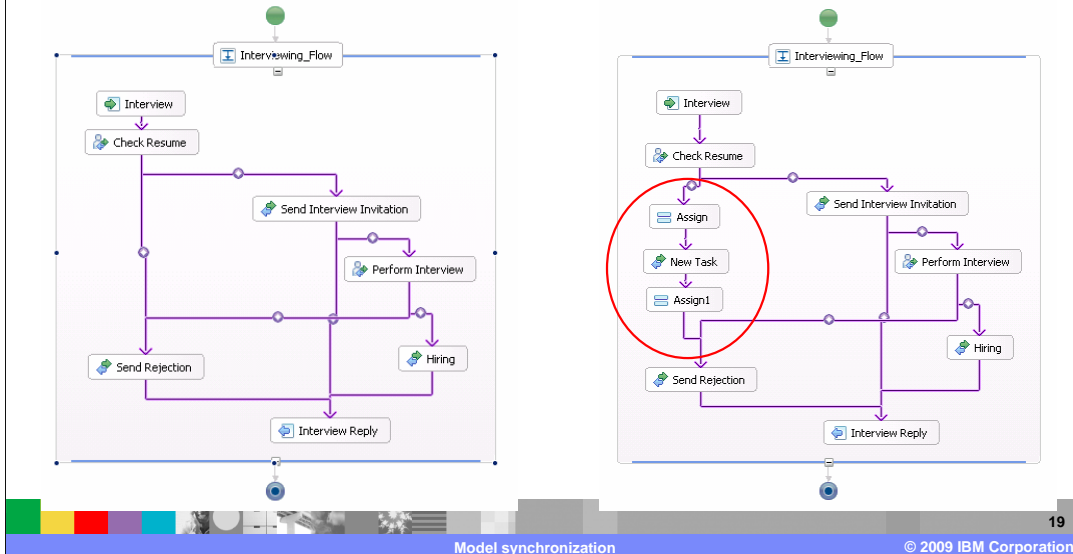


1. In this scenario, Brian creates the initial baseline model and gives it to Ivan.
2. Ivan immediately notices some things that require changes in the business logic and library modules.
3. Ivan makes the necessary changes and then creates the change report by synchronizing with the initial baseline model. This will provide Brian the information he needs to correct the business model based on Ivan's feedback.
4. Notice that while Ivan was making changes in WebSphere Integration Developer, Brian was making changes to the business model in WebSphere Business Modeler.
5. When Brian runs the change analysis in WebSphere Business Modeler, he is notified that there have been changes in the same area that he has been working in but he is not able to see those changes.
6. Brian will have to cancel the analysis, export his current work, and request a new change report from Ivan.
7. Ivan will then run the model synchronization again, and merge in the latest changes to the business model and generate a new report for Brian.
8. Brian can now run the change analysis again, this time he can see Ivan's changes, provided he did not make any new changes to the business process.

Note that Ivan could have prevented the extra iteration by requesting a new export file from Brian before starting to do the synchronization.

Changes to the implementation model - example

- Changes made in WebSphere Integration Developer relative to baseline imported from WebSphere Business Modeler



Model synchronization

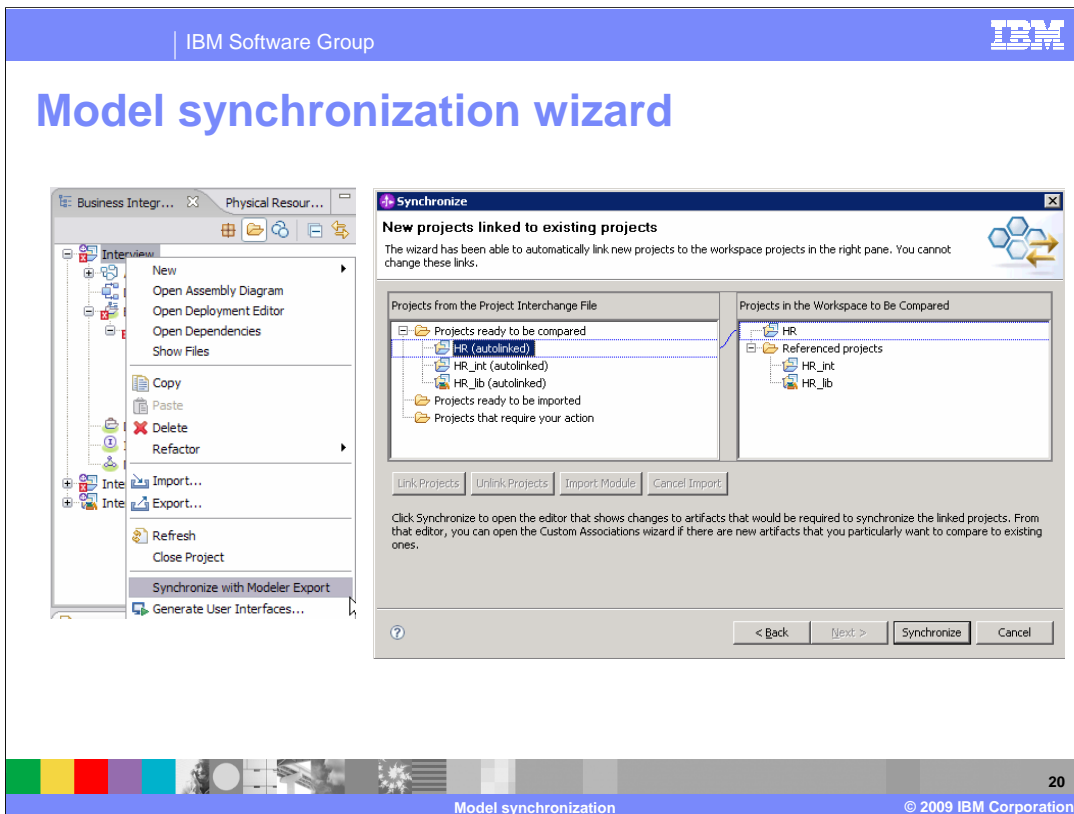
© 2009 IBM Corporation

19

Here is an example of how the synchronization wizard and change analysis tools work.

Shown here on the left is the BPEL representation of the business model as it was initially received by Ivan.

Ivan inserted a new task and had to add a couple of BPEL assign activities to help with the data manipulation.



After making the changes, Ivan invokes the synchronization wizard from the pop-up menu of the business logic module.

Ivan is prompted for the input file, which in this case is the project interchange file that was initially imported, and then sees the dialog on the right.

Note: Ivan should have used the latest project interchange file from the business analyst, but he did not. It was this oversight that caused the rework in steps six and seven in the simultaneous development scenario that was previously discussed.

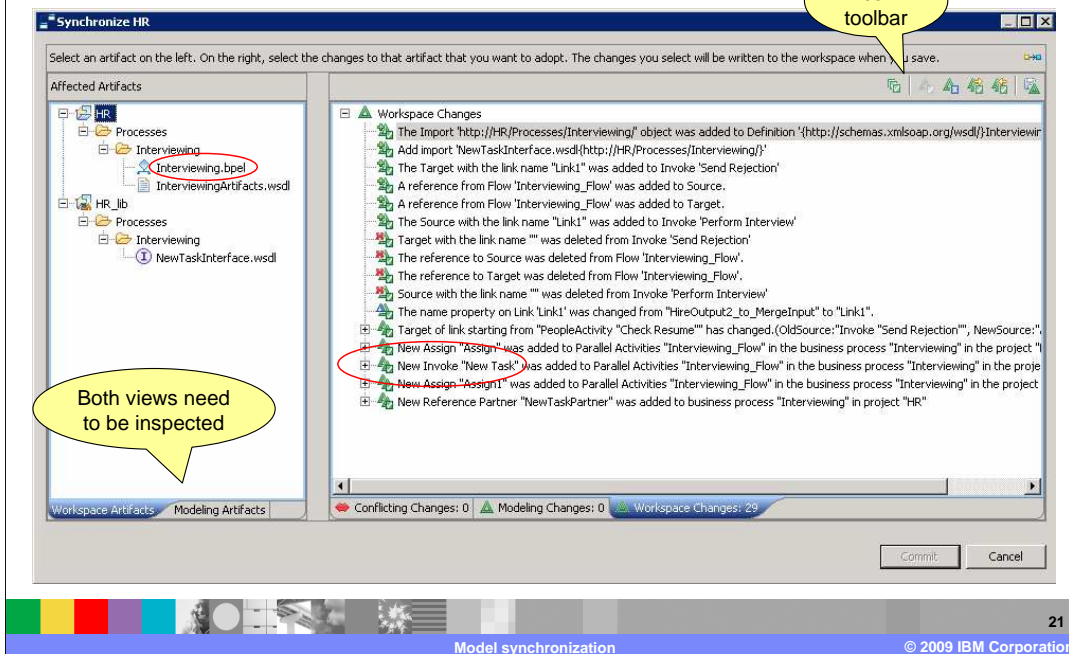
Here you can match up the projects that are in the project interchange file with the SCA modules in the workspace. The synchronization wizard attempts to match up the names and link them automatically when it finds a match. The linkage is indicated by the blue boxes connected with the blue line.

The names of the projects and modules might not match. This is most likely because the modules in the WebSphere Integration Developer workspace were created with a previous version, before this feature was available. In any event, the projects can be manually linked to the modules using the buttons in the center of the dialog. The buttons are not available in this example because all of the projects were linked automatically.

For more information about how to manage the manual linkage to legacy WebSphere Integration Developer modules, see the on-line help available with WebSphere Integration Developer V6.1 or greater.

Click Synchronize.

Results of model synchronization



Clicking synchronize will start the compare and merge; when it is completed you see this dialog.

First, notice the tabs at the bottom of the two panes. There are several views that you can use to understand the nature of the changes. On the left, you can choose to inspect the changes from the perspective of the WebSphere Integration Developer workspace or the WebSphere Business Modeler. It will depend on what you are most familiar with.

In this case, Ivan is doing the synchronization so he wants to view it from the workspace perspective first.

Both views should be inspected for a complete understanding.

The pane on the right has three different views, one for the modeling changes, another for the workspace changes and the third is for any conflicts. By looking at these tabs you can quickly assess the scope of the changes. For example, there are no new changes coming from the modeling side, there are no conflicts and there are 29 changes that were made in the workspace.

The information displayed in the right pane is determined by what is selected in the left pane. If you want to see only the changes related to the interviewing business process, then select the Interviewing.bpel element in the tree on the left.

If you want to know more about the changes related to the new task then you expand the New Invoke element in the tree on the right.

Notice the icons and the icon tool bar. The toolbar is one way that you can accept or reject the changes.

If a change is made and committed, it changes the workspace. In this case, since all of the changes are coming from the workspace, the tool assumes that you want to accept them, so they are all marked as accepted by default. This can be seen by the checkmark on the bottom right of the icon.

In other scenarios, you will need to look at the icons carefully and understand their significance.

Accept or reject, save and export

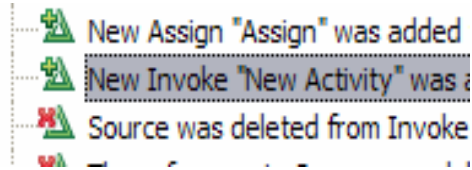
- Icons next to the changed element indicate if it was an

- ▶ Addition (+)

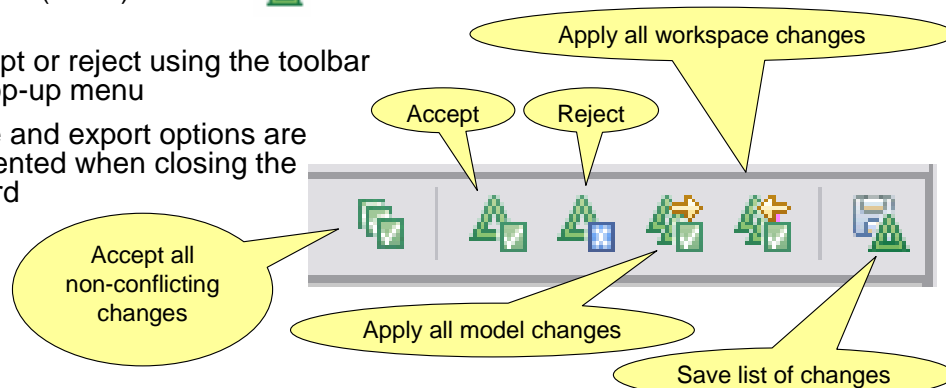
- ▶ Deletion (x)

- ▶ Change (triangle)

- ▶ Move (arrow)



- Accept or reject using the toolbar or pop-up menu
- Save and export options are presented when closing the wizard



Taking a closer look at the icons, you see that there is an adornment for each of the possible cases: add, delete, change, and move.

The adornment on the upper left indicates the kind of change detected. If the change is accepted then there is a checkmark attached to the lower right and if the change is rejected, then there is an x on the lower right.

There are accelerator buttons on the toolbar for accepting all non-conflicting changes, all changes from the model, or all changes from the workspace.

There is also a button to save a list of all the changes.

Changes can also be selectively marked by using the pop-up menu for the individual item.

It gets interesting when you start to selectively apply and reject the changes. You should think carefully before committing the changes.

On line help is helpful

The screenshot shows the IBM WebSphere Information Center interface. The top navigation bar includes the IBM logo and the text 'IBM Software Group'. Below this, the page title 'On line help is helpful' is displayed. The main content area is titled 'WebSphere Business Modeler Advanced' and features a search bar and a 'GO' button. A red circle highlights the search bar area. The left sidebar contains a 'Contents' section with a tree view of topics under 'WebSphere Business Modeler Advanced'. The main content area displays the article 'Analyzing changes from WebSphere Integration Developer', which includes an introduction, a section on 'Recommendations for iterative development', and sections on 'Analyzing model implementation changes', 'Resuming change analysis', and 'Deleting change analysis versions'. The footer of the page includes a decorative bar with the text 'Model synchronization' and '© 2009 IBM Corporation'.

Shown here is the section of the on-line help where you can find detailed information about analyzing the changes coming from WebSphere Integration Developer.

You should locate these topics and bookmark them.

Summary

- Powerful new features for managing and synchronizing business models with implementation models
- Three pronged approach
 - ▶ Separation of concerns
 - Supported by the export best practice
 - ▶ Model synchronization
 - Mangle business model changes that effect the implementation model
 - ▶ Change impact analysis
 - Feedback to the business analysis



Traceability between the business model created in WebSphere Business Modeler and the implementation model created in the WebSphere Integration Developer business logic module is greatly improved.

There is now direct support for iterative model driven development, using WebSphere Business Modeler and WebSphere Integration Developer.

Starting with the export in WebSphere Business Modeler, the business model is partitioned into multiple implementation modules based on function.

After the initial iteration, subsequent iterations of the business model can be selectively merged into the business logic module in WebSphere Integration Developer.

These features are complimented on the Modeler side with the change analysis wizard, which enables the business analyst to be notified of changes in the implementation that effect the business model. The business analyst can then incorporate the changes or reject them.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv62_Modeler_Overview.ppt

This module is also available in PDF format at: ..\WBPMv62_Modeler_Overview.pdf



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Lotus Rational WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

Adobe, are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Internet Explorer, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

