



SELECT

B U S I N E S S S O L U T I O N S

Select Business Solutions

Supporting the OMG Model Driven Architecture® (MDA®) For
Service and Component Based Development



OBJECT MANAGEMENT GROUP

Disclaimer and Trademarks

© Select Business Solutions, Inc. 2004. All Rights Reserved.

Information in this document is subject to change without notice and does not represent a commitment on the part of Select Business Solutions, Inc. to provide or continue providing said functionality. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without prior written consent of Select Business Solutions, Inc. Company names, data and other information appearing in examples are fictitious in nature unless otherwise designated.

The software described in this document is furnished under license or non-disclosure agreement and may be used or copied only in accordance with the terms and conditions of that agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement.

Select Enterprise is a Registered Trademark of Select Business Solutions, Inc., and Select Solution for MDA®, Select Component Factory, Select Component Architect, Select Component Manager, Select Component Portal, Select Solution for MDA, Select JSync, Select CSync, Select C#Sync, Select ForteSync, Select VBSync, Select DBSync, Select XMLSync, Select ORSync, Select Document Generator, Select Object Animator, Reviewer for Select Component Architect, Reviewer for Select Enterprise, Reviewer for Rose, Select Process Director, Select Process Director Plus, Select UDDIServer, Select Application Composer, Select Scope Manager, Select Perspective, Select SE, Select SSADM, Select Yourdon are all Trademarks of Select Business Solutions, Inc.

Because of the nature of the material, numerous hardware and software products are mentioned by their trade names in the publication. In most, if not all, cases these designations are claimed as trademarks by their respective companies. It is not the publisher's intent to use any of these names generically, and the reader is cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

For more information about Select Business Solutions Inc., or to download an electronic copy of this document please visit the Select Website: <http://www.selectbs.com/>

UML, Unified Modeling Language, OMG, Object Management Group, MDA®, Model Driven Architecture are all Trademarks or Registered Trademarks of the Object Management Group, Inc.

Table of Contents

Executive Summary	3
Introduction	3
Principles	3
The Unified Modeling Language (UML).....	4
Model Driven Architecture® (MDA®)	4
About Select Business Solutions	4
This Document	5
What is MDA?	6
Why Model - A Brief History?	8
Supporting MDA	9
Computation Independent Model.....	10
Mapping the CIM to the Initial Solution PIM	11
Solution Platform Independent Model	12
Defining Platform Transformations	14
Component/Service Platform Independent Model	14
Component/Service Platform Specific Model	14
Component/Service Implementation	16
Solution PIM	18
Solution PSM	18
Solution Implementation	19
Component Management	20
Conclusion	22

Executive Summary

Introduction

The Internet, company acquisition and a multitude of technologies are imposing system integration challenges for organizations; IT can no longer afford to deliver stovepipe systems (e.g. product focus or departmental focus) but rather collections of solutions that together support end-to-end process, even across organizational boundaries. This requires an architecture-centric development approach that ensures that IT solutions are resilient to change, easy to integrate with legacy functionality and deliver business benefit quickly.

Quite a daunting prospect but it is possible today by:

- Selecting a development process that emphasizes early and continual business alignment of the IT solution to the needs of the business
- Adopting a service-oriented architecture that enables the ‘loose coupling’ and strong encapsulation that are required to build flexible and integrated solutions
- Adopting standards and integration technologies such as XML and Web Services that provide the essential “glue” with which to build these solutions into an organization-spanning, consistent whole.

However, new technologies can cause loss of productivity as they are embraced due to the learning curve needed and hiring new staff cannot always offset this because of the skills shortage. This would seem to go against the continuing struggle to:

- Reduce costs throughout the application lifecycle
- Reduce development time for new applications
- Improve application quality
- Increase return on technology investments
- Rapidly include emerging technology benefits into existing systems

Principles

Of course one certainty is change; component and service integration technology continues to evolve - yesterdays DCOM is today’s Web Services and legacy systems may at some point be re-implemented to run on a different platform. So building to one standard or platform won’t necessarily protect our solutions from technology change.

How do we become agile and resilient to change? Here are some key principles:

- The ability to define a solution that satisfies the needs of the business as a collaborating set of services and or components
- The ability to define those services and components from a business rather than technology perspective
- The ability to separately define the technology specific mappings for middleware and platform by the skilled staff
- The ability to apply technology specific ‘mappings’ to services and components for a chosen platform by less skilled staff
- The ability to quickly re-apply new technology mappings to services and components
- The ability to re-use mappings to ensure consistency for easier maintenance later

And what does this mean for the organization? - Those that strategize and architect computer systems will need to ensure they have the enabling technology but equally importantly that they adopt a development process, including understandable documentation, which enables them to define their business solutions in a platform and technology neutral form. This will make them flexible and responsive to both business and technology change.

Visual modeling tools play a key role in the development of Service and Component based solutions; helping to elicit, define and communicate:

- The end to end business processes
- The business requirements
- The system requirements
- The specification of the Services and Components that will deliver the business functionality
- The design of Services and Components

The Unified Modeling Language (UML)

The UML is a standard published by the Object Management Group (OMG)¹, which has become the international standard for documenting the requirements and design of a system. UML can be used to define the specifics of that system or to define technology patterns that are to be used for implementing it. The UML also contains elements that may be used to describe system architecture.

Model Driven Architecture® (MDA®)²

The OMG has recognized that business language does not necessarily equate to the terminology imposed by technology and that common system design concepts are implemented in different ways when using different technologies i.e. there is a semantic gap between business, design and implementation that may cause problems during development and later maintenance and extension of a system.

The OMG's answer to this problem is Model Driven Architecture (MDA) in essence describes a series of distinct and linked models whose purpose is to abstract the business functionality of a solution from the technology independent design and the technology platform it is developed and deployed on.

About Select Business Solutions

Select Business Solutions (Select) is the market leader in the provision of Service and Component Based Development (SCBD) process (the Select Perspective) and Software Tools (Select Component Factory, Select Process Director and Select Solution for MDA) for the rapid development of business solutions assembled from components and services.

Just as Select recognized the value in combining Ivar Jacobson's and James Rumbaugh's work before UML, so we recognized the value in separating the business and technical viewpoints in Select Perspective before MDA was proposed and produced tools for the synchronization of models and code.

The Select Perspective

Select Perspective is a thought-leading, practical and proven Service and Component Based Development (SCBD) process and is based on over 100 effort-years of real experience in the delivery of business solutions in object-oriented (OO), component-based and service-oriented architectures.

It is delivered through Select Process Director, which enables users to amend and maintain the definition of the development process to suit their organizations' culture and working practices.

Select Component Factory

Select Component Factory is the suite of tools for the visual modeling, component management and code synchronization that help to implement Select Perspective. It provides support for the UML and MDA as well as specific support for Business Process Modeling (BPM) and data modeling not covered by the UML.

The Select Component Factory suite of tools comprises:

- Select Component Architect; an industry leading visual modeling tool for BPM and UML modeling
- Select Component Manager; providing Component and Service publication, cataloging and management facilities
- Reviewer for Select Component Architect; providing fast quality checking and correction of Select Component Architect models
- Select code, XML and database synchronizers; providing full round-trip synchronization between models and implementation artifacts

In addition Select Solution for MDA is available which offers full round-trip MDA support

¹ <http://www.omg.org/>

² <http://www.omg.org/mda/>

This Document

This document will show how the principles of Select Perspective and facilities offered by Select Component Factory provide practical support for MDA; in particular this document will describe support for:

- The abstraction and refinement of models
- The explicit linkage and trace between different types of model
- The modeling transformation required when targeting business functionality at different platforms
- Providing the benefits available from the use of MDA

Audience

The audience for this white paper comprises the managers, leaders and strategists of IT departments and System Integrators who are considering or are in the process of adopting SCBD and who wish to understand how this approach is used in context with MDA. The audience is assumed to have an appreciation of SCBD and the UML.

What is MDA?

MDA concerns itself with development productivity (which gives short-term benefits) and preserving the long-term investment in application and component development against the all too frequently changing face of technology.

The MDA approach defines first the Business domain independent of any computer system; it then defines an approach to system development that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform. To this end, the MDA defines a modeling architecture and provides a set of guidelines for structuring specifications expressed as models.

The MDA approach and the standards that support it allow the same model that specifies system functionality to be realized on multiple platforms through auxiliary mapping standards, or through point mappings to specific platforms, and allows different applications to be integrated by explicitly relating their models, enabling integration and interoperability and supporting system evolution as platform technologies come and go. The key concepts are:

- Models, which are used to document and communicate designs and architectures
- Abstraction, Refinement and Viewpoint. These are very important points and are generally overlooked.
 - Abstraction refers to the suppression of irrelevant information
 - Refinement is used to relate to lower levels of information, which are covered later in this document
 - Viewpoints are usually considered as relating to higher or lower levels of informational models
- ‘Zooming’ out, is the ability to easily access a higher abstraction; to move from a complex model to a simplified model
- ‘Zooming’ in, is the ability to access a detailed model from a simplified abstraction. This provides the flexibility to move from abstraction to multiple platform specific models
- Platform and Implementation Language Environment. The term platform is used to refer to technological and engineering details that are irrelevant to the fundamental functionality of a software component. This brings out the important MDA concept of Platform Independent Models (PIM)

These key concepts are used to create a set of linked models from high-level abstraction to low-level platform specific implementation. Model linkage and traceability are the cornerstones of MDA are key for to enable the Zooming capabilities. The layered models are shown in Figure 1.

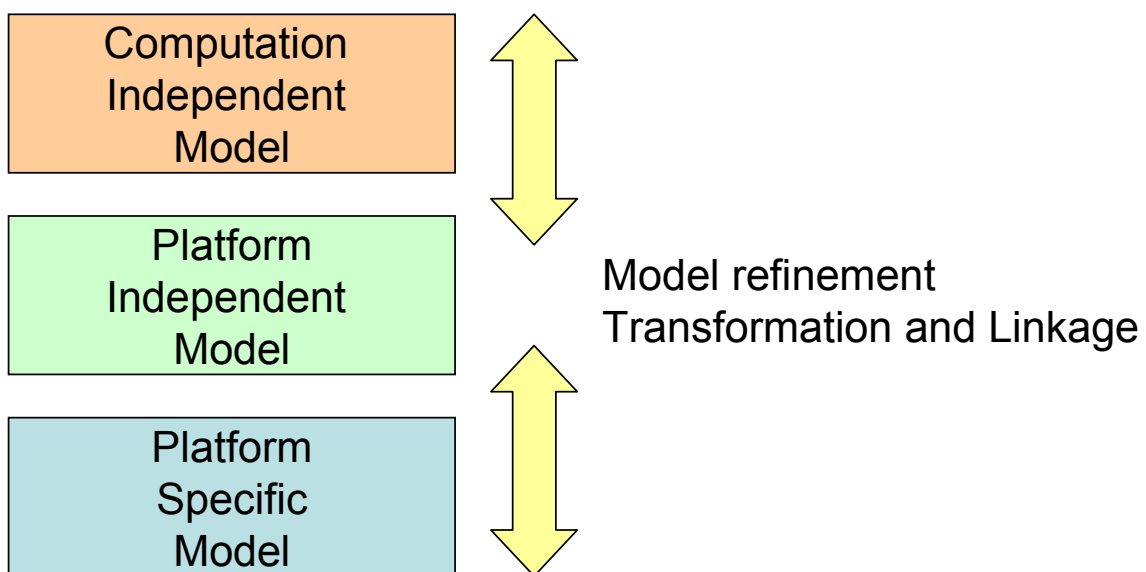


Figure 1 - CIM, PIM and PSM ‘linked’ models

A **Computation Independent Model (CIM)** is a view of a business that is independent of any technology constraints; a CIM does not show details of the structure of systems. A CIM is sometimes called a domain model and a vocabulary that is familiar to the practitioners of the domain in question is used in its specification.

A **Platform Independent Model (PIM)** is a view of a system independent of any implementation technology constraints. A PIM exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type.

A **Platform Specific Model (PSM)** is a technology and implementation specific view of a system. A PSM combines the specifications in the PIM with the details that specify how that system will use a particular type of platform.

In summary, MDA defines modeling abstractions: Computation Independent Models (Business Process Models), Platform Independent Models (Components and their interaction) and Platform Specific Models (which represent implemented code). Traceability links should exist between these models and the OMG suggest that automated generation of one model, from another, is an optional additional feature. MDA implies a business driven component based development approach and provides a robust and flexible approach to modeling that will sustain inevitable technology changes. The approach enables organizations to maximize their intellectual property investment and target multiple and evolving platforms.

Challenges Addressed

MDA addresses the challenges of today's highly networked, constantly changing systems environment, providing an architecture that assures:

- Portability, increasing application re-use and reducing the cost and complexity of application development and management, now and into the future
- Cross-platform Interoperability, using rigorous methods to guarantee that standards based on multiple implementation technologies all implement identical business functions
- Platform Independence, greatly reducing the time, cost and complexity associated with re-targeting applications for different platforms-including those yet to be introduced
- Domain Specificity, through Domain-specific models that enable rapid implementation of new, industry-specific applications over diverse platforms
- Productivity, by allowing developers, designers and system administrators to use languages and concepts they are comfortable with, while allowing seamless communication and integration across the teams

Bottom-line Benefits

The benefits of MDA are significant to business leaders and developers alike:

- Reduced cost throughout the application life-cycle
- Reduced development time for new applications
- Improved application quality
- Increased return on technology investments
- Rapid inclusion of emerging technology benefits into their existing systems



Why Model - A Brief History?

Modeling applications and systems has now become widespread. Models direct the course of understanding during the development lifecycle and provide a means of abstracting often complex distributed systems.

It is worth taking a moment to reflect on some of the specific problems that modeling overcomes, considering not only the advantages but also those aspects that are not solved by basic modeling alone.

There are 3 key problems addressed by a model-driven approach to and system development:

1. **Requirements.** Before modeling there are two routes to gathering requirements:
 - a. Talk to the person who wants the application and code it; checking that the requirements have been met by using the finished product. However, constantly changing the code was found to be time consuming, introduced errors and costly.
 - b. Produce detailed text based documents that the application requestor can approve before the coding work. This produced fewer defects in the code but the documents were large, time consuming to create and difficult to review.
2. **Maintenance.** Whenever a change is required to existing functionality in a system and there is no documentation then the places to change the code have to be found by examination of that code (which may not be commented adequately); This is a very time consuming exercise with no guarantee of success in finding all the places the code needs to be changed. Similarly, even if there is text based requirements and design documentation, finding the places in the text and mapping that to the code is laborious and costly.
3. **Extensibility.** Often, new functionality is required to be added to existing applications that need to access or interact with the existing functionality. With no documentation or text based documentation this exercise is difficult and often leads to duplication of functionality that causes further potential maintenance problems.

Modeling, as a part of application development, has been found to overcome many of the above problems; it is easier:

- For requirements specifiers to see if the requirements have been understood
- For developers to understand the requirements
- To design applications that minimize duplication
- To maintain applications
- To extend applications

Although basic modeling makes all these aspects easier there are still some problems to overcome:

- There still needs to be a mapping from business language to technical language
- There still needs to be a mapping from logical system design to physical implementation
- There is still a need to keep differing views of the system synchronized
- There is still the need to adopt an appropriate development process that employs the right people at the right time

The effort in mapping and synchronization is one reason why some organizations have abandoned modeling as concrete development documentation for maintenance and extension using it only for initial development 'brainstorming'; others have abandoned it completely in favor of such practices as eXtreme Programming.

Select has addressed these problems in the past by producing system animation and code synchronization tools and continues to address these additional problems by supporting the MDA with an SCBD process and the Select Solution for MDA.

Supporting MDA

Select Component Factory and Select Perspective have at their heart the Supply Manage and Consume (SMaC) process that expresses the potential for the separation of the delivery of solutions from the delivery of the components and services supporting the solutions. Detail of the SMaC process can be found in the Introducing Select Perspective available from www.selectbs.com.

The SMaC Framework is shown in Figure 2.

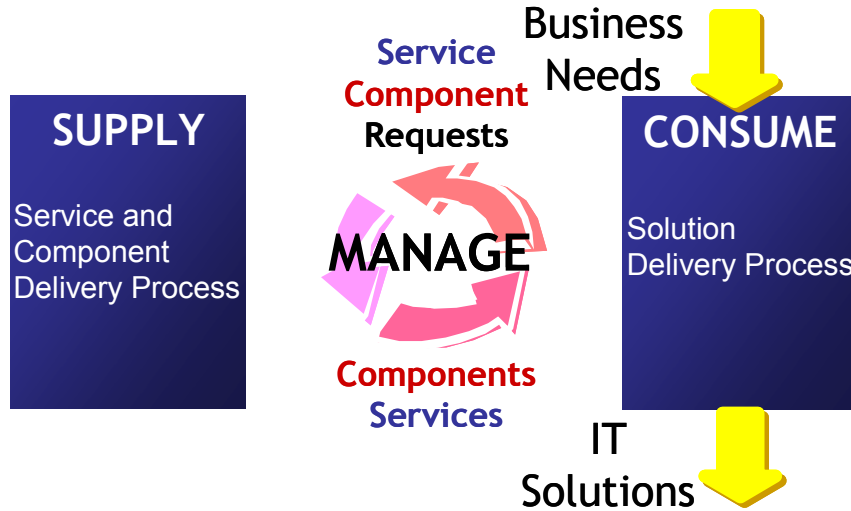


Figure 2 - The SMaC Framework

Key principles and benefits of SMaC are:

- Separate definition of a business model from the solution specification; the CIM
- Separate definition of a project solution, independent of technology implementation; Solution PIM
- Separate definition of components/services to support project solutions independent of technology implementation; Component/Service PIM
- Ease of producing multiple designs of components/services for differing platforms; Component/Service PSM
- Rapid construction of Components/Services for specific platforms; Code Synchronizers
- Ease of producing multiple solution designs for differing channels using differing technologies; Solution PSM
- Rapid assembly of solutions, from services and components; Code Synchronizers

Select have been the leaders with this unique approach to SCBD that aligns very closely with the principles of MDA, i.e. the abstraction and refinement of different models representing the business domain, specification of functionality of the solution as services and components, and the technology specific implementation of those services and components to form the operational solution; the only thing missing until now has been the automated model transformation and synchronization, now available in the Select Solution for MDA; code synchronization has been available for some time.

SMaC and its relationship to MDA are shown in Figure 3.

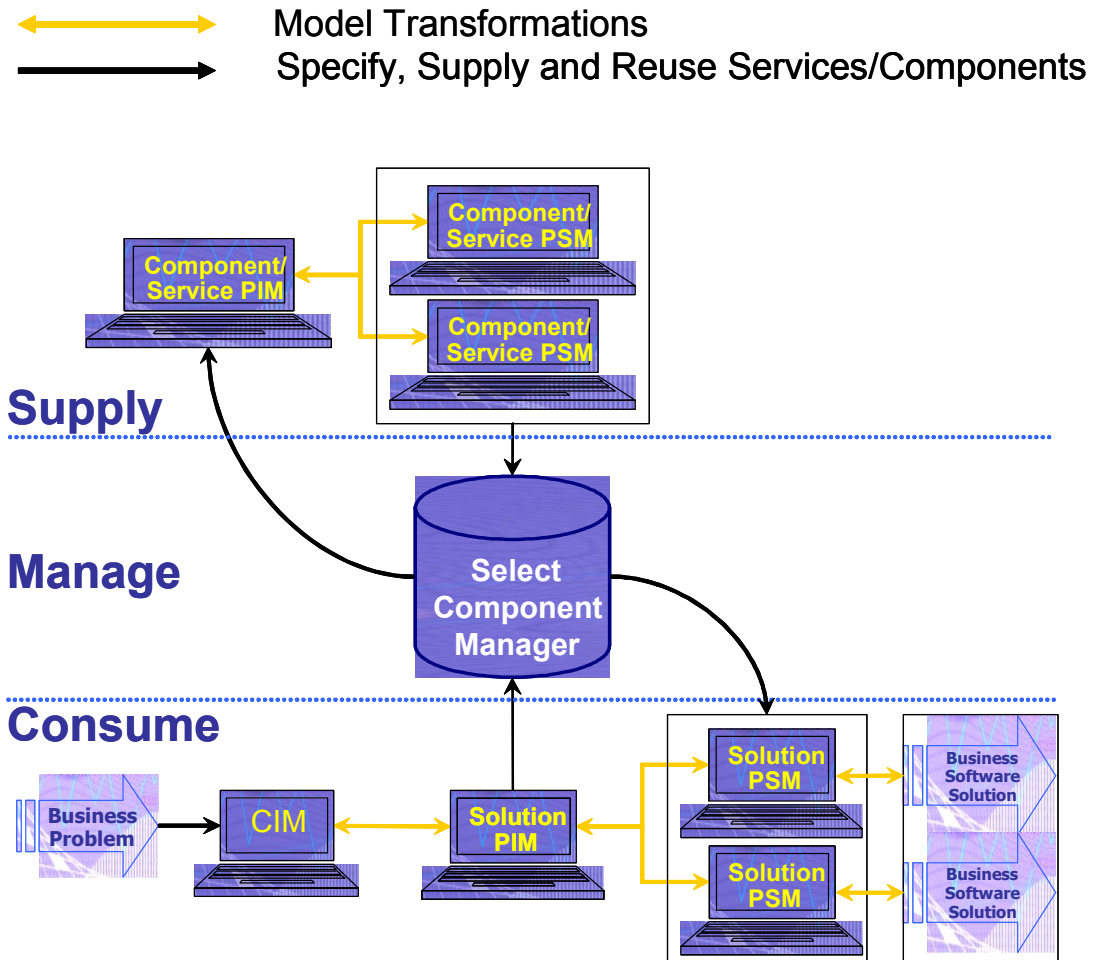


Figure 3 - SMaC and MDA Relationship

The explanations of the different types of PIM and PSM as applied to SCBD are in the following sections.

Computation Independent Model

The CIM is developed within the early stage of the Select Perspective's Consume process in the Business Alignment Workflow, which concerns itself with the requirements, description of the Business Process Model and the statement of business rules that constrain the business processes.

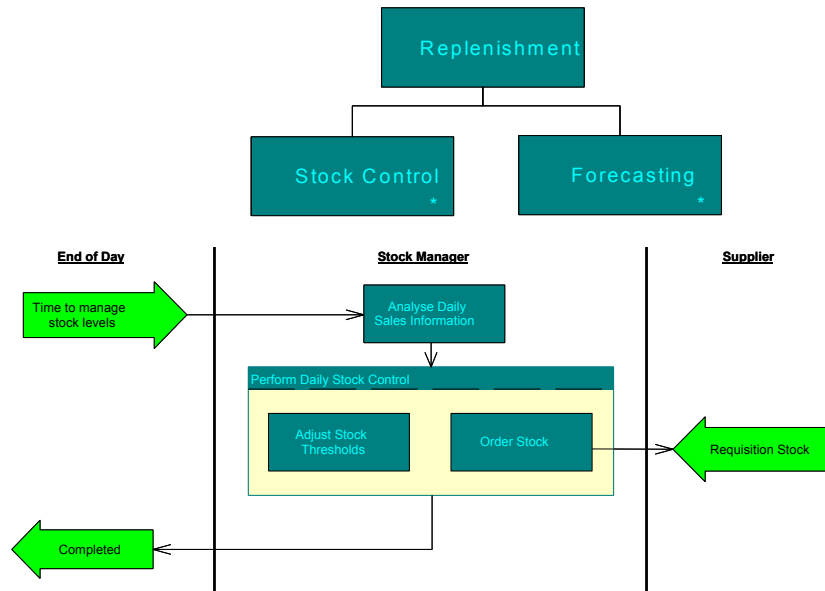


Figure 4 - CIM Process Hierarchy and Process Thread Diagrams

Business Process Modeling (BPM) is provided within Select Component Architect, by a notation based on CSC Catalyst³, see Figure 4. Although UML Use Cases and Activity diagrams can be used to convey the business domain, at Select we recommend using separate notations for the business notation and system definition; this adds clarity and provides a distinct separation.

A key differentiator of Select Component Architect is the ability to record and trace system requirements of all types including Business rules. This is done via the Select Component Architect Requirements Cataloging capability that allows referencing of the source of the requirements from the model as well as linking of the requirements to specific model elements that implement them as indicated in Figure 5.

Mapping the CIM to the Initial Solution PIM

Select recommends that initial BPM be completed as the initial stage for solving specific business problems i.e. as a Solution CIM. Each model is initially constructed from any existing information in a ‘corporate’ business model and the results of the project specific modeling collated back into the corporate model; business processes in the corporate model that will not be changed but are needed for clarification in the project model can be viewed by using the Select Component Architect ‘external elements’ functionality. Thus, Select advocates the maintenance of a corporate CIM with multiple Solution PIMs; one piece of business functionality being implemented by different designs and implementations; this practice is an extension to the OMG MDA concepts.

Some requirements may be given to the project and the BPM may uncover others; both may be recorded in the Requirements Hierarchy in Select Component Architect. Once a decision has been taken about which business processes are to be the subject of the project, they are mapped to UML Use Cases, which will be used to capture the user view of the system behavior. Thus traceability may be maintained from requirements, through the business processes and Use Cases. The emerging requirements definition forms the boundary and scope of the Solution PIM.

The Select Solution for MDA can generate the first-cut Actors and Use Cases needed to develop the Solution PIM from the CIM Business Actors and scoped Elementary Business Processes. Moreover, as the Solution PIM is developed, if other Actors and Use Cases are discovered they may be synchronized with the CIM.

The explicit linkage between the model artifacts of CIM and PIM as shown in Figure 5, business process activities are linked to supporting functionality described by Use Cases, requirements can be linked to both Use Cases and Business Process to imply non-functional constraints or algorithmic requirements.

³ Business Process Modeling notation is part of Computer Science Corporation’s industry accepted Catalyst notation

This model linkage, not only important for impact analysis, also provides the mechanism to **Zoom** from one MDA model abstraction to another.

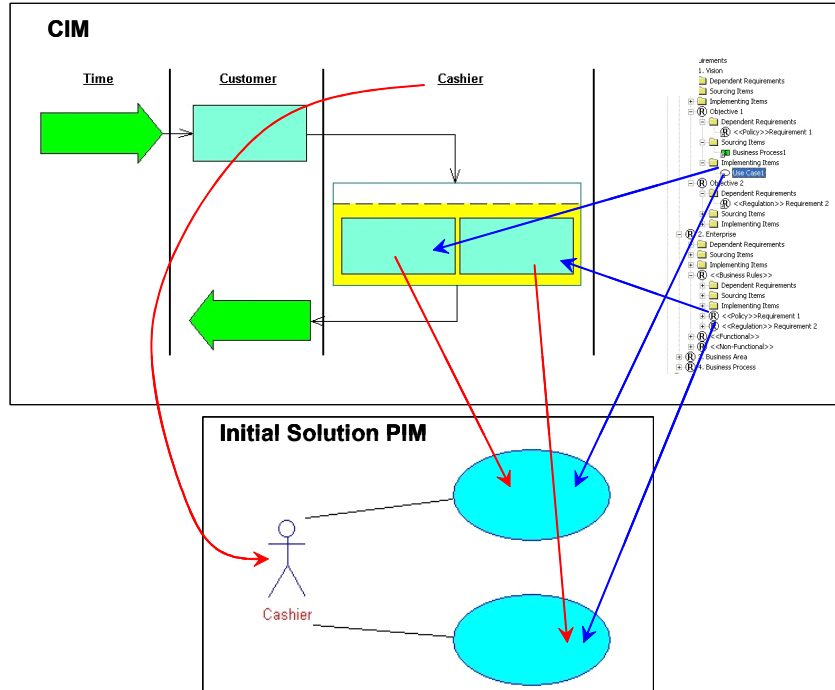


Figure 5 - Select Component Architect links between Processes, Use Cases and Requirements

Solution Platform Independent Model

Within the Select Perspective Consume process, early attention is focused on the identification and specification of the service and component-oriented solution architecture during the Business Architecture Workflow. This is independent of any technology that will be used for the implementation and thus forms the Solution Platform Independent Model.

Select Component Architect uses UML Class Diagrams to model the data requirements of the system as a Business Information Model as shown in Figure 6.

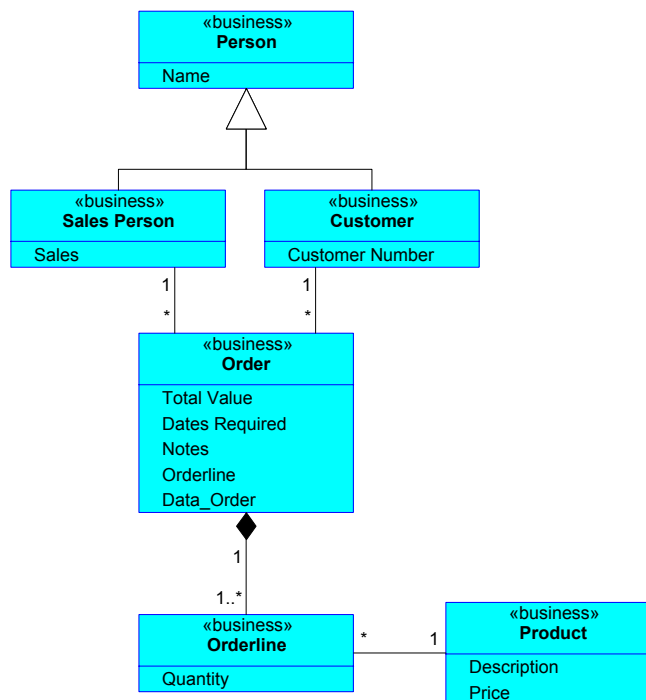


Figure 6 - Business Information Model (PIM)

The Business Information Model is used to identify the candidate required business components and Object Sequence Diagrams are used to identify the services of those components required to fulfill the functionality of each Use Case, Figure 7.

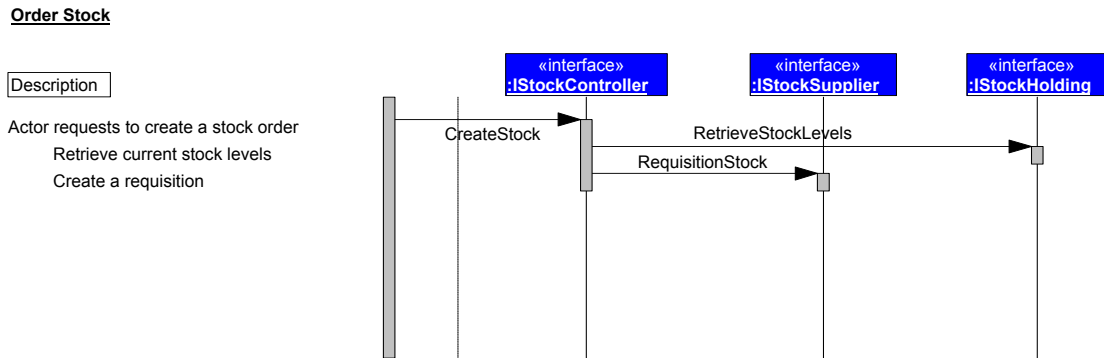


Figure 7 - Logical Specification of Service-oriented Business Architecture (PIM)

Finally, the Business Architecture in terms of components and the service interfaces are shown on a class diagram as UML packages with the corresponding interfaces and dependencies as shown in Figure 8.

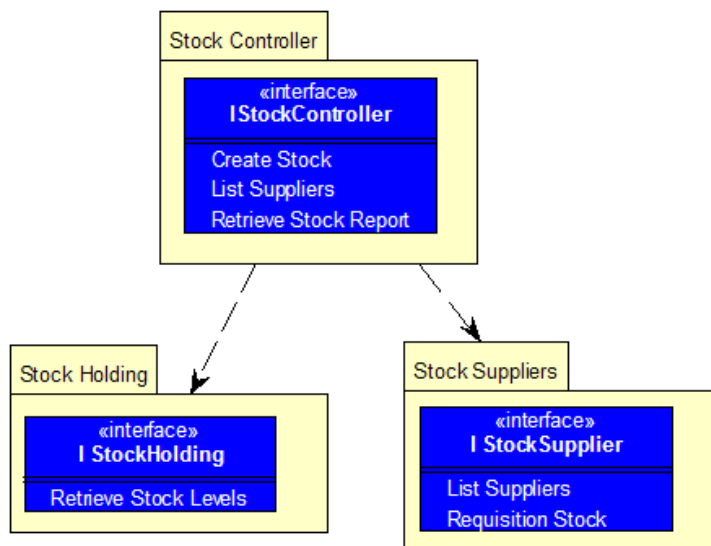


Figure 8 - Logical Specification of Service-oriented Business Architecture (PIM)

The Business Architecture model is recognized as a pivotal and unique deliverable within Select Perspective, unlike other development approaches that consider services and components at physical design and implementation time. As such, this model aligns directly with the concept of an SCBD Solution PIM; the components that will be assembled to form the solution will have their own Component PIMs.

The specification of services and components is the contractual basis for the supply of the physical services and components. At this point in the SMaC lifecycle the Business Architecture specifications are published into Select Component Manager that acts as the brokerage mechanism between the Consumers and Suppliers of services and components.

Select Perspective wholly supports both CIM and PIM, by activities within its Consume process. The Select Solution for MDA directly supports these two model abstractions by proving the ability to capture Business Processes independently of the requirements and the “Business Architecture” for the required solution. Explicit model generation and linkage is also provided from Business Process to Use Case and Requirement and Use Case to Service and Component interactions.

This mapping between CIM and PIM in the Select Solution for MDA is both ways; new Use Cases discovered during the development of the PIM may, where appropriate, be mapped back to business processes and linked with requirements, thus maintaining a consistent BPM for use on later projects.

Defining Platform Transformations

Before considering other ‘MDA’ models to be produced for a Solution we must consider the Technical Architecture. Select Perspective recognizes that without a well-defined technical architecture a consistent and flexible approach to systems development, maintenance and extensibility is almost impossible.

Therefore, as part of the Supply Workflow, outside of, but in conjunction with, the normal project work, activities need to be undertaken to define the technology specific implementation patterns and other artifacts to be used. Once these have been decided upon, Select Solution for MDA may be configured so that these patterns and artifacts may be applied automatically at the appropriate time to the appropriate models. The Technical Architecture may in fact be treated as a separate project in its own right to include the configuration of Select Solution for MDA.

The benefits of this approach are:

- Defining the standards to be used in one place
- Using the specialized skills required to define the architecture appropriately
- Making it simple to apply those standards
- Protecting intellectual property and the investment in design effort
- Simplifying system maintenance and extension effort

Figure 9 shows an extract from a Java Platform Specific ‘Data Access’ pattern, the base persistency class provides general facilities for all business classes to inherit. The <<BusinessEntity>> stereotype represents the mapping that will be applied to the business classes as part of a component model.

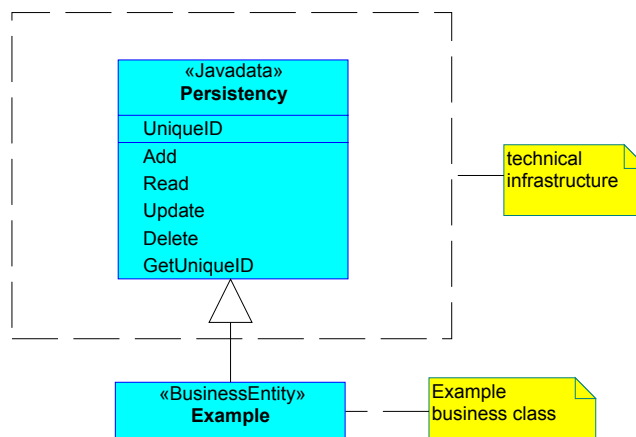


Figure 9 - PSM (Component) mappings for business class design.

Component/Service Platform Independent Model

Once the decision has been taken to build a component/service required by a solution, as opposed to reusing existing assets or renting/buying the functionality, the specification is imported to a separate model for design purposes. At this stage, the standard Platform Independent design patterns may be applied and such things as ‘business’ datatypes mapped to the appropriate ‘design’ datatypes.

In this way either much time is saved from normal practice or, if this is not normally done because of time constraints, a robust model is produced for later use.

Component/Service Platform Specific Model

Once the Component/service design has been completed a decision has to be made about which technology it is to be implemented in. There may be diverse user channels, middleware technologies and distributed databases in place that demand that components/services have to be implemented in different technologies.

A key part of the Supply process is the construction and delivery of components. Select recommends the platform specific modeling of components, unlike other vendors that apply template based code generation to the PIM or others that do not recognize the concepts of the PIM and advocate 'direct to code' from the solution model. This allows for a rich white-box design of the component and, as we shall see, the application of the organization's implementation standards and patterns embodied in the "technical architecture".

The initial component/service PSM is generated from the component/service PIM using the chosen technology transformer to produce a UML diagram, an example of which is shown in Figure 10.

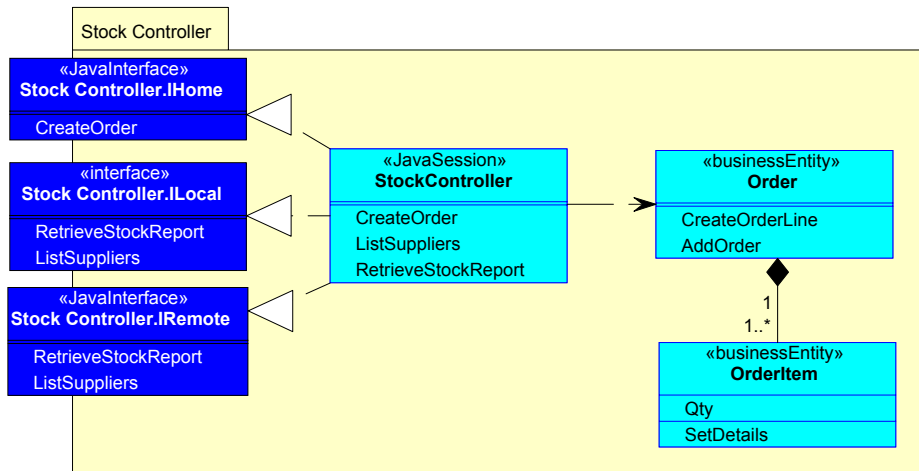


Figure 10 - A Component PSM

The white-box design of the component is automatically elaborated within the component model to realize the service specification, shown by the classes that realize the Interfaces in Figure 10. The technical architecture mappings, in this case the Data Access, are applied during the design of the component.

Applying the entire well formed rules necessary for the technology and the technology implementation patterns may well generate additional model artifacts, as illustrated in Figure 11

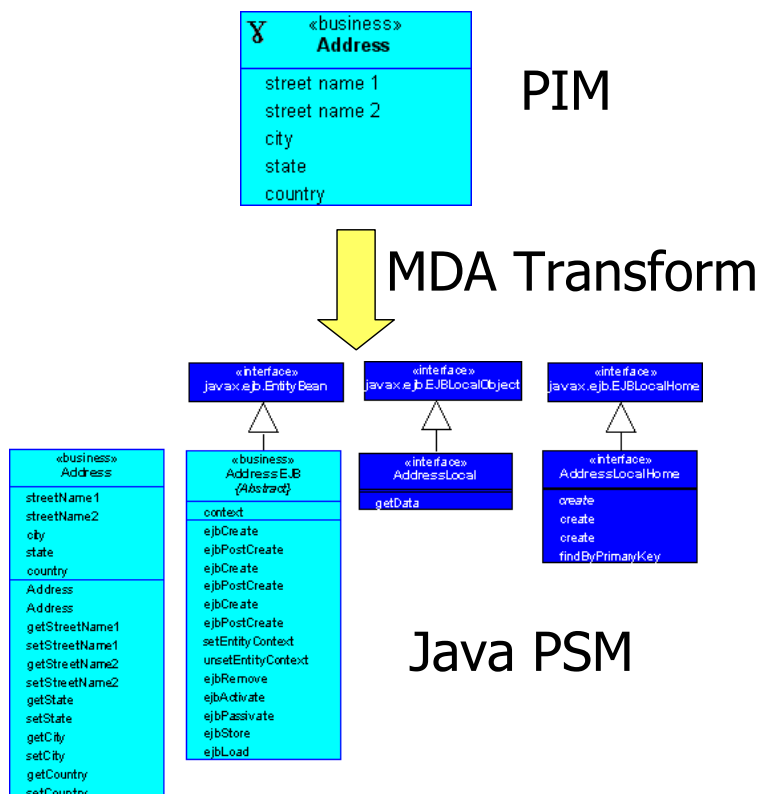


Figure 11 - PIM to PSM Transform

Select experience has shown that an average of at least 40% of platform specific design details may be generated with use of the Select Solution for MDA.

Further technology specific design work can be carried out on each PSM and the component/service can again be republished. Furthermore, it is often found that this design work highlights shortcomings in the PIM and/or patterns. Select Solution for MDA may be used to update the component/service PIM should the PIM be needed for different PSM. Requested changes to the patterns may be communicated to the pattern ‘owner’ via Select Component Manager for consideration and configuration of the Select Solution for MDA.

Again, this mapping is bi-directional in Select Solution for MDA; elements added to or modified in the PSM, where appropriate, may be easily mapped back up to the PIM to keep the design models synchronized, thus saving hours of development time and keeping a consistent PIM for other required PSM transformations.

Component/Service Implementation

It is all very well having a robust and complete technology specific design but the component/service still needs to be coded. What is needed are tools to generate as much of the final source code as possible; tools such as the Select Code Synchronizers. Support is available for generating not only markerless framework code but also the actual code where this can be defined as a pattern. Select experience has shown that, again, an up to 80% of the final code may be generated from the PSM and that does not include re-use of existing components that are necessary ‘lower’ in the technical architecture.

In addition, Extended Markup Language (XML) scripts and Web Service Definition Language (WSDL) may be generated from the model and Select Component Manager respectively.

The completed files (compiled if appropriate) maybe linked to the component/service specification in Select Component Manager as part of the Implementation Folder for the published component/service specification, as illustrated in Figure 12, for re-use by all Solution Teams.

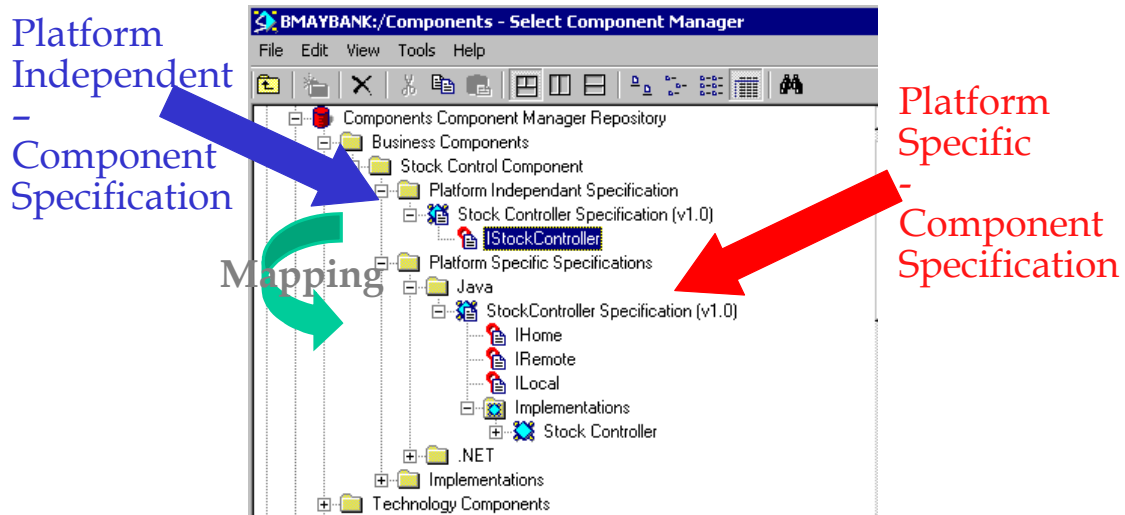


Figure 12 - Implementations in Select Component Manager

Changes and additions to the generated implementation source code may be reflected back into the relevant PSM by the use of the Select Code Synchronizers.



Figure 13 - Select Code Synchronizer

Solution PIM

We left the consideration of the solution above at the point where the Business Architecture had defined the business components/services needed to fulfill the solution. However, work still needs to be done to design the user interface (UI) and overall application management.

Select advises that this design work is done on an incremental basis. Post publishing the business component/service specifications, planning work is carried out to decide which Use Cases will be delivered in which increment; these are tagged as such in the Solution PIM and Select Model Copy is used to copy all the related model elements and diagrams to a separate model, either one for each increment or adding to the same model as each increment is started.

In conjunction with Solution prototyping, the Solution (or Increment) PIM is enhanced, new business component/service information being fed to the suppliers via Select Component Manager.

In addition, the Business Information Model (used up to now for deciding which business components would be needed and for creating structures to be passed as parameters for the services) can now be used to generate logical data structures in the form of tables, columns and the relationships using the Select database synchronizer, Select ORSync.

The platform independent UI and application components/services to be used can follow the same lifecycle as the business ones although it is expected that the Solution will, in the main, re-use those that already exist and have been published in Select Component Manager.

Solution PSM

Just as in the business component/service development lifecycle above, Select Solution for MDA may be used to apply technology specific mappings and patterns to the UI and application management model artifacts in the Solution PIM to generate a Solution PSM.

Figure 14 shows an extract from a Java Platform Specific 'Page Navigation' pattern; the base page class provides general facilitates for all pages to inherit, including the delegation to the navigation control component. The navigation control component is responsible for the redirect to the next page in the sequence. The <<jsp>> stereotype is the mapping tag that will be applied to the User Interface design as part of a solution model.

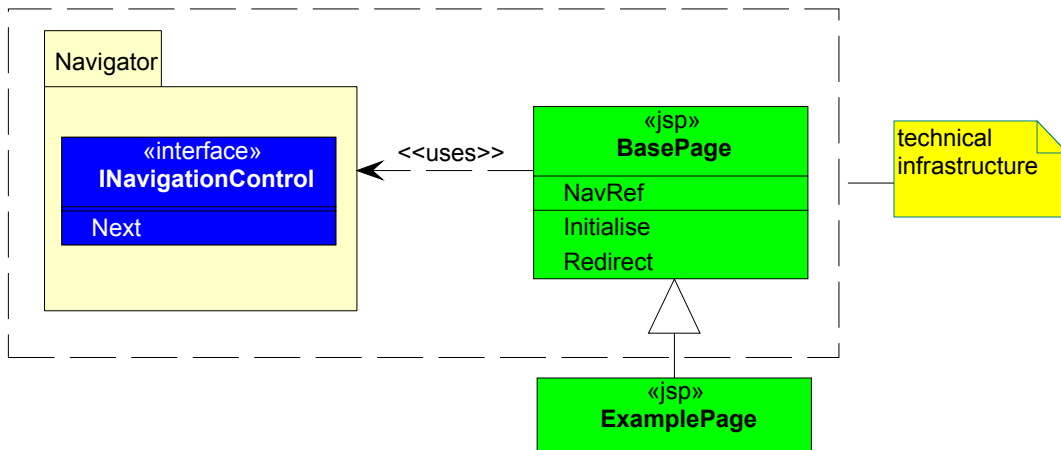


Figure 14 - PSM (Solution) mappings for the User Interface design

Solution Implementation

Solution Implementation is just a matter of generating the code for the UI and application control specific items and deploying them.

If the required business components/services are not ready at the time of testing, these can be stubbed so that at least the UI and application management aspects of the solution can be tested. As the business components become available, each Use Case can be fully tested before deployment.

Changes and additions to the generated implementation source code may be reflected back into the relevant PSM by the use of the relevant Select Code Synchronizer.

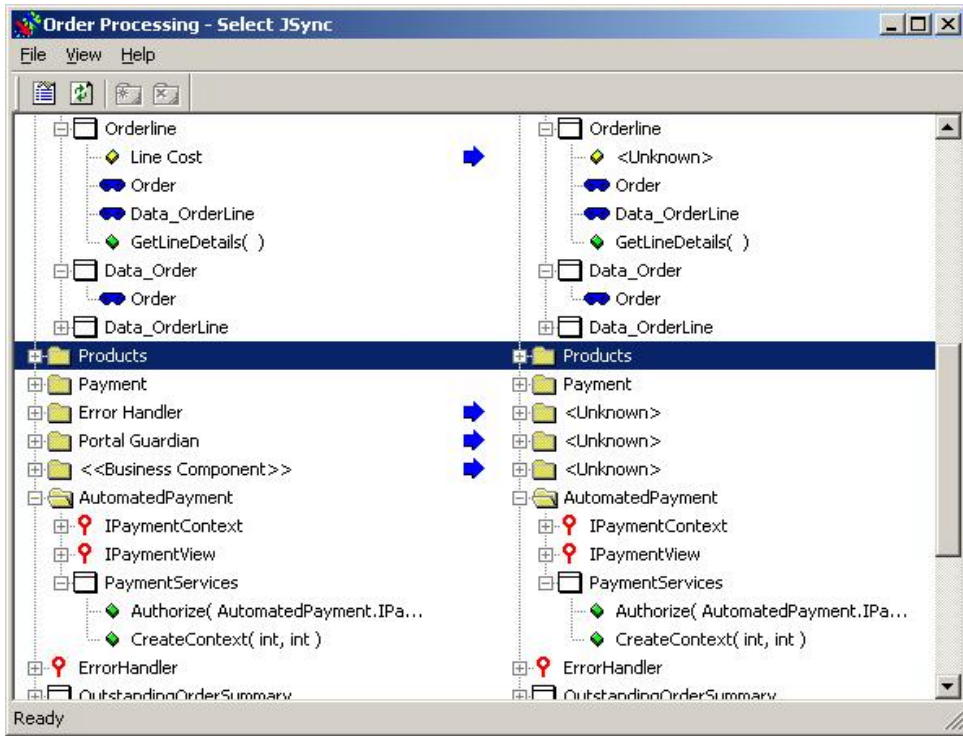


Figure 15 - Select JSync

Component Management

Effective component/service management is the key to success in a service and component based development environment and it is worth considering this subject in the context of MDA.

Component management within the context of SMaC recognizes distinct streams of activity; for example the acquisition, or harvesting of components for re-use. However re-use strategies are beyond the scope of this paper; instead we will concentrate on the role of component brokerage across the lifecycle, from service request independent of technology to platform specific component construction; involving the requisite transformation of PIM specification to PSM specification.

Where the “Business Architecture” specifications cannot be realized immediately by reusing an existing component, the specification forms a design and implementation contract between supplier and consumer, commonly know by the term “Design by contract”. In order to construct the component, the PIM specification must be first transformed into PSM specification as described above.

Select Perspective has long recognized the importance of the separate definition and application technology specific mappings. An example might be the mapping required for a distributed Java component, an ‘Enterprise Java Bean’. In this case the mapping would define the transformation of the logical business service into a Home, Remote and Local interfaces. The data specification for each interface is also transformed; in the case of XML, Select XML Schema synchronizer is used to transform the logical data specification to a physical XSD.

Levels of Component Abstraction

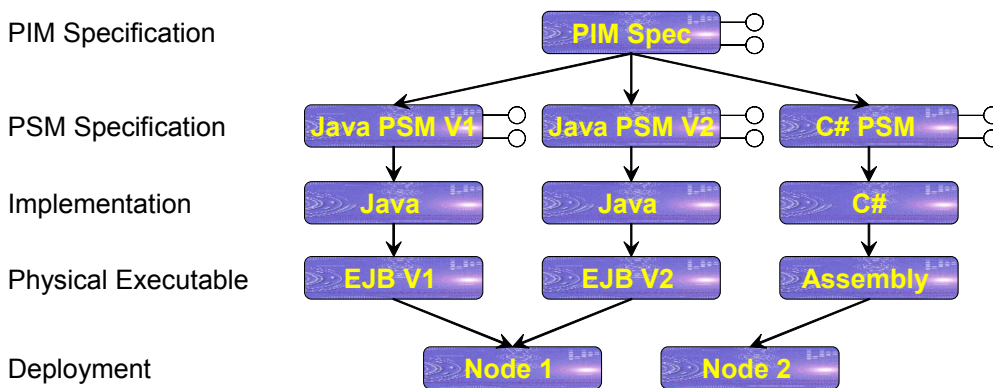


Figure 16 represents the different levels of component abstraction that can be captured within Component Manager, i.e. the transform of PIM to multiple PSM specifications and the implementation of PSM specification, and finally the deployment.

Levels of Component Abstraction

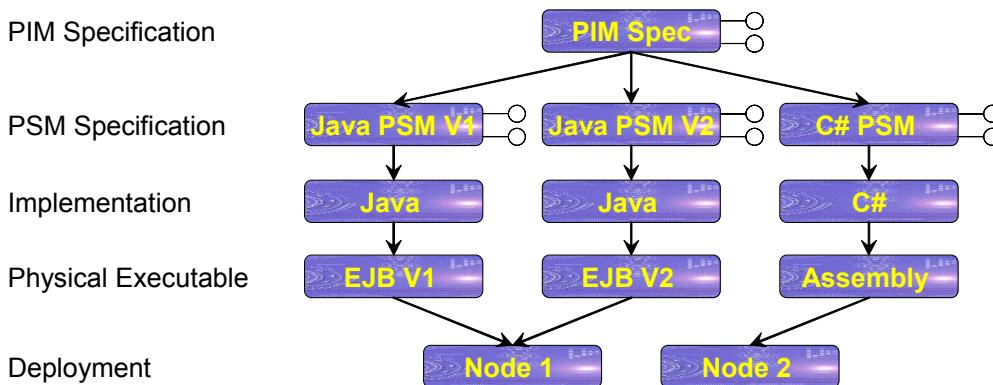


Figure 167 - Component Levels of Abstraction.

Once the component is constructed as part of the Supply side process, it is submitted to Select Component Manager and associated as an implementation for a given specification.

Select Component Manager also supports the acquisition and location of components from other sources, such as UDDI servers, component marketplaces such as ComponentSource⁴ and registered components on any network machine including the .NET Global Assembly Cache. Once the component or Web Service is located it can be published for use, catalogued within Select Component Manager and be immediately ready for use within the modeling environment.

Select Component Manager provides UML component and deployment diagrams to depict the dependencies and physical distribution of components. With its versioning capability, Select Component Manager extends the component modeling supported by many vendors - in that actual component version and configurations can be modeled. An example of the importance of this capability is where different versions of a component are being used by different solutions. Select terms this physical view the Deployment Specific Model (DSM), which is the key to supporting the full lifecycle of a component or service. As yet the OMG does not consider this abstraction.

⁴ ComponentSource is a commercial supplier of component and services <http://www.componentsource.com/>

Conclusion

Select Solution for MDA provides both practical guidance and tool support for the facilities of MDA:

- **Models.** Select Component Factory is a modeling tool suite that provides UML, as well as distinct Business Process Modeling and Database modeling. Select Perspective provides a defined process for using these models in a supplier/consumers SCBD process. Table 1 shows the models defined by MDA, their relationship to that suggested by Select Perspective and Select tools that may be used to construct them.
- **Abstraction, Refinement and Viewpoint.** Using the models defined above Select Component Factory allows clear levels of abstraction refinement and viewpoints.
- Generation and ‘reverse-engineering’ between CIM, PIM, PSM and source code save hours of valuable development time to produce consistent and robust models.
- **Computation Independent Models** are designed within the Modeling section of Select Component Architect. These relate only to the business domain and the business processes.
- **Platform Independent Component/Service and Solution Views** are core to the Select Solution for MDA. This unique strength allows you to design black-box components, their static relationships and procedural interactions. Separation and abstraction from the Platform Specific Models is provided through generation and the model linkage facilities of Select Component Manager.
- Once the component specification contracts have been defined and published to Component Manager, Platform Specific mappings are applied to create **Platform Specific Models**
- **‘Zooming’ in and out.** Select Component Architect allows you to drill down from a Business model, to a Component Interaction model (usually via Use Cases) and then down to platform specific component models, via Select Component Manager. All of these drill-down links are fully traceable and can be modified as the designs evolve over time.
- **Platform and Implementation Language Environment.** At the lowest level of design abstraction within Select Component Architect, code and database synchronizers map directly to the language specific implementation.

Table 1 - Select Component Factory with MDA relationship

MDA	Select Perspective	Select Component Factory
CIM	<i>Consume</i> Business Alignment	Select Component Architect Select Solution for MDA Select Reviewer
PIM with PSM mappings applied	<i>Consume</i> Business Architecture Solution Delivery <i>Supply</i> Component Delivery Technical Architecture Data Architecture	Select Component Architect Select Solution for MDA Select Synchronizers Reviewer for Select Component Architect Select Component Manager
PSM tags and mapping transforms defined	<i>Consume</i> Solution Delivery <i>Supply</i> Component Delivery Technical Architecture	Select Component Architect Select Component Manager Select Synchronizers
PIM specifications PSM specifications and implemented components	<i>Manage</i> Component Catalogue Solution Rollout	Select Component Manager

Throughout Select Component Factory, where appropriate, the mapping and generation between CIM, PIM and PSM is automated not only with the Select Solution for MDA but also with tools such as the Object Relationship synchronizer, which creates a linked database design, based upon a UML class model. A number of other automation tools exist, including XML Schema and WSDL generation.

Many modeling tool vendors are bending their solutions, through template and pattern driven code generation, to claim a degree of ‘compliance’ with MDA. As you can see from the various MDA specifications provided by the OMG, MDA is not code generation; it is a whole abstraction and linking concept. Select Solution for MDA provides the facilities and approach to support and delivers on the MDA concepts.

The benefits to be realized by using the Select Solution for MDA are:

- Reduced development costs throughout the application lifecycle
- Reduced development time for new applications
- Improved application quality
- Increased return on technology investments
- Rapid inclusion of emerging technology benefits into existing systems

This achieved by:

- Requirements specifiers being able to see if the requirements have been understood:
 - Constructing a separate definition of a business model from the solution specification; the CIM
- Developers being able to understand the requirements:
 - Constructing a separate definition of a project solution, independent of technology implementation; Solution PIM
- Designing applications that minimize duplication
- Making it easier to maintain and extend applications:
 - Constructing a separate definition of components/services to support project solutions independent of technology implementation; Component/Service PIM
 - Making it easy to produce multiple components/service designs for differing platforms; Component/Service PSM
- Enabling the rapid construction of Components/Services for specific platforms using the Select Code Synchronizers
- Making it easy to produce multiple solution designs for differing channels using differing technologies; Solution PSM
- Enabling the rapid assembly of solutions, from services and components; Code Synchronizers

All this is enabled by:

- Defining the standards to be used in one place
- Using the specialized skills required to define the architecture appropriately
- Making it simple to apply those standards
- Protecting intellectual property and the investment in design effort
- Simplifying system maintenance and extension effort

To find out more about how The Select Solution for MDA can solve your problems, visit:

<http://www.selectbs.com/products/products>

Or browse our website:

<http://www.selectbs.com/>

Or call our Sales teams who will be pleased to help with any questions you may have:

North America

Englewood, Colorado

Tel: +1.800.717.7761 (US only)

Email: info@selectbs.com

Trumbull, CT

Tel: +1.888.472.7347 (US Only)

Benelux

Belgium

Tel: +32 (0) 16386010

Email: infobenelux@selectbs.com

Nederland's

Tel: +31 (0) 206508040

France

Tel: +33 (0) 141481034

Email: infofrance@selectbs.com

Germany, Austria, Switzerland

Tel: +49 (0) 72198653.0

Email: infogermany@selectbs.com

United Kingdom

Tel: +44 (0) 1452638800

Email: infouk@selectbs.com