

A Federated PIM for Supply Chains

Dr. Bipin Chadha

bchadha@atl.lmco.com

Lockheed Martin Advanced Technology Labs

1 Federal St., A&E 2W, Camden, NJ 08102

ABSTRACT

A federation implies a loosely coupled system distributed across the internet or an intranet, where the participants can join in and leave the federation without breaking the federation. It also implies that the participants can function on their own when they are not a part of the federation. While our business units operate in this fashion for various programs such as Arsenal Ship, our computer based systems do not. We have developed a COTS based federated architecture to support multiple, geographically distributed organizations in a supply chain. This virtual supply chain can cut the cost and cycle time for programs. The key components of the architecture are (1) Federated Product Information Manager (PIM), and (2) Federated Production Planning and Scheduling. The paper discusses the work done to date with commercial tools towards realizing the federated PIM architecture.

INTRODUCTION

Over 60 percent of a product's cost is in its supply chain. To be more efficient each organization needs to look at how it interacts with customers and suppliers for multiple programs. The rapid consolidation and reorganization of the defense industry means a business unit cannot rely on centralized corporate systems for its mission critical needs. The design and manufacturing processes have significant inefficiencies built into them due to lack of collaboration between prime contractors, customers, and their supply chains. This ranges from very little involvement of suppliers in design process to limited supplier visibility into primes' manufacturing plans and schedules. Multiple, intensive iterations are involved in the exchange of technical data during design. These iterations are exacerbated by a lack of seamless and managed data exchange between each organization's data management systems. The processes for managing designs and approvals are ad-hoc and typically are not

synchronized with the information requirements of a design. The process tools currently in use provide limited support to users in relieving them of their data management burden.

Existing tools and architectures focus on enterprise-wide capabilities as opposed to the virtual enterprise capability. The enterprise is seen as a stable entity made up of distributed sites. Existing tools cannot easily handle organizational changes (mergers, spinoffs, consolidations, etc.). Vendors assume their system is the only one that anyone will ever want to use and provide limited support for interoperability standards. Better architectures are needed to address this dichotomy between business needs and COTS capabilities.

APPROACH

The efforts of the commercial vendors must be leveraged to ensure that the results will be quickly deployable throughout the industry without much disruption of the existing and planned information technology strategies. The solutions need to be developed in an IPPD fashion with key vendors as part of the team. One significant aspect of our approach is the distinction between concepts of federation versus integration. While there are several programs aimed at developing enterprise-wide infrastructures, what is needed here is a commercially available federated enterprise capability that lets the enterprise systems of primes and their suppliers to form program specific federations. Federation capabilities we propose go beyond the typical enterprise integration approaches via standards such as EDI, CORBA and STEP that essentially offer connectivity. To achieve true federation between instances of an application, the application itself needs to be enhanced so that it is aware of other instances and has the capability to interact with other instances in a meaningful way. For example the multi-site scheduling capabilities of i2 software cannot be simply duplicated by: wrapping legacy MRP systems using CORBA or other object wrapping technologies; by providing a web interface to the MRP system; or by passing data between modules.

CONCEPT OF OPERATION

The concept of operation for a Federated Environment is centered on the concept of open (but managed) communication through the Internet. It begins with a system established within the

prime contractor organization. The prime sets up a federation of organizations that will participate in the program, including customers. Federations will be setup by different primes for their programs and will be able to block out the organizations they do not want to involve in their federation. The federation setup will involve defining who the members are, what are their roles, what information and functionality they can access. The business relationships with suppliers will be long term strategic relationships built on trust and cooperation, not competing suppliers against each other every time or searching databases for low cost suppliers. These partnerships (federations) will essentially drive the configuration of their respective business tools, i.e., PDM, Workflow, MRP, etc. Once the business tools are configured to act as a federation they will be ready to support the supply chain throughout the life-cycle.

In our envisioned scenario, the entire team will be involved from the beginning to develop high level requirements. They will collaborate with each other in a structured and ad-hoc manner as needed to perform trade-off analysis to make the most cost effective decisions. High level requirements are broken down into design requirements. The design effort will also be collaborative. Members will develop more robust designs that are less susceptible to design changes downstream. The designers will search through part catalogs based on their requirements and find parts based on cost and delivery criteria. Design changes are immediately percolated throughout the supply chain to assess impacts. The Federated PIM (FPIM) will provide automated set-up of the design collaboration process throughout the supply chain. The FPIM system will significantly reduce the cost and cycle time through the inclusion of best-practice processes for collaborative system requirements breakdown, systems partitioning, and detailed design. Concepts like reuse and collaboration will be explicitly built into the design processes.

REQUIREMENTS FOR A FEDERATED PIM

This section does not cover the generic PDM functional requirements. The focus is to understand what is needed for different organizations with different PDM systems to form virtual corporation environments for sharing product data. The ability to interoperate goes

beyond simply exchanging data and files. The systems have to become aware of other PDM systems and recognize them as part of the overall environment. The control and ownership of data may pass back and forth between PDM systems during design. The users in this environment should not have to go through redundant authorization checking. To a user, multiple PDM systems should appear to operate as a single system. This essentially implies that all the usual PDM requirements apply, only with a twist that this mega-PDM is made up of pieces that are supplied by different PDM vendors, and are owned and maintained by different user organizations. The following requirements emerge to achieve this interoperability:

1. Standards Based APIs

For the PDM systems to interoperate, it is critical that they start supporting Standards based APIs. This enables an external application to make these calls without too much dependence on a particular vendor's software. There are several such standard API efforts (WfMC, DMA, etc.) underway looking at different parts of the problem.

2. Standards Based Messaging

The PDM systems should support a standards based messaging approach. The CORBA/IIOP standards provide the most credible and mature set of standards in this area. The PDM systems should be CORBA compliant.

3. Standards Based Data Exchange

The PDMs need to exchange data between themselves and other applications, therefore they need to support data exchange standards. The broadest set of such standards are the STEP standards.

4. Standards Based Configuration Management Model

Since each PDM and each organization have potentially different schemes for configuration management (CM) of data, there is a need to develop and support standards based approaches for CM. RASSP [1] program has developed such a model that can be leveraged by PDM

systems and businesses looking for a structured way for CM. CM here refers to the management of evolving versions and revisions of data objects throughout their lifecycle.

5. Standards Based Product Structure Model

One of the key sets of data a PDM manages and needs to share is product structure. Ability to manage and share product structure data in a standard way is desirable. The relevant standards are STEP AP203 and MIL-STD-2549.

6. Standards Based Authorization Model

Since each PDM and organization have potentially different schemes for authorization management or access control of data, there is a need to develop and support standards based approaches for Authorization Management (AM). The DARPA RASSP program (<http://www.atl.external.lmco.com/projects/rassp/rassp.html>) has developed a model that can be leveraged by PDM systems and businesses looking for a structured and uniform way for AM. The model also needs to be extended to support users belonging to external systems.

7. Standards Based Process/Workflow Model

Process and workflow models face the same set of problems as CM and AM - different PDMs have different notions of what workflow or process management means. A standard way of representing workflows and ability to coordinate their execution within different organizations is a significant challenge. Standards such as WfMC and PIF need to be supported to enable process interoperability.

8. Awareness of External PDM Systems

Typical PDM installation has no notion of any other PDM installation and therefore cannot even begin to start interacting with it. The PDM systems have to be capable of understanding their own identity and their context in a complex environment made up of several PDM installations from different vendors. This awareness includes knowledge of relevant objects in other PDM systems.

9. Manage Cross System Relationships

PDM systems should be able to form and manage relationships between objects belonging to different PDM systems.

10. Support Cross System Queries

PDM systems should be able to generate and process queries across systems transparently. Users should be able to control the scope of these queries.

11. Support Multiple Schemas

PDM systems should be able to work across different schemas in different installations. Although it is highly desirable to have schemas that are similar, it is unlikely that any two installations will have identical schemas. The PDMs should be able to perform most functions across the systems with slightly different schemas.

12. Administer Complex Federations

PDM systems should provide cost effective tools to easily administer and manage complex federations of PDM systems. It should be easy for a systems administrator to create and manage multiple federations of PDM systems.

FEDERATED PIM ARCHITECTURE

The architecture described within this paper will serve the needs of businesses that have existing PDM systems (or are planning to procure PDM) and need to interact with their customers' and suppliers' PDM systems to execute their programs. Clearly manufacturers and the businesses they interact with will not have homogeneous PDM environments. Increasingly, as they form teams to execute a program, they will find product data scattered in a variety of PDM systems. Current PDM systems provide minimal support for interaction between independent PDM environments. It is unrealistic to assume that subcontractors will procure/implement new PDM environments to support every new project. It will also be cost prohibitive to integrate multiple PDMs to other business applications used in an organization. As businesses start using their PDM systems for all project activities, it will become increasingly important to be able to deal with the issue of heterogeneous PDM environments.

The federated architecture enables the primes and their suppliers to function as a federated supply chain (Figure 1). The federated architecture supports primes, large suppliers, as well as small suppliers who have limited resources. A federation implies a loosely coupled system where participants can join in and leave the federation without breaking the federation. It also implies that the participants can function on their own when they are not a part of the federation. This represents the reality of the supply chain member organizations who are independent businesses. These businesses have their own business systems (PDM, ERP, Workflow, Schedulers, etc.) that cannot be dependent on other systems. In order to achieve the cost and cycle time benefits they need the capability to participate in multiple supply chains on an ongoing basis. The federation concept fits very well with these needs. Any detailed architecture or implementation that does not support this high level architecture will fail to provide the needed capabilities for most large programs effectively.

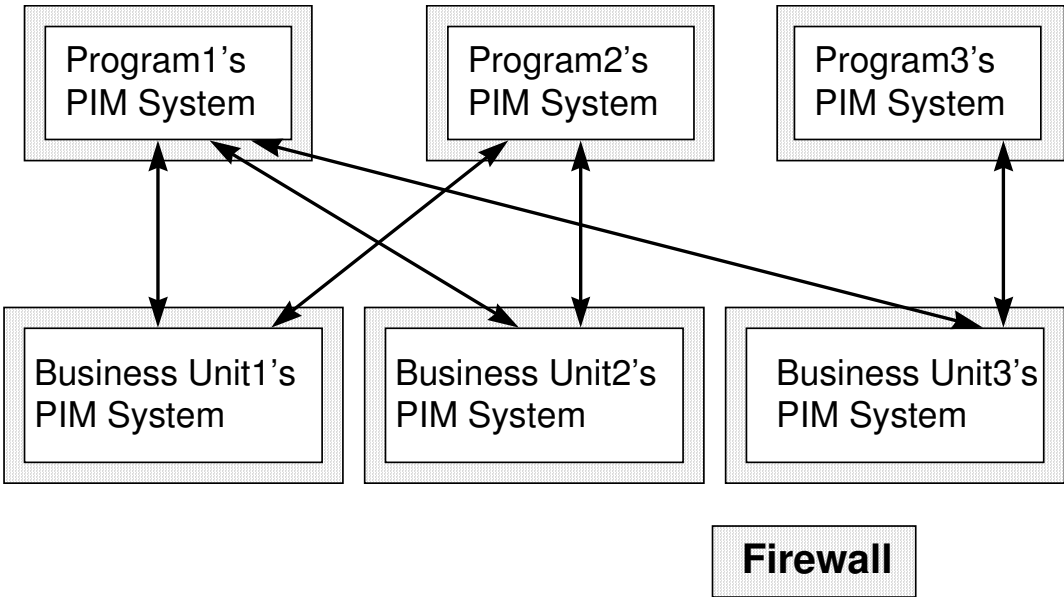


Figure 1 Business Architecture for Programs Requires Federation of PIMs

In addition to supporting the above business architecture, the architecture has to insulate other business applications (to a large extent) and users from the outside PDM environments and applications. This is the key tenet of our approach. The detailed architecture is driven by the desire to insulate a business unit's applications, users, and business processes from those of other participating organizations (Figure 2).

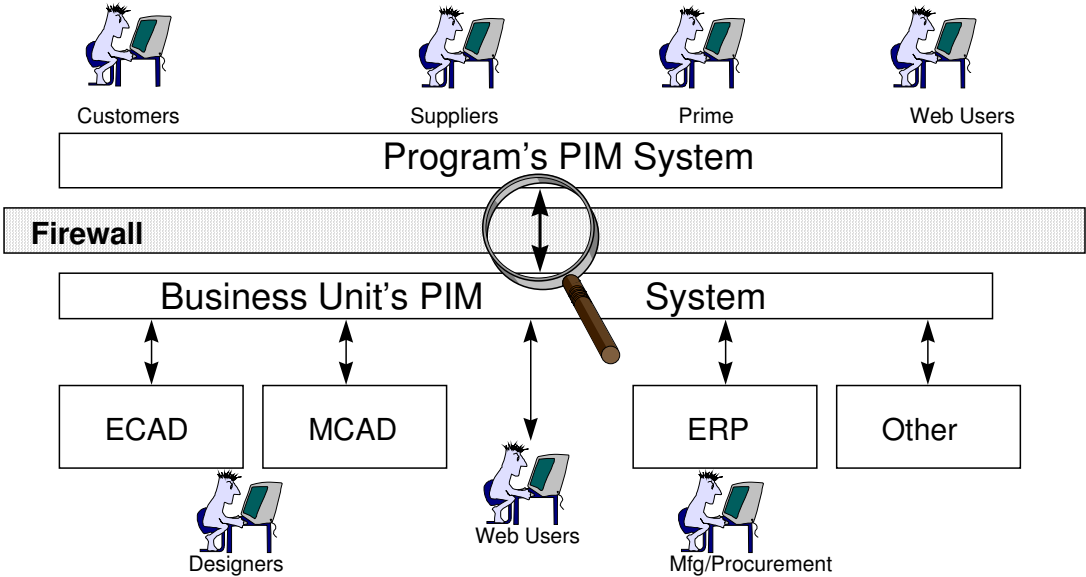


Figure 2 Business Application Integration to Local PIM Only

There are several architectural additions that need to be made to the current infrastructure to make this architecture possible. The basic architectural construct to be used is an Object Request Broker (ORB) that sits between clients making requests and objects that will service those requests [2]. It is expected that the interactions between various architectural components will be via CORBA compliant ORB mechanisms. It should be noted that ORBs do not provide complete data management functions or the object domain functionality. ORBs facilitate the communication without necessarily understanding what is being communicated.

Another concept used in the architecture is that of a “Proxy” object that provides a surrogate or placeholder for another object to control access to it [3]. A proxy object resides in the local PDM environment and represents an object residing in a remote PDM environment. Any requests made to the proxy are handled by the proxy and are forwarded to the real object for further processing, the results are returned to the proxy object (Figure 3). The proxy then forwards the results to the requester. Any clients therefore only work with the proxy object in the local PDM environment and are insulated from the details and how-to of dealing with the objects in remote PDMs. These PDM proxies are not the same as the proxies in the ORB, however, they will utilize proxy objects and services provided by the ORBs for their operation. The ORBs themselves can interoperate via Internet Inter-ORB Protocol (IIOP) that specifies a standardized interoperability protocol for the internet. The ORBs are to provide capabilities, such as:

- Find the right ORB for communication.
- Communicate with that ORB.
- Represent the target object reference.
- Identifying the operations to be performed.
- Translation of data parameters, exceptions, and context.
- Passing authentication and security information.

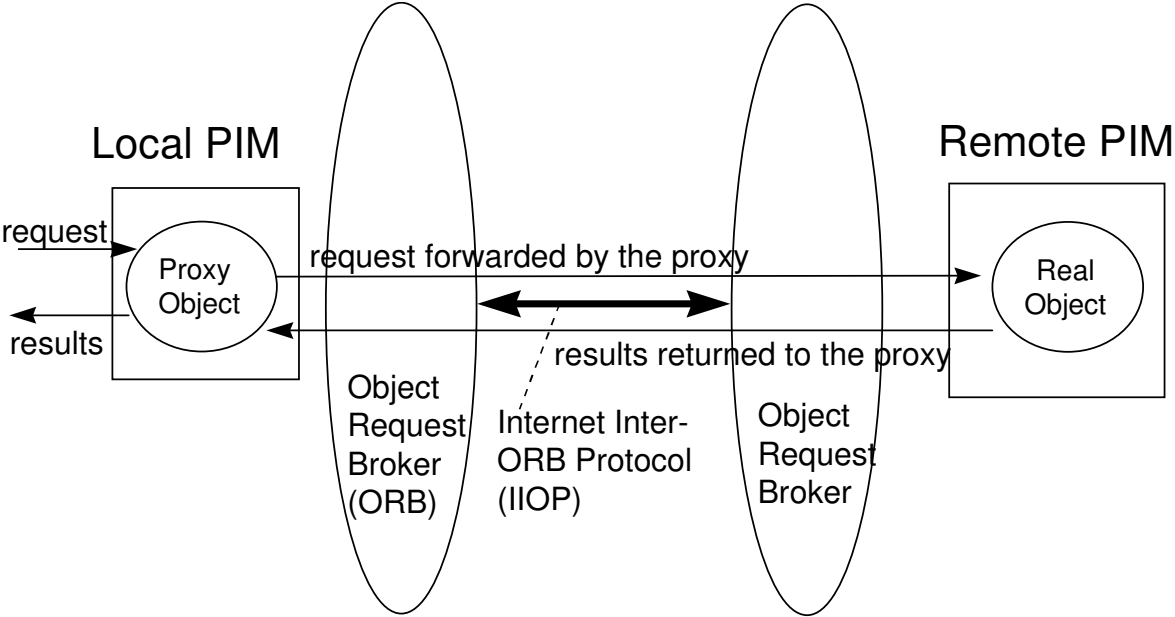


Figure 3 Proxy Object Concept Facilitates Independence from Remote PIMs

The generic structure for the PDM proxy object is shown below (Figure 4). The conceptual structure states that both “Real Business Item” class and “Proxy Business Item” class share a common interface inherited from the “Business Item” class, i.e., these two are sub-classes of the “Business Item” class. Also, there is a one-to-one relationship between the “Real Business Item” class and “Proxy Business Item” class. It is the responsibility of these classes to provide methods to implement appropriate request forwarding behavior as shown in the proxy concept diagram above (Figure 4). The implementation of proxy object will depend on the PDM’s class hierarchy.

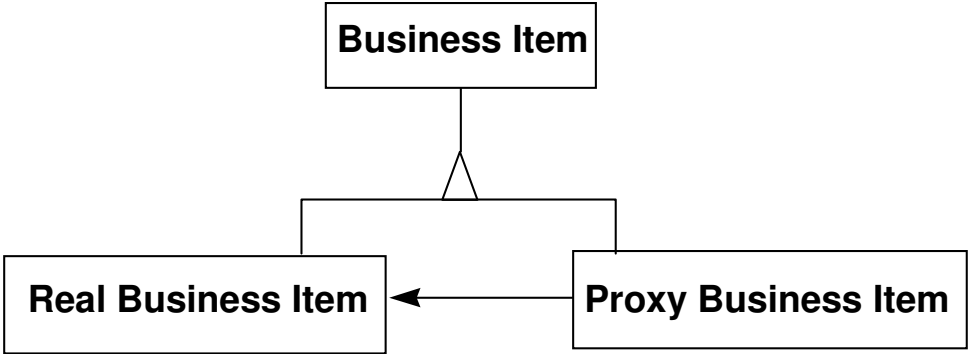


Figure 4 Generic Proxy Structure Provides Efficient Access to Remote Objects

There are several benefits for using proxy mechanisms [3]:

- Ability to defer the full cost of creation and initialization of the object until it is needed. This is valuable when it is expensive to move and create objects in local space, e.g., large design models.
- It is a more versatile and sophisticated reference to an object than a simple pointer.
- A proxy can provide mechanisms to control access to the original object in local space.
- A proxy can provide configuration management control for the original object.
- A proxy can provide enough information in the local environment without having to go to the remote PIM and copying/checking out the entire object.
- A proxy can make location of the real object transparent to the clients.

The following sections discuss the details of the components of this architecture concept.

Architecture Components

The detailed architecture to achieve federated PIM capability is shown below (Figure 5).

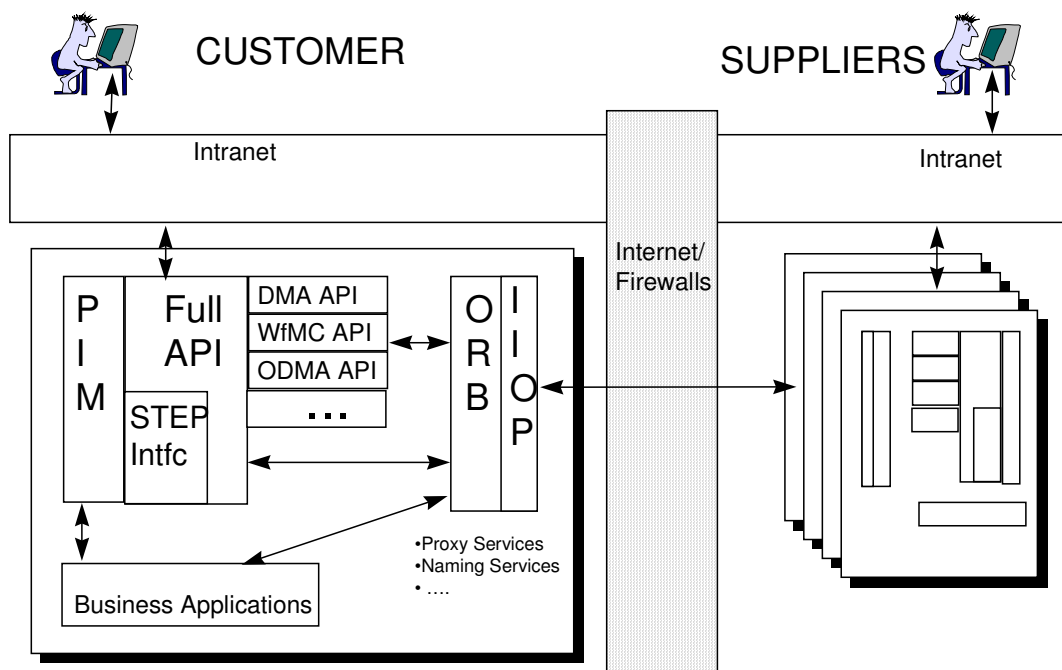


Figure 5 Federated PIM Architecture Enables Collaborative Design in a Supply Chain

The key components of this architecture are:

Product Information Management (PIM) System

The PIM system provides mechanisms for managing all aspects of product data over its lifecycle to facilitate design and manufacturing methodologies. The PIM systems commercially available today cannot support all the requirements described earlier and will have to be extended to support these requirements.

Full Application Program Interface (API)

The PDM system should provide a complete API to enable open access to all of its functionality. This is necessary for third party developers to create value added components, and create other abstract or standards based APIs.

Standards Based Application Program Interfaces (API)

The PDM system should provide abstract APIs that conform to standards (such as those set by OMG, WfMC, DMA, ODMA, etc.) to access some specific functionality.

Standard Data Interface

The PDM system should provide the ability to exchange data using standards such as STEP AP203, and AP232, and MIL-STD-2549. These standards define ways of organizing and sharing metadata between systems.

Object Request Broker (ORB)

The ORB provides several services that are key to achieving the PDM neutral capability. The ORB should conform to the Common Object Request Brokering Architecture (CORBA) specification [2]. An ORB enables objects to transparently make and receive requests and responses in a distributed environment. It is the foundation for building applications from distributed objects and for interoperability between applications in hetro- and homogeneous environments.

Internet Inter-ORB Protocol (IIOP)

The IIOP provides the ability for an ORB to communicate with other ORBs over the internet [2] by specifying a standardized interoperability protocol for the internet. IIOP specifies how inter-ORB messages are exchanged using TCP/IP connections. The IIOP specification consists of a common data representation (CDR) definition, message formats, and transport assumptions. The CDR supports primitive data types and attribute identifiers. There is no support for translations at the semantic level, where we plan to use standards such as STEP to perform intelligent communication between two PDM systems.

IMPLEMENTATION CONSIDERATIONS

The following issues surface when one thinks about implementing such an architecture:

Process Issues

Given the architecture described above, it is critical to rethink the processes that will be used to design and manufacture the next generation products. Key issues to be addressed are: what processes should be used to involve customers and suppliers throughout the lifecycle; what information will be shared; and how will teams collaborate to perform work. This architecture opens up many new possibilities to redefine processes to improve productivity and cycle times. A good understanding of the AS-IS process and a vision of the TO-BE process is necessary to derive the benefits of this new architecture.

Tool Issues

Given the architecture described above, it is necessary to examine current tools and technology to see how much of this architecture is feasible and what new tools or tool capabilities are needed to realize and maintain this architecture for an organization. The tools and tool integrations should be carefully evaluated and selected by considering maturity, availability, supportability, and cost. Many technology providers will claim to be able to do everything. The question to be answered is, what capabilities the currently available tools have to cost effectively support the business requirements.

Organizational Issues

The new architecture, and the process changes it forces will require cultural changes at the organizational level. Historically, design teams have not worked simultaneously with customers and suppliers. Involving customers in the design process, as opposed to getting approval at some later stage, will be hard for customers and the contractors. Use and acceptance of technologies such as PDM and Workflow itself will be hard for design teams not used to the discipline. Benefits of these technologies come years later and therefore require strong management commitment and resolve. Performance and efficiency is likely to go down in the near term before it goes up. Therefore, it is better to not start unless you intend to follow through.

The business reasons and driving forces are compelling and strong to start adopting these technologies. They will quickly become the differentiators in the coming years. The technologies are not completely mature but mature enough that a wait-and-see strategy is not advisable. The business units must start the learning process now to become familiar and master these new technologies.

PROTOTYPE IMPLEMENTATION

A prototype implementation was developed to test and evaluate federated PIM concepts. Two PIMs were selected: Metaphase and Optegra. The demonstration architecture is shown in figure 6. The user belongs to a subcontractor organization that builds a “black box” that fits in the “radar room” on a ship. User’s organization uses Metaphase, while the organization responsible for the “radar room” uses Optegra. Key information that this user needs resides in Optegra and Metaphase (CAD data in “gaf” or VRML format for this demonstration).

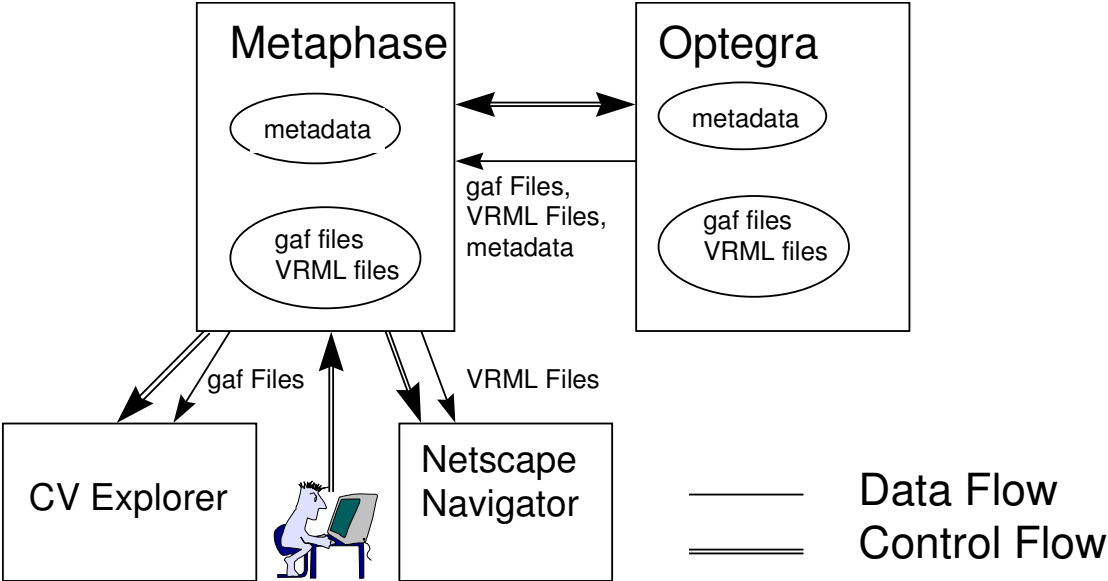


Figure 6 Demonstration Architecture Allows Information Exchange Between PIMs

In the demonstration scenario, the user logs into the Metaphase system, starts work on the “black box” and discovers its relationships to CAD models, documents, and assemblies. User discovers that the “black box” needs to fit into the “radar room”. Further investigation reveals that CAD information for the “radar room” is not available locally, but a proxy object exists that points to Optegra (Figure 7).

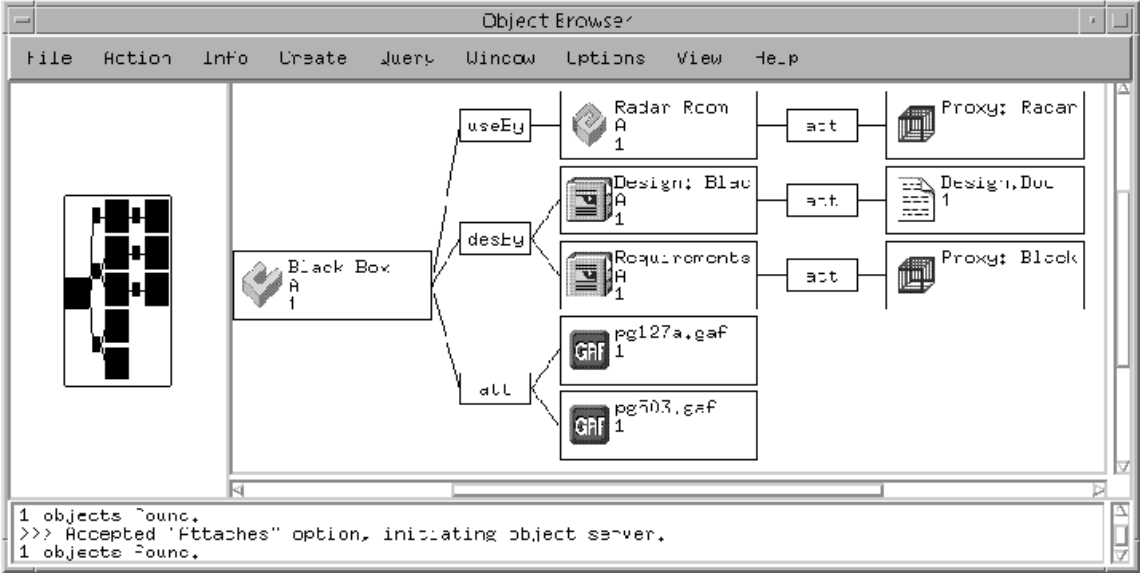


Figure 7 User Discovers a Proxy Object in Metaphase

The user requests the CAD data corresponding to the “radar room”. The proxy object initiates a request to Optegra for the appropriate CAD document. The document is copied out from Optegra using Optegra API and transported to the Metaphase location. The document is then registered into Metaphase and appropriate relationships are created with Metaphase Objects (Figure 8). The user requests to view CAD documents for “black box” and “radar room”. The appropriate CAD tool is launched with the two CAD documents. The user can see how the two fit together and makes appropriate design changes.

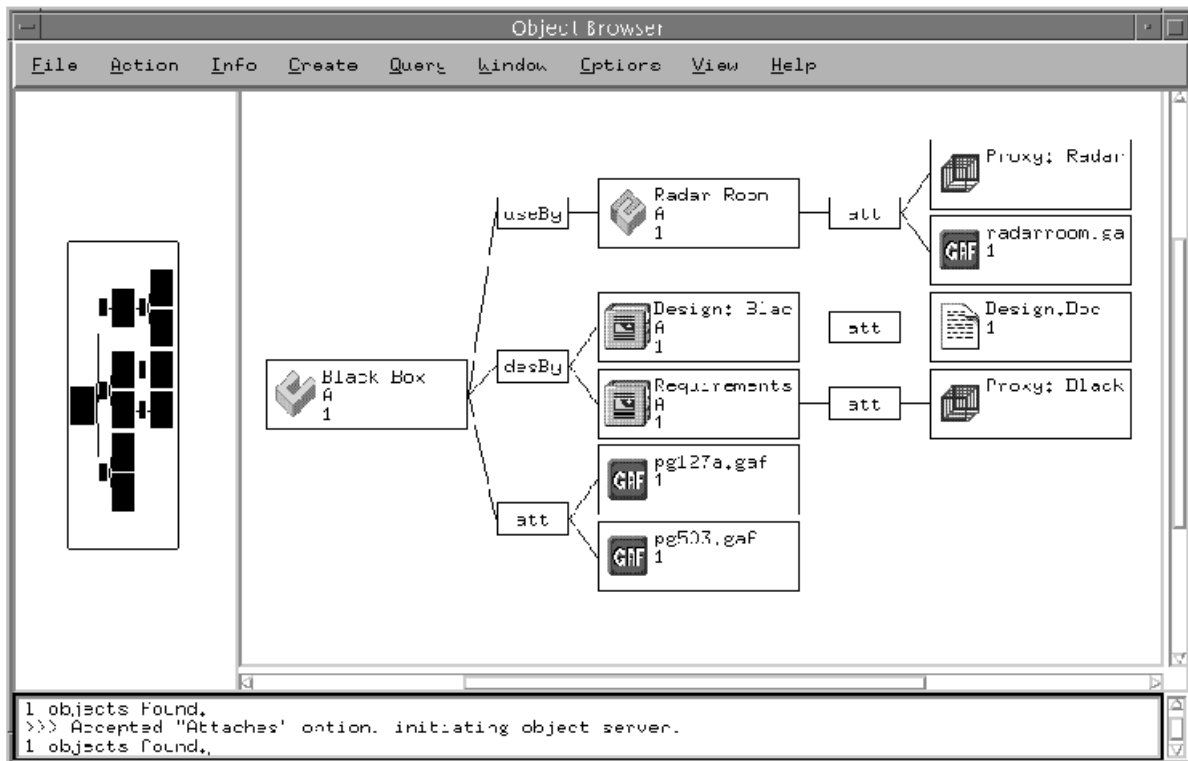


Figure 8 Actual Object is Copied from Optegra and Registered in Metaphase

This prototype essentially demonstrates a user interacting with the local PIM system while accessing information from multiple PIM systems. A similar prototype that lets Optegra users access Metaphase objects is being developed. The concepts described here can be applied to any PIM system.

SUMMARY

The federated approach is inline with the current business trends. It provides a low risk alternative to existing technology investment strategies and enables the organizations to bring new technologies in a modular fashion as opposed to the big bang approach. It provides the capability to protect current technology investments. Key vendors are investing in these technologies and COTS based solutions are deployable in less than 12 months. Any planned PIM deployment should consider the federated architecture proposed in this paper.

REFERENCES

- [1] Chadha, B., et. al., Enterprise Infrastructure Support for Agility, The Fifth National Agility Conference, 1996.
- [2] The Common Object Request Broker: Architecture and Specification, Revision 2.0, OMG, July 1995, Updated July 1996.
- [3] Gamma, E., et. al., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 94.