

The Open Group
Apex Plaza
Forbury Road
Reading, Berkshire
RG1 1AX, England
Tel: +44 1189 508311
Fax: +44 1189 500110

THE *Open* GROUP

Business Scenario: Building Blocks Information Base

Business Scenario

Source: NCR Analysis March 23, 2000

Status: Draft

Author(s): Terence Blevins, NCR Corporation

Last Revision Date: March 29, 2000

Table of Contents

BUSINESS SCENARIO PROBLEM DESCRIPTION3

 BACKGROUND OF SCENARIO3

 PURPOSE OF SCENARIO4

OBJECTIVES4

VIEWS OF ENVIRONMENTS AND PROCESSES.....4

 PROCESS DESCRIPTIONS.....7

 PROCESS STEPS MAPPED TO ENVIRONMENT.....12

 INFORMATION FLOW.....12

ACTORS AND THEIR ROLES AND RESPONSIBILITIES13

 HUMAN ACTORS AND ROLES.....13

 COMPUTER ACTORS AND ROLES.....13

REQUIREMENTS15

 DESCRIPTION OF A USEFUL BUILDING BLOCK INFORMATION BASE15

RESULTING TECHNOLOGY ARCHITECTURE MODEL.....16

 CONSTRAINTS16

 IT PRINCIPLES.....16

 TECHNOLOGY ARCHITECTURE SUPPORTING THE PROCESS.....17

Conceptual List of Components of Building Blocks Information Base.....17

APPENDIX A - ADML DESCRIPTION18

APPENDIX B - BUILDING BLOCKS DESCRIPTION WITH ADML EXTENSIONS19

Table of Figures

FIGURE 1 - A TYPICAL BUSINESS ENVIRONMENT5

FIGURE 2 - BUSINESS ENVIRONMENT VIEW OF A SHOP5

FIGURE 3 - TECHNOLOGY ENVIRONMENT VIEW OF A SHOP.....6

FIGURE 4 - BUSINESS ENVIRONMENT VIEW OF HEADQUARTERS.....6

FIGURE 5 - TECHNOLOGY ENVIRONMENT VIEW OF HEADQUARTERS7

FIGURE 6 - PROCESS STEPS12

FIGURE 7 - COMPUTER ACTORS13

FIGURE 8 - TECHNOLOGY VIEW OF PROCESS17



Business Scenario

Building Blocks Information Base

What a difference a few more bucks for first-rate architecture make to everyone and everything it impacts. - Malcolm Forbes

Business Scenario Problem Description

Background of Scenario

A building block is either an architecture construct such as a blueprint diagram or computer component description that is re-usable in the architecture process. A building block can also be a solution construct, such as a brick or piece of software that is re-usable in the implementation process. A solution building block is an implementation of an element of the architecture.

Architectural Building Blocks (ABBs)

- Define what generally is needed
- Capture business/technical requirements
- Are technology aware
- Direct and guide solution building blocks

Solution Building Blocks (SBBs)

- Define the “how”
- Are the implementation
- Fulfill business requirements
- Are delivered as products

There is an obvious relationship between architecture and solution building blocks. There is also a relationship between architecture and/or solution building blocks and The Open Group Standards Information Base as architecture building blocks point to standards and solution building blocks adhere to standards.

Building blocks make either the architecture process or the implementation process easier by enabling re-use. As long as a building block meets the needs and constraints, it can be re-used regardless of the "internals" of the building block. However in order for building blocks to be useful in these processes, they must be stored and accessible in ways conducive to the processes. Building blocks that are unknown are useless.

Purpose of Scenario

This scenario envisions how the Building Blocks Information Base (BBIB) will be used. It provides an overview of the users of the BBIB, and lists the processes supported by the BBIB. The internal goal is to ensure that building blocks are accessible to all that need them. The external goal is to lower costs by increasing the rate of re-use of architectural and solution oriented building blocks.

Typically, subject matter experts transfer building block-like information with presentations. Less so in the product side and more so in the architecture side, the consumer of this information is left with presentation material to be interpreted over and over again. Much information is lost due to this interpretation. Quantifying this seems impossible, but it is obvious that this "non-architected" approach requires vast amounts of money to recreate redundant and conflicting solution components. This scenario presents the BBIB, a mechanism used to facilitate the storage and access of building blocks information to address these issues.

Objectives

The following are proposed objective categories that this scenario supports. Detailed metrics will be added as we gain agreement on the objective categories.

- Improve re-use within the TOG community.
- Improve the quality of IT systems within the TOG community.
- Improve applicability of IT systems within the TOG community.
- Improve cycle time for delivery of IT systems within the TOG community.

Views of Environments and Processes

Figure 1 below depicts the "typical" business environment relative to the scope of this scenario. Within the business environment there are two main domains. The first is the external domain where partners, or vendors of the software and hardware, and consortia reside. The other is the internal domain that is comprised of a corporate headquarters where procurement typically occurs, and various locations where development projects occur. In the typical business environment there are many development shops, where each shop has between 5 and 200 people involved in development. For procurement there may be very few people involved. Of paramount importance is the notion that the entire business environment is connected.

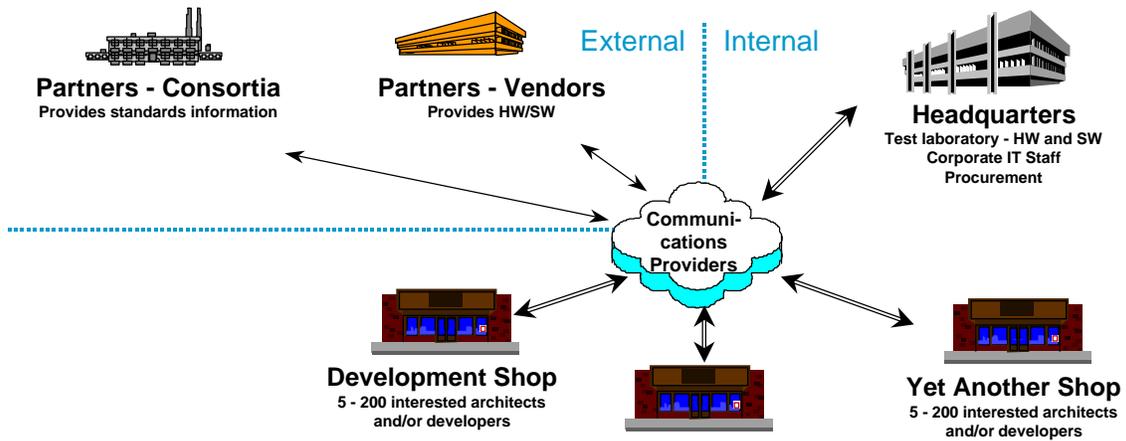


Figure 1 - A Typical Business Environment

Critical to this scenario is the inspection of the typical development shop and the headquarters. Two views are provided below for each of these, a business environment view and a technology environment view. These views are presented at a high level, avoiding unnecessary detail at this point.

Figure 2 depicts a business environment view of a typical shop. In the shop there is the architect area, the developers area, and the IT infrastructure area that provides local resources and connectivity to the "outside" world. In this example the business is tightly coupled to the technology. The business is about the definition, development, and delivery of IT systems.

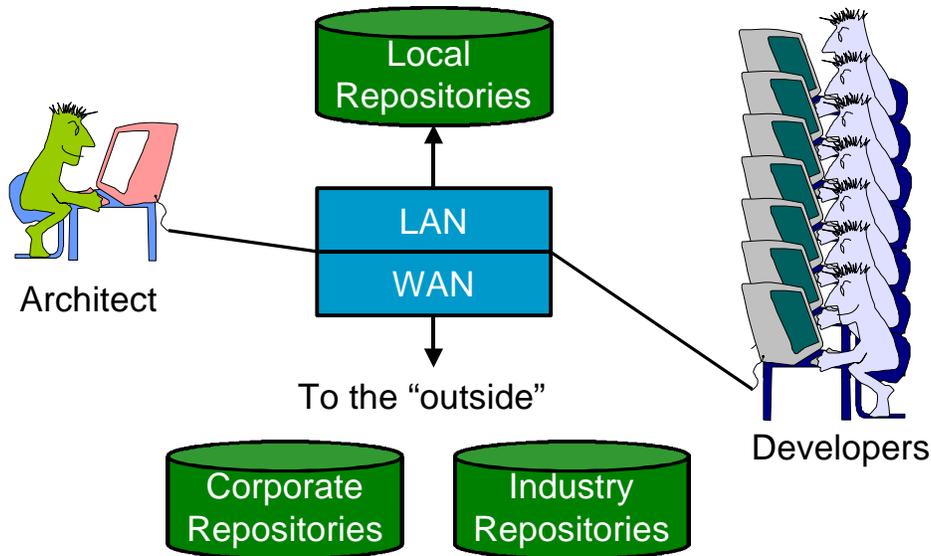


Figure 2 - Business Environment View of a Shop

The architect and developer areas are very similar from a technology point of view. Difference consist of the various tools used by the architect and developer. The architect is more likely to be using high level modeling tools, while the developer is likely to use lower level design and coding tools. However, in the best case scenario, the tools used by the architect should create information useable by the developer, and the tools used by the developer should produce information useable by the architect. Each of these cases should be facilitated by the tools supporting the information exchange. Note that the architect is the

steward/owner of architecture tool output, and the developer is the steward/owner of the development tool output. This ownership must be respected. The remaining area is the IT infrastructure area, which is obviously spread throughout the business environment since it's a combination of PC, laptops, servers, databases, directories, and wires.

The technology environment is depicted in Figure 3. Highlighted are the types of applications that support the developer and architect. This building block information base scenario must handle interoperation of information between the development and architecture tools. In addition to the architect and development workstations, there is a LAN environment which connects the workstations to a Development Shop Processor (note this is a term coined here).

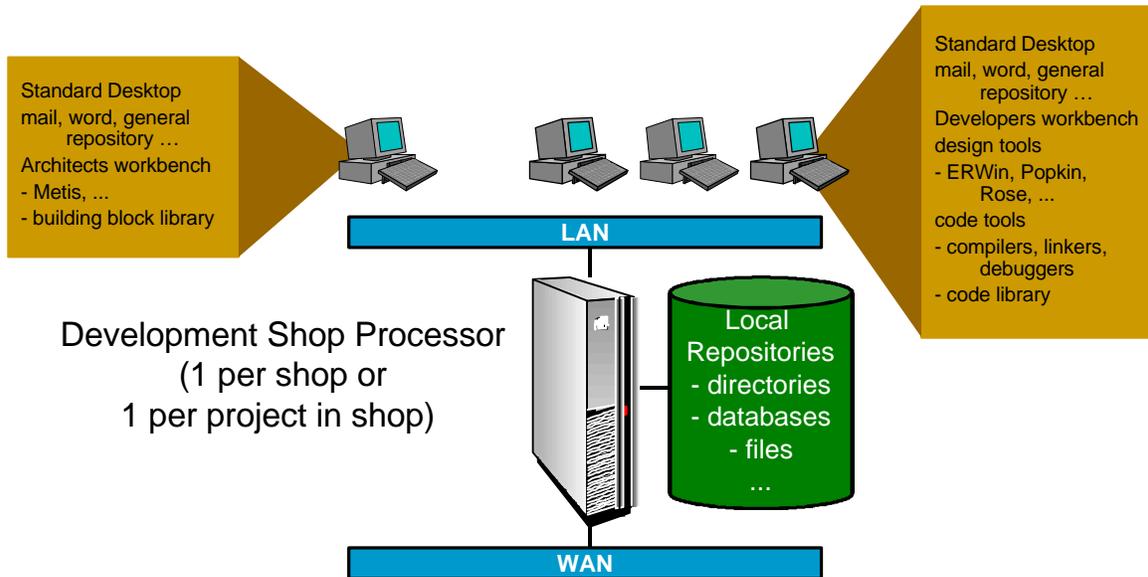


Figure 3 - Technology Environment View of a Shop

The headquarters business environment view is depicted in Figure 4 below. This appears to be too simple, but does seem to depict the typical environment as one being very "paper intensive."

Corporate Procurement

piles of regulations that must be met
 must justify each and every purchase
 typically paper intensive



Figure 4 - Business Environment View of Headquarters

The technology environment view in Figure 5 shows how the procurement department connects with the corporate server. Typically this is done to connect with corporate databases and corporate applications that

support the procurement process, such as financial applications. The Corporate server connects to other departments at other locations through the LAN and WAN.

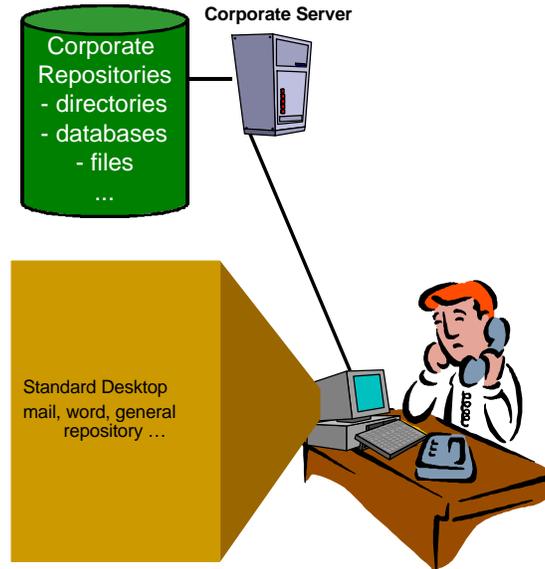


Figure 5 - Technology Environment View of Headquarters

Each of these environment views provides insights into the interactions required to support the architecture, development, and procurement processes. The next section provides details of the actual processes and how they map to the above views.

Process Descriptions

The involved processes are described very briefly in the following table. These processes are implemented differently by different organizations, so one should not read these as a guide for the process, but rather to explain what is required from the building block information base. The step description and input/output is scoped to be relevant to the BBIB.

Architect Process Steps	Step Description and Input/Output
Initiation and Framework	<p>Identify requirements, initiate architecture development cycle.</p> <p>This step produces candidate architecture and solution building blocks that should be stored in the BBIB, marked as candidates, and identified as belonging to this project.</p>
Baseline Description	<p>Capture relevant existing environment.</p> <p>This step requires access to reusable solution building blocks that should come from the BBIB.</p> <p>This step produces a refinement to candidate architecture and solution building blocks that should be stored in the BBIB, marked as candidates, and identified as belonging to this project.</p>
Target Architecture	Define target architecture.

	<p>This step requires access to reusable architecture and solution building blocks that should come from the BBIB. <i>Note that here, and henceforth, when a step requires access to reusable building blocks, this should be considered a research step. In the research step the scope of the search may include going to BBIBs outside of a company, e.g. to partners. In this case a "synchronize" request may result. The "synchronize" request will bring, either in whole or part, a BBIB from one location to another and then would allow localization of the information in the BBIB.</i></p> <p>This step produces approved architecture building blocks that should be stored in the BBIB, marked as approved, and identified with an owning organization.</p>
Opportunities and Solutions	<p>Evaluate and select major work packages.</p> <p>This step requires access to reusable solution building blocks that should come from the BBIB.</p>
Migration Planning	<p>Prioritize work, develop outline plan.</p> <p>This step requires access to reusable architecture and solution building blocks that should come from the BBIB.</p>
Implementation	<p>Develop full plan and execute.</p> <p>This step requires access to reusable solution building blocks that should come from the BBIB.</p> <p>This development process that supports this step produces reusable solution building blocks that should be stored in the BBIB, marked as implemented, and identified with an owning organization.</p>
Architecture Maintenance	<p>Establish procedure for maintenance of new baseline.</p> <p>This step produces architecture and solution building blocks that should be stored in the BBIB, marked as approved or implemented, and identified with an owning organization.</p>

Development Process Steps ¹	Step Description and Output
Inception	<p>Specifying the end-product vision and its business case, defining the scope of the project.</p> <p>This step should produce candidate architecture and solution building blocks that should be stored in the BBIB and marked as candidates, and identified as belonging to this project.</p>

¹ These are the steps of the popular Rational Unified Process

Elaboration	<p>Planning the necessary activities and required resources; specifying the features and designing the architecture.</p> <p>This step requires access to reusable architecture and solution building blocks that should come from the BBIB.</p> <p>This step produces more detail in the design of the architecture building blocks that should be stored in the BBIB and marked as designs, and identified with an owning organization.</p>
Construction	<p>Building the product and evolving the vision, the architecture, and the plans until the product--the completed vision--is ready for transfer to its users' community.</p> <p>This step produces reusable solution building blocks that should be stored in the BBIB and marked as implemented, and identified with an owning organization.</p>
Transition	<p>Making the transition from the product to its users' community, which includes: manufacturing, delivering, training, supporting, maintaining the product until the users are satisfied.</p>

Procurement Process Steps ²	Step Description and Output
Acquisition planing	<p>This step creates the plan for the purchase of some component. For IT systems the following considerations are germane to building blocks.</p> <p>This step requires access to architecture and solution building blocks that should come from the BBIB.</p> <ul style="list-style-type: none"> ▪ The procurer needs to know what architecture building blocks apply constraints (standards) for use in assessment and for creation of RFOs. ▪ The procurer needs to know what candidate solution building blocks adhere to these standards. ▪ The procurer also needs to know what suppliers provide accepted solution building blocks and where they have been deployed. <p>The procurer receives technical data from suppliers and it would be best if this data conformed to the building block definition.</p>
Concept exploration	<p>In this step the procurer looks at the viability of the concept. Building blocks give the planner a sense of the risk involved; if many architecture or solution building blocks exist that match the concept, the risk is lower.</p> <p>This step requires access to architecture and solution</p>

² These steps need honed

	<p>building blocks that should come from the BBIB. The planner needs to know what architecture building blocks apply constraints (standards), and needs to know what candidate solution building blocks adhere to these standards.</p>
Concept demonstration and validation	<p>In this step, the procurer works with development to prototype an implementation. The procurer recommends the reusable solution building blocks based upon standards fit, and past experience with suppliers.</p> <p>This step requires access to reusable solution building blocks that should come from the BBIB.</p>
Development	<p>In this step the procurer works with development to manage the relationship with the vendors supplying the solution building blocks. Building blocks that are proven to be fit for purpose get marked as approved.</p> <p>This step requires an update of the status to "procurement approved" of a solution building block that should be in the BBIB.</p>
Production	<p>In this step, the procurer works with development to manage the relationship with the vendors supplying the solution building blocks. Building blocks that are put into production get marked appropriately.</p> <p>This step requires an update of the status to "in production" of solution building blocks that should be in the BBIB, with the system identifier of where the building block is being developed.</p>
Deployment	<p>In this step, the procurer works with development to manage the relationship with the vendors supplying the solution building blocks. Building blocks that are fully deployed get marked appropriately.</p> <p>This step requires an update of the status to "deployed" of solution building blocks that should be in the BBIB, with the system identifier of where the building block was deployed.</p>

These processes were analyzed and the results are combined in the table below. This represents the general process of accessing the building block information base and this is the process within the scope of this scenario.

BBIB Access Process	Step Description
1) Create	Store approved architecture building blocks in the BBIB, mark as approved, and identify with an owning organization.
	Store candidate architecture and solution building blocks in the BBIB, mark as candidates, and identify as belonging to this project.
	Store detailed designs of architecture building blocks in the BBIB, mark as designs, and identify with an owning organization.
	Store solution building blocks in the BBIB, mark as implemented, and identify with an owning organization.
2) Retrieve	Access architecture building blocks from the BBIB and their constraints.
	Access reusable architecture building blocks that should come from the BBIB.
	Access reusable solution building blocks that should come from the BBIB.
	Access solution building blocks from the BBIB that adhere to standards.
	Access suppliers that provide accepted solution building blocks and where they have been deployed.
3) Update	Update candidate architecture and solution building blocks stored in the BBIB marked as candidates, and identified as belonging to this project.
	Update the status of a solution building block to "deployed" in the BBIB along with the system identifier of where the building block was deployed.
	Update the status of a solution building block to "in production" in the BBIB along with the system identifier of where the building block is being developed.
	Update the status of a solution building block to "procurement approved" in the BBIB.
4) Delete	Update the status of a solution building block to retired.
	Update the status of an architecture building block to retired.
5) Synchronize	Download an entire BBIB from a partner.
	Synchronize local BBIB with partner BBIB.

Process Steps Mapped to Environment

The graphic below depicts where the BBIB accesses map within the environment. The accesses themselves are labeled "A" for architecture process accesses, "D" for development process accesses, and "P" for procurement process accesses. The process label is followed by the number of the access, 1) for create, 2) for retrieve, 3) for update, 4) for delete (retire), and 5) for synchronize.

Below

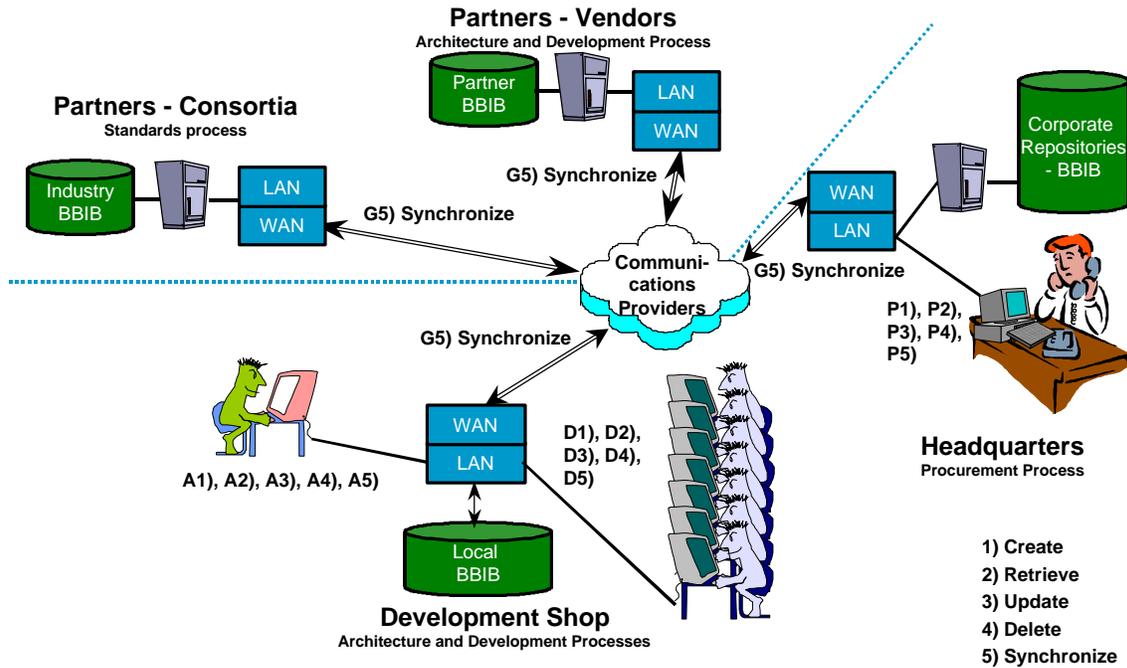


Figure 6 - Process Steps

Information Flow

The key information objects in this scenario are the building block information. The building block information must freely flow from all locations in Figure 6. Information flow across the WAN should be done through the synchronization request.

Actors and Their Roles and Responsibilities

Human Actors and Roles

The most important actors are those that are in bold print. Greater attention should be spent on these actors, in context to the main purpose of this scenario - creating and/or accessing building block information.

Actor	Role
Architects	<ul style="list-style-type: none"> effectively and efficiently creates architectures of IT solutions that support the business
Developers	<ul style="list-style-type: none"> effectively and efficiently creates components of IT solutions
Procurers	<ul style="list-style-type: none"> manages purchasing of sub-elements of an IT solution cost effectively

Computer Actors and Roles

The most important computer actors are those that are in bold print. Greater attention should be spent on these actors, in context to the main purpose of this scenario - providing support for the creation and/or access of building block information.

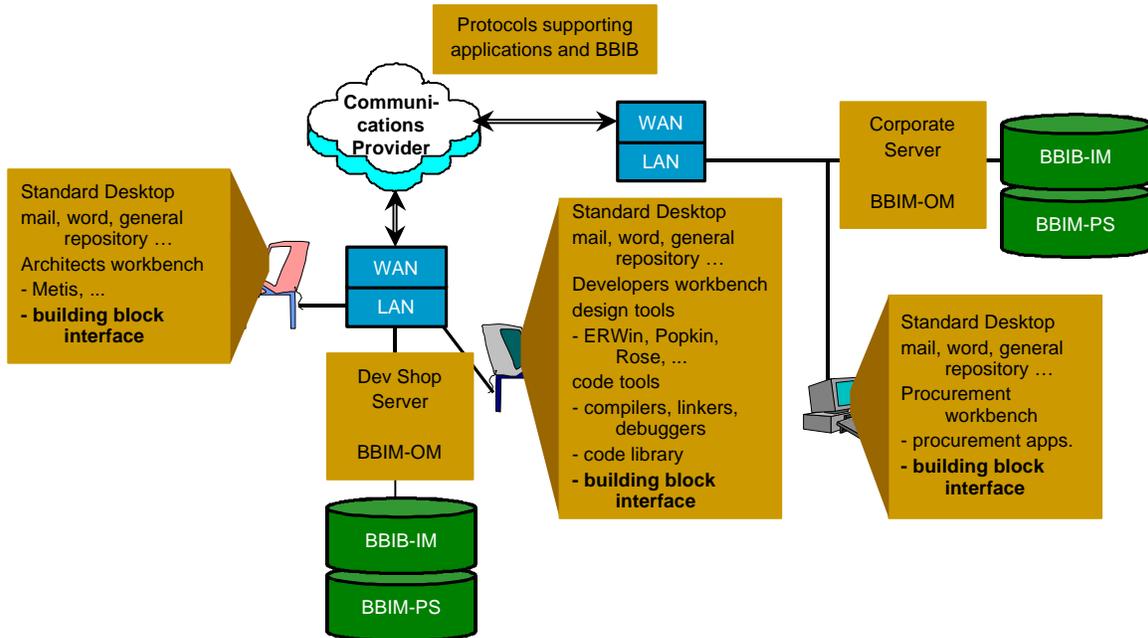


Figure 7 - Computer Actors

Actor	Role
User applications - architecture,	<ul style="list-style-type: none"> User applications support the tasks of the architect, developer and procurer. Optimally

design and development, and procurement	the user applications are integral parts of the processes used by the human actors that perform the roles.
BBIB client side interface	<ul style="list-style-type: none"> • The BBIB client side interface provides the user with access to the building blocks information. • The client side interface process access to both the human user and to applications.
Information model of BBIB	<ul style="list-style-type: none"> • The information model of the BBIB provides the client side interface with the necessary meta-data to understand how to fulfill requests from the user or applications. • The information model provides the object manager with the necessary meta-data to understand how to fulfill requests from the client side interface.
Object Manager of BBIB	<ul style="list-style-type: none"> • The object manager fulfills requests from the client side interface. • The object manager accesses information through the persistent store.
Persistent Store of BBIB	<ul style="list-style-type: none"> • The persistent store provides the necessary services to access and manage the data with high integrity.
Development Shop Processor	<ul style="list-style-type: none"> • The development shop processor serves the user and applications from a local perspective. • The development shop processor hosts the object manager and is attached to the persistent store.
Corporate Server	<ul style="list-style-type: none"> • The corporate server serves the user and applications from a local perspective. • The corporate server hosts the object manager and is attached to the persistent store. • The corporate server hosts the processing to fulfill synchronize requests.
LANs	<ul style="list-style-type: none"> ▪ LANs provide local communication facilities
WANs	<ul style="list-style-type: none"> ▪ WANs provide wide area communication facilities
Application Protocols	<ul style="list-style-type: none"> ▪ Application protocols provide the necessary application level formats and access protocol to access building block information

Further information about these actors is documented below in the requirements section.

Requirements

Description of a Useful Building Block Information Base

A necessary tool that facilitates the architecture or implementation processes is an "information base." The building blocks information base provides a *standard* way to store and access building block information. The building block information base should include:

- An information model
- An object manager
- A persistent storage service

A *standard* building block information base provides open ubiquitous access.

However implementing an information base is not sufficient. A sufficient implementation is one that also provides:

- Tools and applications that encourage and or require new building blocks to be saved and maintained in the information base.

To maximize re-use, the tools and application that support the architecture and implementation processes must use the information base and provide the user with the information in a natural and friendly manner. The users fall into the following classes:

- 1) architects
- 2) developers
- 3) procurers

The following are considered natural requests likely to be made to the information base by any of the given users.

User oriented requests that must be supported from an open client interface, such as a web client:

- The user is likely to request access (search or browse) to solution or architecture building blocks selectively based upon building block elements or properties.
- The user is likely to request access to building blocks that specify or support a specific standard.
- Architects are likely to search for architecture building blocks that meet certain constraints, fit certain properties, match certain requirements.
- Developers are likely to request all solution building blocks that implement an architecture building block, in part or whole.
- Procurers are likely to request all solution building blocks that implement a standard, or are branded.
- The user is likely to request all architecture or solution building blocks related to a specific architecture or solution building block. (In what building blocks is building block A used, and/or what building blocks does building block A use.)

The following describes what it takes to facilitate this process within the information base per the elements of the information base.

The information model:

- The information model should reflect a standard schema that describes the fundamental elements of building blocks (appendix a).
- The information model should support specific properties of a building block, e.g. extensible properties (appendix b).
- The information model itself should be easily extensible.

The object manager:

- Automatic caching of related building blocks should be supported.
- The building block object should be presented to the requester either in part or whole. That is automatic composition of a complete building block should be supported.
- The object manager should ensure that the minimum properties of a building block be present before storing. E.g. required elements described in Appendix A and B.
- Should be globally accessible.

The persistent storage service:

- Must provide versioning for building block objects.
- Must provide audit trail for modifications.
- Must provide backups for recovery.
- Must provide dynamic joining to compose complex building blocks.
- Should provide dynamic indexing to facilitate extensibility.

A standard building block information base provides open interfaces.

- The interfaces and drag and drop metaphor must match those of the standard modeling application to facilitate drag and drop of selected building blocks from the information base to the modeling applications.
- The open interfaces must provide access to web client and server applications.
- The open interface must provide an accepted open query language and programming interface.

Finally, a standard building block information base should be capable of being downloaded, and synchronized within any compatible environment. This will facilitate wide adoption and use by enabling localized security and processing requirements to be met.

Resulting Technology Architecture Model

Constraints

IT Principles

The building block information base will have standard interfaces.

The building block information base will be based upon a standard platform.

There will be open protocols for use of building block information among the tools.

Technology Architecture Supporting the Process

The following figure depicts how the technology environment supports the processes.

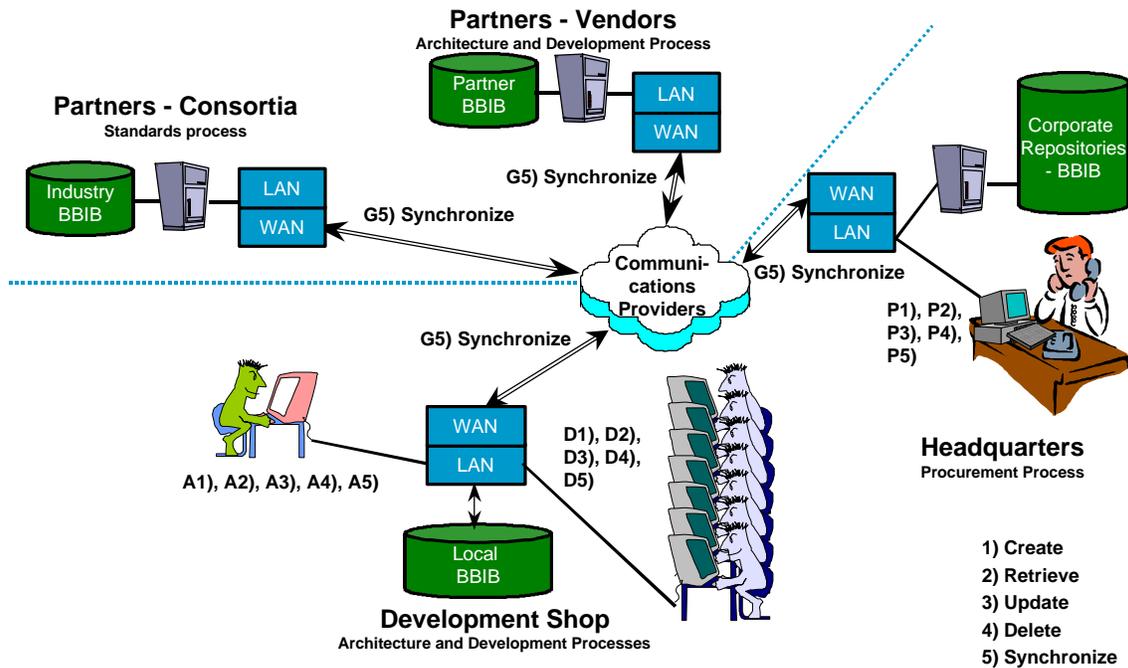


Figure 8 - Technology View of Process

Conceptual List of Components of Building Blocks Information Base

Client side

It is envisioned that a user; an architect, a developer, or a procurer, could access the building block information base from a client side device using a browser. It is further envisioned that the device may have other tools, such as modeling tools, development tools, or purchasing tools co-resident and active at the same time as the building block information base is being accessed. The user should be able to browse or search the building block information base and then selectively pull one or more building blocks into one of the other applications.

Web server side

It is envisioned that the web server provides the required features and functions such as security validation. It is also envisioned that the web server hosts application code that encourages new building blocks to be saved and maintained in the information base, e. g. an incentive program. The web server must also host code that manages the building block information base. The web server must have access to the building block information model.

Database server side

It is envisioned that the database server provides the object management functionality, such as processing the information model, providing caching, and checking integrity constraints. The database server may provide access to building block information in multiple, geographically dispersed locations. Finally the database server handles the persistent store.

Appendix A - ADML Description

ADML can be used to describe components of an architecture and its interconnections with other components of an architecture. The high level structure of ADML documents are provided below to help guide the information model of the building block information base. Please note the reference of record should be the ADML document type definition.

FAMILY

REPRESENTATION

SYSTEM

- Component
 - Port
 - Property
 - Representation
- Connector
 - Role
 - Property
 - Representation
- Port
 - Property
 - Representation
- Role
 - Property
 - Representation
- Property
 - PropertyType
 - PropertyValue
- Attachments
 - Attachment
 - PortID
 - RoleID
 - Representation

Appendix B - Building Blocks Description with ADML Extensions

In essence a building block is an enhanced description of an ADML component. It is an ADML component that has additional attributes described through the property mechanism of ADML³. Additional properties are such items as:

- type of building block - architecture, detailed design, or solution
- status of the building block - candidate, approved, in production, implemented, deployed, ...
- projects where building block are being used
- owner of the building block - organization and person
- constraints - standards required, performance characteristics, ...
- standards supported -
- suppliers -
- date

Below is the high level structure of ADML augmented with property information.

FAMILY

REPRESENTATION

SYSTEM

Component

Port

Property

Representation

Connector

Role

Property

Representation

Port

Property

Representation

Role

Property

Representation

Property

PropertyType

(such as BB type, status, projects, owner, constraints, standards, suppliers, date, ...)

PropertyValue

(per PropertyType, e.g. for BB type - architecture, design, solution; for BB status - candidate, approved, in production, implemented, deployed)

Attachments

Attachment

PortID

RoleID

Representation

³ Note: the properties mechanism in ADML allows for the definition to be described in another XML document. This gives great flexibility in evolving these properties over time.