

The Role of the Enterprise Architect in Business Process Reengineering

Richard Freggi, Senior Supply Chain Architect, HP Inc.

Abstract— Business process reengineering is often a challenging undertaking. This paper is a case study, sharing practical experience of how the Enterprise Architect can help in three ways:

- Provide a common language allowing different organizations, consultants and IT teams to communicate effectively
- Set the right level of abstraction to facilitate analysis and solution of complex questions
- Reconcile user's wants and needs with the capabilities and constraints of IT systems

Reference is made to the Zachman Framework, especially the columns for "Data", "Function" and "People"; and how these columns can be used to interact with stakeholders using UML (Unified Modeling Language).

Index Terms — Business Process Re-engineering, Enterprise Architecture, Methodology, UML, Zachman Framework

I. INTRODUCTION

Currently several industries are undergoing reorganizations in an effort to adapt to new business challenges.

These changes inevitably drive the need for business process reengineering. It is often a long, difficult and expensive undertaking, usually led by line managers with the assistance of external consultants.

Typically the Reengineering work starts with business process models, then documentation of IT system requirements, and finally discussion with the IT department on the systems needed to enable the processes. The enterprise architect is often engaged in this latest stage of IT discussion.

Experience over several-cross functional projects has shown that the architect can use specific artifacts to simplify and speed up business process definition. By using the right artifacts, the architect can also ensure that IT system requirements and constraints are integral part of process definition, and not a separate after-thought.

This paper assumes reader knowledge of the Zachman Framework and UML.

II. MOTIVATION AND PURPOSE

This paper aims to generate awareness and stimulate thinking on how to apply enterprise architecture in business-driven activities such as process development. It is a synthesis of lessons learned during several cross-functional projects with worldwide scope. Emphasis is on practical application of well-established tools and techniques.

III. PROBLEM DEFINITION

Some of the challenges common to Business Process Reengineering are:

A. Understand the context and unique situations impacting stakeholders

Stakeholders may have widely differing opinions and assumptions. There may be some hidden issues, or they may have concerns on risks caused by changes in process and roles and responsibilities. This can dramatically impact business process development, so it's important to identify and understand the context as soon as possible.

B. Reconcile the culture, assumptions and terminology of different organizations

In large organizations it's common to find different operating models and terminologies. A common language is needed to help stakeholders communicate effectively.

C. Generate consensus over diverging goals and expectations

Different organizations may have different opinions on process goals and metrics. Consensus is necessary to develop reliable, stable cross-functional processes.

D. Generate new ideas with significant benefits instead of just rearranging existing Processes

Many reengineering projects aim to achieve radical performance improvements. To do this stakeholders must find ways to promote creativity and think "out of the box".

E. Reduce the cost and risk to implement new processes

Implementation projects can run into cost overruns or schedule delays. For example, unforeseen dependencies may require a process redesign; or IT systems may not perform to required levels. These and other risk factors should be considered and mitigated during process definition.

Project teams tasked with implementing the reengineered processes need a clear definition of deliverables, scope, and system / process dependencies. This helps in development of the project plan and risk management strategy.

IV. ROLE OF THE ENTERPRISE ARCHITECT

The enterprise architect can support business process reengineering in three ways:

A. Provide a common language to communicate effectively between different stakeholders

Stakeholders may include senior managers, line managers and line workers; various organizations including the IT department; external consultants; and sometimes third parties such as business partners. Each has their own background and terminology.

The architect can develop a glossary of key terms, including the names of key roles performing a process and definitions of the information required by each process.

This is useful in several ways: for example stakeholders may talk at length about different processes performed in different ways by different people, without realizing that they are essentially identical. By showing that the two processes consume and generate the same information, it becomes much easier to see that they are probably the same process. This can be a very powerful tool, especially when dealing many stakeholders using different terminologies.

The glossary can be derived for existing architectural documents, or from industry standard models [1], [2].

B. Set the right level of abstraction to facilitate answering difficult questions

Sometimes the reengineering effort can bog down or get stuck on a topic that is difficult or poorly understood. The architect can help stakeholders to look at the problem from the right point of view.

In some cases it might be easier to decide on a specific process change by considering the ‘big picture’, such as the organization areas of accountability; other times it may be better to look at details such as specific constraints of an IT system.

Looking at the issue from the right point of view allows the architect to ask the right questions at the right time, and this helps the reengineering get back on track.

C. Reconcile user’s wants and needs with capabilities and constraints of IT systems

The architect can maintain an overview of process changes vs. system capabilities. This overview can be used to facilitate an ongoing dialogue between stakeholders, including the IT department.

When a process change is under discussion, the architect can support good decision-making based on present or future system capabilities.

If an IT system is under discussion, the architect can support good decision making based on the present or future process requirements. This can help to rationalize the workload of the IT department, for example by aggregating and postponing system enhancements until a consistent group of requirements is finalized, and all dependencies are understood.

V. METHODOLOGY AND TOOLS

The architect can use artifacts from mainstream frameworks and methodologies. In this case study we use the Zachman Framework [3],[4].

If the business process is simple, well understood and with few dependencies, then the reengineering may be successful with no need of architect’s intervention. This case is shown in Fig. 1 as a single cell highlighted in the Zachman column ‘Function’.

However if the process scope, complexity and functional /

system dependencies are extensive, then the architect can leverage additional Zachman Framework cells. In general, the benefit of using more cells increases in proportion to the complexity of reengineering.

	DATA what	FUNCTION how	NETWORK where	PEOPLE who	TIME when	MOTIVATION why
CONTEXT General Manager	Things important for the business	Business Functions	List of business locations	Organization structure	Business strategy timeline	List of business priorities
CONCEPT Line Manager	Semantic data dictionary	Business Process	Business Logistic System	Roles and responsibilities	Business event timeline	Business Plan and Budget
LOGICAL Architect	Logical Data Model	Application Architecture	Distributed System Architecture	Software interface	Processing timing and sequencing	Business Role Model
PHYSICAL IT Manager	Database Schema	Software Specification / Configuration	Hardware and network infrastructure	Presentation Architecture	Control structure	System Role Design
OUT OF CONTEXT Subcontractor Implementor	Database Definition Language	Program code	System components	Access and security system	Timing definition	System Role Configuration

John A. Zachman, Zachman International (810) 231-0531

Fig. 1: The Zachman Framework for Enterprise Architecture (cell for Business Process Reengineering is highlighted)

Derived from: "Zachman Framework Detailed" by Marcel Douwe Dekker based on earlier work of Phlegg2 et al. - self-made, combination of File:Zachman Framework Basics.jpg and File:Zachman Framework.jpg. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Zachman_Framework_Detailed.jpg#/media/File:Zachman_Framework_Detailed.jpg

Each Zachman Framework cell can be represented by artifacts; the choice of artifacts depends on the specific circumstances and the architect’s preferences. Fig. 2 summarizes the most common Zachman artifacts for business process reengineering.

BPR Challenge	Enterprise Architect Role	Artifact	Notation
Understand context	Common language Right level of abstraction	Business Functions Organization Structure Semantic Data Dictionary Business Priorities	UML Package Diagram UML Package Diagram UML Class Diagram Text (MS Excel sheet)
Reconcile assumptions	Common language Right level of abstraction	Semantic Data Dictionary Functional Model	UML Class Diagram UML Use Case Diagram
Generate consensus	Common language Right level of abstraction Needs vs system capabilities	Semantic Data Dictionary Logical System Architecture	UML Class Diagram UML Component Diagram
Generate new ideas	Common language Right level of abstraction	Standard Operat. Procedures Logical System Architecture Process timing	UML Sequence Diagram UML Component Diagram UML Timing Diagram
IT innovation	Common language Right level of abstraction Needs vs system capabilities	Semantic Data Dictionary Functional Model Logical System Architecture	UML Class Diagram UML Use Case Diagram UML Class Diagram
Reduce cost and risk	Common language Needs vs system capabilities	Semantic Data Dictionary Functional Model Logical System Architecture Project Charter	UML Class Diagram UML Use Case Diagram UML Class Diagram Text (MS Word)

Fig. 2: UML artifacts for Zachman Framework cells supporting business process reengineering

Using the Unified Modeling Language (UML) to document the artifacts for Zachman Framework cells [5], [6] has some advantages over more traditional notation such as flowcharts, Data Flow Diagrams etc., for example:

- UML Class Diagram, Sequence Diagram and Component Diagram are fully consistent with each other (any change in Class Diagram will be reflected in the other diagrams and vice versa, as shown in Fig. 3). This helps us move laterally across the Zachman Framework rows.
- UML allows to easily drill down to detail or zoom out to abstraction as needed. For example we may want to look at a Sequence Diagram to discuss the detailed responsibilities of an Actor, then consider the Package Diagram for high level overview of dependencies. In other words, UML Objects relate to the same Classifier as we change the level of abstraction: this helps us move vertically along the Zachman Framework columns.
- UML has excellent documentation and many Open Source tools, so anyone who wants to participate can do it quickly and cheaply.

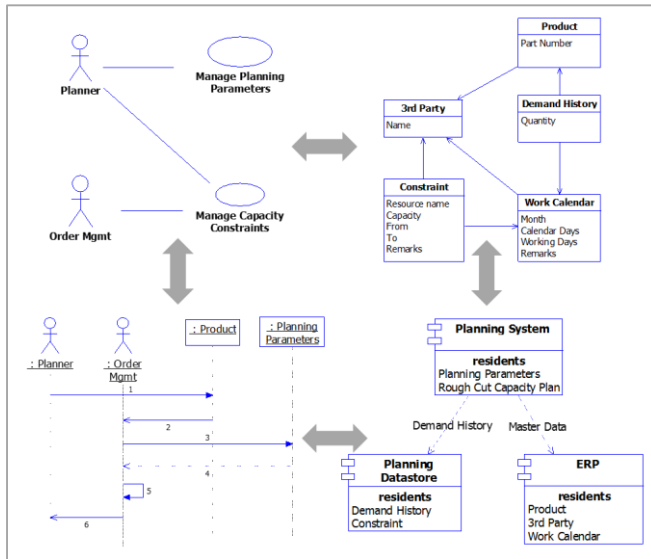


Fig. 3: UML can represent most Zachman Framework cells, and also provides mechanisms to maintain consistency across cells

Below are some practical examples of how the artifacts support process reengineering.

A. Understand the context and unique situations impacting stakeholders

	DATA what	FUNCTION how	NETWORK where	PEOPLE who	TIME when	MOTIVATION why
CONTEXT General Manager	Things important for the business	Business Functions	List of business locations	Organization structure	Business strategy timeline	List of business priorities
CONCEPT Line Manager	Semantic data dictionary	Business Process	Business Logistic System	Roles and responsibilities	Business event timeline	Business Plan and Budget
LOGICAL Architect Project Manager	Logical Data Model	Application Architecture	Distributed System Architecture	Software interface	Processing timing and sequencing	Business Role Model
PHYSICAL IT Manager	Database Schema	Software Specification / Configuration	Hardware and network infrastructure	Presentation Architecture	Control structure	System Role Design
OUT OF CONTEXT Subcontractor Implementor	Database Definition Language	Program code	System components	Access and security system	Timing definition	System Role Configuration

Fig. 4: Zachman Framework cells used for "Understand context"

Business priorities can be listed as text entries in MS Word or Excel. They provide a useful criteria to facilitate decision-making during process definition.

Package Diagrams are useful to facilitate discussion on topics such as what are the major functional areas in scope and what are their interdependencies.

Information subject areas impacted by the process can also be expressed as Packages.

The resulting Package structure can be used to drive agreement on governance, for example by discussing what Actors, Use Cases and Components should be owned by what Package.

In this case study, stakeholders agree on a contextual model with four Business Functions, namely: Operations, Planning, Fulfillment and 3rd Parties as shown in Fig. 3.

Dependencies can be agreed on – for example 3rd Parties receives Exception notices directly from Fulfillment instead of via Operations.

Well-constructed Package Diagrams with clear dependencies are a powerful tool for communication and facilitation with cross-functional teams.

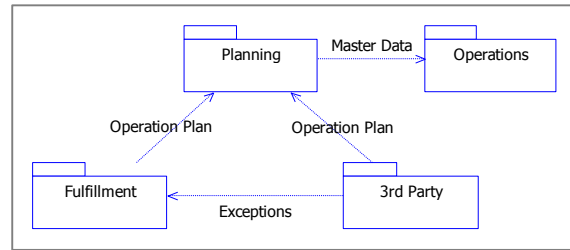


Fig. 5: Artifact for "Understand context" (UML Package Diagram)

B. Reconcile the culture, assumptions and terminology of different organizations

	DATA what	FUNCTION how	NETWORK where	PEOPLE who	TIME when	MOTIVATION why
CONTEXT General Manager	Things important for the business	Business Functions	List of business locations	Organization structure	Business strategy timeline	List of business priorities
CONCEPT Line Manager	Semantic data dictionary	Business Process	Business Logistic System	Roles and responsibilities	Business event timeline	Business Plan and Budget
LOGICAL Architect Project Manager	Logical Data Model	Application Architecture	Distributed System Architecture	Software interface	Processing timing and sequencing	Business Role Model
PHYSICAL IT Manager	Database Schema	Software Specification / Configuration	Hardware and network infrastructure	Presentation Architecture	Control structure	System Role Design
OUT OF CONTEXT Subcontractor Implementor	Database Definition Language	Program code	System components	Access and security system	Timing definition	System Role Configuration

Fig. 6: Zachman Framework cells used for "Reconcile culture"

Business process goals the Actors involved in achieving the goals can be expressed with Use Case Diagrams.

Class Diagrams are useful to identify the key business information needed to perform the processes. Both diagrams provide a clear, concise and flexible basis to develop glossaries.

For example as shown in Fig. 7, we identify the key business information and eliminate misunderstandings or ambiguity by looking at Attributes and Associations. For example, we consider the concept of 'Work Calendar' and understand the process need of having the Demand History, Constraints and Rough Cut Capacity plan expressed in terms of exactly the same Calendar to avoid errors and delays.

The Class Diagram also helps us validate the Dependencies between Packages and adjust them if needed: Work Calendar is part of the 3rd Party Package, while Demand History, Constraints and Rough Cut Capacity are in different Packages.

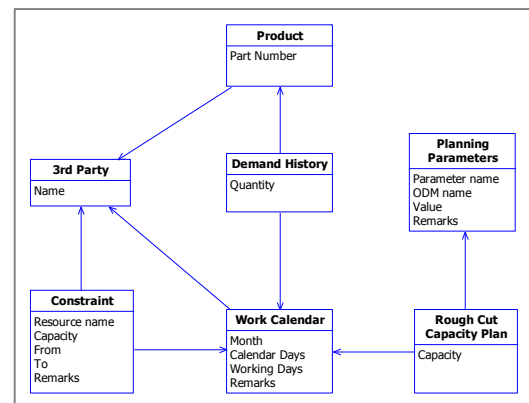


Fig. 7: Artifact for "Reconcile culture" (UML Class Diagram)

C. Generate consensus over diverging goals and expectations

	DATA what	FUNCTION how	NETWORK where	PEOPLE who	TIME when	MOTIVATION why
CONTEXT General Manager	Things important for the business	Business Functions	List of business locations	Organization structure	Business strategy timeline	List of business priorities
CONCEPT Line Manager	Semantic data dictionary	Business Process	Business Logistic System	Roles and responsibilities	Business event timeline	Business Plan and Budget
LOGICAL Architect Project Manager	Logical Data Model	Application Architecture	Distributed System Architecture	Software interface	Processing timing and sequencing	Business Role Model
PHYSICAL IT Manager	Database Schema	Software Specification / Configuration	Hardware and network infrastructure	Presentation Architecture	Control structure	System Role Design
OUT OF CONTEXT Subcontractor Implementor	Database Definition Language	Program code	System components	Access and security system	Timing definition	System Role Configuration

Fig. 8: Zachman Framework cells used for "Generate consensus"

Consensus can be achieved by cycling through the appropriate Zachman Framework cells and working with stakeholders to gradually resolve gaps and contradictions.

For examples, the list of business priorities can be used as criteria to evaluate the Use Cases and Class Diagrams (e.g. what needs to be done to satisfy the priorities; what information is important and what is not).

Actors and their responsibilities can be harmonized to the Package Diagrams; and prioritization of IT systems can be determined by mapping their support for the Use Cases and Classes, as shown in Fig. 9.

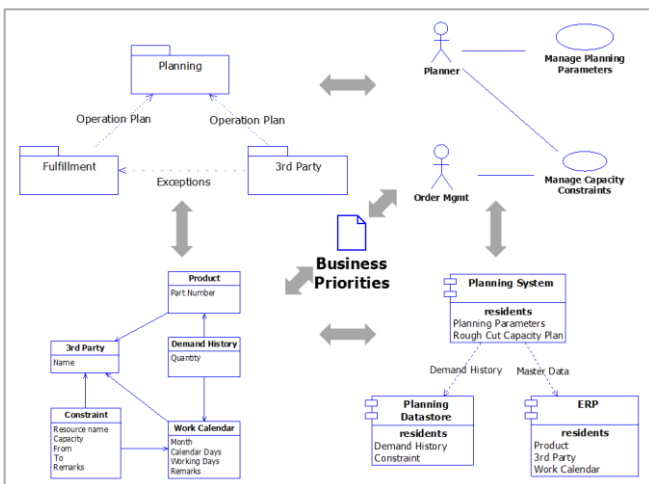


Fig. 9: Artifacts for "Generate consensus" (Business Priorities List and UML Package, Use Case, Class and Component Diagram)

D. Generate truly new ideas with quantifiable benefits instead of just rearranging existing Processes

	DATA what	FUNCTION how	NETWORK where	PEOPLE who	TIME when	MOTIVATION why
CONTEXT General Manager	Things important for the business	Business Functions	List of business locations	Organization structure	Business strategy timeline	List of business priorities
CONCEPT Line Manager	Semantic data dictionary	Business Process	Business Logistic System	Roles and responsibilities	Business event timeline	Business Plan and Budget
LOGICAL Architect Project Manager	Logical Data Model	Application Architecture	Distributed System Architecture	Software interface	Processing timing and sequencing	Business Role Model
PHYSICAL IT Manager	Database Schema	Software Specification / Configuration	Hardware and network infrastructure	Presentation Architecture	Control structure	System Role Design
OUT OF CONTEXT Subcontractor Implementor	Database Definition Language	Program code	System components	Access and security system	Timing definition	System Role Configuration

Fig. 10: Zachman Framework cells used for "New ideas"

Consider definition of the context (Package Diagram); business goals (text); process goals (Use Cases), Actors; glossary (Class Diagram). Stakeholders can use any of these

elements to explore and better ways to perform a process.

Fig. 11 shows an example of how a Sequence Diagram can be used to determine how Actors could use information provided by new IT systems to work faster and with less errors.

An important benefit is that this innovation does not occur in isolation, but is already integrated with other processes and IT systems, as already shown in Fig. 3.

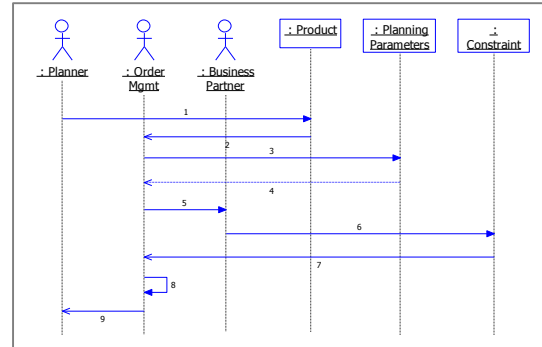


Fig. 11: Artifact for "New ideas" (UML Sequence Diagram)

E. Reduce the cost and risk to implement the new Processes

	DATA what	FUNCTION how	NETWORK where	PEOPLE who	TIME when	MOTIVATION why
CONTEXT General Manager	Things important for the business	Business Functions	List of business locations	Organization structure	Business strategy timeline	List of business priorities
CONCEPT Line Manager	Semantic data dictionary	Business Process	Business Logistic System	Roles and responsibilities	Business event timeline	Business Plan and Budget
LOGICAL Architect Project Manager	Logical Data Model	Application Architecture	Distributed System Architecture	Software interface	Processing timing and sequencing	Business Role Model
PHYSICAL IT Manager	Database Schema	Software Specification / Configuration	Hardware and network infrastructure	Presentation Architecture	Control structure	System Role Design
OUT OF CONTEXT Subcontractor Implementor	Database Definition Language	Program code	System components	Access and security system	Timing definition	System Role Configuration

Fig. 12: Zachman Framework cells used for "Reduce cost and risk"

Class Diagrams, Use Case Diagrams and Component Diagrams provide a solid starting point to kick off implementation projects. If several projects are launched in parallel, these artifacts help with overall coordination and management of intra-project dependencies.

VI. CONCLUSION

This paper presented a few examples of practical application of architectural artifacts to common challenges in business process reengineering.

Similar principles can be used to increase the business value of enterprise architecture in a wide range of reengineering engagements.

REFERENCES

- [1] An example of industry reference models are available at <http://www.apics.org/sites/apics-supply-chain-council/frameworks/scor>
- [2] Another example of an industry reference model is available at <https://www.apqc.org/>
- [3] John Zachman, <http://www.zachman.com>
- [4] Overview of the Zachman Framework is available at https://en.wikipedia.org/wiki/Zachman_Framework
- [5] J. Gorman, UML For Managers, 2005, ch. 4 page 15, available at <http://www.codemanship.co.uk/parlezuml/e-books/umlformanagers>
- [6] H. Poor, Mapping the UML to the Zachman Framework. AMCIS 2005 Proceedings