Business-Centric Methodology Specification Version 0.05

OASIS BCM Technical Committee

September 28, 2003

1 Status of this Document

This document specifies a BCM SPECIFICATION for the Business community.

Distribution of this document is unlimited.

The document formatting is based on the Internet Society's Standard RFC format.

This version:

http://www.oasis-open.org/committees/BCM/documents/

Errata for this version:

http://www.oasis-open.org/committees/BCM/documents/Errata.shtml

***Previous version*:**

4 June 2003, Version 0.01
29 July 2003, Version 0.02
19 August 2003, Version 0.03
14 September 2003, Version 0.04

2 Technical Committee Members

Bruce Peat, eProcess Solutions (Co-Chair)
Mike Lubash, Defense Finance and Accounting Service -DFAS (Co-Chair)
David RR Webber, consultant (Secretary)
Eric Okin, Defense Finance and Accounting Service -DFAS
Carl Mattocks, Checkmi
Hans Aanesen, Individual
Sally St. Amand, Individual
Laila Moretto, MITRE
Dan Pattyn, Individual
Paul Kirk, Hewlett-Packard
Bob Greeves, DOJ
Murali Iyengar, SAIC

3 Table of Contents

4 Introduction

The Business-Centric Methodology (*BCM*) for Enterprise Agility and Interoperability is a roadmap for the development and implementation of procedures that produces effective, efficient, and sustainable interoperability mechanisms. The methodology emphasizes '*Business First*'; shifting power and responsibility to the users -- customers and business domain experts. Business is defined for this specification in broad terms as the reason for an organization's existence – their functional domain. The *BCM* task is to provide an overall roadmap for developing interactions between collaboration partners and within *Communities of Interest* (*CoI*). The roadmap can be used for new development, providing guidance in defining requirements for the procurement of products, and for providing the structure for interfacing to extend legacy application and services. The *BCM* offers an approach for managers facing the problem of tying together disparate systems and services. The approach extends the traditional Enterprise Application Integration (EAI) model which only provides internal viewpoints and reengineering of an organization's processes.

Exploiting the Common and Mitigating the Differences

The critical *BCM* take-away is that of providing a holistic solution to the interoperability quandary business and technical mangers face today by providing an *organizational memory* that is persistent. This memory is also agnostic to the implementation architecture and enables business personnel to understand, direct and manage the operations. This approach is at the heart of the *BCM* and is implemented as a series of *BCM Templates* for each of the *architecture layers* that the *BCM* defines. The *BCM Templates* prompt for the information artifacts required for proper control, understanding, and building of a shared information architectural foundation. The *BCM Templates* provide for the precise communication required for not only business understanding but also for directing and controlling the application implementation. (an example set of *BCM Templates* are provided in Appendix A) . Templates can be used both internally and externally. Ideally collections of *BCM Templates* are shared across a *CoI* to foster adoption, promote re-use and align implementation efforts. The *BCM* is not intended to be an end-point solution but rather a point-of-departure for, and enabler of, downstream analysis, development and implementation.

The intent of the *BCM* is to provide flexible guidance to those tackling the difficult challenge of interoperability at both tactical and strategic levels. For instance, alignment of financial *events* between organizations take prime importance when developing an enterprise accounting architecture, whereas '*verbs*' or services take center stage when developing a series of shared core capabilities for an advanced logistics distributed solution. The *BCM* provides template prompts for a prescribed set of views, with the business manager determining the applicability of each such view to the specific business requirements. There is no pre-determined order of completion or particular emphasis to the *BCM Templates*. Instead managers are encouraged to extend the *BCM Templates* and/or create new *BCM Templates* as the need arises. As a roadmap the use of the *BCM* is dependent on the philosophy, conditions and constraints of the deployment environment and to the degree which one can integrate vs. interoperate.

The *BCM* employs an opportunistic strategy that fosters organic growth and enables self-correction by adding mechanisms for shared experiences, guidance and intelligent decisions. For instance, the *BCM* highlights the need for proper interpretation of the business language and its semantics, in context and in relation to shared domain concepts. The *BCM* uses classifications, ontology, and patterns to clarify and align the business context. By not relying on formal language syntax, the *BCM* moves the business semantics from the application into the infrastructure layer. As a result, the *BCM* provides standard mechanisms with templates that deliver a sound base to effectively negotiate operational differences and achieve information agility. In short, the *BCM* supplies the missing link that provides the Enterprise with the means to track and control information artifacts through their life cycle[1] from business vision to implementation.

The *BCM's* focus is on increasing best value within an e-Business[2] environment, by establishing precise communications between multiple communities to conduct business transactions and align their infrastructures in a timely manner as exampled in the following chart. The *BCM* reduces development time, integration resource requirements and maintenance costs through reuse and coordination of efforts.

Perspectives	"As Is"	"Can Be' (NetCentric)
Business Operations	Long-standing, stove-piped business process	Integrated business lines; addressing the whole value-chain to extend past the Enterprise
Information	Islands of information supporting isolated solutions	Manage metadata as information asset; knowledge-centric, interoperable solutions
Technology	Technology-driven, proprietary solutions	Declarative processing, open vendor solutions (i.e., open source code)
People	Crisis-driven, single focus mentality	Collaboration – Communities of Interests

In essence, the *BCM's* advantage arises from its simplicity; by adopting and following an intuitive approach for [1] unconstrained conceptual alignment, [2] *authoritative source* collaboration, [3] layering of business constraints and constructs, and [4] the capture of rationale through templates. By applying these techniques one gains *pragmatic interoperability,* as well as *semantic interoperability.*

Sharing semantics across domains and between *authoritative sources* requires an effective means to uniquely label individual artifacts. Implementers can therefore incorporate [5] *Unique IDentifier* (*UID)* references during analysis, or development, or make alignment later, to

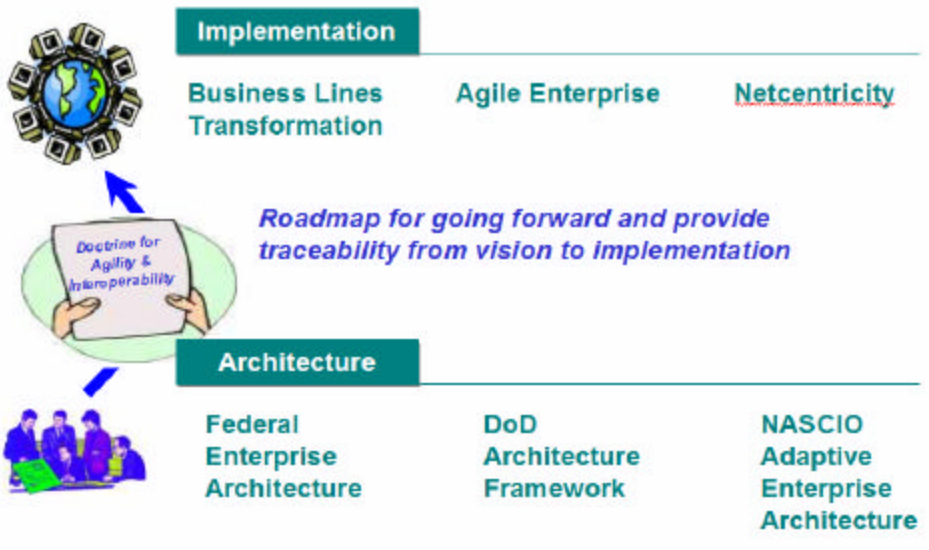
[1] Life cycle includes concept, requirement, information exchange mapping and physical application manifestation and support.
[2] The term 'e-Business environment' includes traditional legacy systems through to modern netCentric systems.

exchange precise semantics that then meet their business objectives. The *BCM Templates* provide the means to track and document these cross-reference *UID* links.

The *BCM* captures and communicates requirements in several architecture layers that simplify the understanding for each stakeholder by organizing how the complexity of e-Business applications is addressed and how each of the *BCM Layers* relates together. The effective management of *BCM Templates* (the *'what'*) proves to be the basis for reusability of the automated code (the *'how'}* and thereby enhances reusability and comprehension.

The challenge of interoperability and enterprise-coordinated development is very large, complex, and extremely critical. The cost of developing and maintaining information systems is a considerable portion of any Enterprises' expenses today – with maintenance costs continually on the rise. The *BCM* can significantly reduce the resulting friction resulting when transitioning from "as is" to "can be" environments. The resulting Enterprise will support the semantic and *pragmatic interoperability* envisioned. The semantic artifacts of this Enterprise are constructed using open declarative mechanisms that allow for mass customization of diverse vocabularies and models within heterogeneous environments. Furthermore, the Enterprise will be able to adapt readily to the effects of rapid technological change, reduce complexity more easily and promote reuse. Most importantly, the Enterprise will be prepared for and better able to respond to new business opportunities.

During the last century science has learned much by decomposing itself down to root concepts. The *BCM* reverses this trend, adding to traditional development decomposition by addressing the phenomenon of a linked network of *Communities of Interest*. The *BCM* effectively integrates these *CoIs* developed upon heterogeneous Enterprise, technical and information architectures; and at the same time provides a roadmap for migration from concept to implementation. As a result as depicted below, the *BCM* is the key to getting from Architectures to Implementation.



4.1 Summary of Contents of Document

This specification covers the requirements associated with the Phase 1 implementation of the *BCM* which is limited to defining the *BCM* vision and sets out to define a methodology which allows business users and experts to participate in the development process. Therefore, this document is limited in technical details or implementation specifics, but every attempt possible has been made to cite possible complementary efforts that are currently underway.

4.2 Audience

The target audience for this specification includes technically-minded business managers, and subject matter experts interested in electronic business (*eBusiness*) solutions as well as the system designers, and software developers that support them.

4.3 Caveats and Assumptions

It is expected that the reader has an understanding of eXtensible Markup Language (XML) and is familiar with the concepts of e*Business* including *Web-based services* and transaction management, *netCentricity*, registries/repositories, and templates.

4.4 Versioning of the Specification and Schema

Specification drafts will have version numbers of the form: ***Version 0.xy***, where ***xy*** represents a two-digit, positive whole number starting at 1. Once finalized, this specification will have a version number of the form: ***Version x.y***, where***x*** is a positive, whole number beginning with 1 and *y* is a positive, whole number beginning with 0. Minor revisions of a particular version, resulting from typographical errors and other edits that do not significantly change the meaning of the document, will be indicated by incrementing the ***y*** value. Major revisions that significantly change the content or meaning of the document will be indicated by incrementing the ***x*** value. This specification will not involve schemas; therefore, no schema versioning is provided at this time.

4.5 Concepts

Technical concepts in this specification are defined in Appendix D, Terminology Alignment Appendix E, and Abbreviations in Appendix F.

4.6 Related Documents

See Section 10 for the complete list of references.

5 BCM Overview

5.1 Introduction

The *BCM* can be viewed as three distinct steps that together provide the cycle that enables business users to formalize their needs and then deploy these into operational environments. The *BCM* enables this in such a way that they can manage the operational rules as well as the design of their processes and information exchanges. The three major parts to the BCM:

1. **BCM Layers** - Formalizing the business needs into *BCM Layers* and supporting *BCM Templates* and other optional models. The first step in this process is the understanding of the use of *BCM Layers* to qualify aspects of the business solution. Once the business user has understood the boundaries and the scope, they can then review their own needs and categorize them accordingly using the templates that the *BCM* provides and extending these to fit each unique situation. Defining common semantic concept definitions, mechanisms and align to *Communities of Interest*.

2. **BCM Information Pyramid** - The business analysts develop the semantic details of the *Information Pyramid* (aka Lubash Pyramid). This provides the roadmap to all the semantic mechanisms that describe the complete information process. This model provides the key foundation on which the actual software implementation is built.

3. **BCM Operational** - Ensuring that the software implementation technology directly leverages those semantics through a consistent context driven information architecture. The *BCM* operations are driven by a 'Contract' metaphor between stakeholders which in turn vector *BCM Templates*.

Provided is an overview of these three parts, the synergy and transitions, and the critical success factors for each of them.

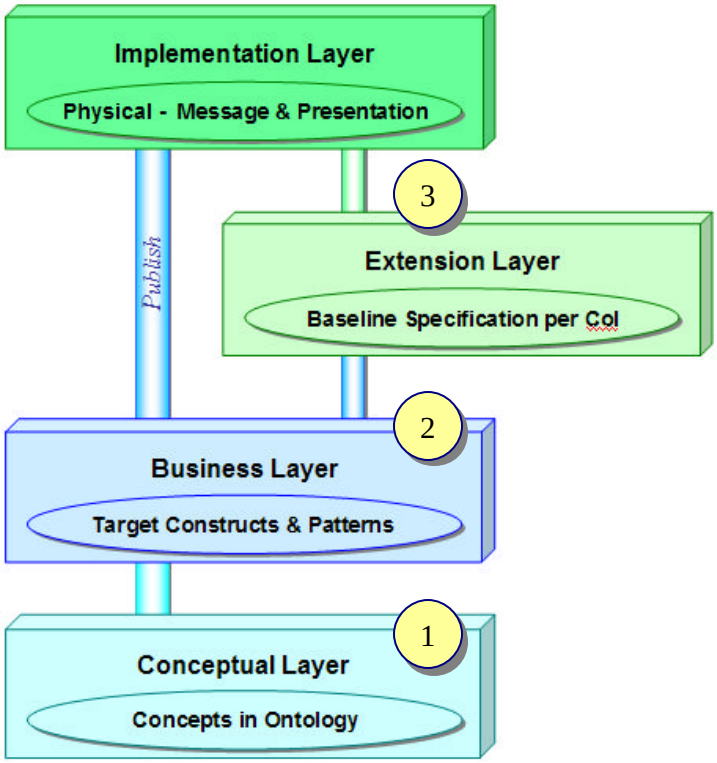
5.2 BCM Layers

The *BCM* provides a layered view of the enterprise information world. Each layer is designed to encompass a complete and discreet set of semantics and to enable the business implementer to segment their understanding of the problem. By focusing on one layer at a time this provides critical organization and structure to solving the complexity of e-Business information integration.

Central to the information architecture and the *BCM Layers* is the ability to pass context across boundaries and retain context state within processes and expose the *Choice Points* associated with these. The *BCM* uses linking and switching control throughout the layers driven by *Choice Point* services to accomplish this. [Choice Points are further described in section 8.5]

The figure 5.2.1 shows an overview of the *BCM Layers*, and each is summarized.

Figure 5.2.1 The BCM layers overview



1 - *Conceptual Layer*

The foundation layer is the *Conceptual* and provides a high level view of the solution requirements. In this layer the business managers determine the solution requirements and classify the business area that is the appropriate owner within the enterprise. The *BCM* provides templates that acquire the necessary business collaboration information within this layer. These include such items as the business goals, the project boundaries, the participants, the *Community of Interest*, use case, business events and the classification of the domain and any associated ontology. The classification and ontology provide the means to relate a particular set of components and to ensure the correct alignment and network is known within the particular business domain.

The *BCM Template* approach is designed to provide business managers and users the ability to create the template content in business terms they can readily understand. This avoids the need to learn arcane modeling tools and similar technologies that are founded primarily in computer-centric philosophies that business users cannot assimilate easily. The approach also allows implementers to use familiar desktop tools such as word processing and spreadsheet software to manage the actual template content. Also dynamic wizard based Web interfaces or handheld content editing allow for lightweight clients which can be applied almost anywhere.

Note: The Conceptual Layer isn't synonymous with database conceptual models where attributes are collected into entities or business objects. This design process takes place in the *Business Layer*.

2 - *Business Layer*

Within this layer you should decide either to select an existing industry model, or the need to build or extend a new model based on the organization's requirements. These provide the target constructs and patterns. A model includes templates for business processes and the associated transaction exchanges with the context parameters. It also classifies these components within the business domain by the area of use or interest.

Examples of industry models include the work of such groups as the OAG canonical model for Business Object Documents (BODs), RosettaNet PIPs, OASIS industry technical committee specifications, legacy EDI industry models, the US Government FEA (Federal Enterprise Architecture), the Canadian Government EDAT.

The *Business Layer BCM Templates* provide the means to tie these industry components together in a consistent way, to manage the critical context drivers for those components and to ensure that interoperability and agility is enhanced. Typically industry groups provide only the raw mechanisms for their members, so the *BCM Templates* here provide the means to orchestrate these across domains in a consistent functional way and to apply context driver mechanisms to enhance the ability to re-use common components.

Again the *Business Layer BCM Templates* also address the need for business managers and analysts to be able to express the requirements, transactions, context parameters, business rules and process steps.

3 - *Extension Layer*

Once the industry model is determined, it is extended out to the particular enterprise environment and *Community of Interest*. The *Baseline Specification* is then determined from knowing that context. The *Extension Layer* includes defining communities and selecting partners around the information exchange requirements. Also included are common a problem definition and connecting to the organization's partners' eBusiness infrastructure. This requires resolving the differences between various solution requirements.

Again the means to manage this process are defined in *BCM* extension templates and supporting technology such as OASIS CAM templates. Easily identifying and resolving differences is a new area of work that *BCM* is leading including the work on *Choice Points.* Catalogs of processes supported by registries and industry vocabularies and dictionaries are also an important part of this aspect of the *Extension Layer*.

The *Extension Layer* further refines these by assigning specific roles to participants, liabilities and responsibilities, schedules, and mapping the interchanges to the specific local applications. This leads to the *Implementation Layer* where the fine-grained semantics of individual information points within the transactions are defined (length, datatype, content values,

meanings), the structure point use (mandatory/optional/paired), and strict validation rules and calculations (see OASIS CAM template specification for approach to implementing this level of template detail).

4 - *Implementation Layer*

At the top of the stack of *BCM Layers*, is the *Implementation Layer* where the business solution is interpreted by the software systems. The rendering of formal business interoperability as XML allows the software layer's behaviors and processes to be formally controlled and directed. The core aspect of this is the *BCM* linking and switching mechanism of *Choice Points* and that are enabled by management and driven by the business context parameters.

Software implementers can therefore choose the mix of technology components that will best fulfill the business needs since the *BCM Templates* are agnostic to the *Implementation Layer*. However this does not mean that the software implementation can choose to ignore the *BCM Templates* completely. The *BCM* requires that the software architecture fully support dynamic application of business context parameters, as exemplified by the OASIS CAM specifications, and also fully support the use of *BCM Choice Point* technology. It is therefore somewhat of a paradox that an agnostic implementation approach actually requires deep support for the principles of that approach within the *Implementation Layer*. The *BCM* calls for strong liaison within the OASIS family of specifications to ensure that support wherever practical.

Conversely the business users can now redress the balance where previously they were excluded from active involvement in the *Implementation Layer*. While software engineers may configure the physical implementation components, the behavior of these can be controlled from the *BCM Templates* and rule definitions that the business users manage and maintain. This coupling is essential to ensure that the implementation exactly follows the business requirements and model in a living and active way. This ensures that information agility is built-in to the software solution.

Related work in the area is the OASIS CPA specifications and is further defined in the *BCM Information Pyramid*.

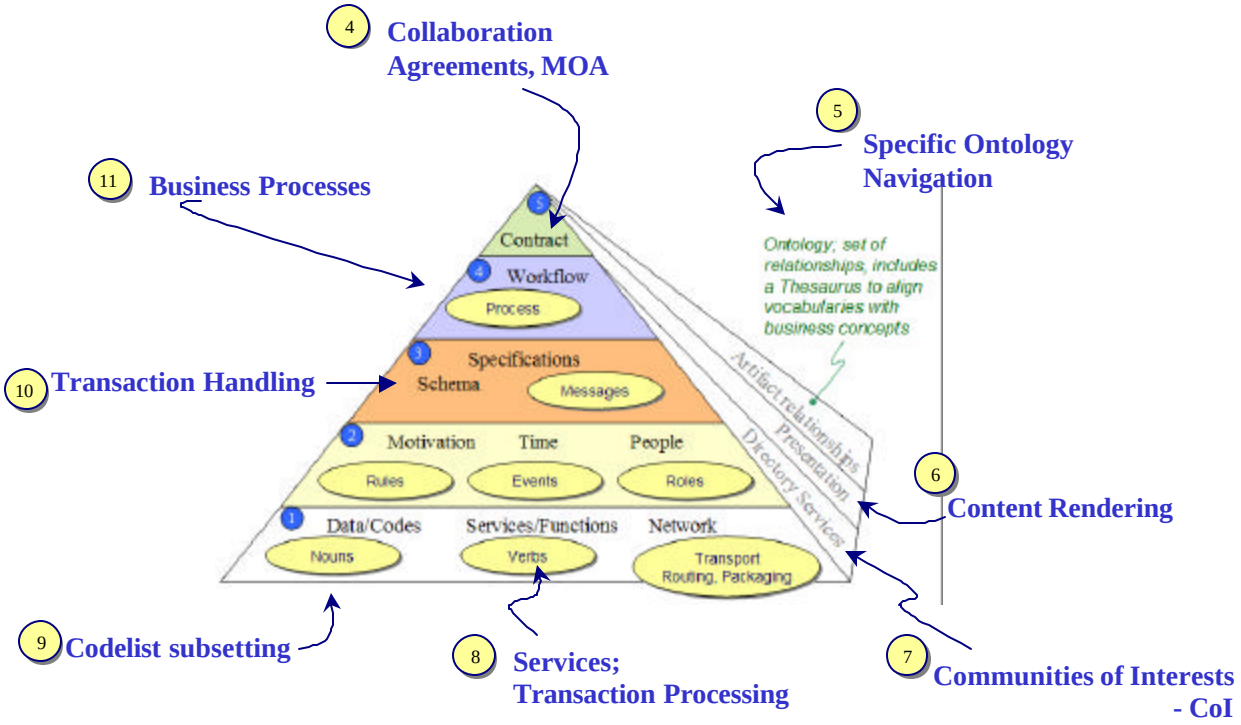
5.3 BCM Information Pyramid

The second major part of the *BCM* is to align the information semantics and process definitions across the implementation domain and *Community of Interest*. Historically business implementations have been viewed as content-centric development by the software developers. However the critical need is not to exchange data content, but to be able to process the semantics and context as well as the data and thereby obtaining complete information exchanges.

To achieve context driven information exchanges requires that the processes themselves within the *Implementation Layer* be driven dynamically by representations of those business interactions. The components detailed in figure 5.3.1 show the *Implementation Layer* breakout that together creates a typical set Enterprise services.

From the template definitions in the *Business Layer*, the *BCM* establishes the template collection of a collaboration agreement and optionally a traditional memorandum of agreement (item 4 in 5.3.1). Once the business collaboration details are agreed upon they can be assigned to a domain and ontology. While not essential for a local enterprise implementation, within an industry group or *Community of Interest* it is important to understand the relationship to different implementation areas (item 5 in 5.3.1). This aids the re-use of existing collaborations later on by providing directories that can be used to discover potential collaborations (item 7 in 5.3.1).

Figure 5.3.1 The Information Pyramid



Control over the rendering (item 6 in 5.3.1) ensures that the business users can configure the deliverables and outputs as determined by the business needs. Again the templates provide a guide to the realization of these parts and subsequently their representations, e.g. XML structures. The OASIS ebXML CPA work is an example of existing implementations in this area (item 4 in 5.3.1).

Once the collaboration is agreed upon, the associated information exchanges to implement that collaboration can be defined (items 8, 9, 10 in 5.3.1). The information transactions require careful detailing of the semantics. There are *verbs*, *nouns*, *roles*, *rules* and message structures to quantify. In traditional software development this is the place most people begin. The question frequently asked is "Do we have a XML schema to use?" with the assumption that if so then the participants are ready to start exchanging XML conforming to the schema and facilitating eBusiness. In order to engage in effective information exchanges, especially across an industry group with multiple participants, experience has shown and the *BCM* expects a greater depth of semantic knowledge than a simple schema provides.

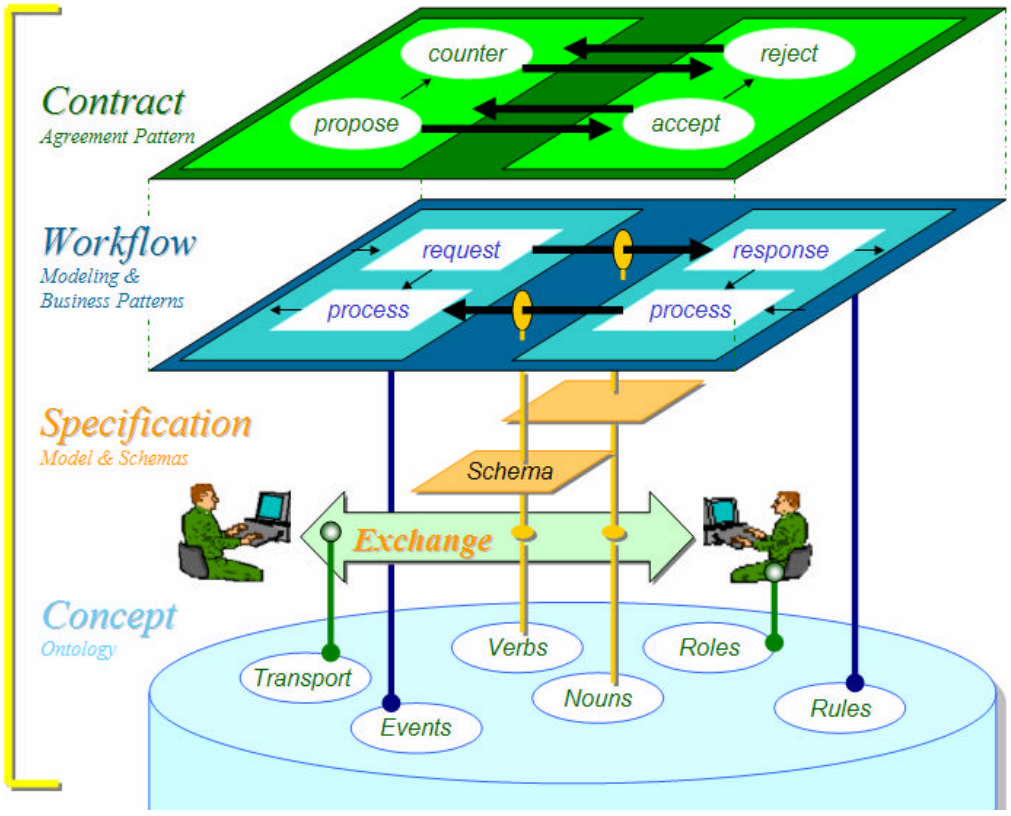
This completes the summary of the second step of the *BCM*, and with the business semantics defined and the ability to render these to XML enabled, the next step is to provide the physical information architecture layer to complete the delivery of the solution.

5.4 BCM Operational

The third major part of the *BCM* looks at the operations and functionality of agile information systems. Again the overarching principle here is that the architecture is agnostic and can be implemented with a variety of software applications as needed. The constraints on those applications are that they must support the key ability to have dynamic context driven business mechanisms through the use of external templates and associated semantics as shown in figure 5.4.1.

Therefore the *Implementation Layer* software applications have to support the use of *Choice Point* services in this manner. Furthermore the *Implementation Layer* also must support the use of business context parameters to control the behavior of local components. These aspects are essential to ensuring that the business users can manage and configure the rules and behavior of the deployed applications.

Figure 5.4.1 Information Architecture Components



Referring to figure 5.4.1 the business goals and agreement patterns relate to the corresponding templates previously discussed in the figure 5.3.1 the *Information Pyramid*, and so on relating

each level in figure 5.4.1 accordingly. Therefore in a physical implementation that supports the *BCM* it is anticipated that the software applications will utilize each of the artifacts in the corresponding way with the relationships between them. For example, transaction processing exchange (shown in the bottom level) will include a schema definition for the structure variants and simple content typing. It can also use a context driven assembly mechanism to create the actual content that is exchanged based on the roles and rules for those participants and process workflow details. The levels can be traversed in this way, and at each boundary the appropriate *BCM Template* can be used to control and direct the behaviors and outcomes.

A registry tool is also highly recommended to manage the semantic content and XML representations and provide the ability to locate content by classification and ontology. This leads into the last part of the semantics, that of process definition. The collaboration is presented as a set of discreet steps with associated information exchanges between the participants. The ebXML BPSS specification is an excellent example of this rendered as XML, and the new OASIS BPEL work is also applicable as a means to execute and process business interactions.

Examples of the depth of semantic information are shown by the OASIS CAM work on content assembly and it provides a benchmark specification that should be referred to here. At each step of the process one or more transaction templates can apply depending on the operational needs. It is also conceived that the OASIS CAM can provide the mechanism to map registries entries. In a traditional eBusiness implementation proprietary information mapping interfaces are used or application program components written. Clearly the rules embedded in these systems cannot be externally directed or verified. However it is conceivable that a CAM template can be used to dynamically direct a mapping component.

Other work in this area includes the OAGIS work on BODs and the use of XSLT scripts and Schematron templates to provide sufficient semantics. This is only partially successful as they are not re-usable nor context driven, and also are extremely difficult for business users to comprehend. Similarly vendors providing integration services have sophisticated semantic integration systems that can be considered provided they support dynamic context mechanisms. Conversely an OASIS CAM template definition provides the entire *noun, verb* and context semantics for complete transaction management including integration into a registry vocabulary dictionary without the need for highly specialized software.

By providing this complete set of functionality the software applications will conform to the *BCM* requirements and provide agile information exchanges that are manageable through business accessible mechanisms.

6 BCM Objectives

6.1 Goals

The *BCM* becomes an explicit driver for all design and implementation decisions using layers of appropriate constraints that make it easier to respond to changes both during and after implementation. The *BCM* focuses on the needs of the implementation team while supporting a structure management methodology that also addresses integration tasks to the implementation level. The benefits include:

- **Faster time to implement exchanges** - due to understanding the semantics of each message and its intent,
- **Dynamic discovery of efforts across the Enterprise** - due to the sharing of lessons learned concerning management of interfaces, concepts, information flows, and metadata,
- **Reuse of work products** – resulting from an architecture framework and methodology geared toward providing reusable components and templates,
- **Extension of work products** - such as internal applications, COTS, and GOTS to meet requirements where asking vendors to modify products has proven to be ineffective,
- **Management of linking and switching through *Choice Points*** – implementation mechanisms that provide the ability to create agile information networks across the Enterprise.

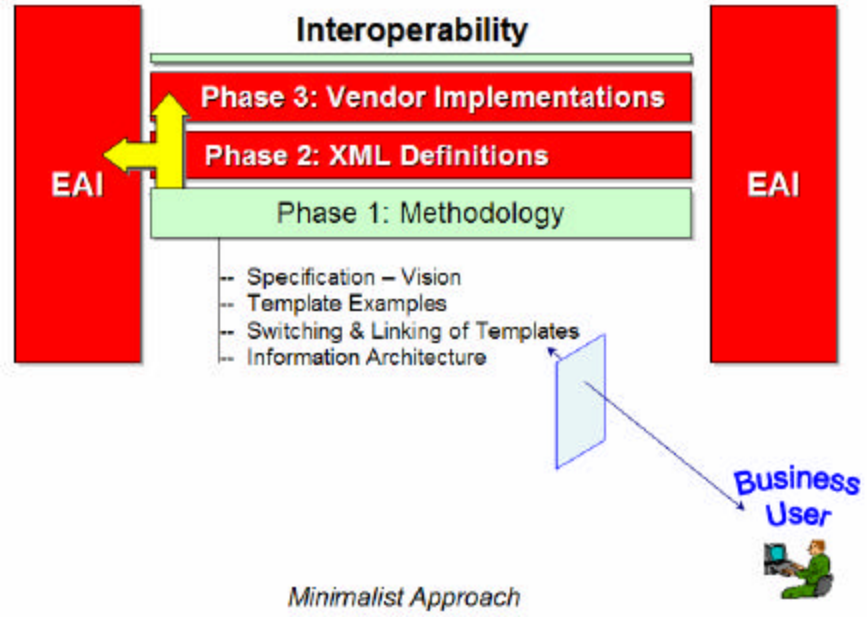
Ideally, the goal is to establish common services that span the entire Enterprise and exchanges that allow for common structures while also allowing for varying business payloads. Solutions like these have been elusive until now. The underlying theme is simply to make the business users, customers, vendors, and developers task easier through declarative ('what' not 'how') mechanisms that facilitate communication, discovery, and management at the right level of alignment.

The process is constrained by the *BCM* that outlines management criteria to assist with the myriad of choices and trade-offs that have to be made in order to achieve the organizations' tailored vision. The results of these choices are transformations of business communications among business partners' using *desired semantics and syntax*. The integrated information architecture can enhance an organization's *performance* and *agility* to deliver the ultimate business metric, "Customer Best Value".

6.2 In Scope – Phase 1

The implementation of the *BCM* is planned in a phased approach as depicted in the figure below between today's systems and the transition guided by the *BCM* to the new agile systems. The three phases include a) Definition of the Methodology, b) Establishment of XML Definitions, and c) Vendor Implementation. This specification document is focused on the Phase 1 part of this approach and will define and explain the *BCM* and its emphasis on interfaces, interoperability, and Enterprise agility.

BCM Solution



Phase 1 provides the foundation of the *BCM* vision with template examples, their linking and switching, and with the information architecture having the general boundaries as follows:

- **Providing Enterprise Agility** – defining the steps required for adopting the *BCM* reduces the risk of change paralysis later for an organization by providing agnostic mechanisms. Defining the supporting information assets required and the approach to acquiring them.

- **Interoperability vs. Enterprise Application Integration** – *BCM* will focus on the exchange of information between business stakeholders with their various *Communities of Interest*. The audience is business users, business managers and technical managers, and developers. For contrast, EAI might deal with all requirements for a business object throughout its life, where as the *BCM* will focus on how to subset this information in sharing with an organization's partners or internal exchanges.

- **Linking and Switching Mechanism** – a business context implementation mechanism that allows determination and management of parameters that control a process. Specifically to allow external context drivers to be implemented across an Enterprise.

- **Information Architecture** – defines the semantics of an enterprise business solution as a set of coherent layers of the *Information Pyramid*.

6.3 In Scope – Later Phases

In later phases rendering of templates as specifications in XML Schemas will be provided and demonstrated with vendor implementations. At that time an industry based interoperability conformance pilot may demonstrate the exchange of *BCM Templates* to produce agile information exchanges.

6.4 Out of Scope

This specification will not establish a list of specific requirements or guidelines for exactly designing or implementing a *BCM*-oriented software or systems solution. Instead constructs and mechanisms are provided that can be purposed as needed for applications that utilize the *BCM.* In addition, the *BCM* supports but does not directly address:

- portfolio management
- simulation
- configuration management
- data management from an operational viewpoint
- business reporting

The *BCM* also seeks to leverage and re-use existing technologies and to identify these where applicable, (see section above on related work).

6.5 Doctrine

The following are attributes of developing with Business-Centric Methodology (*BCM*), an approach that requires business users and managers to accept the responsibility for issues that many times do not get addressed from a strategy perspective, but manifest which problems within organizations.

✍ Business First

- Shifting accountability and power to the users; customer and business experts, e.g. self-service
- Provide traceability from business vision to implementation (and status)
- Managing information assets to ensure: visibility, accessibility, interoperability, and understandability through metadata
- Semantic-driven; technology-agnostic context supported by classifications, ontology and patterns for semantic alignment
- Moving the semantics from applications to the infrastructure layer
- Objective; not standard language - but instead standard reusable mechanisms to better negotiate differences

- Capture rationale for *pragmatic interoperability*; Templates and models to define 'what' not 'how';

✍ Multi-Faceted Architecture

- Choice: Web (human), data, process, services
- Modular and layered to address complexity; leverage open initiatives such as XML
- Service-oriented; loosely coupled interfaces
- Wrap legacy systems with services
- Provide structure for business patterns
- Defer physicalization as long as possible
- Function-centric; not system or entity

✍ Strong Business Case

- Clear defined goals with success metrics
- Supported by proof of principles; e.g. pilot project, spiral approach, applying Pareto's Principle to task
- Have a short and long term migration strategy
- Can't wait for a perfect solution
- Continuous integration process

6.6 Adoption Approach

- Take a business user's perspective rather than a technical viewpoint:
  - Take a minimalist approach as to the scope; promoting enterprise agility and interoperability, not attempting to address all of the organization's needs at once
  - Combine the strengths of *Communities of Interest*, architectures and ontologies to allow focus on the part (decomposition), yet leverage the sum of the parts (composition) as an organization's information network
  - Define in business constraint terms templates to be applied in a methodology. The templates provide for business users to define in precise communication the requirements, rationale, assignments, relationships and definitions of the organizational functional aspects of the business. This assures sufficient constraints are defined to achieve the level of interoperability participating stakeholders require.
  - Develop an open mechanism; *Choice Points* for (1) switching the templates/services, (2) computing/using values, and (3) workflow paths based on constraints. In particular a state(s) of a *Choice Point* does not need to be known at the time of development, such as defining subparts or even during runtime.
  - Develop an information architecture viewing information as an enterprise asset using an agility model as the base with a 'contract' driven model for selecting particular uses for resources

7 Connections - Relationships to Other Efforts

The four *BCM Layers* provide the scope for relations to other work. Each *BCM Layer* has associated with it appropriate existing work, or ongoing new work. The *BCM* does not seek to discriminate specific technologies however. Instead the approach is to provide a set of requirements that can be fulfilled or supported as needed. Where examples are provided they are intended to be illustrative, not normative.

The following lists based on the *BCM Layers* provide a directory of technology and work that is appropriate for consideration by implementations using the *BCM.*

Conceptual layer:

Each domain has its own *Community of Interest* for harmonizing terms for exchange. For example excellent baseline points for address and customer information, can be adopted and extended within the communities of OASIS CIQ (Customer Information Quality) specifications or Electronic Commerce Code Management Association (ECCMA) to meet the needs of the UPU (Universal Postal Union) and US Postal Service.

Other such sources include the UCCNet, OAG, RosettaNet, EAN, DISA.org, HL7, OTA, Accord, PIDX and similar industry reference associations. As the whole arena of eBusiness transactions matures along with business process definitions and templates more catalogs will be available from *authoritative sources*.

The infrastructure work in this area includes the techniques described in IDEF 5, XFML; eXchangeable Faceted Metadata Language for publishing and sharing hierarchical faceted metadata and indexing efforts, WebOnt; Web Ontology Language used to define a common set of terms that are used to describe and represent a domain, OWL; is a semantic markup language for publishing and sharing ontologies, Topic Maps and ebXML registry and management and representations.

Business Layer:

Within this layer it is decided to either to select an existing industry model, or to build or extend a new model. The models in this area includes the work of such groups as the IDEF3, OAG Canonical model for Business Object Documents (BODs), RosettaNet PIPs, the National Association of Convenience Stores (NACS) architectural model, legacy EDI industry models, OASIS UBL, OASIS industry models, the US Government FEA (Federal Enterprise Architecture) the Canadian Government EDAT project and CEFACT core components semantics. These models capture the precedence and causality relations between situations and events in a form natural to domain experts by providing a structured method for expressing knowledge about how a system, process, or organization works.

Extension Layer:

Once the industry model is determined, it is extended out to the particular enterprise environment. This layer includes defining communities and selecting partners around the information exchange requirements. Also included are common problem definition and connecting of partners' eBusiness infrastructure. This requires looking at their solution needs and resolving the differences. The means to manage this process are defined in *BCM Templates* and supporting technology such as OASIS CAM templates. Easily identifying and resolving differences is a new area of work that *BCM* is leading including the work on *Choice Points.* Catalogs of processes supported by registry are also important along with industry vocabularies and dictionaries.

Implementation Layer:

The work in this area includes the W3C XML and Schema work, ebXML BPSS, CPA, Messaging and Registry, OASIS BPEL and CAM, and *Web service* work such as WSDL and UDDI. Also included is modeling and design tools such as OMG UML, CEFACT UMM, ebXML FSV and BSV models, the *Service Oriented Architecture (SOA)* work and the W3C *Web services* architecture work and the OASIS/CEFACT work on ebXML architecture.

For capabilities updates one excellent source is '*Cover Pages*', hosted by OASIS at: http://xml.coverpages.org For links relating to these technologies please refer to this directory site – http://www.xml-acronym-demystifier.org.

The *BCM* presents an interoperability methodology that complements …

- Organization's efforts in linking its vision to implementation
- Architecture frameworks
- Reference models
- Documentation and knowledge capture efforts
- Interface specifications
- Modeling and modeling language preference
- Technical approach e.g. object-oriented, Rapid Applications Development (RAD)
- Controls and metrics
- Technology-Agnostic methodologies

8 Applying the BCM

This section discusses key areas for *BCM* implementation.

8.1 Determining Communities of Interest

In building interoperable agile information systems one of the first needs is to select common formats for the information. To achieve consensus the participants can either seek out existing formats or develop their own. In either case it is important to determine the *Community of Interest (CoI)* into which the information domain falls and *authoritative sources* within that domain. While this is often overlooked in local application system development, where the focus is totally on internal information, as soon as any external interaction occurs (typically this is accounting related first) it becomes apparent that those internal systems need to conform to external requirements and that *authoritative sources* for those are needed. Therefore it is best to plan immediately to understand the *CoI*, not just the immediate local business.

There is much existing work around *CoI* classifications. Examples include DUNS and EAN classifications, government codes such as SIC and NAICS and international systems such as the UNSPSC groupings. Also trade and industry associations provide existing networks of *CoI* groups. Such larger standards bodies have already developed extensive dictionaries, vocabularies and semantics. However, acquiring access to these is often problematic, with restrictions of membership, copyright and software versions adding complexity.

Nevertheless building coherent *CoI* domains with consistent representations of specifications in open formats that can be utilized by a variety of software technologies is part of the challenge. Clearly technology like OASIS *BCM*, OASIS CAM and OASIS ebXML Federated Registry provide mitigations that will help solve these disparities.

Once the broad *CoI* has been established, the next classification is within the *CoI* itself, and development of ontologies and classifications to promote re-use by enabling the purpose and function of artifacts to be clearly determined. Again this is often overlooked and artifacts are poorly organized, or placed within too broad a grouping.

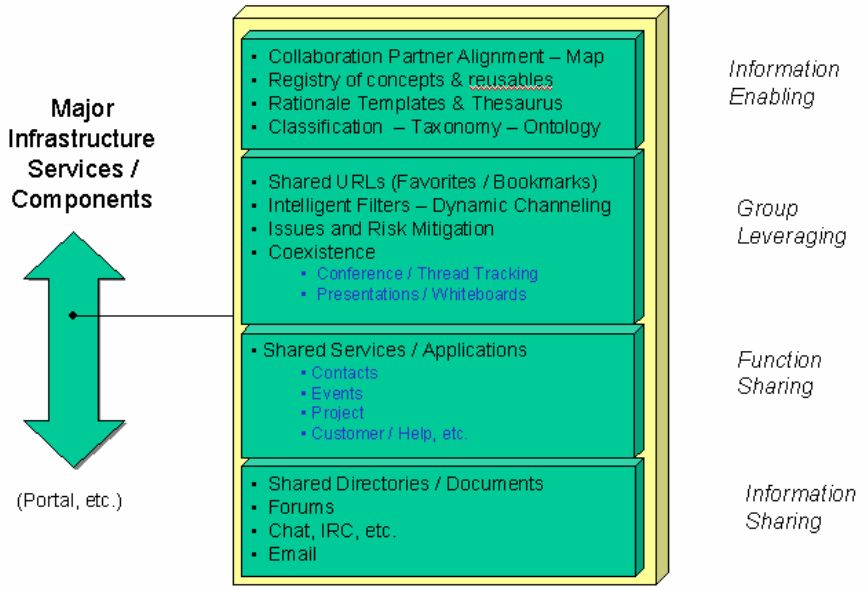
By identifying the task of *CoI* facilitation the *BCM* helps focus business attention on the need to improve *CoI* alignment. By providing templates to address these needs the *BCM* allows individual enterprises to effect change and improve within the *CoI*s. Technology such as federated registries and shared directory services are the other metrics in improving discovery and re-use of coherent standards. The next section considers in more detail collaboration mechanisms between enterprises within a *CoI*.

8.2 Collaboration Mechanisms

Once the *CoI* metrics are determined, two things are needed to more effectively interact with enterprise partners within a *CoI*; (1) *BCM Templates* to formulize the information configurations consistently, and (2) methods of interacting with and distributing those in a shared environment.

Figure 8.2.1 shows the technology aspects of this.

Figure 8.2.1 Collaboration technology metrics



From a business perspective this amounts to either leveraging existing technology infrastructure such as email systems and collaboration tools, or deciding that more extended technology is required such as a federated registry or a shared Web based content management system. The investment in these is balanced against the complexity and cost of the systems implementation requirements.

Traditionally collaboration has also occurred within standards organizations thorough physical meetings and verification of specifications. While this can be effective it is also slow. Today's standards are developed cooperatively using networked communications to move agreement forward in real-time.

Production systems also require real-time access to specification artifacts rendered as XML. This includes schemas, business process instructions, context parameters, communications profiles and business semantics. It may also include XML renderings of *BCM Templates* that can be referenced directly by the *Implementation Layer*.

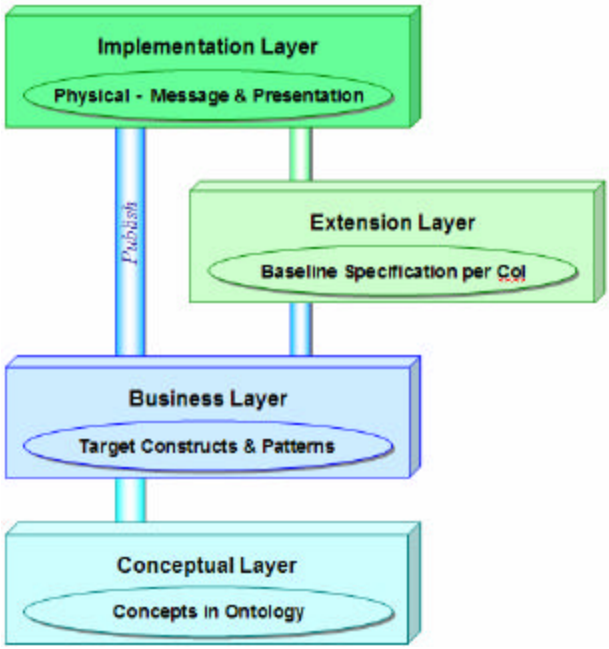
8.3 Layered Approach Details

The layered approach within *BCM* also helps significantly in improving collaboration across a *CoI*. Participants can relate to the requirements of a particular layer using consistent templates.

Particular benefits and goals of this layered approach include:

- Strategic management of artifacts and constraints
- *Semantic Interoperability*
  - Lexical alignment at *Conceptual Layer*
  - Identification of *Authoritative Sources*
  - Use of or mappings of business *Target Constructs*

Figure 8.3.1 Review of BCM layers

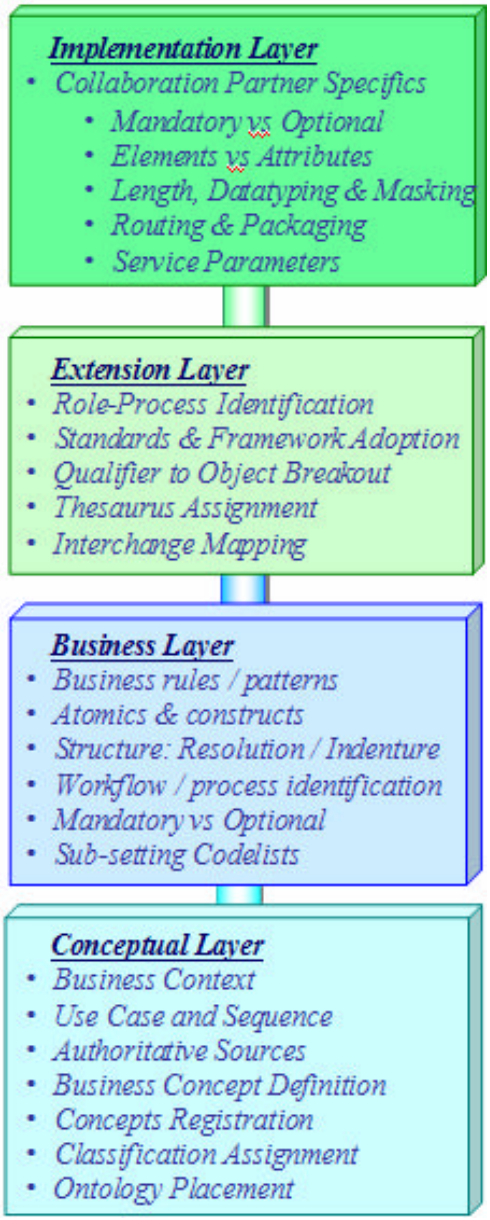


The next section details the specific *BCM Templates* associated with each layer and how they are utilized.

8.4 Templates

Particular benefits and goals of this template approach include improving communication between the business domain experts ('what') and the technologist views ('how') to maximize a coherent and consistent understanding of the requirements and semantics. This includes the ability to deploy directly from the templates to the *Implementation Layer* based off business rules rendered as XML artifacts. The figure 8.4.1 shows aspects of each layer that are candidates for resolving as templates.

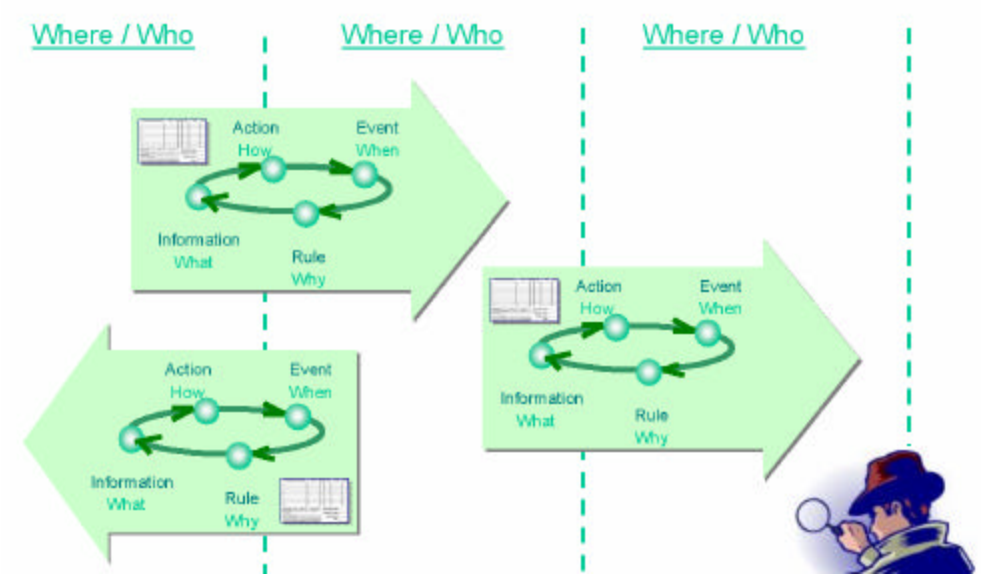
Figure 8.4.1 BCM template factors by layer



BCM Templates are designed for use with familiar desktop software tools, such as word processors, spreadsheets, and forms in a visual environment that can manage the hierarchies and relationships. The emphasis is on delivering a solution that business personnel can understand directly and using business terminology. This contrasts to formal modeling information technology methodologies that require complicated software tools and technical training in their use.

Figure 8.4.2 Partner Agreement Templates

Using the same template mechanism to communicate with collaboration partners



The *BCM Templates* are going to prompt for the same 6 questions, at different layers, from different points of view (with each view being from a dominant question). These prompts are:

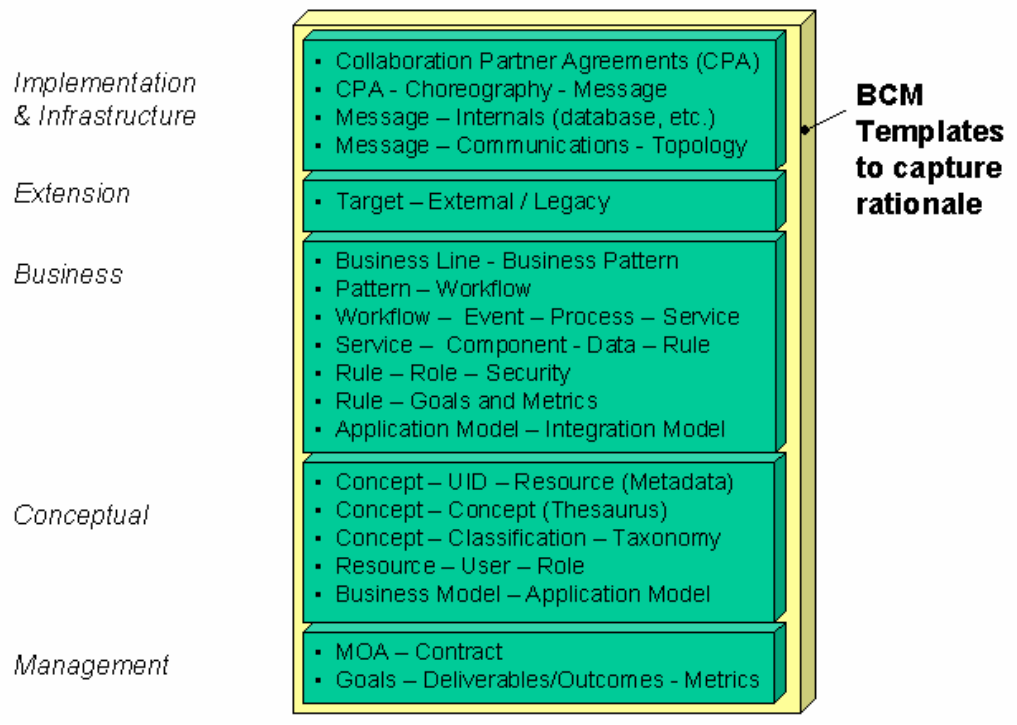
- **Why** – motivation and business rules
- **What** – information, data, codes
- **When** – timing & events
- **Where** – relation to landmarks
- **How** – services and functions
- **Who** – stakeholders and their roles

This leads to the notion of an *Agreement Template* that can be applied for exchanging information successively at each layer level that is then completed with appropriate information. For example at the *Conceptual Layer* the notion of business transaction defines the overall transaction document and any context level parameters. While at the *Business Layer* the transaction template needs to capture the rules, optional and mandatory use of the transactions, and business reference codelists such as to international or local regulation requirements.

The result of these steps is a collection of templates (figure 8.4.3) that contain the orchestration details for the required business collaboration and the associated process(es). These templates can be rendered into XML content that can then be processed by *Implementation Layer* software applications as needed.

Figure 8.4.3 shows a selection of typical metrics associated with the template detail from each *BCM* layer.

Figure 8.4.3 Template Products Summary



8.5 Choice Points

The *BCM Layers* represent major points of interface where choices must be made. But there are many more physical interfaces within an organization, and how these separations work impacts its business functions. Within large organizations, decisions involve thousands of variants of business choices, business rules, business patterns, and data permutations. Organizations need to manage these *Choice Points* in a proactive manner, capturing both options and their rationale. The results can then be stored and reused with efficiency and refinement. [*Choice Points* are briefly discussed here with further description in Appendix B]

The explicit identification and management of these *Choice Points* significantly aids comprehensibility and alignment, while promoting tracing and accountability. In large organizations, the vectors at each decision point and their interrelated linkage are often complex. An agile organization extracts these relationships as business patterns and separates the *Choice Point* vectors out as parameters for each context.

The declarative approach provided by the use of *BCM Templates* improves comprehensibility and reduces the probability of errors, as processes are orchestrated based on a selection of options within a template. Understanding those options and providing them into a template

based on the business knowledge of the domain is the skill that the business analyst delivers. Enabling such developing for choice is the challenge businesses face.

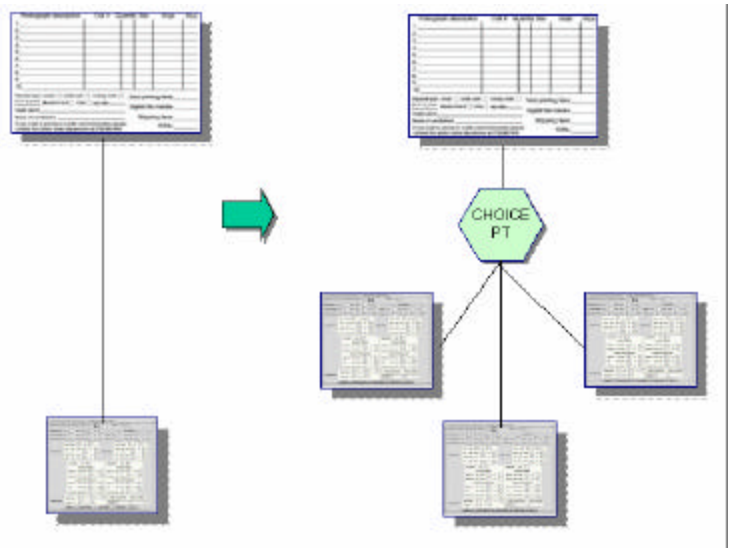
8.5.1 Developing for Choice

The *BCM* utilizes a 'contract' to formalize the combination of workflow, processes, schema, maps, rules, etc. into *BCM* artifacts. The underlying principle is that each *BCM* layer solves the problem at that level, and only that level, based on a focused set of constraints. Information that is not available or relevant at that point is deliberately deferred up to the next layer – thereby simplifying the overall solution. This approach is also in alignment with *Service Oriented Architecture (SOA)* technologies built around *Web services* where service points deliver solutions to discreet requirements, and therefore often function like "help from above" from the users perspective.

The gathering of *Choice Point* parameters and control requirements (inputs and outputs/outcomes) occur around the boundaries of layers, as well as within layers themselves at the intersection of process paths.

The specific combination of *BCM* products and their interrelationships determines the *BCM Templates* needed to generate decision points and variables across an identified pattern. For example contract instantiation creates objects at runtime that interact as described by the contract; e.g. *Web service* components in the *Implementation Layer*. By using such contract driven techniques, dissemination of change from the requirements through to implementation is greatly simplified.

Figure 8.5.1 Template Contract Choices directed via Choice Point



Choice Points can be seen as providing three enablers for agile information exchanges:

- Context criteria, where the scope of the context extends beyond the local decision point, and can also require persistence of decisions
- Determining context by refining criteria dynamically, and that may include undetermined start points
- Where the context requires a thread manager to establish and track the state of a process.

There are other significant aspects to the implementation of *Choice Points*, such as consistent semantic definitions for the context rules and robust process control syntax that allow the user business requirements to be precisely defined. A further significant benefit of the *Choice Point* approach is that it exposes and makes available the context parameters within a given application layer. This allows business decisions and choices to be clearly known, classified and selected. This serves to highlight the difference with today's systems that lack *Choice Point* technology. Such non-agile systems are therefore static inflexible 'stovepipe' solutions that cannot support dynamic linking and switching based on *context* and are thus hard to re-purpose and change. These previous applications were built as a "black box" that could not be easily re-purposed or their suitability to task quickly determined.

Experience indicates that today's organizations are too complex to be modeled and easily understood with lines and boxes in a CASE tool. Current modeling techniques are adequate for showing sub-classing, path options, sets of codelists, or object-role variances; but they fall short in tracing the thread of user choices. This is where the *BCM* differs significantly from current methodologies as it directly embraces and provides support for choice.

8.6 Unique Identifier (*UID*)

To complete this section the need for and use of *Unique IDentifier* (*UID*s) are reviewed. In order to provide a consistent reference system across templates and between layers the *UID* is preferred. Any artifact or semantic fragment may be labeled with a *UID* reference attribute. Also *UID* references may be added later to resolve cross-referencing issues, or to facilitate the *Implementation Layer* details.

Some examples of *UID* use within *BCM Templates* are pointing to:

- A concept definition
- A concept linked to an external registry vocabulary dictionary system
- Another *BCM Template* such as a business collaboration agreement
- An explicit information point within a *BCM Template* (e.g. currency, country)
- A codelist reference value set
- A business process script component (e.g. CPA, BPSS, BPEL, or CAM instance)
- An industry transaction format definition (e.g. XSD or CAM or EDI definition)
- A company's partner information

The *UID* should consist of the follow parts wherever applicable:

- Steward
  - Registration authority that controls the *UID* to assure there are no conflicts
    - Reference <dc:publisher> in Dublin Core Element Set v1.1
- ArtifactName [*or autonumber algorithm*]
  - Name of the "quasi" root, for example, USSGLAccountType
- Version [*or release sequence*]
  - Date of creation or last modification, for example, 2002-12-17 with a letter sequence for multiple versions on the same day
    - Reference <dc:date> in Dublin Core Element Set v1.1
- FileType
  - Internet Media or Mime types, for example, xml, xsl, xsd, dtd, etc.
    - Reference <dc:format> in Dublin Core Element Set v1.1

Therefore, one example of a valid *UID* is:

DFAS.USSGLAccountType.2002-12-17a.xsd

Another example is an element reference such as: **OAG010309:001:000** where the *UID* is described in the OASIS CAM TC specification to depict an OAG BOD transaction element that references element 010309 and version 001. In this case the *UID* reference system also supports versioning and sub-versioning. In this case the *UID* is an alphabetic character prefix (aka alias) followed by 6 numeric digits, followed by optional version information in the format colon (:) number suffix, and then sub-version as colon (:) number suffix.

The *UID* references can then be rendered into the XML instances of the *BCM Templates* and accessed by the application systems accordingly. The *UID* system is designed to provide a unique coding system for a *CoI* domain, and with codes that are easy for human manipulation and verification. This contrasts with the machine generated UUID system that produces 128 byte keys, or complex URL unique identifier based code schemes that are intractable to human use[3].

[3] Notice however that a UID can be assigned to such complex references to make them also easy for human use.

9 Layered Analysis Approach

This section details each layer and the tasks associated with its use. Also discussed is how the particular analysis techniques within the *BCM Layers* enable the implementation and a better understanding of the problem. It also serves to explain the rationale and goals for each layer within the *BCM.* This section serves as a starting point for establishing a collection of templates and descriptions of their application in a *BCM Template* library. Such a collection should provide a focal point for implementers. The foundation of this BCM Template library is extracted from best practices gathered from industry and government sources and projects. The Template library itself is in Appendix A, (and also accessible online), and contains a directory of the initial set of tasks detailed in this section.

In addition to the individual sets of *BCM Templates* and tasks, these individual items can be grouped and referenced into sets for given scenarios to achieve particular business results. These sets offer choice to the business manager depending on the environment of the project. And just like individual *BCM Templates* and tasks, the sets can be tailored to suit a given need as well. Hopefully as you read this section it will bring to mind both new ideas, and good "templates" that worked in the past that make sense to contribute as a *BCM Template* now and share within the BCM community.

9.1 Conceptual Layer

Conceptually what does the business manager want to achieve, and does the solution make business sense? These seemingly simple questions drive the BCM and provide the underlying foundation from which interoperability will develop. One will need to answer such questions as, "Which standards or business frameworks to adopt?" as one decides conceptually to address the problem holistically; and often the answers are driven by one's customer base. As one takes the appropriate steps through the *Conceptual Layer*, other questions will provide telltale signs of interoperability, such as understanding the organization's collaboration partners' business concepts. With this said, the *Conceptual Layer* has an internal focus addressing the needs of the enterprise and not necessarily the external *Community of Interest*.

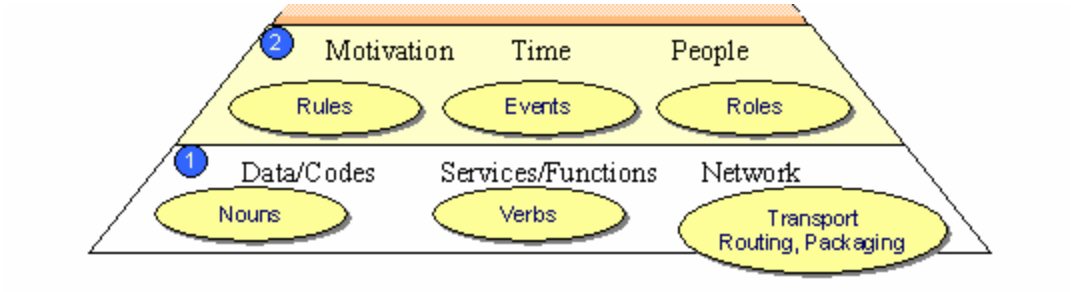
For instance, if an organization uses an off-the-shelf accounting package that has no notion of a 'contract' (where resources are subtracted as work is accomplished), and then attempts to interface with its customer, (and 'contract' is the standard business practice); mitigation in the upper *BCM Layers* of the project will certainly be necessary. At best this can provide an adjunct to the processing in the accounting package; or in a worst case scenario an alternative accounting process must be used. This may even involve manually computing results and front ending the off-the-shelf package.

In addition, in the *Conceptual Layer* the task is to fully understand the concepts of the business, including the business terminology of the domain, but excluding conceptual models of the business from software ER perspective and terms. The concepts are independent and tend to be atomic; in that one doesn't attempt to make business objects from these with attributes, rules,

roles, events, services (verbs), concepts (vocabulary – nouns) etc. combined together. The *Conceptual Layer* deals with the bottom two portions of the *Information Pyramid* in its pure form (figure 9.1.1), and no attempt is made to link the various pieces of the puzzle together to solve the enterprise interface challenges. This provides the business with the lowest common denominator with which to align, giving the best chance for agreement.

The *Conceptual Layer* builds the foundation of the *Information Pyramid* illustrating the required types of artifacts needed for eBusiness. Enterprises need to extend their base from *Data* Management to *Metadata* Management. It is important that these artifacts are therefore as unconstrained as possible by application context.

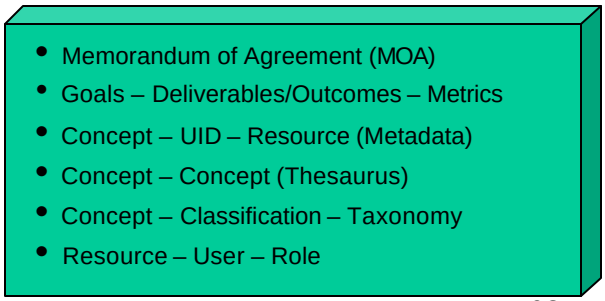
Figure 9.1.1 – Information Pyramid



:

One gets a first-cut of products in the iterative top-down process. One shouldn't expect these to be final, but should have a start in each of these areas (figure 9.1.2).

Figure 9.1.2 – Conceptual Layer Products



9.1.1 Drivers and Constraints

9.1.1.1 – Drivers – Business Goals

Many projects and products though technically feasible simply are not business successes. This is because they don't meet the business user's need, and are typically created with insufficient customer input along the way – much like starting the car without first deciding where to go. An organization needs to ask, "What are our objectives and what do we measure to achieve our

goals?" They also need to know that they are doing the right thing at the right time. If the object of the implementation is to address deficiencies, have these deficiencies been collected from all stakeholders? Have they been analyzed from an impact and dependency standpoint, assuring the root causes are to be addressed and not simply the symptoms?

The *BCM* vision is focused on communication. Specifically how the information architecture that is built to service the organization can be the conduit for business exchange. The vision is to unify many of the pieces that are in place today, address these pieces from a strategic viewpoint, add a few missing components, and assure that the organization thereby becomes a world-class service or product provider.

Perhaps more importantly is how the following items link within the organization and its collaboration community, at all levels, such that they are accountable for them:

- Vision Statement
- Balanced Scorecard
- Goal Patterns
- Targets, Measures & Assessments
- Policies
- Strategic Plans
- Performance Agreements
- Architectures

To become world-class one needs the vision of the particular Enterprise leaders to be adopted and enhanced through implementation. One needs to involve an organization's 'political' leaders as well as business experts in declaring the organization's metadata and business rules in a precise manner in order to make the intentions clear to developers and implementers. The organization's business goals, currently located in various forms, need to be the raw materials for guidance in the operation of the business.

Just as important is to bring the developer's awareness up to the requirements of the business using a methodology that promotes the sharing of ideas and concepts. An application developed where the implementers know the reason why something needs to be done will provide better results than one where there is no idea what the business drivers are.

The *BCM* revolves around the people and how collaboration expedites the capabilities of the organization. The underlying theme is; "Its not just about the technology, it is about the people". This translates to their understanding of the information. It is only when one considers the organizations' human capital that true business intelligence in systems will ever be attained. People can also have unstated goals; understanding these is important to perceiving the terrain for overall success. This translates into the difficult task for a good analyst, of knowing the right questions to ask to obtain the correct answers. Often information may be withheld that is critical, either because someone is so familiar with the domain, they assume everyone else is, or thorough a fear of potential vulnerability.

By accomplishing the accountable tasks listed above the business experts aren't dependent on the technologist to achieve their objectives. The technologists can then understand better what the business needs are and this increases the probability of the business users getting what they need to do their tasks. With accountability there will be less disconnects as everyone understands each other's objectives. With accountability, developers will experience more stability without moving objectives. In short the enterprise will operate out from under the '***Policy*** ✍ ***Myth*** ✍ ***Implementation***' syndrome as business experts and managers take back the steering wheel of the details.

9.1.1.2 Frameworks and Standards

Emphasis on open systems is a step in the right direction – organizations need to encourage vendors to move from proprietary to open mechanisms and interfaces. As organizations move toward opening up their interfaces one finds a cost decrease for deployment as well as maintenance. Removing proprietary software application shackles is a win for the organization, and required to build foundational constructs of the information architecture.

Horizontal standards (all industries) and vertical standards (specific industries) come in various flavors: sanctioned bodies, consortiums, a few leading companies, or if the company's product is widespread, one company. The problems in choosing standards are that some initiatives are complete frameworks; others are just focused areas, while many standards overlap and are duplicative.

Organizations need to take charge of their business information artifacts, managing them as critical business assets. Taking control isn't just defining an approach such as the single enterprise architecture, with a single message structure – for the world is too complex for a 'one size fits all' strategy. The organization's past experience with data standardization and EDI has shown that a system, a mechanism, or protocol that doesn't include extensibility, that doesn't include flexibility, that doesn't bend - will eventually break. For more, refer to the Section 8 Connections topic where it discusses a subset of the underlying frameworks that may be applicable to the organization and meet the organization's requirements.

9.1.2 Tasks

9.1.2.1 Define Business Context

Understanding the context of the project or interface, its size, and its complexity, is as important to know as how to apply the *BCM Templates* themselves. Also knowing what is not in context is just as important, and should not be underestimated. One needs essentially to go *From* Business Goals *To* concepts, constructs, and communication by performing the following tasks:

- Business Case Analysis (BCA)
  - Align with Balanced Scorecard - are we addressing the enterprise's needs?
  - Identify overall issues - prepare problem statement(s)
  - Feasibility, Risk, Cost Benefit
  - Understand organizational drivers (pain, opportunity) from each stakeholders' perspective
- Define what is in and out of scope – prepare scope statement

- Research pattern/capabilities base for leveraging prior efforts
- Coordinate with other project planning tasks
- Timeline Decision?: 'Link Now' vs 'Link Later'
  - Link Now = Use BCM Templates as best practice guidance throughout development
  - Link Later = *"Fast Track"* where time overrides costs, expedite & align UIDs after the fact
- Begin *iterative* process…

It is helpful to also think of the following *BCM Focus Cycle*:

Decision ✍ Rule ✍ Information ✍ Action ✍ Event ✍ Decision✍

The *BCM Focus Cycle*; decision, rule, information, action and event will be viewed from several angles during the *BCM* tasks. At this level the "reason, justification, motivation or excuse" that drives the nature of the project is captured. "Why are we doing this and what is the scope? Does it align with our leadership direction? Does it align with an enterprise-level design? Is there a strong business case? Is it deemed a top priority?" If a project doesn't define its business context properly – it takes on unnecessary risks and enhances its probability of going off course or becoming infected with scope creep.

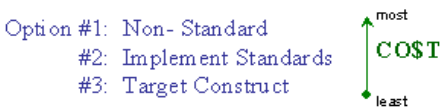
It is important that everyone knows 'why' in terms of Return-On-Investment (ROI) that an interface or project has been given the green light, both in hard and soft terms. This will tend to keep scope from increasing, easing developer's frustrations, and certainly management's. If an ROI can be given the team can come to an understanding and development doesn't take place just because it is technically feasible. Also from an enterprise perspective (figure 9.1.2.1.1), items that may be accomplished at earlier nodes in a value chain and not downstream where costs are higher may provide a least-cost alternative. This needs to be rewarded and metrics applied with the entire organization in mind.

Figure 9.1.2.1.1 – Assessing costs and risks compared to approach

More on the "Fast Track" Alternative --
Because we are [1] developing an alignment infostructure, [2] incorporating UIDs, [3] aligning at concept vs 'standard vocabulary' we are afforded a 'Fast Track' option because the link isn't tied into programming structures and thus can be easily linked into the ontology as a separate development process.

- ***Option #1: Metadata Management as a Natural Aspect of the Process***
- ***Option #2: 'Fast Track' Alternative***

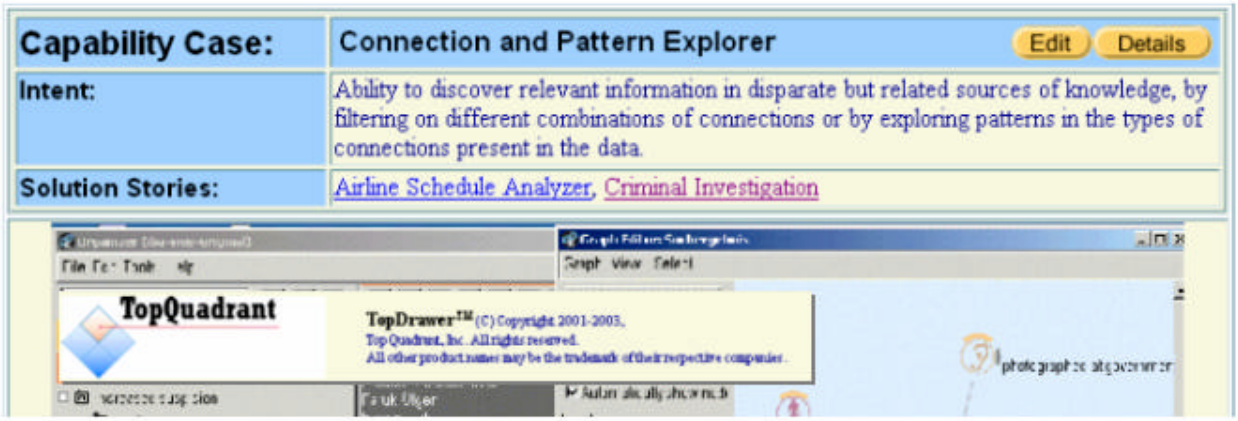
Keep in Mind: 'Fast Track' Alternative maybe at a higher cost to the enterprise than Option #1 for the resulting service defaults to *Extension - Outreach*, rather than opting for the opportunity to build from the *Target Construct* base. Also the loss of rationale is probable as decision criteria and tradeoffs are not documented along the way.



Costs to the Enterprise are based on interoperability opportunities.

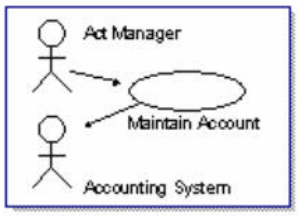
Patterns of the business should be researched so as to leverage prior initiatives. In large organizations this requires a procedure and sometimes a service to handle the magnitude of information to be able to extract a pattern. Over time, the organization realizes gains in reuse and obtains advantages based on the lessons learned of prior efforts. This base becomes the organization's best practice when solutions help to create a unifying vision and implementation. These practices can be published as 'Capability Cases' and exercised in "design by example" workshops where analogies and brainstorming make for the best possible solution. The patterns allow for workshop members to say, "What we want is something like this" (figure 9.1.2.1.2).

Figure 9.1.2.1.2 – Identifying Patterns through quantitative classification.



9.1.2.2 Develop Use Case

The use cases become the storyteller for the project; coordinating and identifying all (1) stakeholders, (2) identified dependencies, (3) identified contingencies, and (4) success metrics into specific scenarios. The use cases, or conceptual operations (CONOPS) prevent the team from being blind-sided later; by increasing scope and costs and by assuring critical that small but critical items are not overlooked, (such as the need to use business transaction acknowledgements).



If relationships aren't fully defined, unnecessary pressure is put on the team with a cycle of ever changing requirements. A mixing of use case techniques for requirements expression along with traditional methods of documenting specific requirements provides an efficient means to record the complete set of rationale drivers at this level.

The *BCM* supports service-oriented architectures (SOA) for loosely-coupled solutions agnostic to platform environments. The methodology promotes the 'Event' as a critical metadata artifact which makes loosely-coupled interoperability solutions successful. The use case development and the cataloging of events (both business and technical implementation triggers) are documented at this early stage. An event is defined as a process that triggers changes in another

process or processes, such as 'receive purchase order' or 'receive payment' (where business events are key to the accounting domain). The trigger occurs at the publisher to signal that an internal state or information has changed. The subscribers respond to the input to change its internal state and are processed accordingly. In a *netCentric* environment these events are used in a publish/subscribe collaboration mechanism where the initiating process need not know the processing details of the downstream subscribers. The events are processed in this manner for all collaborations in the value-chain. In developing the *BCM Templates* this event-driven approach divides the information required for development into manageable pieces and removes the need at this stage of development to be concerned with the diverse applications in eProcess.

Referencing the *BCM Focus Cycle*:

Event ✍ Decision ✍ Rule ✍ Information ✍ Action ✍ Event ✍✍

The *BCM Template* for Event provides the focal point for Event Reconstruction allowing for the determination of what one needs to manage, the identifying of sources for all events, and starting on determining the flow of events. The template supports the optimization analysis by providing for organizing the events into groups, analysis for elimination of unnecessary events, and to accelerate critical information flow.

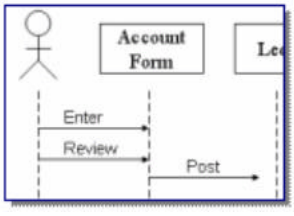
Event management provides the framework for further tasks in fully understanding the domain processes. The Event template allows for determining the impact of business events and defining how processes interact with the information flowing through the organization and identify critical issues to each event.

Business and information models are created following the selected organization's business process and information modeling methodology. It makes no effort to force the application of specific information technology techniques such as object-oriented principles. The diagrams are deliberately free structured (as with a UML diagram) to complement the flexibility inherent in both the *BCM Layers* and the *BCM Templates* approach.

One needs to accurately define scope and transactions between stakeholders. This may start with a preliminary interaction sequence diagram, which shows how the objects collaborate over time. Once the use case is initially sketched out the magnitude of the problem that is being considered will be known and the level of effort approximated.

9.1.2.3 Prepare Sequence diagrams

Sequence diagrams are useful for making message structure explicit by outlining how stakeholders and their modules (services, systems, applications, etc.) interact with each other; defining both "Happy" and "Sad" paths. Sad paths detail what the sequence is when something goes wrong, and requires error notification and recovery. The paths provide action to the design, on which to later hang information (such as message structures, data tables, or program classes).



The diagrams can simply be flowcharts formatted with swim lanes for stakeholders to allow for analysis/design issues between members. The messaging interactions can get very complex, and sequence diagrams are one tool to provide the required communication between stakeholders to reduce the difficulty of understanding and achieving consensus on the functionality.

Referencing the *BCM Focus Cycle*:

Action ✍ Event ✍ Decision ✍ Rule ✍ Information ✍ Action✍

The sequence diagram illuminates the 'action' aspect of the *BCM Focus Cycle*. But if designed properly one will also return to enhance the Business Goals or determine additional clarification – such as proper response to business events.??remember this is an iterative process, so perhaps the usage cases need to be enhanced as well, to tell the full story and clearly convey it to the right stakeholders).

Working between the various aspects of the *BCM Focus Cycle* not only makes for a better end product, but also avoids "analysis paralysis" by providing various views. Today's Integrated Design Environments (IDE) are beginning to include a canvas for capturing features; the trick is to find an approach or tool that includes the business users in that process and thereby leveraging *BCM Templates + diagrams* to support it.?

✍

9.1.2.4 Identify Authoritative Sources

From an enterprise perspective an "Order of Authority Preference" per *Community of Interest* should be developed and maintained. This will simplify much of the guesswork as to who is the lead on the definition of the concept. For integrity, the enterprise must clearly identify the prime *authoritative sources.* This includes the location in which they can be found, and how they can be retrieved; repository, Webpage, *Web service*, etc.

Agreement on the authorative source at the business experts level eliminates mapping later in the process, so attempts should be made to discover and use the proper sources as early as possible. A note of caution; internal concept and/or vocabulary definitions certainly appear to be the quickest to market, but may cause alignment challenges downstream, and lead to the expending of valuable resources needlessly.

Unfornuately, there are often multiple authorities/sources/registrations for the same concept or entity, i.e. FIPS v. ISO, demonstrating that having multiple enumerations as well can be a problem. A context driven preference order needs to be defined that guides the selection of definitions, and existing *UIDs*. Keep in mind, that definitive sources can also be found in the legacy forms of policy and trading partner agreements.

The parameters for such as list can be faceted using some basic rules:

- **Established / Emerging / Legacy / COTS**
- **Technology Independent / Technology Included**
- **Standards Organizations / Consortiums / Proprietary / Federal / State / DoD / Enterprise Internal**

9.1.2.5 Develop Business Concept Template

The idea is to define concepts and align to an associated vocabulary which becomes the basis for communication. Stakeholders need to agree at this level, or they can't do business. The key here isn't the '*Term*' as much as it is the '*Definition*' which needs to align.

Here the aliases and multiple *authoritative sources* for the definitions between partners is fleshed out. Don't be surprised to discover what appears to be redundant or dependent sources, or most often five or more terms for the same concept within the organization. This is particularly likely to be true if the organization is the result of multiple mergers or acquisitions. It is suggested that the enterprise build a network of business concept/ term stewards as part of a tiger team to assist with this complex task.

Quite simply, if collaboration partners can't agree at the Conceptual layer then business can't happen. If agreement occurs later at the Business or Extension layers then we achieve reuse.

Instead - today, much of the effort is tactical, and takes place at the Implementation layer where the opportunity is least and redundancy is at its maximum.

Normalized libraries are essential in performing business concept mapping to an enterprise's own interpretation(s). The presumption is that mapping is unavoidable in most cases, and that concept matching is based on identical concept definitions and characteristics such as determined by an *authoritative source*.

Business transaction vocabularies each have different resolutions depending on the stakeholder, (the interest of detail for one party is greater than the interest of another). For example a car parts company may only be interested in ordering a door handle, and is interested only in its product identifier. Their trading partner, the manufacturer, on the other hand interprets the product identifier into multiple fields, which means something to the manufacturer only.

Aliases: Tres	Aliases: DFAS	Physical XML Tag	Business	Definition	Source	Steward	Constraints	Example Data
alternate names for business term	alternate names for business term	the agreed to XML nomenclature to be used in this instance	as discovered in the reference documents (OMB, Treasury, DoD, etc.) and are used by	the precise definition (either verbatim or paraphrased) from a definitive source; the preference order of research is the documents of OMB first, then Treasury, then DoD, then DoD agency, and finally an external source such as an English dictionary or other authoritative source	source of the definition; exact as possible	party responsible for definition and maintenance of concept	Limits on valid values, ranges, etc.	14
DEPARTMENT REGULAR	*A1, DPT, Department ID*	*DepartmentCode*	*Department Code*	*The Department Code represents the current government department or agency responsible for a fund or account and is the highest governmental organizational level at which appropriation, fund, deposit, clearing and receipt accounts is summarized (unless the funds were transferred to it by another government department or agency. (e.g., Department of Labor, Office of Personnel Management)*	*FEDERAL ACCOUNT SYMBOLS AND TITLES, PART IV, Page I-1 (NOTE: This file is most recent list)*	*Department of the Treasury*	*On rare occasions and for short periods of time, the FAST Book may not be current and may not include a recent addition or deletion to the list of Department Codes*	

Business Concepts Definition Template

Note: In general, the metadata capture should be kept to a minimum. Keeping resolution decisions in line with the business is one key and capturing as much system/application generated metadata as possible is another. However, the process should permit users to add extra information, beyond just automated metadata capture though the use of templates, in order to meet a particular business requirement.

Returning to the *BCM Focus Cycle*:

Information ✍ Action ✍ Event ✍ Decision ✍ Rule ✍ Information ✍✍

✍

This aspect of the cycle focuses on the question, *"What do you call...?"*. As gained from Shakespeare, *"A rose by any other name smells just as sweet*." Organizations need resolution on the problem so that when stakeholders use different labels each can still understand the meaning of the exchanged information. However, if the same label is used yet is understood differently depending on context, then that needs to be flushed out at this step – early in the *BCM.* Identifying context is a critical success factor. The *BCM Focus Cycle* needs to be on information, in business terms, and not defaulting to system or technical vocabulary. The *BCM* calls for concept definitions with use of a thesaurus mapping rather than enforced rigid vocabulary (data) standardization.

9.1.2.6 Register Concepts

Concepts should be promoted, and managed so that everyone can discover the artifacts, much like using the use of yellow pages for products and service concepts. Both external vs. internal concepts should be registered; linking external concepts to *authoritative sources* and storing internal concept definitions.

It is important that external concepts can be referenced as needed internally. If not they will have to be learned and 'adopted' by the organization, not for business purposes, but for control and access purposes alone. Hopefully as definition registries come on line, this problem will be ameliorated.

The *BCM* promotes an architecture that supports the idea of *global knowledge*. Architectures such as Service Oriented Architecture (SOA) may read and/or write to common registry/database(s). This knowledge is used to represent a *world-view* of what the service does in its environment – its context. The advantage of having global knowledge is that different services may share their information and abilities for more intelligent combined behavior making for more a more modular and effective architecture. Also it is easy to determine suitability to purpose and facilitate re-use when the context of the original use is known and documented.

Business knowledge is captured in a registry and forms the business library above. The registry contains data, process, and other business artifact definitions including relationships and cross-references as expressed in business terminology. The registry is the bridge between the specific business or industry language and the knowledge expressed by the organization's models in a more generalized industry neutral language.

Building and maintaining point-to-point translators between applications is expensive and usually specific to a particular process or use within a project. Consequently, they are not very flexible or adaptable to new projects or changes within existing projects. A common object-oriented engineering data repository solution that takes advantage of advanced data modeling techniques has significant promise. However it must support industry data standards, provide data translation to and from tools, and provide discovery of repository capabilities, distributed communication and notification mechanisms. The solution should also address issues with communicating semantically, not just syntactically, by supporting varying levels of abstraction and detail of data/information representations.

9.1.2.7 Classification Assignment

Classifications ready the information with the proper structure to be understood and have intelligence applied; thereby providing the critical groupings and links to allow for querying the information as input to business decisions. Library and information science professionals have provided the foundations of an alternative to traditional classification techniques: faceted classification to characterize information-intensive changing business environments.

Tomato – Fruit or Vegetable?

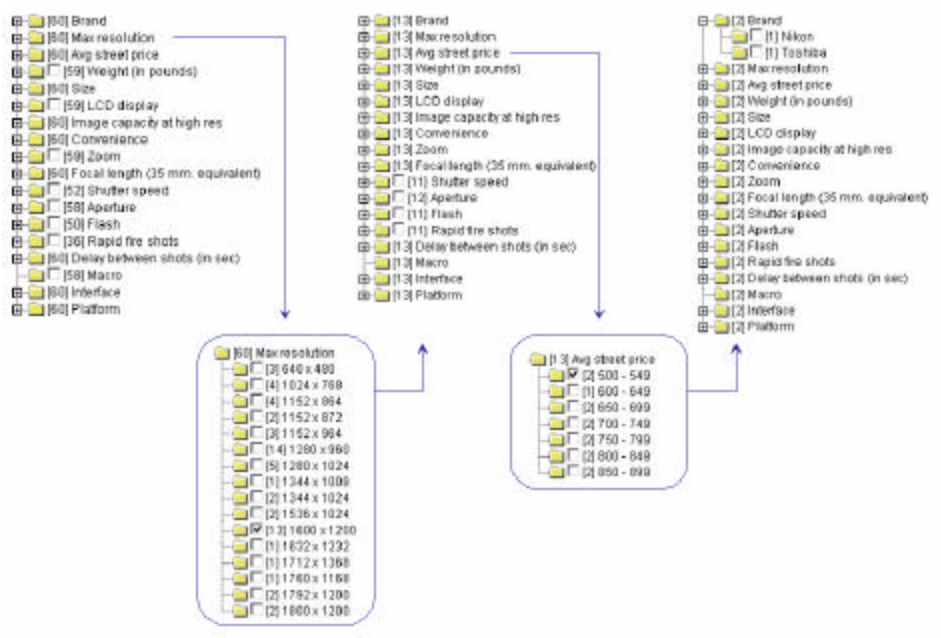
A ripened ovary of a seed plant, the tomato is by definition a fruit, but in 1893 the U.S. Supreme Court overruled Mother Nature declaring that tomatoes were not fruits, but vegetables. The court ruled in the case of "NIX v. HEDDEN" that tomatoes were to be considered vegetables.

Context: **Mother Nature – Fruit**
Taxes – Vegetable

Once registered one needs to be able to effectively search/view the collection of like items; it is this linking which is imperative to understanding generic terms and identifying patterns. It is these generic patterns where one is most apt to find reuse and gain convergent thinking. Faceted classifications aid in searching much like the library Dewey Decimal or Library of Congress mechanisms; applying these with characteristic-specific aspects for each concept will determine the facets.

Key to the above is providing the facilitation infrastructure for artifact discovery and navigation, using faceted classification and ontology to cluster like terms, and at the same time differentiate business term usage through decomposition. Consistent classification greatly increases the probability of discovering concepts by grouping them in a constant manner. Below in figure 9.1.2.7.1 are graphic representatives of facets and how they can be applied so as to complement full-text searching.

Figure 9.1.2.7.1 Faceted Classifications



One can see how a faceted classification differs from a traditional classification scheme in that it does not assign fixed slots to subjects in sequence, but uses clearly defined, mutually exclusive, and collectively exhaustive aspects, properties, or characteristics of a class or specific subject. Such aspects, properties, or characteristics are called facets of a class or subject.

These controls provide for navigation and clarity, supporting taxonomical views:

Multiple faceted taxonomical views -

- *Domain(s) Discipline* — most stable
- *Information Architecture*
- *Business Line* — least stable

Good analogies of taxonomy are shared folders/directories on an organization's network, but with axioms for detailing each node, animal classification, and Yahoo structuring of website entries.

9.1.2.8 Ontology Placement

By combining ontology and faceted classification with search, users gain a map of the resources available to the eProcess. The ontology is a network of concepts (as well as other supporting artifacts of the information architecture) that allows for various taxonomy-based views into the business with the capability of defining thesaurus (e.g. synonyms, alias) relationships, residing in a registry.

Ontology = Set of Relationships = Classifications + Taxonomies + Codelists + Schemas + ….

The registry provides for storing information about the supporting classifications and metadata artifacts. These are (1) link references to external artifacts or (2) links to stored artifacts in the

content management system(s). The links and relationships assist the discovery/search and notification services by providing a mechanism for cooperative actions. Metadata in many cases provides the critical controls and metrics of the enterprise. If this is the case, then only by using the above ideas in concert does the enterprise have a holistic solution for integration. The ontology supplements other search mechanisms, and allows for the quick navigation of artifacts and understanding of the morass of information by providing the 'big' picture.

Ontology provides meaning to data because it puts data in a structured *conceptual network* that is implemented directly from an understanding of the particular information domain. In contrast, a typical application schema is a structured concrete representation of data points that actually exist within a system's scope and therefore only has limited implied context and use information. In addition to navigation, and searching, the ontology is used to resolve semantic conflicts where information appears to have the same meaning, but does not, and naming schemes differ significantly (e.g., synonyms and homonyms). The ontology is meant to answer the what- and-why questions about its domain or common functionality, as opposed to the how-questions.

Primary relationship types:

- Association - denotes a semantic connection.
- Inheritance (generalization, specialization, is-a)
- Has (aggregation, whole/part, decomposition, has-a)

The two areas of needed research are (1) understanding how to best automate the interpretation of a trading partner's ontology and (2) developing industry based common, global ontologies while reflecting the multiple and diverse needs and the evolving nature of ontologies.

The methods for reconciling differences with conflicting ontologies are not well understood – as one attempts to translate and align the semantic concepts and decision trees of each. For the latter, reaching group consensus on "what to represent" in a dynamic, distributed environment is a challenge that should not be underestimated. Work is being done to bring automation for these tasks to reality, but one must have patience working with what they have today, taking one step at a time. Also, the Pareto principle (the 80:20 rule) often applies where substantial progress can be made rapidly by accepting a reduced level of thoroughness to the task, as the overall ROI on the project may not justify a massive information harmonization effort. Limited harmonization of mission critical content may be sufficient.

9.2 Business Layer

9.2.1 Drivers and Constraints

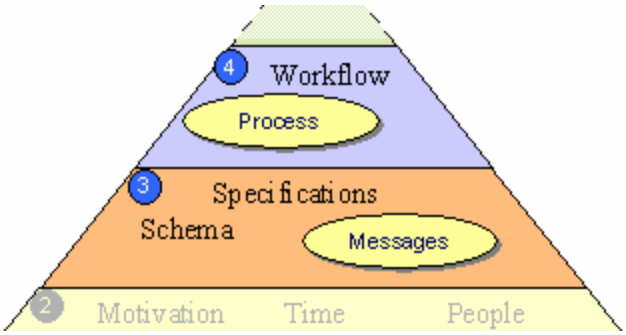
The Target Constructs will fall into two basic types of Use Cases:

EAI - requires that the participants share each other's stores creating a comprehensive data model and process model – an all requirements or *Superset* approach. In the most ideal situation software venders will equip their packages with export and import facilities to a neutral comprehensive data model format. Even then loss of information is unavoidable, because there will be differences between the application data structure and the neutral data structure.

B2B - information that is exchanged within the context of the system that uses it. This implies that the information changes if the context changes. All efforts must be taken to develop common mechanisms to exchange information rather than data. This is a focused data *Subset* approach, but yields exchanges with maximum constraints that are difficult to align with all participants needs.

From a mechanism viewpoint, it is the inverse, the B2B is the superset approach for if an organization solves the B2B problem set through services, etc. the organization can certainly handle EAI requirements - EAI mechanisms are a subset of B2B mechanisms.

Reviewing the artifacts here, the next layers are added on the *Information Pyramid* – opening up for collaboration context specific entries of business processes (workflow) and the Target Constructs (schemas).

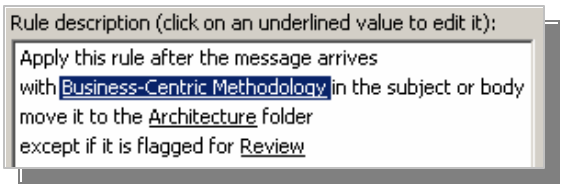


Just like in the *Conceptual Layer*, one gets a first-cut of products in the iterative top-down process – this time the previous *BCM Layer* products should be more stable, as one completes this layer.

- Business Line - Business Pattern
- Pattern – Workflow
- Workflow – Event – Process – Service
- Service – Component - Data – Rule
- Rule – Role -- Security
- Rule – Goals and Metrics

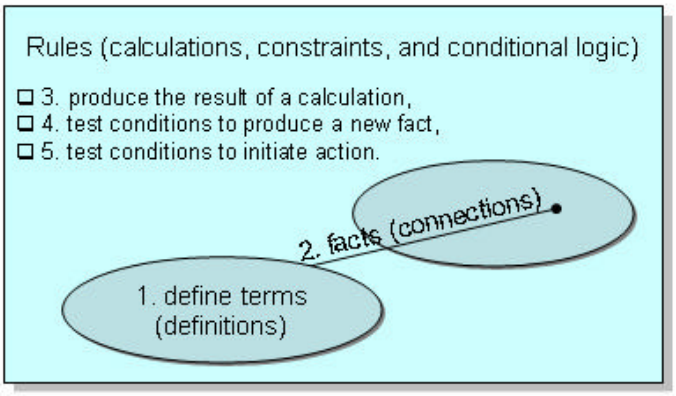
9.2.1.1 Define Business Rules

Business rules answer the question 'why'. Rules guide the behavior of the enterprise and instruct how to use information in carrying out a business action. Rules are the heart of an organization's decision-making capability. Some rules are imposed on the organization from external authorities while other rules are crafted by the organization itself so that the organization functions as its leaders intend – defining its value system. With *BCM Templates* the rules are in a declarative form, not buried and fixed in software application code. As an analogy with which many are familiar, the Microsoft Outlook's rules are described in this manner, for routing and processing of mail messages as shown below:



Defining the business rules and constraints are indispensable aspects of business semantics. Even though system interfaces may be defined, much of the time the precise meanings of the data elements produced by a system have been lost or is indeterminate.

Business rules can be thought of falling into five primary types:



In addition to focusing on the collaboration sequences, *BCM* promotes the sharing of business rules and the decisions of the business; rather than burying such rules in procedural code. Rules buried in procedural code are difficult to find and expensive, if not impossible, to change over time. Rules need to be extracted out and exposed to business users and experts in automated templates for maintaining, checking, and rethinking the business at hand - bridging the gap between the business and technical community.

As with databases, referential integrity implies that defined relationships between data elements and data structures are maintained when data content is added, updated or deleted. The *BCM* extends these rules to achieve wherever possible the appropriate rules for target constructs context relationships and between metadata atomics with templates. If referential integrity within a database breaks down then the data content quickly becomes unusable; likewise with metadata within the *BCM*. Loss of context will quickly lead to unreliable retrieval and the target construct will no longer be viable. With a loss of context, a business strategist can not refine the way existing rules offer business opportunity by changing, adding, or deleting business rules for its business opportunities.

Returning again to the *BCM Focus Cycle*:

Rule ✍ Information ✍ Action ✍ Event ✍ Decision ✍ Rule

and applying that to the *Business Layer*, one come next to the topic of patterns.

9.2.1.2 Capture Business Patterns

A business pattern has been described as the business nature in specific context in order to understand and abstract best practices, or capture the essence of repeatable processes for reuse. Another common definition of a pattern is: "a solution to a problem in a context; especially clever and insightful way of solving a particular class of problems." Without making a consorted effort to identify the organization's business patterns, the organization is destined to 'repeat history' developing stovepipe systems and unable to build an *organizational memory* that learns from past mistakes.

In addition to ontological generalized concepts, patterns are the closest artifacts that organizations have for attempting to document a level higher than information with the *BCM Templates*. Why is this so? Patterns are attempting to capture 'experience' into the mix. After repeating circumstances, one begins to combine like instances in a general form that one can leverage the next time one addresses 'like' tasks. That is, it provides multiple viewpoints of a problem, which have been considered, with the result being the most general and flexible solution for this particular need that can be leveraged from the *organizational memory* to aid with the task.

Software programming has had the most success, perhaps because it allows the programmer to prefer composition over inheritance – by adding a layer of abstraction. Programming design patterns success reaches across horizontal domains, but one certainly can envision some business

patterns that cross multiple domains, such as 'agreement' or 'reconcilliation'. Much can be gained with community-based patterns or even enterprise-based patterns even if to a lesser degree. Enterprise matadata strategy should include mantainance of patterns.

Below are examples of patterns for business.

Verb-oriented

If workflow is described as a process in whole or in part, then a pattern is one level of abstraction or the "best practice" of a process as learned from experience.

- Contract (Check for serviceability)
- Negotiation (Check and variable for pricing eBay Auction Proxy/Agent)
- Reconciliation
- Document (outline… edit… signoff)
- Business Reference Architecture
- Information Aggregation (Rollups)
- Procurement(s) (simple, large, services, products) (Buy, Sell)
- Meeting (finding a room, invite, agenda… notes)
- Shipping (to carrier, track, accept, call reconciliation pattern)
- Travel Reservations
- Publish/Subscribe
- Integration (verb/services, noun/edi…)

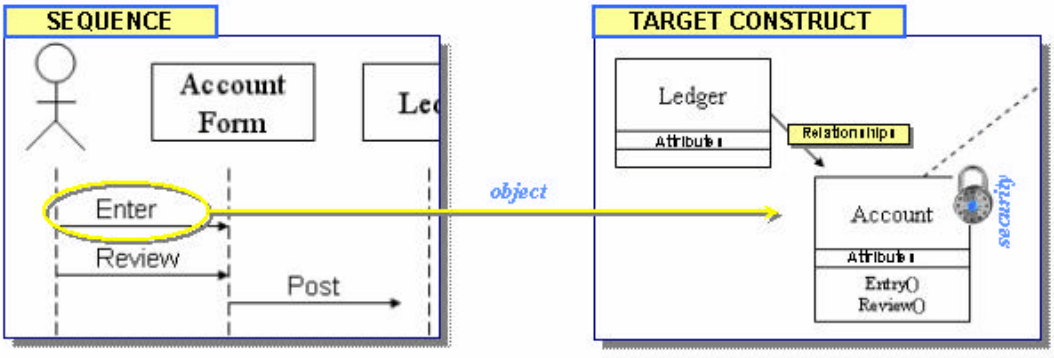
Noun-oriented

By using declaratives rather than procedural logic one begins to see 'forms' or structures in the nature of the business.

- BCM Template approach: Feasibility, Risk, Cost Benefit, Business Rule, Workflow, CAM…
- UID, unique key
- Header / Payload
- HTML page with META components (somewhat the same as above)
    - Verb to this: Download form, complete, submit, next hyperlink page
- Tree (Hierarchical/"Composite")
- Status Log
- Classes (groupings) e.g. Long-Line of Accounting, DoD Classwords

9.2.1.3 Atomics and Constructs in Exchange Scope

The task is to develop further the sequence diagrams and for each message or message set in the sequence set, identify the organization's business objects/constructs that are being exchanged. Business users should attempt to collect like objects and understand that from a developer's perspective universal constructs allow for common functions, thus reducing the overall cost.



The need is to extend the sequence process to a formal description of the information flow and capture that in a *BCM Template.* The *Target Construct* needs to trade-off application specific metadata with adaptation to new standards. For maximum flexibility an enterprise needs to provide a strategic view – or *Target Construct* - where business transaction data structures and application data structures can be mapped. The enterprise *Target Constructs* need not be implemented, but will serve as a stable reference.

If required from a business point of view security attributes are placed on constructs, per their role, at this step as well.

9.2.1.4 Structure: Resolution / Indenture

A common problem in managing resolution is determining what resolution fits the business best. For example, finding which resolution provides for the greatest flexibly without leading to a dizzying array of options that are often unused, misused or just not useful. Most users appreciate *specific construct* rather than *general constructs* (that do not always serve them precisely). Of course, it is also entirely possible that the simpler solution is the more general construct.

Keep in mind that *BCM Templates* can select (switch) *Target Constructs* or aspects, where a technology such as XML Schema does have support today. Also XML can handle indentures well, whereas this may require multiple joins that would slow down a relational database. It is quite possible that the *Target Construct* is the same as the relational database, if the database design was done properly.

9.2.1.5 Workflow / Process Identification

To assure a streamlined process an organization needs to think in terms of its entire value-chain as being customizable – 'the customer can have it their way'. Quite simply, organizations that do this are proactive and those who do not are reactive. A workflow of the exchange needs to be developed or adopted and provided with easy access for all parties. Understanding and including the organization's business metrics allows for managing by exception, a very powerful position. Managing by exception allows the organization to get its "heads out of the trees and see the forest." Workflow isn't only for automation but to provide visibility into the process, assuring business goals are clearly managed and customers get what they need.

One can think of workflow as presented in a UML diagram such as the Component, or Activity diagrams of IDEF products. With the key difference that one may want to address the *value-chain* that includes the organization and its collaboration partners. This view is enlightening, especially if this is the first time reviewed. One may find duplicate processes, double or triple checking of values unnecessarily, or collaboration of sources to increase integrity.

The business meaning of a data element is defined by the ways in which it may be used. Business rule metadata helps end users understand the lineage of the data as it flows through the Enterprise. As information progresses through multiple systems and processes, various business rules apply based on context of the information. The roadmap will need to call for a common enforceable mechanism to address the semantics of their data flows and varying information models.

Other than data modeling, process or workflow has a rich heritage from which to draw. With *Web services* there is now much interest in bringing a choreography aspect to simple remote procedure calls. The next few years should provide enterprises some very exciting opportunities for defining and executing their flows both internally as well as external to trading partners.

Beware that UML hasn't gained the acceptance at the speed first envisioned. This is due to the following reasons, as cited in a recent IT survey of software developers:

- Don't see any benefit
- Not supported by the organization's tools
- Too expensive to implement
- Too complex to use
- Not production ready
- Too complex to learn

Returning to the *BCM Focus Cycle*:

Action ✍ Event ✍ Decision ✍ Rule ✍ Information ✍ Action

9.2.1.6 Focus on Attribute Details

Experience tells us that the final decision of optional vs. mandatory needs to be defined in *BCM Templates* and be based on context and nothing else. Each collaboration partner will view the same information definition and requirement differently – a tracking number for one is absolutely critical for reconciliation of shipments, where as the number is meaningless to the other, and is only asked to be returned for use in subsequent exchanges. However the collaboration itself applies to internal as well as external entities, and therefore the context must be able to support all instances and usage.

Likewise codelists are specific to the needs of the collaboration partner. This is especially true if the same definition is to be used by multiple partners. This leads into another thorny problem affectionately labeled *"multi-field challenge"* where the code sets are used in conjunction with other fields to carry the full semantics to be exchanged. This is a complete discussion by itself;

suffice it to say that the *BCM* with a registry base for resolving values in context seems to be the best solution that organizations have today.

9.3 Extension Layer

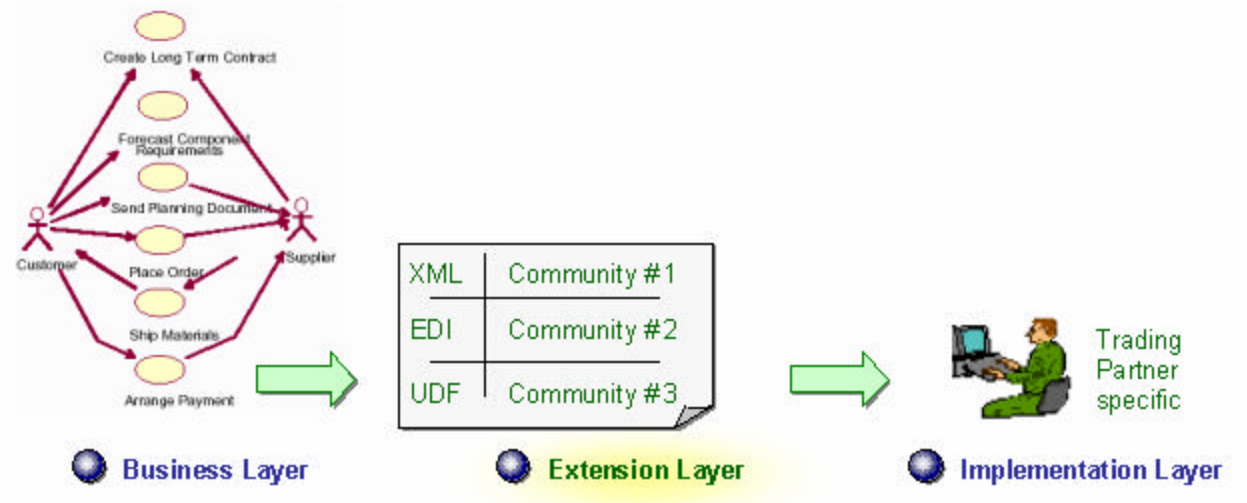
9.3.1 Drivers and Constraints

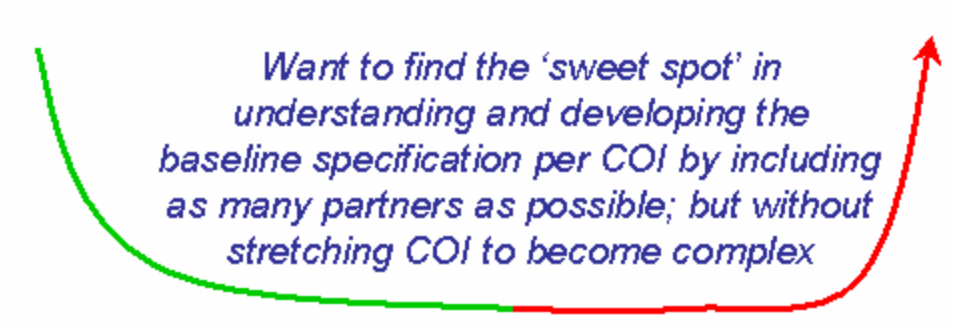
The previous *BCM Layers* focus was on internal requirements, building from the needs of the organization almost exclusively. In this layer the focus is to support heterogeneous collaboration partner environments; preferably within the existing application capability, while supporting moving to future needs. Legacy applications can become reusable components through encapsulation, such as by using *Web services* or proxy servers. There is no technical reason to throw away valued applications, especially if one considers the risks involved in precisely replicating critical business processes. It is relatively easy, inexpensive, and low risk to encapsulate rather than the alternative of completely new developments. *Web services* can apply to legacy batch processing and message-oriented online applications. Therefore, if the legacy applications are still fulfilling their business purpose, encapsulation may be the best strategy, particularly if you can also resolve any other structural issues during the revised implementation.

9.3.2 Tasks

9.3.2.1 Role-Process Identification

From previous defined Use Cases, stakeholders need to be identified, and grouped accordingly. The grouping can be based on any parameter that makes sense to the business, and offers opportunity for reuse, e.g. type of data feed, type of system, geopolitical – business flow patterns and how the community will implement them. In the previous stages in the *BCM Layers*, one generically identifies processes and roles. As one discovers the '*who*' and '*how*' - *verb* aspect one specifically identifies each based on the legacy system or framework in terms of their outreached stakeholder community.





9.3.3 Standards & Framework Adoption

As the definition progresses, the organization aligns its concepts and target constructs to external partners or legacy systems. The alignment analysis (toward the noun aspect) addresses the '*what*' in the communication equation as shown in the example below:

Legacy PDM*	MIL-STD-2549	X12 (EDI)	STEP AP 203
Part No	Part Product Identifier	Product/Service ID	Part Number
	Part Product Name	Product/Service Name	
Supplier	**	Entity (Supplier) Name	Supplier Name
Contract No	Contract Document Identifier	Buyer's Contract Number	Contract Number
	Component Product Quantity		Component Quantity
Doc Type	Document Type Code	Report Type Code	

The ISO5964 standard is an area for further research into the documentation and establishment of multilingual thesauri and identifies the following types of relations:

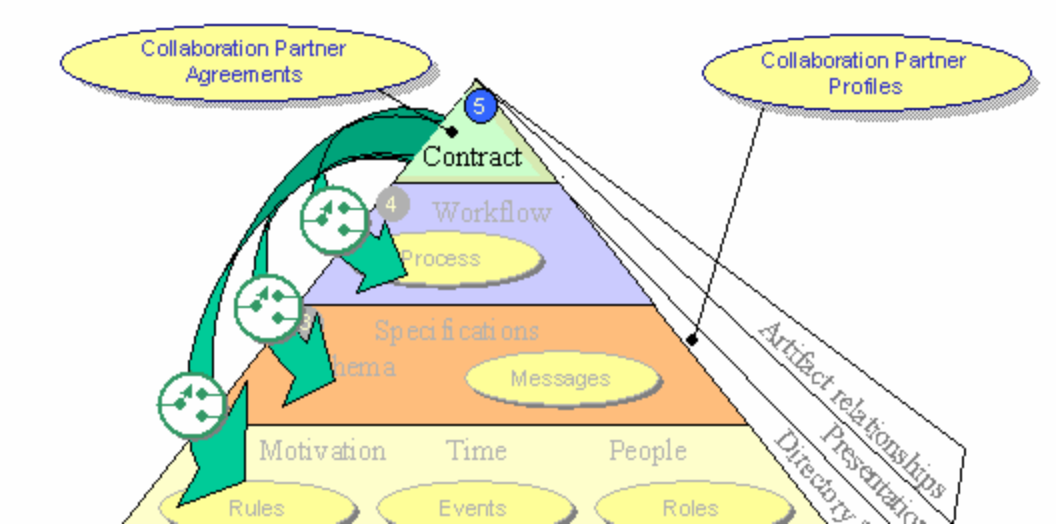
- ? exact equivalence
- ? partial equivalence
- ? single to multiple equivalence
- ? inexact equivalence

These relations indicate that the semantic relations between terms from different metadata vocabularies are likely to be much more complex than one-to-one exact equivalence and that even "exact equivalence" will be an approximation. The ontology and thesaurus base is extended for each community. Because the scope of the challenge is limited to business relations the solution is manageable in comparison to that of a general natural language thesauri. The product at this layer is the mapping between target constructs and that of external standards or legacy systems.

9.4 Implementation Layer

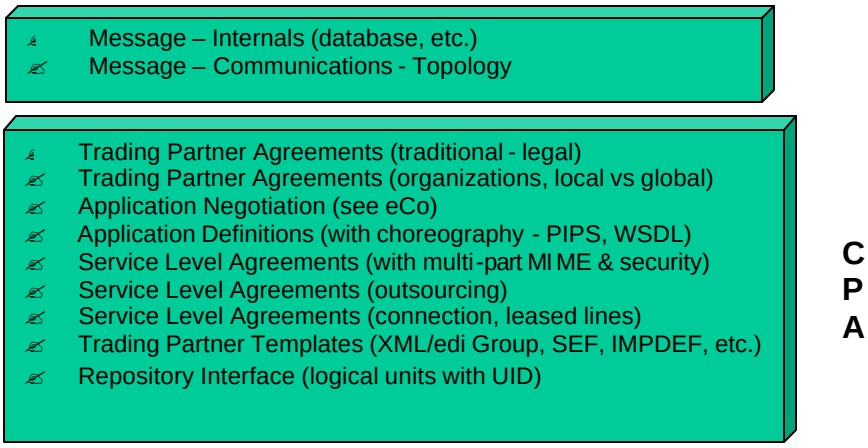
9.4.1 Drivers and Constraints

For each stakeholder (or group of stakeholders if possible) a *Contract* is established based on the Memorandum of Understanding or Agreement (MoU/MoA). The Contract is the formalization and linking of supporting *BCM Templates* for that business deliverable.



In essence the process has come full circle, as the Contract with a collaboration partner or community provides the detailed definition from a business viewpoint, as they should be incorporated. The Contract is viewed differently than the MoU/MoA. At this stage the *Contract* template turns on (selecting/invoking) a chain of linked *BCM Templates*, and sets the overall context of the processes.

The types of deliverables can vary on circumstance, and there are many that a large enterprise will need to manage, listed here is but just a few.



9.4.2 Tasks

9.4.2.1 Tailor Collaboration Partner Specifics

Technologists develop interchanges and user interfaces using *Target Constructs* or *Baseline Specifications* and their supporting products within partner constraints.

One simple example is converting the representation of data from numeric to a character string. These conversions are well known and the problems documented. Many of today's data sources, such as databases and applications can automatically export information into standard formats, such as eXtensible Markup Language (XML), by using built-in data transformation with code-free mapping tools. The accessibility of the information, or transport problem, has been reduced to routine engineering tasks due to widespread investment in messaging infrastructures.

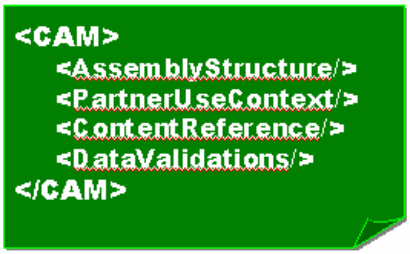
9.4.2.2 Content Assembly Mechanism (CAM) Template

The OASIS CAM defines the structural formatting and the business rules for the transaction content. This drives the implementation step of linking the derived final contextual details to the actual application information and mapping between components stored in the Registry. The declarative approach states the input and output path locations. The CAM Template uses plain XML to describe destinations, which all XML-based tools can understand.

Reference OASIS CAM TC: http://www.oasis-open.org/committees/cam/

CAM Template attributes can be summarized:

- Uses well-formed XML structure with in-line directives to describe content model and supports legacy formats
- Uses XPath, *UIDs* and declarative predicates to state the MIG (Message Implementation Guidelines) or IC (Implementation Convention) in machine accessible format.
- Allows for localization and substitution structures
- Provides referencing to component semantics in registry or inline locally.
- Makes consistent assembly possible, and drives adoption of *Target Constructs* for transaction structures.



9.4.2.3 Ontology Providing Interpretation Support

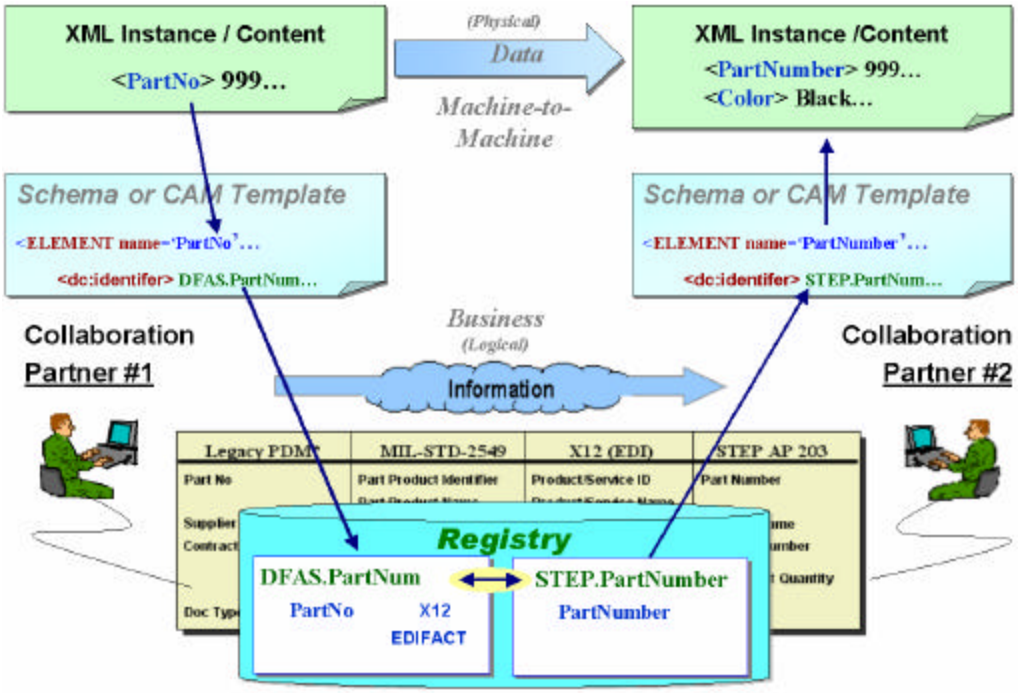
The ontology provides mitigation support allowing for Enterprise–level crosswalks and light transactions. With business artifacts keyed using a *UID* in transactions that allow referencing

into repository instead of having to repeatedly carry the same information. Crosswalk information such as the link that states Collaboration Partner #1 vocabulary of *PartNo* is equivalent to Collaboration Partner's nomenclature of *PartNumber* allows each domain to work and grow their vocabulary independently of each other. Thus each domain can grow and adapt faster.

Context everywhere through 'help from above' (provided by previous layer definitions):

- It is impossible to unambiguously define information for all potential uses unless the proper metadata is defined in context
- Context eases integration and reduces cost
- Metadata accessible throughout the workflow for interpretation

Also additional information that is stored in the registry is available, such as *Color*. The diagram depicts the XML instance being light, with the *UID* reference in the Schemas which link to the registry. The registry stores information about the business artifact other than crosswalk information to assist in the exchange.



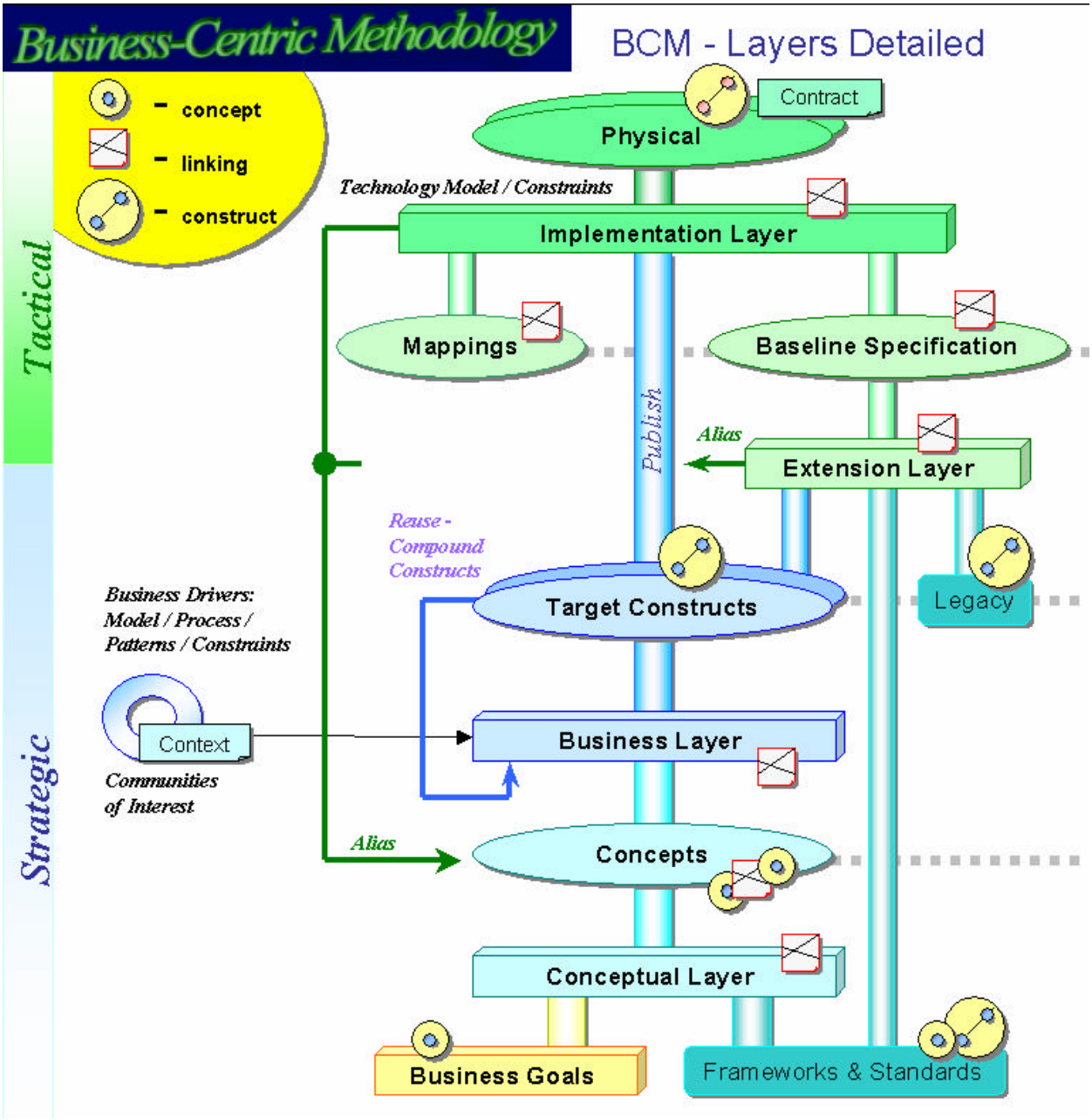
The benefits of the Registry are:

- Allows for discovery of processes – for function and service which to build applications
- Promotes reuse – system developers can locate a business object in the Registry will save time and effort, and reduce the number of required crosswalks
- Enables efficient version control – the Registry enables tracking multiple versions of a business object efficiently
- Promotes unified understanding of registered objects - metadata for registered objects are accessible from a single location, a unified understanding of the purpose and rationale can be maintained

- Allows for collaboration – finding partners (internal or external) connected to the metadata to share ideas and receiving notifications as to configuration changes
- Enables navigation of business – with metrics assigned via processes or users, management can see at an enterprise level operations at a glance
- Assists with impact studies – provides input as to changes and how it impacts the organization, also benefits gap analysis as well
- Collect independent metadata – which is separate from COTS tools to supplement capture of required business information that can not be housed in the products
- Organization's methodology – through the use of consistent templates and information-driven wizards for capture of user's input
- For orchestration of services – by taking a information-driven approach to sequencing and invoking functions throughout the enterprise, and at the enterprise level

Alternately, if two entities register independently or the registry is federated (combined) with others then a linking of *UID*s will be required for the look-up.

1934 For reference the following diagram is shown below:
1935



1936

10 Infrastructure and Implementation Support

This section considers the *Implementation Layer* and the infrastructure components needed to fulfil the requirements directed by the *BCM.* First off one needs to understand and quantify what those are.

The goals of the *BCM* can be summarized as follows:

- addresses the root cause rather than just symptoms of the organization's integration problems by providing *semantic* and *pragmatic interoperability*

- is *business-centric;* shifting power to the business experts; managing Enterprise artifacts and governance through *Communities of Interest*

- directly enables the model; provides coupling between the *BCM Templates* and the *Implementation Layer* via *Choice Points* to ensure that the *linking and switching* occuring in the deployment environment matches the actual business requirements.

- exposes *context* instead of embedding it; provides visibility, accessibility, understandability, using open *declarative mechanisms* that allow for *mass customization* of diverse vocabularies and models within *heterogeneous environments*

- insulates business from the high rate of change of technology by dividing the problem into multiple levels and applying constraints properly to reduce complexity and promote reuse

- provides for Enterprise agility and prepares the Enterprise for new opportunities in doing business

Following on from these statements one can then begin to understand the support required for each item. It is important to note that the *BCM* is agnostic to the implementation technology itself and only directs that whatever technology is selected that it supports the fundamental capabilities needed above. Each of these items will now be considered in turn and assessment made of what technology components and capablities are required to deliver on each.

Following that is presented an overall feasible information architecture diagram that combines all these components synergistically. Again, this diagram is intended to be agnostic to technology but is obviously orientated toward todays Service Oriented Architectures and solutions since it is intended to point at what is feasible today (see figure 10.6.1).

10.1 Providing Semantic and Pragmatic Interoperability

10.1.1 Approach

Key to the above is providing the facilitation infrastructure for artifact discovery and navigation and the classification and ontology for the clustering of like terms and to differentiate business terms usage through decomposition.

The prime shift components are:

1. Taxonomy/Ontology,
2. Registry,
3. Workflow, and
4. Content management system.

The ontology is comprised of various facetted taxonomy views of the business with the capability of defining thesaurus (e.g. synonyms, alias) relationships that reside on a registry. The registry provides reference assistance and stores information about the supporting classifications and metadata artifacts. This occurs independent of them being link references to external artifacts or links to stored artifacts in the content management system(s) and processed workflow.

The workflow allows for the status of the enterprise's value-chain 'pipelines' to be analyzed and corrections made quickly (see section below on *linking and switching*). The links and relationships assist the discovery, search, and notification services by providing a mechanism for cooperative actions. Metadata in many cases provides the critical controls and metrics of the enterprise (directed through the use of *Choice Points*) and only together with the ideas above does the enterprise have a holistic solution for integration.

10.2 Shifting Power to the Business Experts

10.2.1 Approach

Following on from 10.1 and providing the means to manage the domain and its semantic representation, it then follows that this allows the managing of Enterprise artifacts and governance through *Communities of Interest*. Most significantly this includes the linking of business goals, to concepts, and exact business requirements, through mappings, and physical implementations using the *BCM.* The business partners are then able to reuse their own declarative community semantics in loosely-coupled machine readable mechanisms like: ontology's, classifications, industry vocabularies, patterns, etc. within their normal business processes with precise context when business opportunities arise. The advantage is that they are not required to learn a new technology every couple of years. However, business is capable of rapid response to emerging opportunities because the technology is "clear boxed" through the use of *BCM Templates* and *netCentric* technologies.

10.3 Directly Enabling the Model

10.3.1 Approach

In traditional information technology development there is a separation between the architects and the implementers. So that the original 'blue print' designs are disconnected from the build-out process and are never updated and maintained to reflect the final product(s).

In the *BCM* the *BCM Templates* capture the 'blue print' of the business requirements and design. The information and semantics in the templates is exposed as XML rendering to the application *Implementation Layer*. This enables the business experts to direct the technology solution from the *BCM Templates*.

This same approach has of course been promised using CASE technology prior to. However there is a fundamental difference between the representations in CASE tools (such as UML) which are tailored to information technology requirements, as opposed to *BCM Templates* that are focused on "*Business First*". Consequently business users do not require specialized training to utilize *BCM Templates*; since the templates use business terminology from the *Community of Interest* directly. (Note that UML tools have their applicability to the software engineering tasks of the solution and providing representations and understanding the ontology between components, as has previously been noted).

10.4 Exposes Context Everywhere

10.4.1 Approach

Everywhere one turns today one sees people developing XML vocabularies for business transactions. There are basically two schools of thought.

a) the standard defines a bespoke set of information unique to the specific industry – and one will build and extend as necessary. Here are the XSD schemas for the current set, and the data dictionary.
b) The standard defines a carefully collected set of core components of nouns and verbs that are assembled into transactions and are reusable across domains. Here are the XSD schemas built up using core components that are carefully designed to fulfill all needs exactly. Alignment on core component dictionary ensures interoperability.

Both suffer from the same limitation in that they both fail to take sufficient account of *dynamic context* as the fundamental driver behind all information exchanges. Transactions contain only data unless the context is known as well, and then it becomes information.

The *BCM* focuses on the need to provide visibility, accessibility, understandability, using open *declarative mechanisms* that allow for *mass customization* of diverse vocabularies and models within *heterogeneous environments.*

The two examples above can be ameliorated if context can be applied globally across their solutions. The OASIS CAM (Content Assembly Mechanism) specification illustrates one way of engineering for context as the foundation of the organization's transactions. It provides a mechanism to retroactively apply context to existing transactions. CAM templates also enable registry components to direct the semantics across the transactions from a single declarative mechanism through its use of content references linked to registry aliases.

These techniques for transaction content management should be studied and understood. In addition to transaction content there is also a need to expose context in the business processes themselves. Fundamentally this is driven from business collaboration agreement in the *Conceptual Layer*, where the business context is agreed and captured into the *BCM Templates*. This then transitions across the remaining *BCM Layers* providing that context. As shown under the discussion of context, there are many context types that need to be managed. As summary is provided here:

- *Community of Interest* determination
- Business agreement context
- Business agreement roles
- Classification of artifacts context
- Process selection context
- Process tracking context
- Transaction context
- Exception handling context
- Decisions context
- Rules context

By enabling the exposing and control of these context parameters through declarative mechanisms in the *BCM Templates*, this fulfils the business requirement to engineer agility into the *Implementation Layer*.

Further more *Choice Points* can be seen as providing three enablers for agile information exchanges:

1. Context that extends beyond the local decision point, and if persistence of decisions is required
2. Context by refining criteria dynamically, and that may include from undetermined start points
3. Context requires a thread to establish and track the state of a process.

Full details and discussion of *Choice Point* implementation is provided in Appendix B.

10.5 Using Layers to Reduce Complexity and Promote Re-Use

10.5.1 Approach

The *BCM Layers* are designed so that refinement can be deferred to the level above as the method is applied and the *BCM Templates* completed. The result of this approach is that within each layer itself the templates contain sufficient information only. Multiple benefits derive from this approach. Most important is that you only ask questions of practitioners that you know they can understand and answer. The next benefit is that this enhances re-use since the context has been exposed and therefore it is much easier to re-purpose the particular artifact knowing that there is not a lot of embedded logic that might otherwise fail or be out of context.

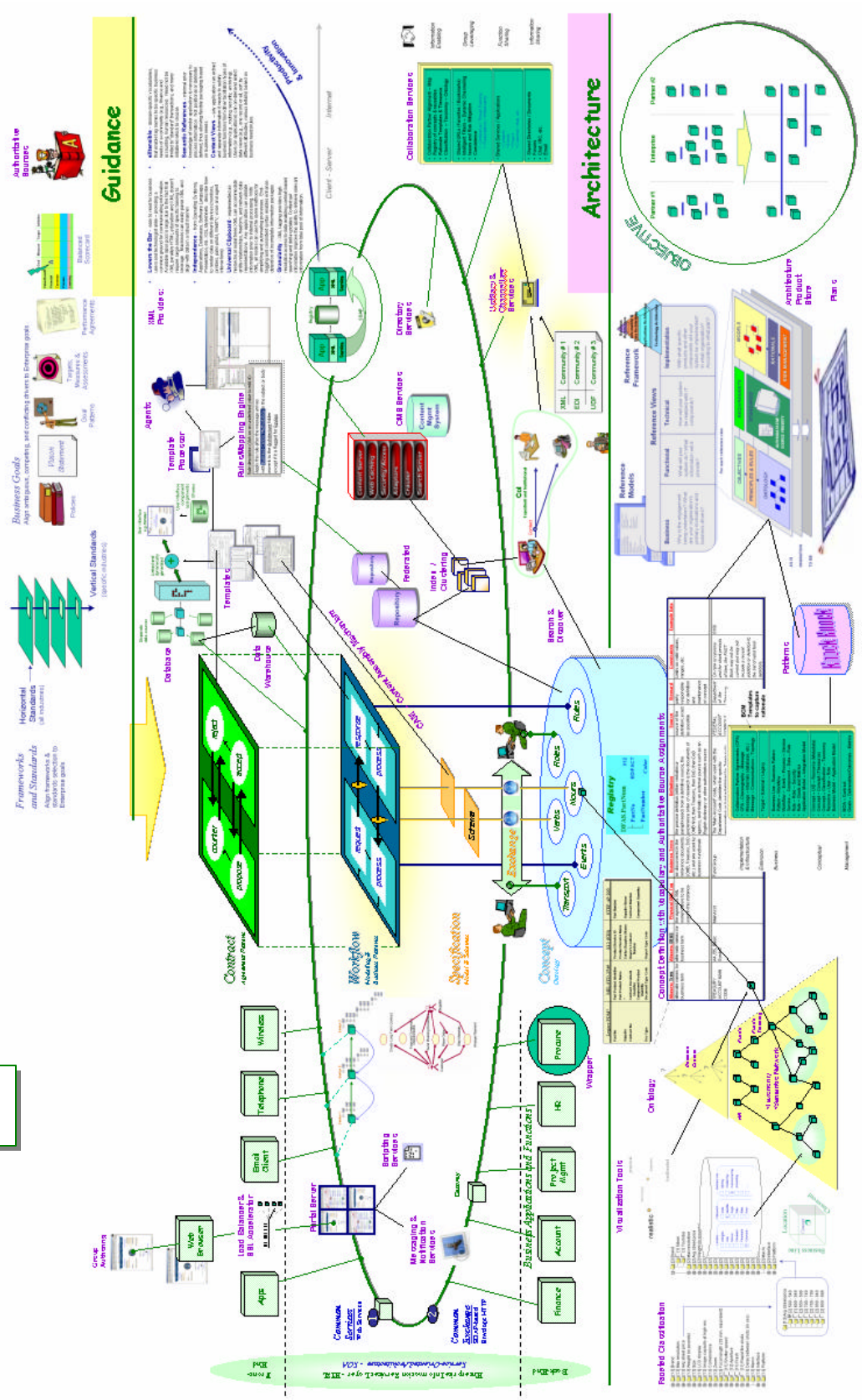
It is therefore key that the *BCM Layers* only resolve the semantics applicable to their focus and that they externally reference and derive all other semantics into the layer above them. When constructing the *BCM Template* tools and mechanisms implementers should enable this as a fundamental ability across a project of templates.

10.6 Architecturing for Enterprise Agility

10.6.1 Approach

The following diagram is presented as an overall feasible information architecture diagram that combines all the components listed above synergistically. This diagram is intended to be agnostic to technology but is obviously orientated toward todays Service Oriented Architectures as the focus is what is feasible today (see figure 10.6.1).

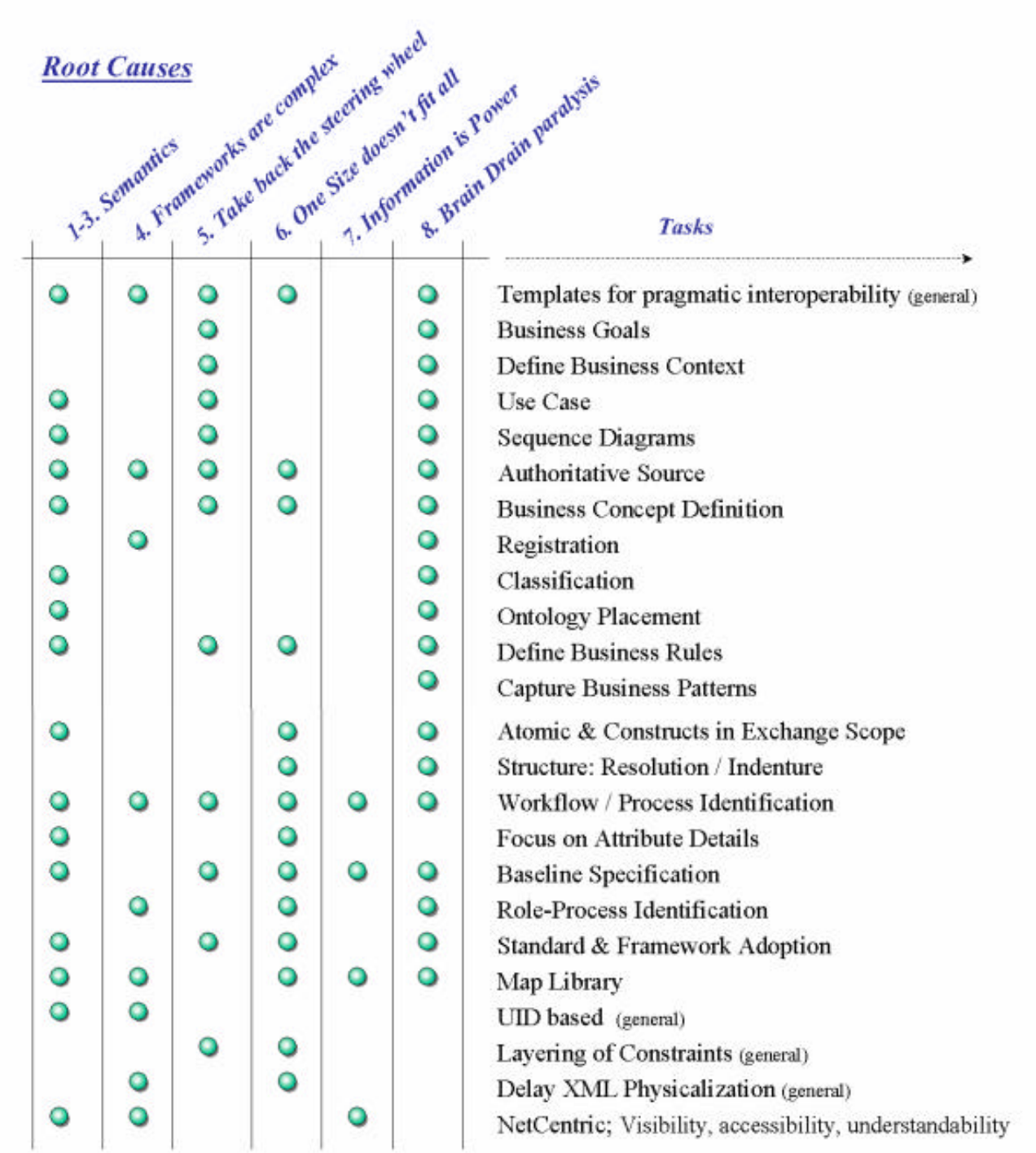
Figure 10.6.1 – Feasible Information Architecture.



Posters can be found at:
http://dfas.info

2129

2130 **Checkoff List:**



2131
2132 **Figure 10.6.2 – Factors for implementation approach**

10.6.2 Further Considerations

A tactical-only solution is a waste of money – organizations need to adopt an Enterprise solution that addresses business context and people.

Organizations need to build with existing infrastructure and have 1, 2, 5, 10 year plan

- Leverage portal efforts to derive organization's ontology
- Develop support network of part-time metadata managers and teams
- Apply methodology to proof-of-principles and new developments

Long term, the goal is to provide an approach that will weather continual industry rolling changes to the physical *Implementation Layer* technologies. With the correct framework the Enterprise can focus on the operational requirements instead of having the implementation tactical details cloud the overall delivery. Better yet, the Enterprise can not only take advantage of technology innovations that complement and enhance the architecture, but also provide the environment to foster vendor development of technology that exploits instead of attempting to make obsolete the deployed systems. In short, *BCM* provides the base for mass customization required - supporting the enterprise's stakeholders and customers.

11 References

Applicable references are listed below:

In a few cases, the only available specification for a function is a proprietary specification. These are indicated by notes within the citations below.

a. [ccOVER] ebXML Core Components Overview, http://www.ebxml.org/specs/ccOVER.pdf.

b. [ebBPSS] ebXML Business Process Specification Schema, http://www.ebxml.org/specs/ebBPSS.pdf.

c. [ebMS] ebXML Message Service Specification, http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf.

d. [ebRS] ebXML Registry Services Specification, http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrs.pdf.

12 Disclaimer

The views and specification expressed in this document are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this design.

13 Contact Information

Bruce Peat
eProcess Solutions
Phone: 301.613.1521
email: BPeat@eProcessSolutions.com

Mike Lubash
Defense Finance and Accounting Service (DFAS)
Phone: 703.607.1166
email: Mike.Lubash@dfas.mil

David RR Webber
Gnosis, Inc.
Phone: 301.693.1000
email: Gnosis_@compuserve.com

Notices

Portions of this document are copyright (c) 2003 OASIS.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights.

Appendix A Template Examples

Example of Templates; Fields and definitions

Appendix B Tempate Linking and Switching

Choice Point Service

Appendix C Information Architecture

Expand out the *Information Pyramid*

Appendix D Concept Definition

Concepts of the *BCM* and their definitions

Appendix E Concept / Terminology Alignments

Alignment to other initiative's vocabulary

Appendix F Abbreviations

Making sense of coded language

Abbreviation	Term
BCM	Business –Centric Methodology

Appendix G BCM Rationale

BCM Story and Opportunities