

CBSE in the Realm of Computing / Information System Life Cycle

Lana Kuzmanov , kuzmanov@sympatico.ca

A position paper for Second International Workshop on Component-Based Software Engineering in the category "Practices for Adopting CBSE".

1. Introduction

In the never-ending quest for efficient and high quality solutions to the IT problems that industry organizations experience in a wide range from strategic to day-to-day, they tend to gladly embrace new techniques. For its known features and benefits, CBSE falls into the category of emerging, promising approaches. However, in the effort to position itself for its adoption, organization inevitably faces questions such as: How to fit CBSE into its business process life cycle? How should the CBSE process itself work? How to enable and support CBSE process execution? How to approach CBSE process implementation? This paper is addressing the above questions.

2. Positioning the CBSE in an organization's process life cycle

Position Statement 1. An organization must position the CBSE process within the framework of its business process life cycle by defining its relations with other processes (Figure 1). The rationale for this is implied by the fact that success of an organization is directly proportional to the quality of business processes design and management of those processes. For example, the quality of results (products and/or services) and the efficiency of execution (least time with least resources) are crucial measurement parameters of a particular process. Applying the approach where activities are executed concurrently and providing utilities for information assets reuse throughout the process, contribute to high rating of these parameters.

Process Architecture.

1.) Business Process (BP) life cycle: *Scope / Problem Context*. Includes business drivers definition, BP modeling, BP planning, BP management, and BP evolution. Results of business drivers definition activities are used in the architecture life cycle at the strategic design level. Relevant deliverables of BP modeling activities for our consideration are data requirements, computing/ information services requirements and computing/ communication requirements. These requirements represent a base (necessary input) for Component-Based system design as a part of architecture life cycle process.

2.) Architecture life cycle: *Solution / Design Context*. Component of Computing / Information System life cycle. System engineering activities, such as component-based analysis, design, and development (i.e. CBSE), are embodied within this context. Detailed in Section 3 (Figure 2).

3.) System life cycle: *Deployment/Operations Context*. Component of Computing / Information System life cycle. This is the context of actual system operations comprising the activities of system management, component (data, software, computing and communication) distribution, production implementation, system administration and support. Detailed in Section 3, Figure 2.

4.) Supporting systems life cycle: *Support / Enabling Context*. Our scope (this paper) only covers Computing / Information Models Process Life Cycle details of which are addressed in Section 4.

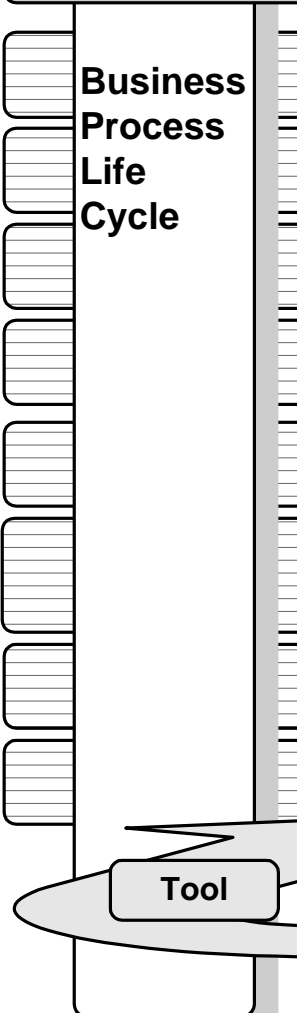
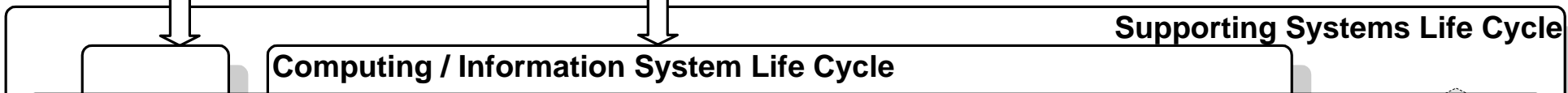
Process Enablers.

In order to execute and deliver product or service, process requires support of enabling components. These components are methodologies, organization, environment, budget, culture, skills/competence and tools and are called process enablers.

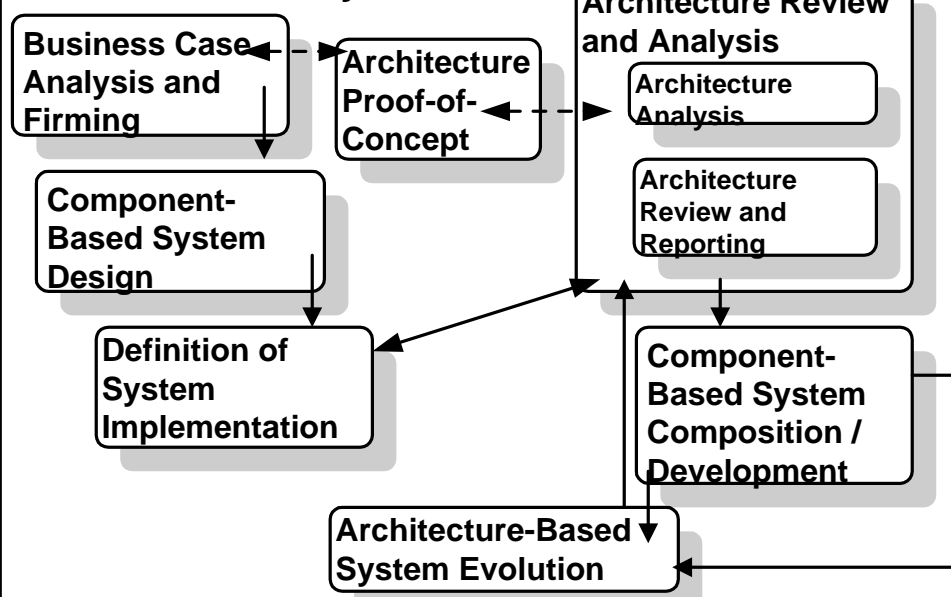
Management of Process.

Business process requires management in all of its segments. It is critical that the management of process be synchronized across Business, Computing / Information System and Supporting Systems. Management of process includes models, trends, planning, metrics and alignment.

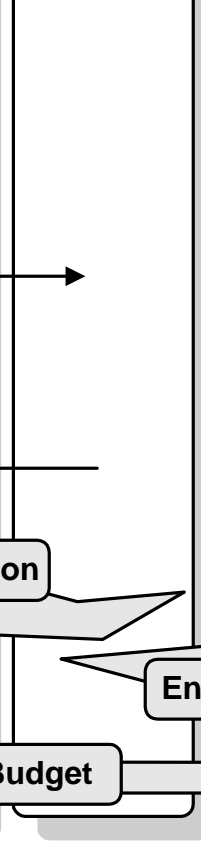
Figure 1. System's Process Life Cycle (*figure appears on next page*)



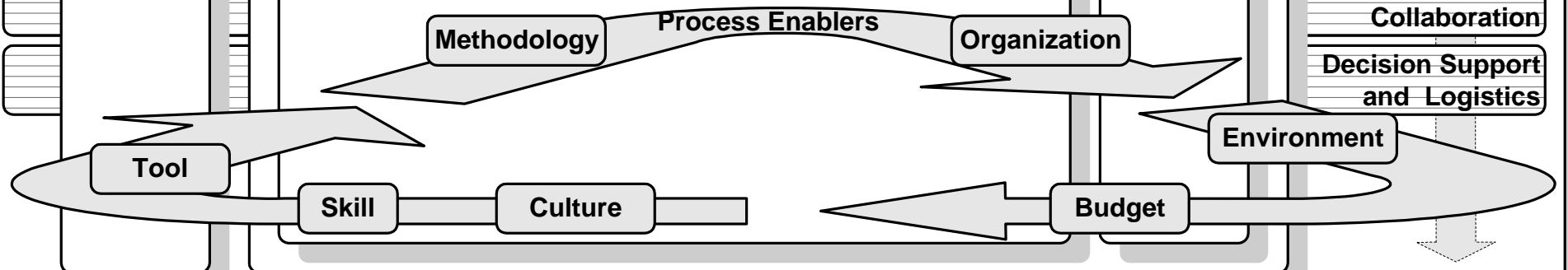
Architecture Life Cycle



System Life Cycle



- Project / Program
- Research and Analysis Support
- Documentation Management
- General Reporting
- Service Level Management
- Problem and Change Mgmt Support
- Work Group Collaboration
- Decision Support and Logistics



3. Defining CBSE Process

Position Statement 2. An organization must adopt (choose, choose and customize or create) CBSE process that suits its needs (Figure 2) focusing on the fulfilling the following: 1.) Applying an architecture-centered approach [BCK]; 2.) Compromising between requiring rigorous procedures (that introduce process complexity, require over-documenting, make achievement of process phase exit criteria difficult, etc.) and manageable, user-friendly procedures; 3.) Requiring concurrent execution of activities throughout analysis, design and development processes; 4.) Standardizing process supporting and enabling framework (computing / information models base, toolkit, etc.).

Figure 2. CBSE Process Diagram (*figure appears on next page*)

Business Case Analysis and Firming

1. System Business Scope Analysis and Firming
2. Business / Functional Requirements Analysis and Firming
3. Quality Requirements Analysis and Firming

Component-Based System Design

1. Architecture alternatives design (if applicable)
2. Define / Qualify Components (from organization's Components Repository or COTS from external repositories)
 - Define services/interfaces (mapping "Import - Service-Export / Behavior") for component "A" "B" "C"
3. Design Components Architecture
 - Specify interaction of component "A" (interfaces to the Infrastructure & Specialized Data Access Service) "B" "Z"
 - Design components / controls structure
 - Design data flow structure
4. Design Data Model (Choose or Create)
5. Design Computing and Communications Platform Architecture (Choose or Create)

Definition of System Impl'tion

1. Define System Life Cycle
 - System Development
 - System Deployment/Operations
 - System Evolution (Transition, Migration)
2. Define Data Model Implementation

Architecture Proof-of-Concept Measurement Technique:

Prototype Experiment

Simulation

Architecture Review and Analysis

Architecture Analysis

- Checklist-Based Technique
- Questionnaire-Based Technique
- Scenario-Based Method (SAAM)

Architecture Review Reporting

1. Architecture Presentation/Description
2. Issues, Dependencies, Constraints and Risk Factors
3. Architecture Opportunities

Component-Based System Composition / Development

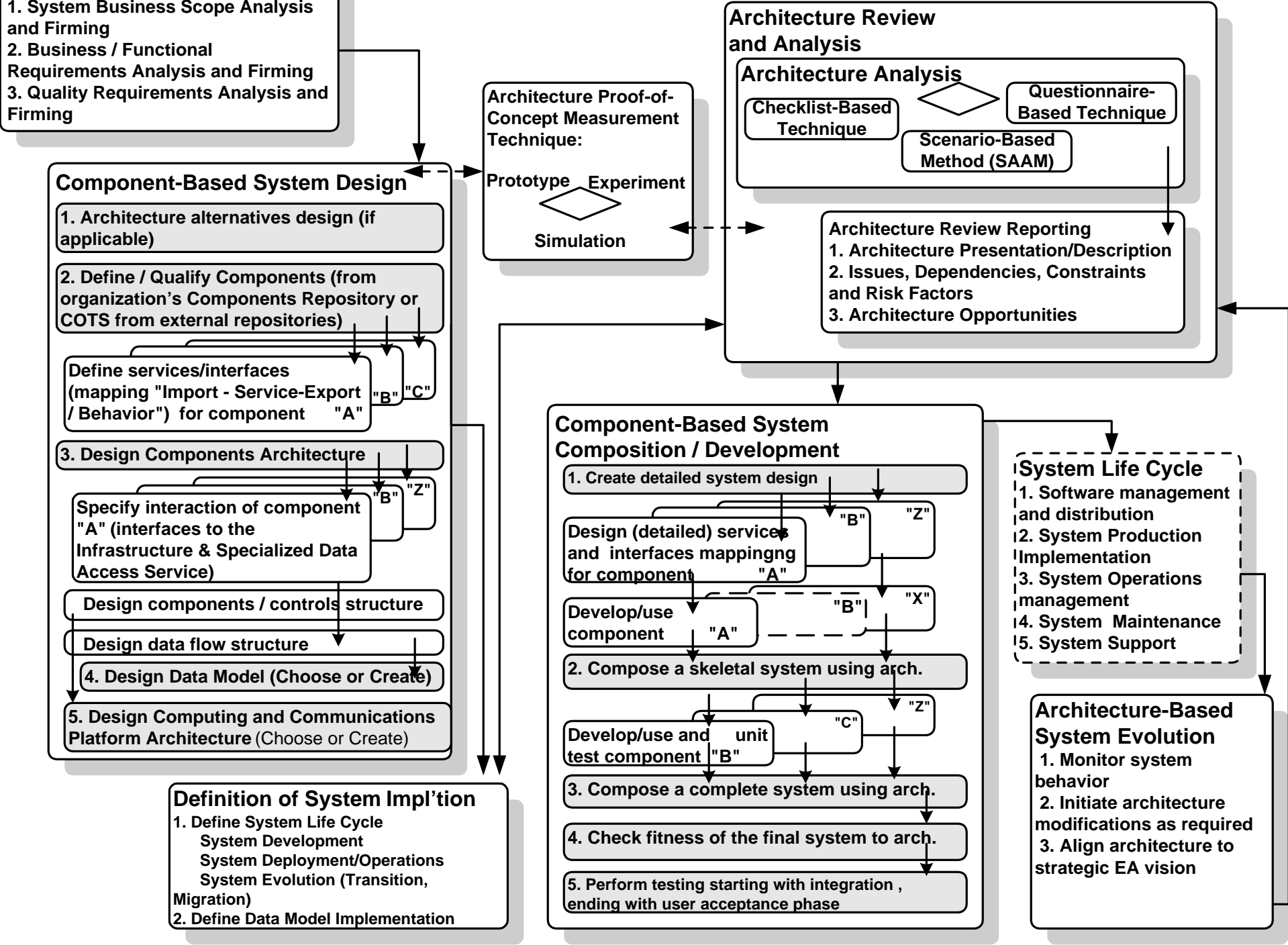
1. Create detailed system design
 - Design (detailed) services and interfaces mapping for component "A" "B" "Z"
 - Develop/use component "A" "B" "X"
2. Compose a skeletal system using arch.
 - Develop/use and unit test component "B" "C" "Z"
3. Compose a complete system using arch.
4. Check fitness of the final system to arch.
5. Perform testing starting with integration, ending with user acceptance phase

System Life Cycle

1. Software management and distribution
2. System Production Implementation
3. System Operations management
4. System Maintenance
5. System Support

Architecture-Based System Evolution

1. Monitor system behavior
2. Initiate architecture modifications as required
3. Align architecture to strategic EA vision

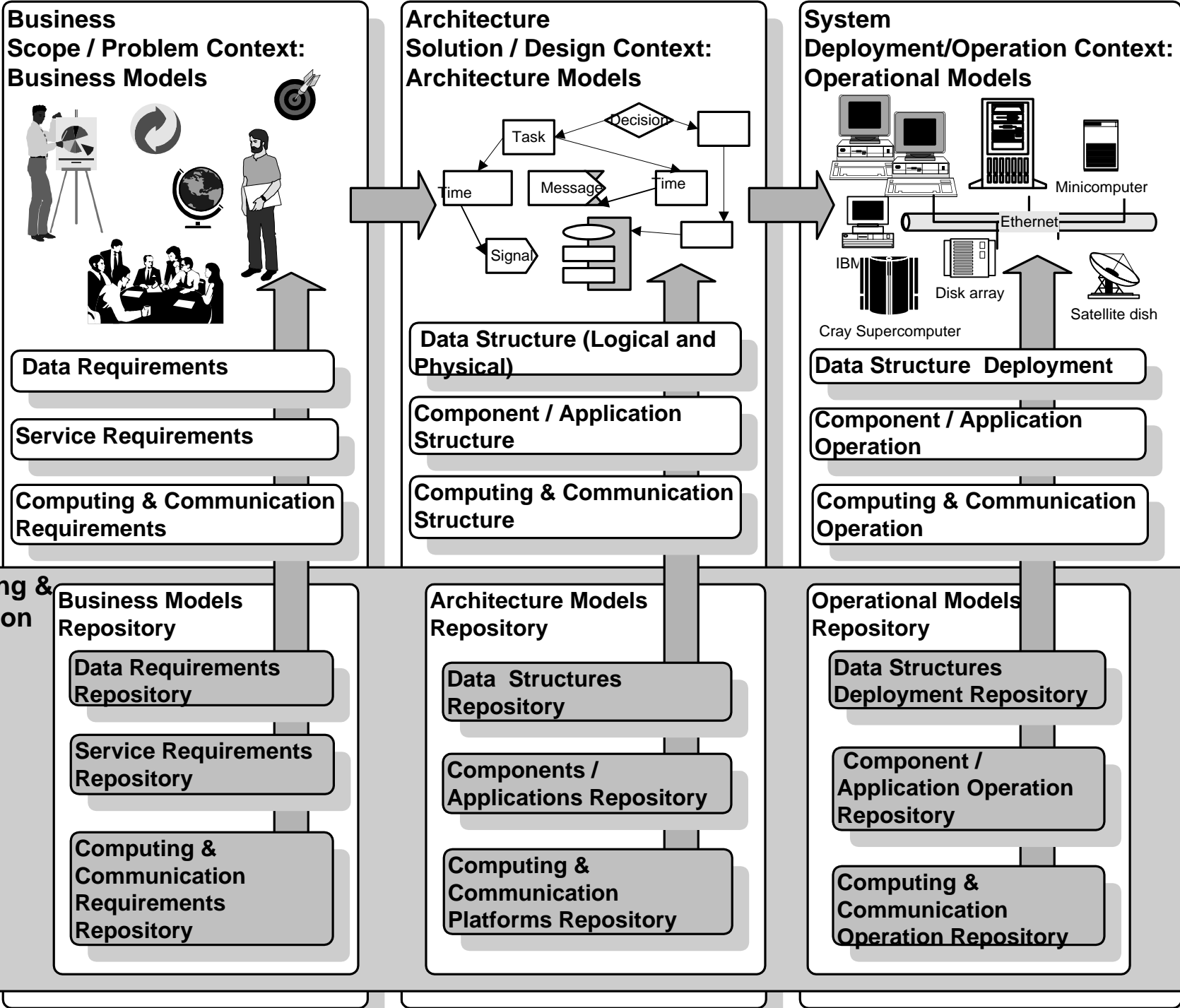


4. Managing the "Building Blocks"

Position Statement 3. An organization should implement Computing / Information Models life cycle approach as supporting process to CBSE (Figure 3). The process is data-centered i.e. its main objective is managing data about organization's information assets (also enabling links to information about COTS). The underlying database is comprised of repositories containing "building blocks" for requirements engineering (Business Models Repository), component-based designing and developing (Architecture Models Repository), and operating and managing system that performs computing and/or information processing (Operational Models Repository). This is achieved by providing management of information about components of corresponding nature available for reuse in all relevant contexts:

1. Business Models Repositories. Data, Services and Computing and Communication Requirements Repositories, provide necessary support for those processes completion in efficient manner. The principle is: "Specification of problem defined once, reused whenever alike problem identified."
2. Architecture Models Repositories. Data Structures, Components / Applications and Computing and Communications Platforms Repositories are used to retrieve, analyze and qualify components for reuse in particular system solution specification. The principle is: "Solution for the problem developed once, reused for alike problem."
3. Operational Models Repositories. Data Structures Deployment, Component / Application Operation, and Computing & Communication Operation Repositories, provide necessary information about various components configurations and setups. The principle is: " Configured and setup onetime, operated and managed anytime."

Figure 3. Information Models Base (*figure appears on next page*)



5. Components-Based Architecture

Position Statement 4. An organization must develop target component-based architecture of computing / information system (Figure4). The Component-Based architecture is natural in designing computing / information services for an organization. This is due to the fact that Component-Based architecture is aimed at resolving problems typical for legacy (current) computing / information systems such as: 1.) Specialization of component services and interfaces standardization not implemented, 2.) Separation of component specification from component implementation not present, and 3.) Component interactions implemented through non-standard interfaces and in a point-to-point mode (creating a look of components integration "spaghetti").

Figure 4. Component-Based Architecture of an Organization (*figure appears on next page*)

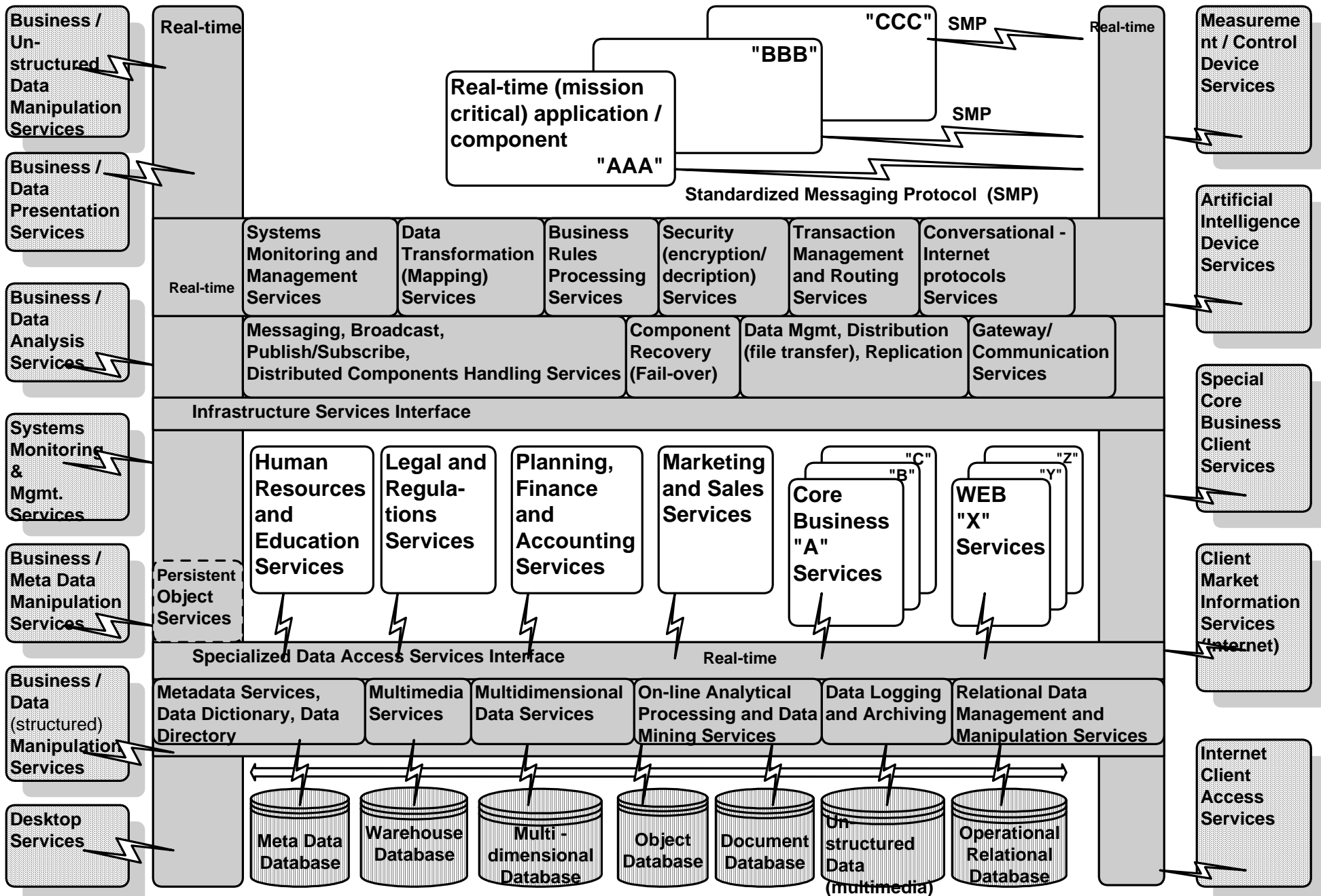
The solutions can be achieved by: 1.) Strict specialization of component services and its interfaces standardization allowing clustering of components by their nature and services providing for clear separation of components concerns and supports clear definitions of component interactions within the architecture through the standard interfaces; 2.) Separation of component specification from component implementation for all newly integrated components allowing component behavior to be described independently of its implementation and supports the possibility of multiple alternate component implementations for the same specification., and 3.) Building a common "plumbing infrastructure" i.e. Interface Infrastructure for integration of components providing for components integration by the common interface enabling the integration at the higher level than DCOM [DCOM], CORBA [OMG] or RMI [RMI].

The Components (Building Blocks) for a typical organization (Figure 4) fall into the following four categories determined by their service specialization: 1.) Business Domain Service components (business applications); 2.) Client Business & Presentation Service components (human / device and computing systems interactions services); 3.) Data Stores (data structures and data access), and 4.) Infrastructure & Specialized Data Access Services components.

First three categories of components will connect into the enterprise architecture through the common **Infrastructure Interface** for integration called Infrastructure Services & Specialized Data Access Services which we consider critical for the success of CBSE. This infrastructure is providing for components interactions at all layers from the basic network to the application through the high level, business-oriented, user-friendly interface. This is accomplished through the following services:

1. Messages, Broadcast, Publish/Subscribe, Distributed Components handling, and Conversational - Internet protocols handling.
2. Business Rules and Events processing. Handle logic of sharing information between multiple components / applications. Support definition of rules and events, and manages associated actions.
3. Data Streams and Messages Transformation and Routing services. Enables data streams and messages mapping, formatting and routing in dynamic and adaptable manner by supporting configuration of transformation and routing parameters based on the format and content of the data stream or message. E.g. a message from a specific component in one format can be routed to another and arrive in the form and context that is understandable and actionable.
4. Components and Transactions services including: Transaction management and monitoring, Security (encryption/decryption), Component fault detection, error handling and recovery, Communications handling.

Specialized Data Access Services is a structure comprising of components providing for the following: Meta-data services, Data Dictionary, Data Directory and Component Repository services, Relational data management, Data Logging & Archiving, Data management, data distribution and data replication, Multimedia management, Multidimensional data handling, On-line analytical processing and data mining, and Persistent objects handling.



6. Summary and Conclusions

This paper has explored some of the questions related to CBSE in the context of its adoption and implementation in a typical business organization. The ideas expressed have originated from our belief that success of the CBSE approach in an organization is directly proportional to fulfilling the following prerequisites:

1. Position the CBSE within the system's process life cycle which is based on relevant context areas covering business processes, architecture processes, operational computing / information system processes and supporting systems processes.
2. Define framework for the CBSE process which is architecture-centered, compromises between insisting on rigorous procedures and using manageable, user-friendly procedures, is based on concurrent execution of analysis, design and development activities and has standardized framework for process support and enabling (computing / information models base, toolkit, etc.).
3. Adopt the Computing / Information Models life cycle approach using Computing / Information Models Base repositories aimed at supporting Business, Architecture and System Life Cycle processes in delivering their products and/or services.
4. Develop target component-based architecture of computing / information services based on component services specialization and interfaces standardization, applying separation of component's architecture and component's implementation, and implementing common interface (plumbing) infrastructure.

References

[BCK] Bass L., Clements P., Kazman R., Software Architecture in Practice, Addison Wesley, 1998

[DCOM] <http://www.microsoft.com/oledev>

[OMG] <http://www.omg.org>

[RMI] <http://java.sun.com>

