

Implementing EA

May 24, 2005

Steve Olding
Project Director – Enterprise
Architecture & Reuse, Everware

**Creating the
Service-oriented
Enterprise**

Agenda

- EA requirements and issues
- Transformation needs
- Understanding services
- Getting from EA to SOA

Enterprise Architecture Status

- EA relates organizational mission, goals, and objectives to the IT infrastructure required to support them
- Defines artifacts that describe the current "as-is" state and desired "to-be" state of the organization
 - Organized according to some standard structure
 - Business models, data models, organizational models
 - Application portfolio
- Identifies IT capabilities needed to support the enterprise

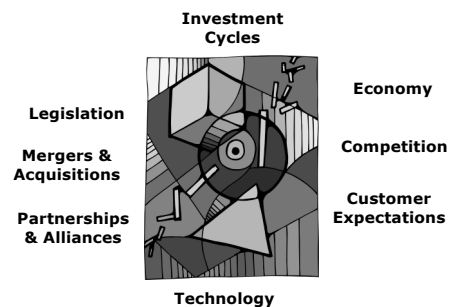
EA Implementation Requirements

- Modernize business processes to deliver services that meet the real needs of stakeholders
- Develop an agile organization that is responsive to changing business needs
- Leverage IT to optimize process efficiency (rather than being constrained by high costs and slow speed of IT development)
- Reduce costs and reduce process inconsistencies by eliminating duplication of effort (reduce redundancy)

Characteristics of Current Systems Architectures

- Many different stovepipe systems used to support the enterprise
- Slow to change in response to changing business needs
- Poor collaboration and information sharing
- Complex interdependencies between systems
- Multiple technologies and technical standards
- Independent system development, repair, revision and enhancement life cycles
- Expensive to change and maintain

Business Change

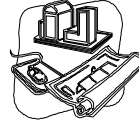


Dealing With Change

"Change is inevitable. In a progressive country change is constant."

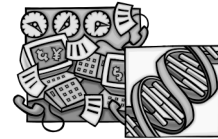
Benjamin Disraeli, former British Prime Minister - Edinburgh (1867)

Engineering Challenges



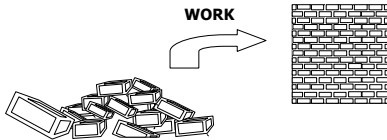
- Civil Engineering
 - High complexity
 - Standardized 'building blocks'
 - Some customization
 - Limited interdependencies
 - Limited changes after initial construction
 - Long time horizon

- Software Engineering
 - High complexity
 - Heterogeneous
 - Highly customized
 - Strong interdependencies
 - Frequent change and reconfiguration
 - Short time horizon



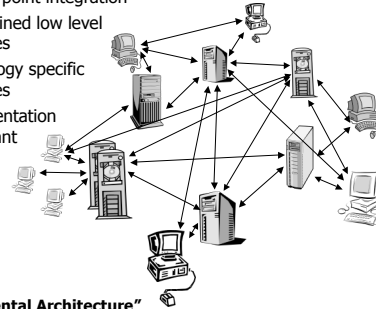
Entropy, Disorder and The Second Law of Thermodynamics

- The concept of entropy and the second law of thermodynamics suggests that systems naturally progress from order to disorder.
- Any system which is free of external influences becomes more disordered with time. This disorder is expressed in terms something called entropy.
- Order can only be produced with an expenditure of energy.



Tightly Coupled Applications

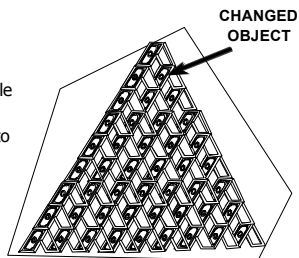
- Point-to-point integration
- Fine-grained low level interfaces
- Technology specific interfaces
- Implementation dependant



The "Accidental Architecture"

Tightly Coupled Applications

- Tightly bound relationships
- Complex interdependencies
- Difficult or impossible to change
- Costly and difficult to operate
- Inherently fragile
- Constrains business change



What we need?

- A system architecture that translates the IT requirements from the EA into the specific system structures (including hardware, software, and technology standards) required to implement them.
- An implementation plan that articulates how we will transform the current IT infrastructure into the desired state.

Issue: many EAs designed to meet compliance requirements. Incomplete - describes where we are and where we want to go, not how to get there. Not a plan for transformation

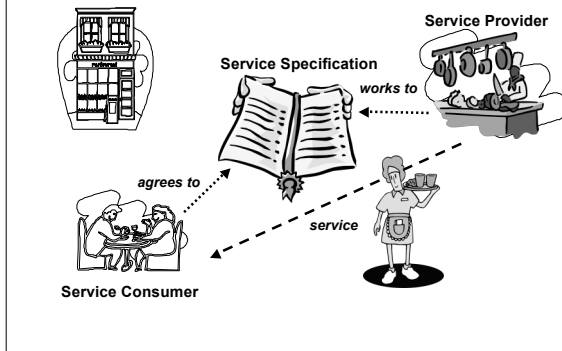
IT Transformation Needs

- IT capabilities better aligned to business needs
- Systems that enable collaboration and information-sharing
- More flexible systems - agile systems that are adaptable to changing business needs
- Systems infrastructure that leverages investment in existing assets and allows legacy applications to collaborate with newer systems
- Technical infrastructure that is adaptable to new hardware and software technology
- Improved ROI (cost savings)

Loose Coupling

- Avoid point-to-point integration... reduce and simplify dependencies
- Reuse rather than replicate
- Coarse-grained interfaces offer generalized business-level services
- Separate software implementation from external representation
- Common services available to many consumers through common infrastructure
- Location independence
- Synchronous and asynchronous processes

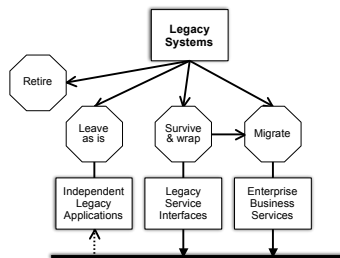
Services



Key Service Characteristics

- Service consumers delegate responsibility for performing some function to service providers
- Services are generalized and not specific to a particular consumer
- Service consumers insulated from the underlying implementation of the services
- Services are made available to service consumers through universally available interfaces

Transforming Legacy Systems

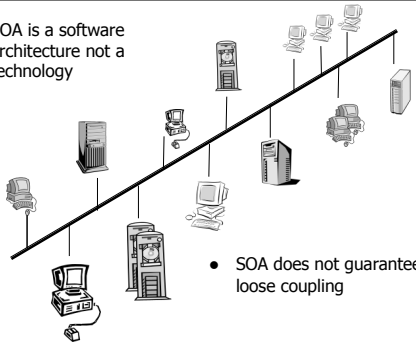


Getting to SOA from EA

- Identify notional business components and component architecture (if not already done in EA)
- Identify common service components
- Group components into logical 'chunks' based on affinity analysis
- Complete legacy portfolio analysis
- Map legacy application functionality to notional component architecture
- Develop legacy transition plan
- Prioritize new development

Service Oriented Architecture

- SOA is a software architecture not a technology



- SOA does not guarantee loose coupling

Why Develop With Services ?

- Containment of complexity
- Parallel development
- Incremental Testing
- Sourcing flexibility
- Unique copy versus cloning
- Greater agility
- Easier information sharing
- Ability to leverage existing assets
- Reduced cost (and risk)

Summary

- Traditional architectures do not provide the agility needed by modern organizations
 - Accept that change is inevitable and prepare for it
- Loose coupling of business processes and IT systems are key to business agility
 - Think 'services' and service-oriented architectures
 - SOA is a software architecture not a technology