

ICTEC

Service-Oriented Computing

31.3.2003



Introduction

- Service-Orientation is a hot, but often misunderstood or even misused term and even a “buzzword” in IT today.
- There is a lot of interest, because it promises to transform IT within organizations, making the business more adaptable to change, reducing costs, and increasing efficiency
- ... but there exists many different, overlapping and conflicting views and definitions for Service, Service-orientation and Service-Oriented Architecture
- ... so what it is really about ?



Business Environment

- The IT industry has matured
 - Modest growth (GNP + 1 – 2 %), balance of supply / demand, commoditization of products and services, ...
- The role of IT has changed from support to enabler (or inhibitor)
 - IT as a transformation agent; business change → IT change
- Business cycles are shorter: year → month → week
- Cost efficiency / productivity; ROI, TCO, BVIT, ...
- Change in inevitable
 - The number and complexity of systems grows
 - Emerging technologies open new opportunities and challenges

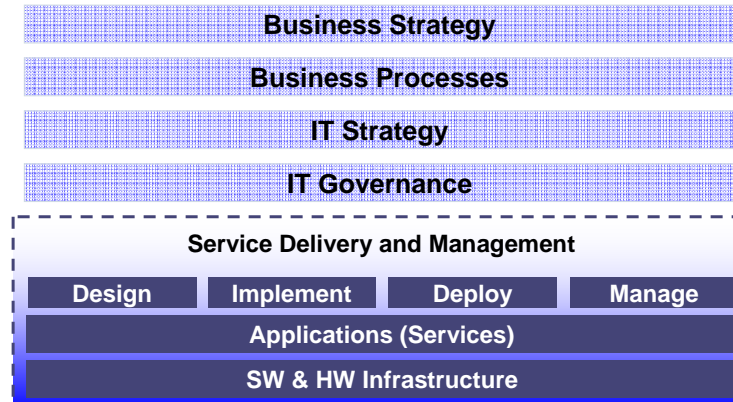


Enterprise IT

- ***“Enterprise”***
 - In this context is any collection of organizations that has a common set of goals/principles and/or single bottom line.
 - In that sense, an enterprise can be a whole corporation, a division of a corporation, a government organization, a single department, or a network of geographically distant organizations linked together by common objectives.



Enterprise IT



Enterprise IT

- Operative IT
 - Support business functions, processes and organizational work
 - Goal is to make processes easier, faster, more automated, ...
 - Benefits are coupled with the process improvements; cost-efficiency, quality, speed, changeability / process flexibility ...
- Strategic IT
 - When margins are tight – revenue management matters more
 - Improving productivity (in the business) is essential but not enough
 - Helping business to grow is the key IT contribution in the future
- Role, value and benefits ?
 - IT is just a cost center → cut costs, maximize value, outsource, ...



Enterprise IT

- Enterprise IT is tightly coupled with the internal organization, processes and business model of the enterprise.
- When looking at enterprise IT, we are not looking at isolated systems, but a large number of systems with cross-dependencies, high level of heterogeneity, redundancies and a long history.
- Today's enterprises depend heavily on IT
 - Responsible of running almost all processes in an modern enterprise; manufacturing, sales, accounting, day-to-day office work, ...
 - ERP, CRM, SCM, HRM, PDM, BI, DM/CM, GIS, ...
 - Automation of manual labor / processes → enabler of new business



Enterprise IT

- Underlies both cross-departmental dependencies and external business relationships
- Must deal with a large number of different requirements
 - Conflicting, unclear, constantly changing, ...
 - Internal & external conflicts and politics
- Rarely contain a large number of complicated algorithms or data structures
 - A piece of business logic is usually very simple to implement
- Long history and very different lifecycles
 - Business data and content ~10-20 years – Front ends ~1 year



Enterprise IT

- Business processes are constantly changing.
 - Enterprises must constantly sense changes in the market environment and adapt their strategies.
 - IT must accommodate these changes quickly and efficiently.
 - Move from "production line" to "marketplace economy";
 - Networking, Outsource non-core competences, ...
 - Enterprise IT dilemma:
 - Global competition and faster cycles; cost cutting and productivity
 - Enterprise software development almost always suffers from the lack of agility and from inefficiency
 - Not able to match business requirements onto IT fast enough
- A new approach is needed – Services !



Agility ?

- Business reality
 - Constant change; structures, tech's, people, regulation, ...
 - Survive and keep profitability during the change
 - Differentiate and seize competitive advantage
 - "Corporate Agility"
- Agility ?
 - The ability to continually sense and explore customer and Marketplace enrichment opportunities and respond with the appropriate configurations of capabilities and capacities to exploit these opportunities with speed, surprise, and competitive success.



Services and Service Concepts

■ A Service ?

- *"useful labor that does not provide a tangible commodity"*

"a facility supplying some public demand"

(Webster's Dictionary)

1. A unit of work done by a service provider to achieve desired end results for a service consumer
2. A meaningful activity that a computer program performs on request of another computer program
3. An abstraction of a business resource or application into a unit of repeatable functionality with standardized (and publishable) interface



Services and Service Concepts

■ A Service: one academic definition

- Services are self-describing, open components that support rapid, low-cost composition of distributed applications.
- Services are offered by service providers—organizations that procure the service implementations, supply their service descriptions, and provide related technical and business support.
- Since services may be offered by different enterprises and communicate over the Internet, they provide a distributed computing infrastructure for both intra- and cross-enterprise application integration and collaboration.

(Papazoglou & al. 2003)



Services and Service Concepts

- A service view of business resources and applications.
- A service is simply an abstraction of a business resource or application into unit of repeatable functionality with standardized and publishable interface.
- An architecture starts with an interface definition and builds the entire application topology as a topology of interfaces, interface implementations and interface calls.
- Services can be distributed and interwoven to support existing and new business processes by changing the sequence of execution.



Services and Service Concepts

- In Business Context
 - a **set of services** that a business wants to expose to their customers and partners, or other portions of the organization.
- In Architectural Context
 - an **architectural style** which requires a service provider, requestor and a service description.
 - a **set of architectural principles**, patterns and criteria which address characteristics such as modularity, encapsulation, loose coupling, separation of concerns, reuse, composability and single implementation.
- In Technology / Application Development Context
 - a **programming model** complete with standards, tools and technologies.



Principles of Service-Orientation

- Services are reusable
 - Designed to support potential reuse (in unknown context)
- Services share a formal contract
 - Describes service and defines the terms of information exchange
- Services are loosely coupled
 - Interact without the need for tight, cross-service dependencies
- Services abstract underlying logic
 - "Black-box" design, only the interface (contract) is visible



Principles of Service-Orientation

- Services are composable
 - Services can participate in composing other services
- Services are autonomous
 - Executes self-governance
- Services are (mostly) stateless
 - The amount of state information is minimized (to promote reuse)
- Services are discoverable
 - Well defined, easy-to-use, somewhat standardized interface, designed to be discoverable to promote reuse.



Service-Oriented Architecture

- Service-oriented architecture (SOA) is not a new idea
 - One of the first descriptions of "modern" SOA by Gartner in 1996
 - A lot of the recent interest in the architecture has been fuelled by the industry trend toward utilizing XML-based Web services.
- It has become an important technology solution because it promises the agility and flexibility enterprises have been looking for.
- In this context, a service is simply an abstraction of a business resource or application into unit of repeatable functionality with standardized and publishable interface.
 - A service view of business resources and applications.

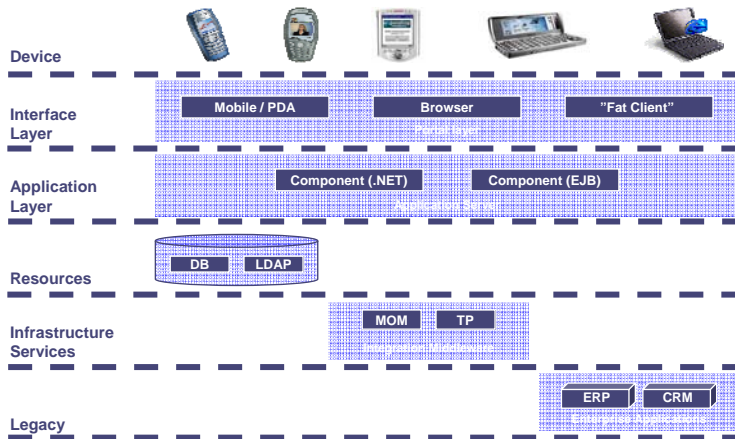


Service-Oriented Architecture

- "A set of business, process, organizational, governance and technical methods (...) to quantifiably measure business value of IT while creating agile business environment for competitive advantage."
- "An enterprise wide IT-architecture that promotes loose coupling, reuse and interoperability between systems."
- "An approach to loosely coupled, protocol independent, standards-based distributed computing where software resources available on the network are considered as Services."
- "An application architecture in which all functions or services are defined through a description language and have interfaces that are called to perform business processes."
- "Synonym for solution architectures making use of Web Services technologies such as SOAP, WSDL or UDDI. <...> Any product and project architecture conforming to W3C Web Services architecture."

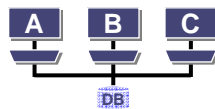
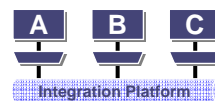
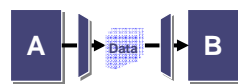


Traditional N-tier architecture

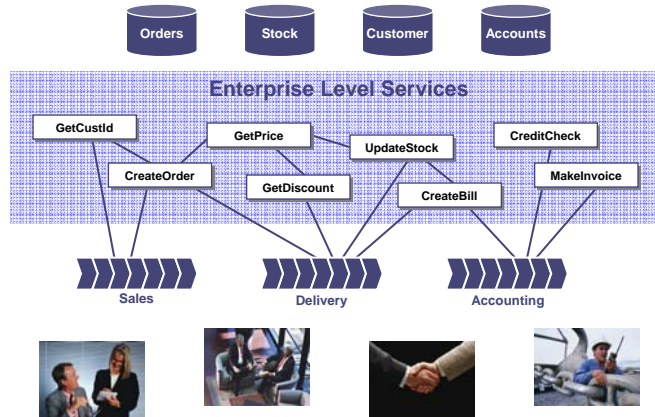


Integration Architecture

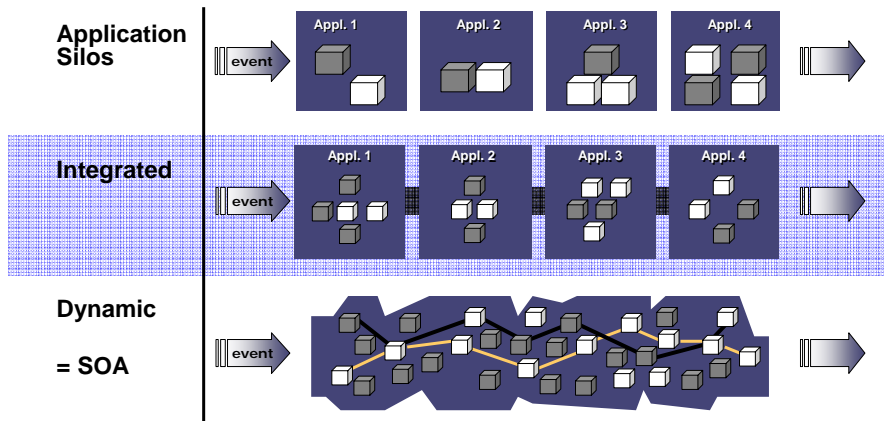
- Many different approaches & models
 - Data (e.g. file transfers, databases)
 - Messaging (e.g. MOM)
 - Process-integration
 - Point-to-point, Hub-Spoke, ESB, ...



Service-Oriented Architecture



Service-Oriented Architecture



Service-Oriented Computing

- Service-Oriented computing (SOC) is the computing **paradigm** that utilizes services as fundamental elements for developing applications.
- SOC involves the service layers, functionality, and roles as described by the Service-Oriented Architecture (SOA).
- Basic services, their descriptions, and basic operations (publication, discovery, selection, and binding) that produce or utilize services constitute the foundation of SOC.



Service-Oriented Computing

- Is an alternative model to the more traditionally tightly-coupled object-oriented models
 - Common Object Request Broker Architecture (CORBA)
 - Distributed Component Object Model (DCOM)
- Individual services can be build with object- or component-oriented style or component designs but the overall design of is service-oriented
 - The key concept is using services that can communicate through standard protocols (e.g. SOAP) allowing a more loosely coupled architecture.
 - Services can be distributed and interwoven to support existing and new business processes by changing the sequence of execution.



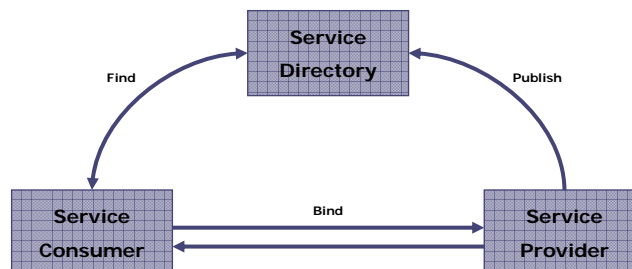
Service-Oriented Computing

- The idea of service can be seen as a natural progression in the evolution of application development.
 - From the earliest procedural code, to structured code, then the concept of objects in parallel with the trend away from mainframes to distributed client-server systems, 3-tier and then n-tier application architectures, and so into the modern era of distributed objects and components.
 - Each advance builds on the previous ones; object-oriented programming uses the procedural and structured coding style within encapsulated methods. Application components combine distributed objects, and now application components are being exposed as services.

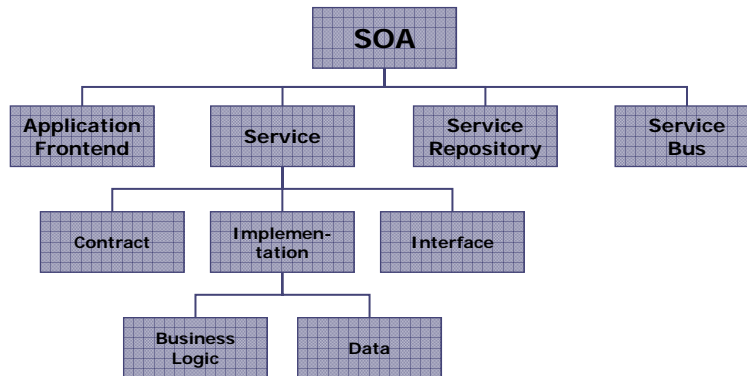


Service-Oriented Architecture

- Simplest definition: consumer, provider and broker.
= "basic" SOA / WebServices-model



Modern SOA



Approaches to SOA / SOC

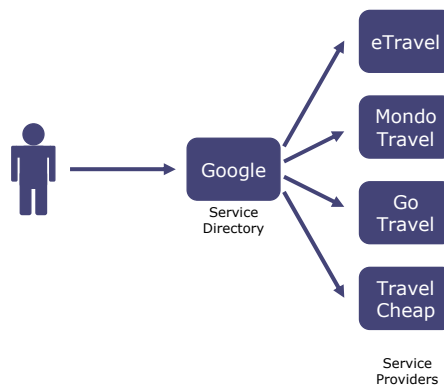
- "Inside Enterprise"
 - An approach to the design, construction, deployment, integration and management of IT solutions through Service-Oriented principles, policies, standards, and models that reflect business strategies, requirements and constraints.
- "Outside Enterprise"
 - An enabler for building Dynamic Business Networks, distinct systems of participants (customers, suppliers, competitors, complimentors, service providers) that provide extended services to customers and where participants and relationships dynamically evolve over time.
 - Loosely coupled business networks change interrelationships between nodes both quickly and frequently, providing little or no notice for planning, implementing, or changing the supporting applications – thus requiring high level of IT agility and flexibility.



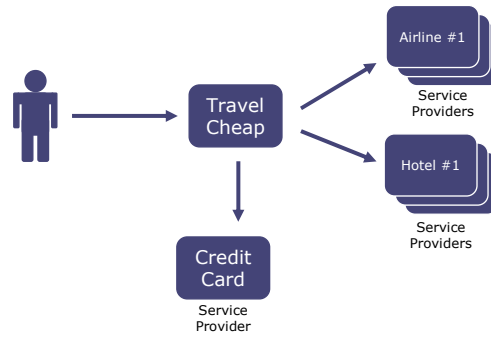
Service-Oriented Business Models



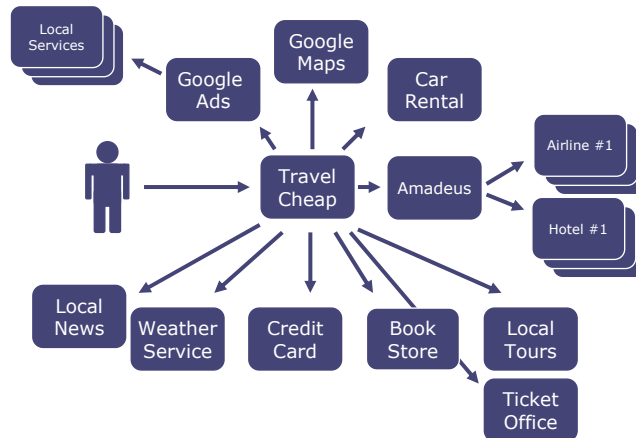
Service-Oriented Business Models



Service-Oriented Business Models



Service-Oriented Business Models



About the Example

- Developers are utilizing various Web Services from a range of service providers to deliver users their very own "experience" (→ *Web 2.0*).
 - A software mashup is comprised of different components or content from different services but presented to the user as a single, unique experience (service).
 - Mashups represent a new development approach to building composite applications using a set of Web 2.0 technologies.
- This new way of creating composite applications is a part of Service-Oriented Computing phenomenon where different service providers share services with each other in new, inventive ways.



Other Related Terms

- IT Services ≠ SOA
- SaaS – Software as a Service
 - SaaS is a model of software delivery where the a company provides maintenance, daily technical operation, and support for the software application provided over network.
 - There are some academic papers which use term 'SaaS' when talking about SOC / SOA, but for the mainstream SaaS is not SOA !
 - SaaS is an evolution of ASP model (Application Service Provider)
 - Also: on-demand software / applications
- Services on-demand
 - Utility computing: processing, storage, ...



The ideal SOA

- Establishes a universal model in which automation and business logic combine to a layer of abstraction that introduces commonly accepted standard for business and process level integration and interoperability
- Coupled with enabling technologies and a set of common principles, service-orientation is emerging as a new computing paradigm and architecture distinct from its predecessors
- Can provide a significant benefits to many traditional challenges of changing business-IT environments



The false SOA

- A product or something else you can buy or a new version of vendor X's product supporting WS
- A technical architecture, characterized by Web Services and their related technologies
- An extension of Object or Component-oriented software reached by wrapping them with XML/WS access layer
- Distributed application computing using XML/WS
- Solely an integration or BPM approach
- A revolutionary transformation of IT



Today's SOA

- Represents architecture that promotes service-oriented principles at all levels, from infrastructure to business models and processes
- Autonomous in services and information flows
- Based on open standards and supports vendor diversity
- Promotes discovery, federation and interoperability
- Promotes inherent reusability and composability
- Promotes loose coupling
- Supports service-oriented business modeling



Today's SOA

- Open, agile, extensible, federated, composable architecture consisting of QoS capable, vendor diverse, interoperable, discoverable and reusable services
- Establishes abstraction of business and application logic resulting loose coupling between these domains
- Evolution of past, building on successful characteristics of previous architectures bringing a distinct set of service-oriented principles to all levels of architecture
- Ideally standardized across an enterprise to fully realize potential benefits



Summary

- In today's business environment, agility and ability to change are critical for any organization
 - Proactive development of business processes and services
- The abstraction level of IT is rising
 - From HW and SW to business processes and services
 - Composition and orchestration of services instead of "coding"
- "Internal" Service-orientation provides enterprises a good foundation for creating more agile and flexible IT systems.
- "External" Service-orientation provides new means for building new innovative services and value networks.



Questions ?

