



Web-Based Information Systems for the Agile Economy: Enablers and Challenges

Prof. dr. Herwig Mannaert



Contents

- The Agile Economy
- Enablers for Information Systems
 - The Classics
 - Service Modules
 - Commoditization
- Challenges for Information Systems
 - Methodology Jungle
 - An Inconvenient Truth



Contents

- The Agile Economy
- Enablers for Information Systems
 - The Classics
 - Service Modules
 - Commoditization
- Challenges for Information Systems
 - Methodology Jungle
 - An Inconvenient Truth



The Agile Company

- The Agile Company
 - Continually scans its ecosystem
 - Stakeholders
 - Technologies
 - Markets
 - ...
 - Reacts quickly to opportunities and is innovative
 - Customer-led production



The Agile Company

- Agile Companies are flexible, for which ICT is a decisive factor
 - Services Sector
 - “Door modularisering van dienstenproducten, standaardisatie van inputs en de stroomlijning van het voortbrengingsproces valt veel productiviteitswinst te behalen. De combinatie van ICT met niet-technologische innovaties vormt hiervoor de sleutel.” (Van Ark, 2004)
 - “Information about the package is just as important as the package itself.” (Fred Smith, FedEx)
 - Industrial Sector
 - Ubiquitous Computing will increase the amount of ICT in industrial products.
 - ICT captures knowledge about the usage of products by customers, and the efficiency of the production process.
 - “IT is not just transformational. It is also a central determinant of the successful business models and industry structure of a growing fraction of the economy.” (Dhar et al., 2007)



The Agile Environment

- COMPLEXITY
 - At Business Level
 - Distribution
 - Multi-channel vs. single channel
 - Diversify offerings/Additional services
 - At Technical Level
 - Compare JEE/.NET to COBOL
 - ...



The Agile Environment

- And if the growing complexity is finally under control, there is always... CHANGE !
 - At Business Level:
 - “These things are changing so fast it’s invention in the hands of the owner.” (Hansen et al., 2007)
 - At Technical Level:
 - Structured Development – 70’s
 - Object Oriented Development – 80’s
 - Component-Based Development – 1995-
 - Service-Oriented Development – 2000-
 - The Next Hype...



The Agile Environment

- In this complex, quickly changing environment, Information Systems need to be:
 - Very flexible
 - Reliable (even mission-critical)
 - Totally secure
 - User friendly
 - Portable
 - Preferably affordable !
 - ...

Let's ask the *IT-department* !



Contents

- The Agile Economy
- Enablers for Information Systems
 - The Classics
 - Service Modules
 - Commoditization
- Challenges for Information Systems
 - Methodology Jungle
 - An Inconvenient Truth



Technology Standards

- Open standards
 - SOAP, WSDL, UDDI
 - BPMN, WS-BPEL, WS-Security, WS-Messaging
 - RDF, OWL, CL
 - SOA, AOP, MDA
 - LMS, SCORM
 - NGN, Web 3.0
- New products
 - ESB, ESOA
 - BPM, ARIS



Agile Methodologies

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

© 2001, the above authors this declaration may be freely copied in any form, but only in its entirety through this notice.



Enterprise Architecture

- The Architecture *is* the alignment !
 - *By definition*, the architecture is the high-level design for current and future requirements of the end users in the agile company, based on domain-specific and technical standards.
 - The architecture is based on business processes which are becoming increasingly modular and inter-organizational
 - The evolvability of the architecture determines to a large extent the evolvability of the organisation.



Contents

- The Agile Economy
- Enablers for Information Systems
 - The Classics
 - Service Modules
 - Commoditization
- Challenges for Information Systems
 - Methodology Jungle
 - An Inconvenient Truth



Modules – advantages

Complexity Reduction

Reuse

Evolvability



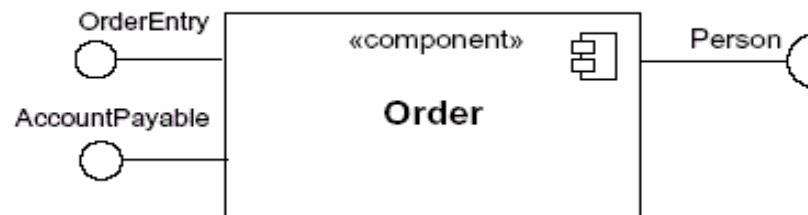
Constructs

- Modules are implemented in constructs, which are becoming increasingly powerful
 - Structures/Procedures,
 - Objects,
 - Components...
 - Services !
- *We are making progress !*



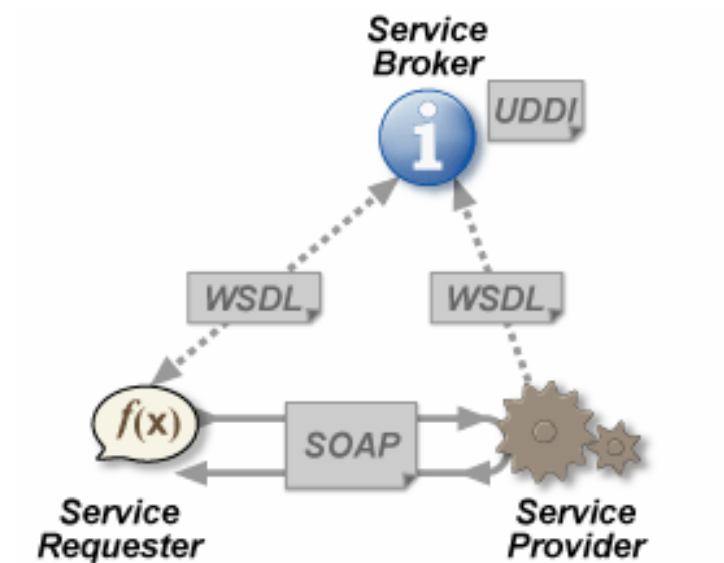
Component-Based Development

- Components
 - Have larger granularity than objects
 - Are independent of underlying technologies
 - For communication
 - For persistence
 - Less 'infrastructure'-code is needed
 - Typically .NET or JEE





Web Services



Bron: Wikipedia, 1 oktober
2007



Services

- Services were initially linked to 'web services', but many authors now describe them as technology-independent modules, implementable in components, OO, web services and so on. Services are the most recent, and **most powerful form of modules**.
- Characteristics
 - They exhibit loose coupling
 - They are 'network-available'
 - They can be easily assembled, also at runtime
 - They are autonomous
 - ***They have a business meaning***



What is a Service *exactly* ?

- Technical definition
 - Services are **autonomous, platform-independent** computational entities that can be used in a platform independent way. Services can be **described, published, discovered**, and **dynamically assembled** for developing massively distributed, interoperable, evolvable systems. Services reflect a "service-oriented" approach to programming, based on the idea of **composing applications** by discovering and invoking **network-available** services rather than building new applications or by invoking available applications to accomplish some task. Services are most often built in a way that is independent of the context in which they are used. This means that the service provider and the consumers are **loosely coupled**. (Papazoglou, 2006)

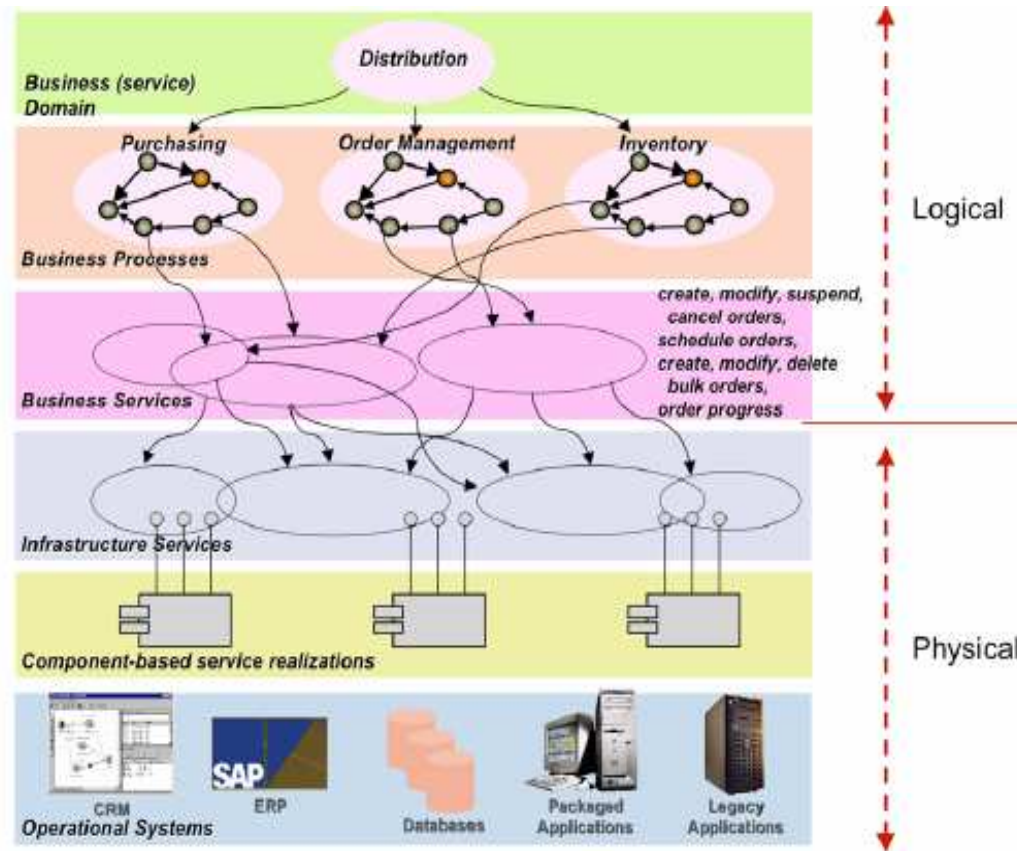


What is a Service *exactly* ?

- Business definition
 - “When we use the term service, we have in mind a **business service** such as making airline reservations or getting access to a company’s customer database. Actually, the SOA must **decouple** business applications from technical services and make the enterprise independent of a specific technical implementation or infrastructure.” (Krafzig et al., 2004)



Service-Oriented Architectures



Bron: Papazoglou et al. , 2006



Contents

- The Agile Economy
- Enablers for Information Systems
 - The Classics
 - Service Modules
 - Commoditization
- Challenges for Information Systems
 - Methodology Jungle
 - An Inconvenient Truth



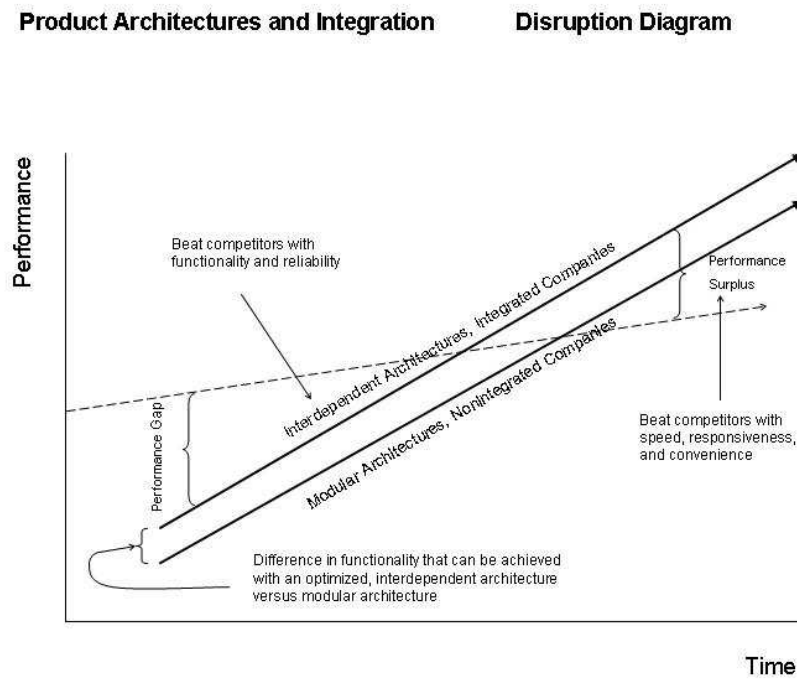
Commoditization of IT

- What is a “commodity” ?
 - a. Undifferentiated
 - a. “commodities are things of value, of *uniform quality*, that are produced in large quantities by many different producers” (Wikipedia, 3 september 2006)
 - b. The fact that agreement or standards exist which define quality, defines the “commodity” (Wikipedia, 3 september 2006)
 - b. Price
 - a. Uniform quality puts pressure on price. Differentiation does not provide profit margins on commodities.
 - b. But, a commodity is not for free (oil, electricity,...)



How does IT become a Commodity ?

- Christensen en Raynor (2003)





Is Hardware a Commodity ?

- Undifferentiated
- Price
- Standards/specs exist
 - hardware description languages
- Modular



Is Software a Commodity ?

- Undifferentiated ?
 - Standard packages
- Price
 - Open source software
 - Price of 1 function point decreases
 - Off-shore Outsourcing makes software cheaper
- Architecture
 - Middleware, JEE...
- Modularization
- Standards
 - Non-domain specific: TCP/IP...
 - Domain-specific: ontologies
- Trend towards end-user computing



Are Business Processes a Commodity ?

- There is a trend towards standardisation
 - ERP-packages
 - Business Process Standards
 - Ontologies



Business Process Standards

- Domain-specific
 - Supply Chains: SCOR (www.supply-chain.org)
 - Insurance: IAA (www.research.ibm.com/iaa/)
 - Automotive: STAR (www.starstandard.org)
 - Banking: SIZ Banking Data Model (www.innotivum.de)
 - Manufacturing: ISA-95 (www.isa-95.com)
 - Telecommunications: eTOM(enhanced Telecom Operations Map)
- Domain-independent
 - OAGI (www.openapplications.org)
 - APQC (www.apqc.org/pcf)



Business Process Standards

- Contain Domain Models of
 - Data, and
 - Processes
- Will they be used in the future for all processes, except core competences ?
- They are a new source of requirements collection; analysts become less dependent on end users and interviews.



Example - STAR

Standards for Technology in Automotive Retail | Main / HomePage browse - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.starstandard.org/>

Google Go Links

Backlinks - Print - Main Recent Changes - News Feed

STAR

Standards for Technology in Automotive Retail
8400 Westpark Dr. MS#16 McLean, VA 22102

STAR MISSION

The [goal](#) of the **Standards for Technology in Automotive Retail (STAR)** Organization is to use non-proprietary information technology (IT) standards as a catalyst in fulfilling the business information needs of dealers and manufacturers while reducing the time and effort required to support related activities.

STAR NEWS

STAR Web Services Quickstart Guide

STAR has published a [Quickstart Guide](#) to help organizations get started with STAR Web Services. This document was created by several member organizations, to help the user community implement the STAR Web Services specification.

STAR BOD Validation Site

STAR has constructed a web site where the community may upload a STAR BOD, and validate it against a STAR BOD Schemas. The community may use this site to verify that their implementations are compliant with the STAR specifications published on the web site. The BOD Validation site is open to everybody and can be found at the [STAR BOD Validation](#) link on the side bar. A user-id and password is required to use the site, and the email address will not be distributed or sold.

Files sent to the site are not kept, they are immediately deleted after validation.

Search Site

Quick Links

- HomePage
- Specifications and Guidelines
- Forms
- [STAR BOD Validation](#)

General Information

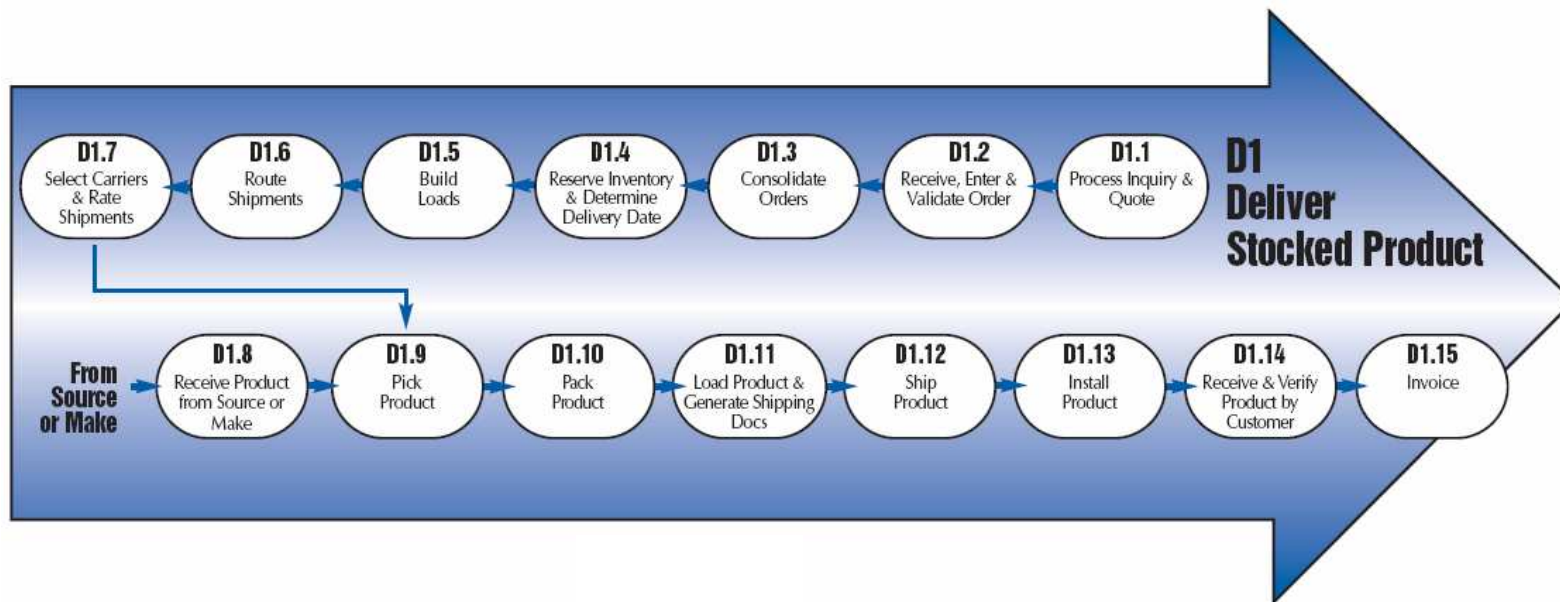
- About
- Membership
- Events
- Short Codes
- Industry Links
- 2007 STAR Implementation
- Survey Summary Report

Special Interest Groups

- XML
- Data Transfer Specifications
- Architecture
- Dealership Infrastructure
- Dealers



Example- SCOR



© Supply Chain Council



Ontologies

- Human Resources
 - HR-XML (www.hr-xml.org)
- Financials
 - IFX (www.ifxforum.org)
- E-business transacties
 - ebXML (www.ebxml.org)
- Financial reporting
 - XBRL (www.xbrl.org)



Contents

- The Agile Economy
- Enablers for Information Systems
 - The Classics
 - Service Modules
 - Commoditization
- Challenges for Information Systems
 - Methodology Jungle
 - An Inconvenient Truth



Scientific Literature

- Paradigms
 - Structured development
 - Object oriented development
 - Component-based development
 - Service-oriented development
- Methodologies
 - Processes
 - Techniques
 - Tools
- Principles
 - Information hiding
 - Coupling, cohesion
- Frameworks
 - Quality Management (CMMI...)
- ...



Which Methodologies ?

- More than 1000 exist...
 - BON, Booch, BOOM, Catalysis, CBD/e, Coad/Yourdon, COMMA
 - CRC, Convergent Engineering, Demeter, DOORS, DOOS
 - EPA, EROOS, Fusion, Goofee, HOOD, IDEA, ION, KISS
 - MERODE, MOSES, MWOOD, Object COMX, Objecteering
 - Objectory, OEP, Octopus, OMT, OOAD/OOIE, OOA/RD, OBE
 - OOCL, OOHDM, OOram, OOSC, OOSD, OOSE, OOSP
 - Open, OSA, PAUD, ROAD, ROPES, RUP, Scrum, Skill-Driven Design
 - SDL, Shlaer & Mellor, Softstar, SOMA, SOMT, Syntropy, XP
- Which one do we use ?
 - The "methodology jungle"...



Adoption of Methodologies

- Huisman & Iivari, 2002:
“Many organizations claim that they do not use any systems development methods.”
- Riemenschneider, 2002:
“Only about half of all organizations actually follow a methodology”
- Interpretation
 - Explicit use
 - Implicit use
 - Ad hoc



The Gap Business <-> ICT

- Seems to exist already 40 years
- A very fundamental problem ?
 - Knowledge gap
 - Cultural gap
 - Personality gap
 - ...
- Other gaps seem to exist:
 - Theory and practice
 - Architects and developers



Heavyweight Management

- Examples
 - Waterfall
 - Methodologies for systems development
 - Operational standards (ITIL)
 - Project management standards (PMBok, Prince2)
 - IT governance
 - Audit (Cobit standard)
- Contribution
 - High-quality execution leads to flexible organisations and information systems



Lightweight Management

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

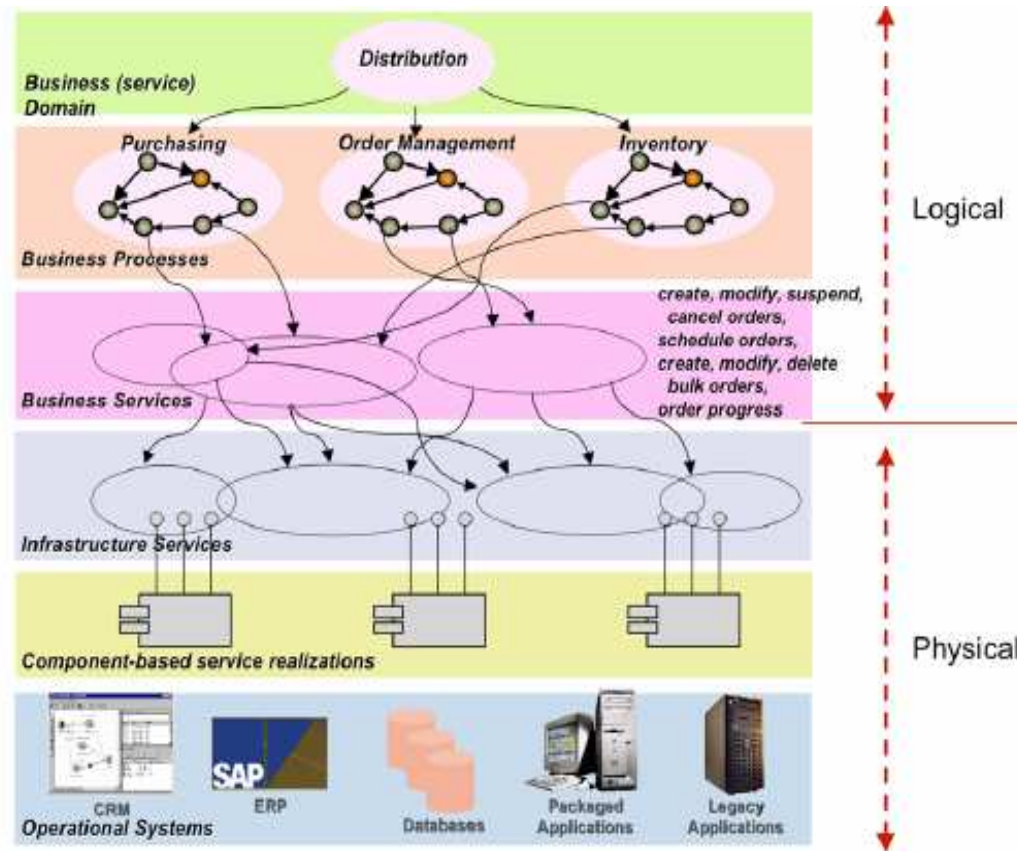
That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

© 2001, the above authors this declaration may be freely copied in any form, but only in its entirety through this notice.



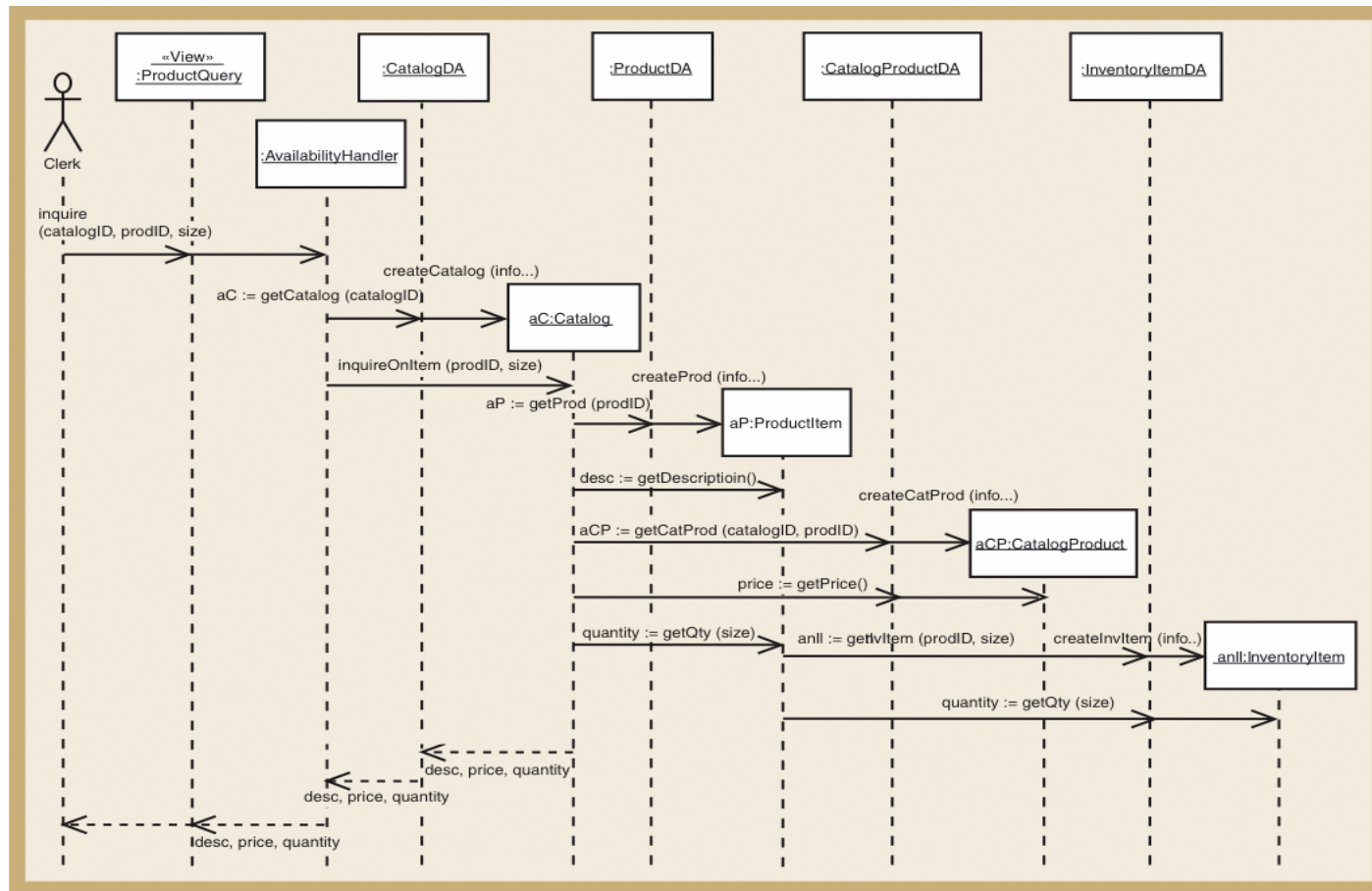
Service-Oriented Architectures



Bron: Papazoglou et al. , 2006



4-Tier OO Development





Contents

- The Agile Economy
- Enablers for Information Systems
 - The Classics
 - Service Modules
 - Commoditization
- Challenges for Information Systems
 - Methodology Jungle
 - An Inconvenient Truth



Doug Mc Ilroy 1968



“expect families of routines to be constructed on *rational principles* so that families fit together as **building blocks**”

uit: McIlroy, *Mass Produced Software Components*,
1968 NATO Conference on Software Engineering, Garmisch, Germany.



An Inconvenient Truth

- Philippe Kruchten notes in 2005 that we haven't found the fundamental laws of software like in other engineering disciplines
- Johnson and Ekstedt argue in 2007 that we should not accept this fragmented situation and should strive for a unified theory
- Johnson and Ekstedt analyze theories that could constitute an embryo of a unified theory and give highest rating to Dijkstra's GOTO theory of 1969
- Basic concepts such as information hiding (Parnas, 1972) have been formulated a long time ago, and are still waiting to be refined



An Inconvenient Truth

- Modern organizations are faced with changing environments, creating a major challenge for the evolvability of information systems
- Empirically, software evolvability is described by **Manny Lehman's** law of increasing complexity and decreasing structure as systems age
- Perfective maintenance is becoming more costly as systems grow and age, implying an ever-increasing marginal cost for changes of information systems
- IT departments seem to be growing exponentially, confirming in an informal way that Lehman's law is indeed hanging as a Sword of Damocles over IT



Conclusions

- Amidst the confusion, we are making progress in a number of important areas
- Real progress is only made where proven engineering concepts are consistently applied, e.g. modularity and standards
- The challenge is to derive deeper, more fundamental knowledge, independent from technology hypes, in order to surpass the confusion and the current inefficiencies