

“Web Services Will Succeed
Where CORBA Failed”
True, False Or Irrelevant?

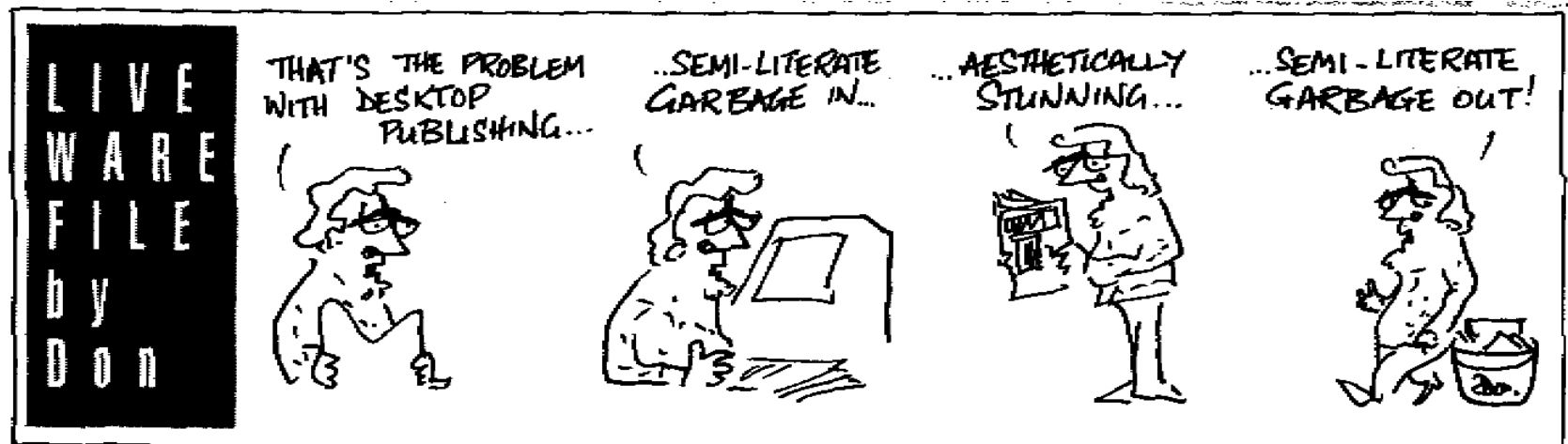
Tom Welsh
tom@tom-welsh.co.uk

OMG Technical Meeting
26 June 2002

Who Is This Guy, Anyway?

- Freelance software analyst/consultant
 - Cutter Consortium
 - www.cutter.com
 - ComputerWire
 - www.computerwire.info
 - MiddlewareSpectra
 - www.middlewarespectra.com
- Digital Equipment Corp
 - Tech support, presales, consulting, marketing
 - Software engineering, middleware

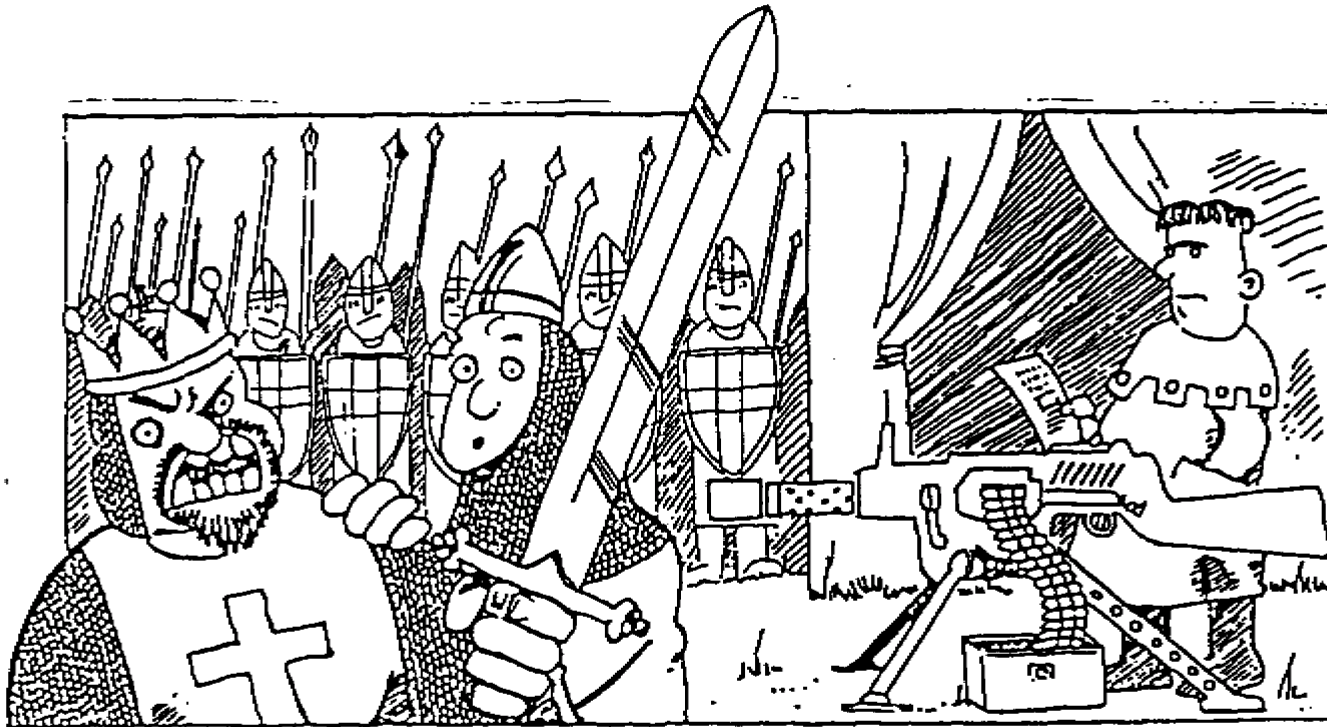
There Is More To Life Than An Attractive User Interface



Agenda

- Innovation And Progress In IT
- “CORBA Failed”
- “Web Services Will Succeed”
- Where Next?
- What Could We Do Better?
- Conclusions

ABOUT ENCOURAGING CREATIVE COMMUNICATION

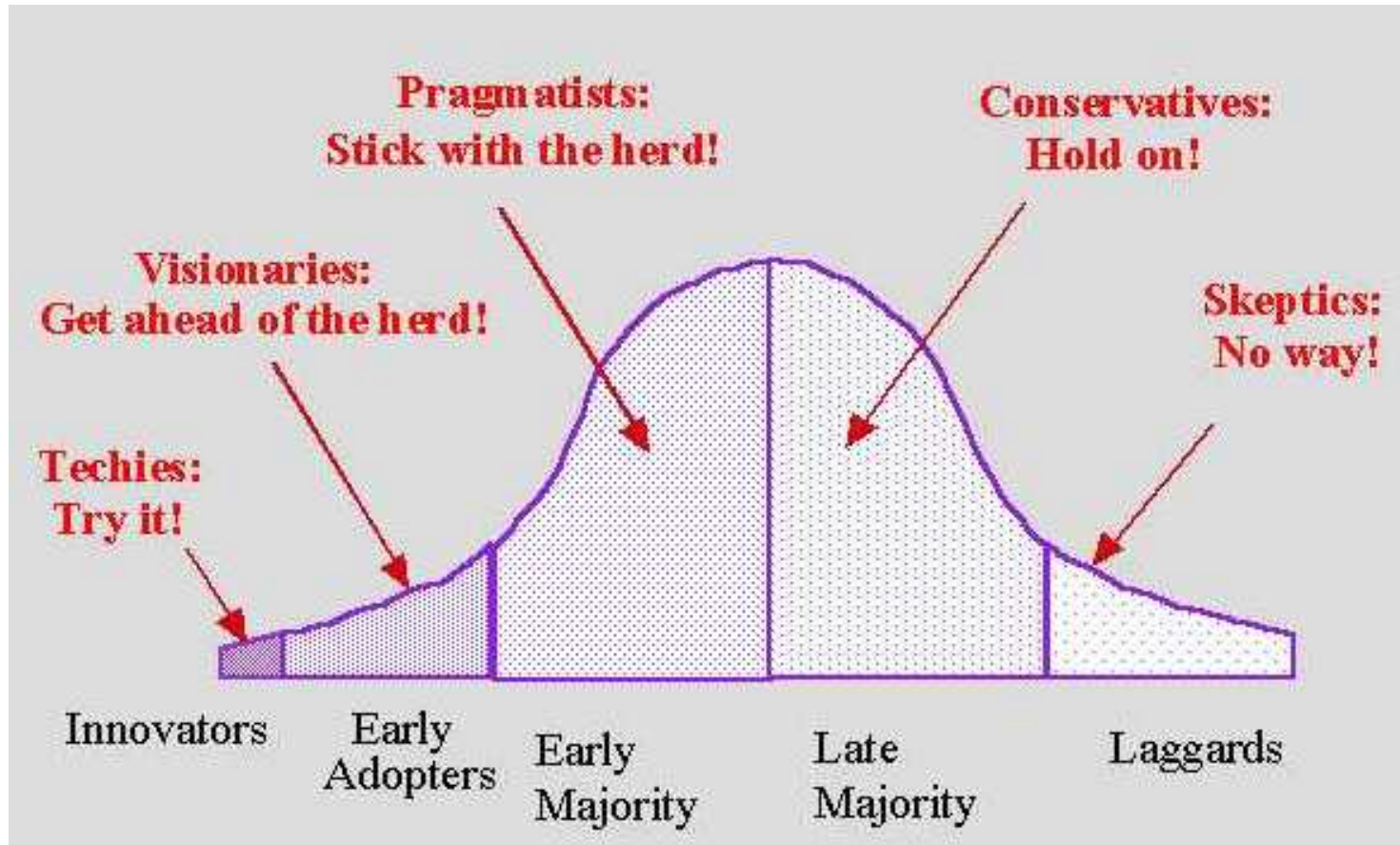


**“No, I haven't got time to see any crazy development engineers,
we've got a battle to fight”**

Innovation And Progress In IT

- State of practice is the resultant of
 - perceived needs...
 - ...partially satisfied by state-of-the-art technology...
 - ...under influence of media, analysts, “culture”
- Experts almost always over-optimistic
 - Babbage was an extreme case
 - *Megamistakes*, Steven P Schnaars
 - Start small, take lots of tries
 - Most systems are counterintuitive (i.e. too hard for us)

Technology Adoption Life Cycle



Innovation And Progress In IT

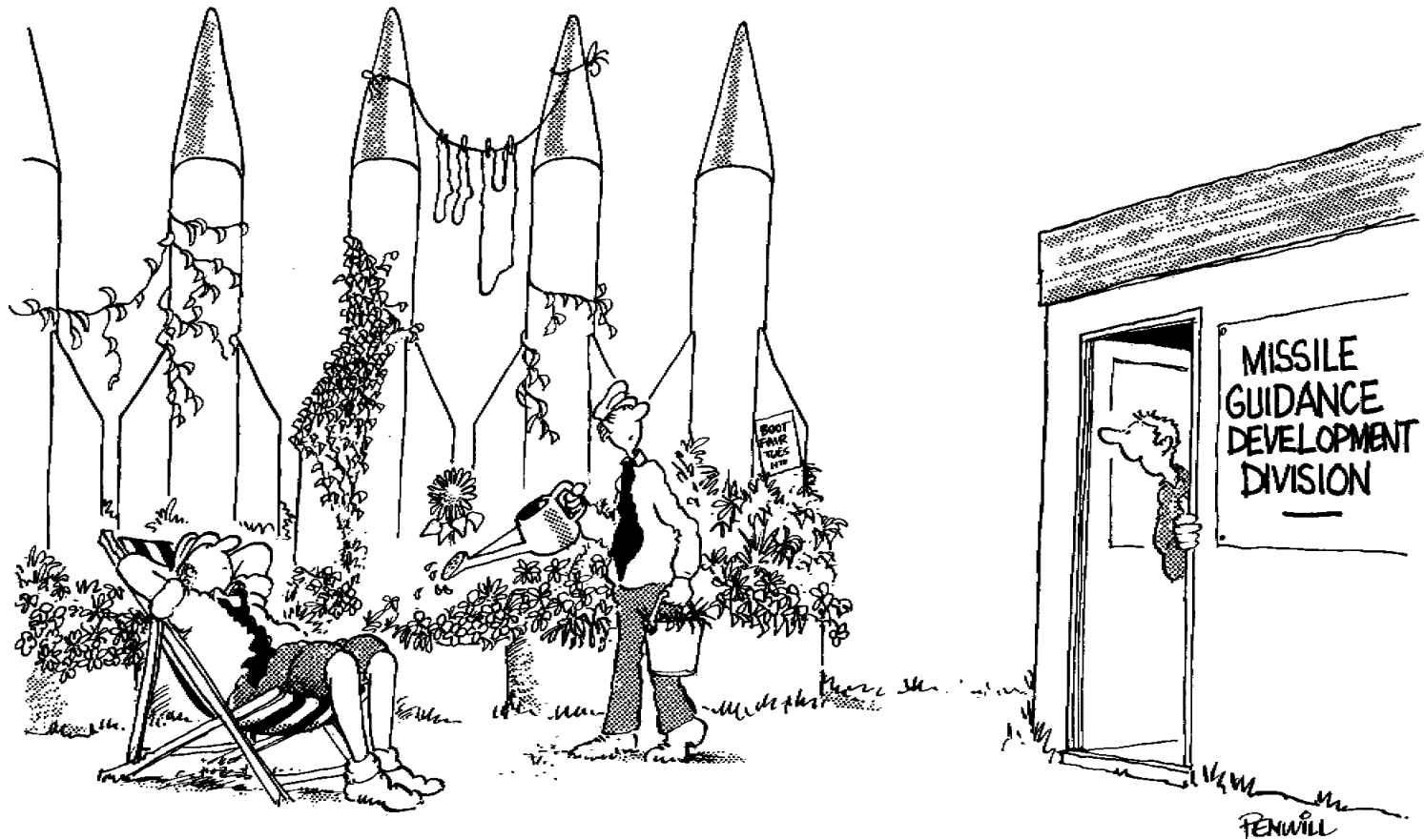
- The software industry has always reused ideas
 - Like Stanley Miller’s “organic soup”
- Consider the Web...
 - Precondition 1: ARPAnet, alias the Internet
 - Precondition 2: Graphics workstations (PARC)
 - Precondition 3: Hypertext (Bush, Nelson, etc.)
 - Precondition 4: SGML
- Now consider using the Web programmatically!
 - “No, but this is where the light is”

“CORBA Failed”

- Why would anyone say that CORBA failed?
 - Tried to learn it and gave up
 - Expected too much
 - Impatient
 - Too expensive at the margin
 - Nobody likes a committee
 - Analysts looking for contrast
 - Journalists doubling their mileage
- Back-handed compliment (like being “legacy”)

“CORBA Failed”

- Plenty of CORBA products
 - 5-6 general purpose commercial ORBs
 - Several realtime, embedded ORBs
 - 20-30 free ORBs
 - Object Transaction Managers (OTMs)
 - CORBA Component Model (CCM)
 - J2EE/EJB application servers
- Not much profit in vanilla ORBs
 - PrismTech model: free ORBs + CORBA services



" NOT LONG NOW CHAPS "

gec software

GEC Software Limited 132-135 Long Acre London WC2E 9AH
Telephone: 01-240 7171 Telex: 21403 GEC LAG Fax: 01-240 9329

“CORBA Failed”

- CORBA has gone underground within Java
 - J2SE
 - RMI-IIOP, JavaIDL
 - GIOP, POA, Interceptors, INS (J2SE 1.4)
 - EJB
 - Containers must support RMI-IIOP invocations, INS
 - JTS includes a Java mapping of OTS
 - CSIV2 Level 0, IIOP/SSL
 - J2EE
 - JTS, JNDI, JMS interoperable with CORBA

“CORBA Failed”

- No hard figures for successful projects
 - Never a mass-market product
 - Tens of thousands of licences sold (not millions)
 - Relatively small part of big projects
- After ten years, just maturing nicely...
 - CORBA 3 specifications being implemented
 - Design patterns understood
 - CORBA adapting to various ecological niches

“CORBA Failed”

- Over 1,000 organizations using CORBA, e.g.
 - CNN Interactive
 - American Airlines
 - Hong Kong Telecom
 - Charles Schwab
 - Port of Singapore
 - McKesson
 - Boeing
 - Wells Fargo

“CORBA Failed”

- **OMG Process**
 - As great an achievement as any specification
 - Arguably the best blend of standards and enterprise
- **Multiple Feedback Loops**
 - Application fails? Get training, mentor
 - Product fails? Call vendor
 - Specification inadequate? Revise at OMG
 - Process weak? Modify OMG process
- “Use standards as a ratchet” (Deming)

“CORBA Failed”

- Strengths
 - Open, vendor-neutral standard
 - Platform, language, location transparency
 - Compatible with J2EE
 - Wide range of implementations
- Weaknesses
 - Complexity, cost unjustified for simple problems
 - Practically no validation
 - Unfashionable

“Web Services Will Succeed”

- “URL-addressable resources that programmatically return results to clients”
 - Letting applications act like Web browsers
- Shift in thinking about remote communication
 - No need to conform to rigid interface (API)
 - Send XML message and let the receiver parse it
 - Contracts published in standard format
 - Relatively low-tech, ubiquitous, hard to break
 - Defeats firewalls by passing through port 80

“Web Services Will Succeed”

- Stone Soup
 - With everyone working on it, success is assured
 - It may take some time
- Convergent motives
 - For Microsoft, an alternative to Java everywhere
 - For IBM, a differentiator against Sun, etc.
 - For developers, blissful simplicity (perhaps)
 - RPC through firewalls (port 80)
 - “Test” by exposing services on the Internet

Web Services Marketplace

The screenshot shows a Netscape browser window displaying the XMethods website. The browser's address bar shows the URL <http://www.xmethods.com/>. The website features a navigation menu with links for [Services List](#), [About This Site](#), [Getting Started](#), [XML Resources](#), [Service Listing](#), [Add a Service](#), and [Notes](#).

What is this site for?
Emerging standards such as SOAP will enable a new generation of "web services" that allow systems to make remote procedure calls to other systems over the Internet. For example, a corporate inventory management system might publish a service that allows a customer system to check real-time inventory levels. This site lists publicly accessible web services.

Updates:

- All services now have WSDL descriptions
- All XMethods services are now callable on [page 80](#)
- Fixing SOAP-Lite Clients? [Read this](#)
- Don't have a server? We can [host your service](#)
- [Sign up](#) to be notified of new services via email

SOAP Service List
(0 = New)

Service Name	Description	Server Implementation
Pacific Bell XML Messaging Service	Sends SMS messages to Pacific Bell cellular phones.	SOAP-Lite for Perl
FedEx Tracker	Access to FedEx Tracking information.	SOAP-Lite for Perl
BakeFish	Interface for AltaVista's BakeFish service.	SOAP-Lite for Perl
XMethods Filesystem	Virtual Filesystem service with IMML quotas.	Apache SOAP 2.0
Ping	Performs a "ping" from the methods server to the specified host.	SOAP-Lite for Perl
Weather - Temperature	Current temperature in a given U.S. zip code region.	Apache SOAP 2.0
Amazon and Mobile Price Quotes	Returns price of a book at Amazon.com given an ISBN number.	Apache SOAP 2.0
Domain Name Checker	Checks whether a domain name is available or not.	XMethods Server
Currency Exchange Rate	Exchange rate between any two currencies.	XMethods Server
California Traffic Conditions	California highway conditions.	Apache SOAP 2.0
Delayed Stock Quotes	30 minute delayed stock quotes.	XMethods Server
S&P Price Watcher	Checks current bid price of an S&P stock.	Apache SOAP 2.0

Enter your email address and we'll notify you of new services:
We will keep your email address private. Thank to anyone who suggested a mailing list!

“Web Services Will Succeed”

- Myriad products
 - Microsoft .Net = DNA 2000 + Web services (roughly)
 - Apache XML-SOAP
 - IBM WebSphere Application Developer, alphaWorks
 - Cape Clear
 - Bowstreet
 - J2EE vendors...
 - EAI vendors...
 - Many free and low-cost packages

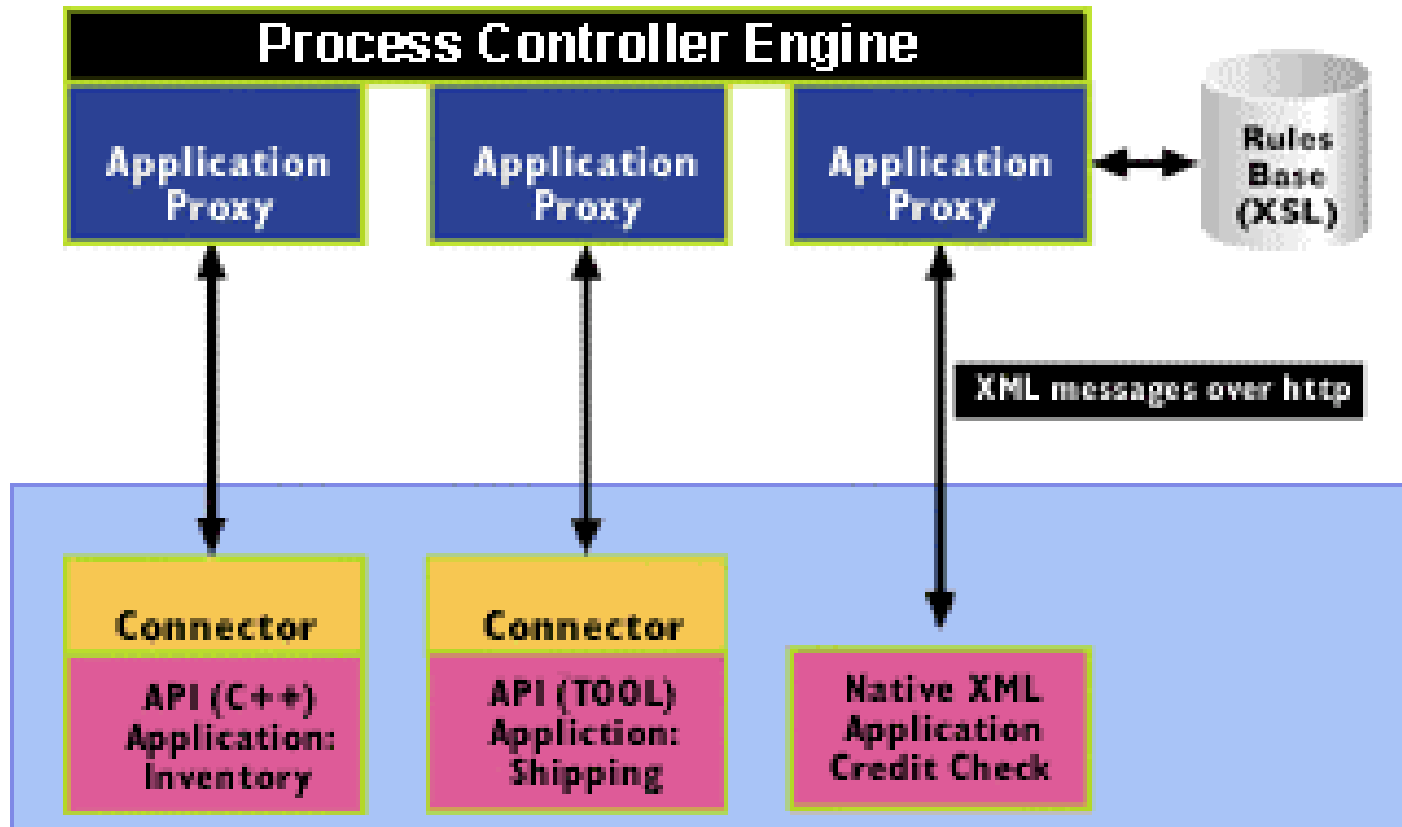
“Web Services Will Succeed”

- Three core standards
 - SOAP, WSDL, UDDI
- Others agreed by IBM, Microsoft
 - WS-Inspection, WS-Security (+ 6 related standards)
- Essentially de-facto
 - W3C working to enhance SOAP
- Many associated specifications being advanced
 - HTTPR (IBM), WSCL (HP)...

“Web Services Will Succeed”

- Two workflow languages
 - IBM’s Web Services Flow Language (WSFL)
 - Microsoft’s XLANG
- IBM’s Web Services Endpoint Language (WSEL)
 - *“I’m a bank. I can respond in less than 5 seconds”*
- Pending Microsoft candidates
 - WS-Referral, WS-Routing, WS-License
- Distributed transaction management
 - “Dependency spheres” model proposed by IBM

Web Services and EAI



“Web Services Will Succeed”

- ebXML
 - Electronic Business XML (from OASIS)
 - Universal language for B2B
 - One of many possible uses for Web Services
- BPML
 - Business Process Management Language
 - From BPMI.org
 - Control and data flow constructs for transactions

“Web Services Will Succeed”

- Java community sees XML as complementary
 - Java for algorithms, communications
 - XML as a universal data interchange format
- JCP has multiple JSRs relating to XML
 - JAXP, JAX-RPC, JAXM, JAXR
 - J2EE + Web Services Developer Pack
 - Will all be included in J2EE 1.4
- Only for interoperation outside Java world
 - JMS, RMI, RMI/IIOP between Java applications

“Web Services Will Succeed”

- Web Services Interoperability Organization
 - Set up in February 2002
 - BEA, IBM, Microsoft, Oracle, SAP, etc.
 - Sun has still not joined (Why not?)
- Explicitly not responsible for creating standards
- Three main goals
 - Provide implementation guidance, education about WS
 - Promote consistent and reliable interoperability
 - Articulate/promote a common WS vision

“Web Services Will Succeed”

- Risks and pitfalls
 - Performance
 - Integrity
 - Security
 - Shared vocabularies (ontology)
 - Functional, not object-oriented
 - Configuration management
 - Ownership

“Web Services Will Succeed”

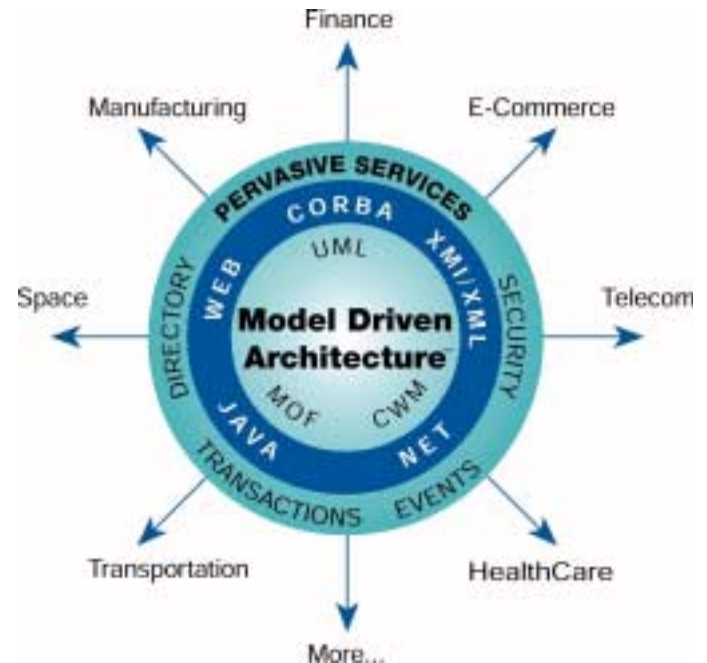
- Strengths
 - Backed by IBM and Microsoft
 - Perceived as cheaper, easier, more flexible
 - Associated with Semantic Web, .Net vision
- Weaknesses
 - Barely specified as yet
 - Inferior to CORBA, RMI, JMS for existing roles
 - Process dominated by competitive factions
 - Overpromised

Where Next?

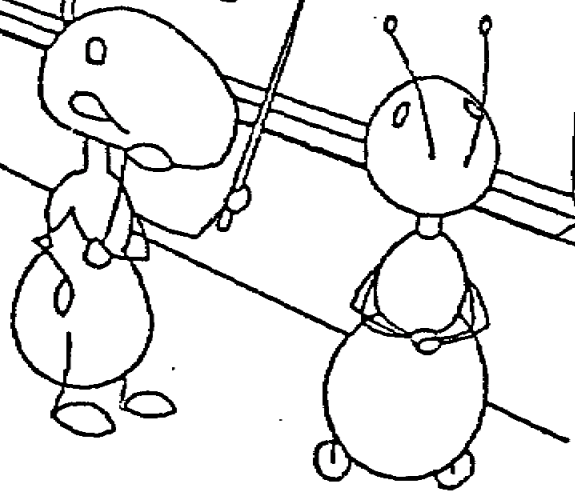
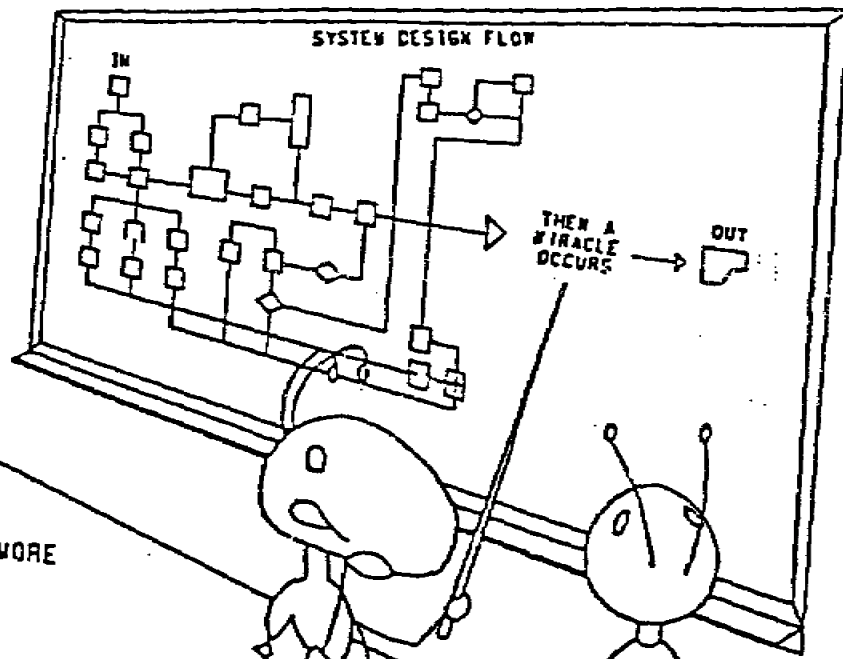
- We have several viable application server models
 - Java (J2EE, EJB, servlets...)
 - .Net
 - CCM
 - LAMP (Linux, Apache, MySQL, Perl/PHP/Python)
- These can interoperate by Web services (inter alia)
- All may continue to coexist
 - More choice
 - More competition

Where Next?

- Platform choice committal...
 - ...unless MDA adopted
- Design PIM once
- Refine to PSMs for
 - CORBA
 - J2EE
 - .Net
 - Web services
- Platform choice becomes an implementation detail



GOOD WORK
BUT I THINK WE
NEED JUST A LITTLE MORE
DETAIL RIGHT HERE!



What Could We Do Better?

- Stop burying our failures
 - Possibly main reason for ongoing “software crisis”
- Openness is a prerequisite for progress
 - Scientific publication, peer review
 - Bridges, buildings, aircraft, etc.
 - Open source, open specifications...
 - ...but secrecy about internal projects
- As at OMG, better to pool knowledge
 - Standardize what everyone is willing to share

Conclusions

- CORBA did not fail
 - Many implementations
 - Multiple platforms, languages, rich services
 - Thousands of projects, many successful
 - A brick in the wall of progress
- Web services will succeed
 - But not in the same space as CORBA
 - Several years to cross the chasm
 - Far bigger problems to solve