Service-Oriented Development: Changing How Software is Written

Jason Bloomberg
ZapThink LLC

zapthink

Web Services in the Present…

zapthink

Web Services are in the *horseless carriage* phase

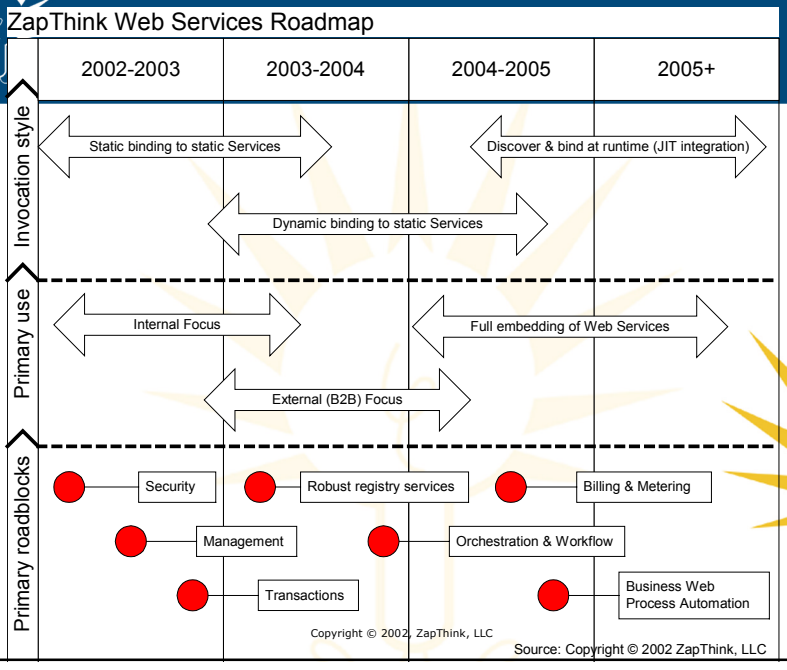
- Where new technology is applied in the patterns of the earlier technology
- Web Services are used to simplify integration

Web Services in the Future…

zapthink

New approaches to software development, engineering, architecture, and management

Connect the Dots



Source: Copyright © 2002 ZapThink, LLC

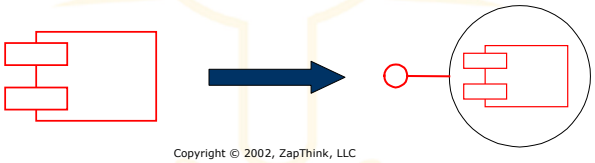
Static Components → Dynamic Services

Today:

- Component-based development: build & integrate components
- Assemble to customer's specifications

Tomorrow:

- Web Services dynamically described with WSDL files
- Developer indicates where WSDL file can be found



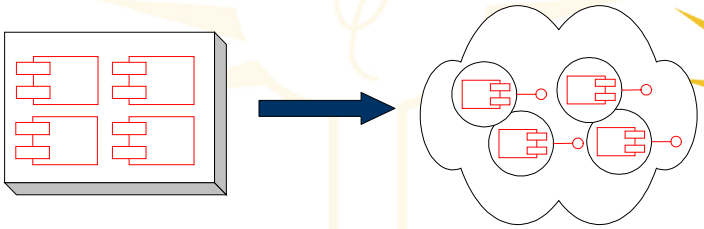
System Integration → Service Exposure & Reflection

Today:

- System requirements → architecture → component specifications

Tomorrow:

- System requirements → Service assembly
- Access dynamic descriptions



Coding for Reusability → Coding for Broad Applicability

Today:

Reusability is a pillar of OO programming, but…

- Takes more time
- No guarantee of reusability
- Developer working beyond requirements

Agile methodologies (like Extreme Programming – XP)

- Code only what customer needs at the time
- Refactor whenever extra functionality creeps in
- Resulting code is *broadly applicable*

Tomorrow:

- Develop Service-oriented architectures following Agile principles
- Ongoing, iterative process that involves customers/users
- Services constructed to be simple and broadly applicable

Disruptive Upgrades → Ad Hoc Upgrades

Today:

Modularity is a great idea, but largely a myth

- Components often not fully encapsulated
- APIs have semantic ambiguities

Tomorrow:

Modularity → loose coupling

- Expose dynamic Service descriptions
- Web Service consumers adjust to changes at runtime
- System upgrades will be *ad hoc*

Top-Down Scalability → Bottom-Up Scalability

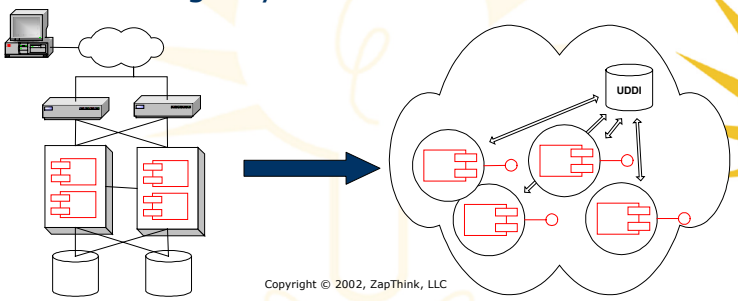
Today:

Scalability carefully planned from top down

Tomorrow:

Scalability grows from bottom up

- Use UDDI registry to locate extra resources



Platform Dependence → Platform Irrelevance

Today:

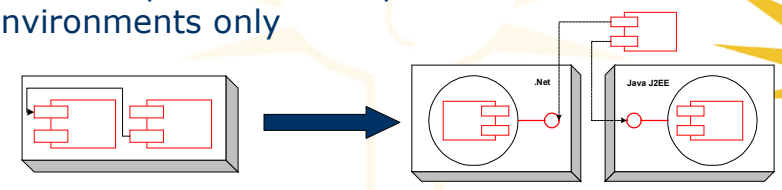
Platform independence is an unrealized dream

- DCOM & CORBA not cost-effective cross-platform
- EAI very expensive & complex

Tomorrow:

Systems communicate via Web Services interfaces

- Platforms provide development & execution environments only



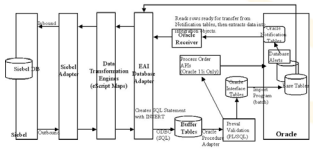
Dictatorship Model → Federation Model

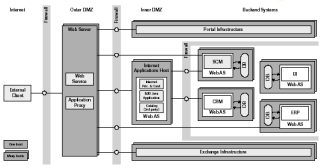
Today:

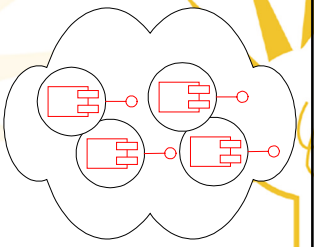
- Tightly coupled enterprise application suites

Tomorrow:

- Loosely coupled collections of federated Services



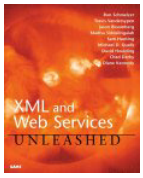




Take Away

- Thought exercises for developers, architects & IT managers
- Many roadblocks yet to overcome
- Need vision of destination in order to point in right direction

Thank You!



Jason Bloomberg & Ron Schmelzer of ZapThink are coauthors of *XML and Web Services Unleashed*.

zapthink

ZapThink is an industry analysis firm focused exclusively on XML and Web Services.

The report for this presentation is available for only $495 at www.zapthink.com.

Jason Bloomberg

jbloomberg@zapthink.com

zapthink