

Architectural Techniques for Interoperability and Coexistence

Monday, June 9, 2003

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
<http://www.rlgsc.com>

Enabling Coexistence and Interoperability

- Longevity
- Completeness
- Seamless Extensibility
 - "The universe is stranger than we can imagine"

Achieving Longevity

- Bad – Inertia, Fiat
- Good – Conceptual Integrity

Inertia – An Excuse

- low quality solution
- fear of change
- long term problem, entrenchment
- the longer inertia continues, the worse it gets

Positive – Conceptual Integrity

- conceptual integrity
- small base of good code
- little need for modification
- contract between implementation communities

Good Architecture is Good Architecture

- Architecture is Architecture
- Fads come and go; Style is timeless
- Critical examination of non-software architectures are well established and understood
- Software Architectures are seductively maleable
- Exotica is more important than it appears

Good Architecture is Good Architecture (cont'd)

- However, there are intellectual precedents
 - Buildings
 - Ships
 - Aircraft

Software is not Different

- software and systems are deceptively maleable
- but, the maleability is not real
- code, once developed is not maleable
- once systems are built to an interface, changes are expensive in effort, and schedule

Real Examples of Architecture:

- The USS Yorktown, Charleston, SC
 - Aircraft
 - Takeoff
 - Landing
 - Elevator
 - Innovative within architecture: F4U, E/C-2

Example:

- Boeing 7x7 Commercial Airlines
- Engines on Pylons
- Manueverability
- Depressurization
- Overall Structural Engineering

What an architecture does:

- balance needs of different constituencies
- contract between communities
- aesthetics of utility
- form follows function

Computer v Real World

- changeability is deceptive and illusory
- traditional architecture is better guide
- good architecture seems effortless
- good architecture minimizes interminable changes and scaffolding
- flexibility is not same as changeability

Positive Examples:

- RFC 821/822, & revisions – SMTP
- IBM System 360/370/...

Common Features of Positive Exemplars

- Demur on issues that are unneeded
- Documented ways to extend the architecture
- Specify what is needed – no more
- Avoid Hubris
- Special cases are indicia of weakness

Negative Example:

- MS Windows 3rd Party
 - Stores GUID of CD-ROM Installation device
 - Doesn't deal with multiple CD devices
 - GUID unique to Manufacturer/Model of drive
 - Other common alternative – drive letter
 - Search for volume label is trivially different yet dramatically increases robustness

Conclusions

- Good architecture is vital to leverage
- Flexibility is greatly enhanced by proper architecture
- Proper architecture is not a straight-jacket
- The most successful architectures are tremendously enabling
- Interoperability is dependent upon well designed architectures

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
<http://www.rlgsc.com>

Session Notes & Materials:

<http://www.rlgsc.com/ieee/charleston/2003-6/index.html>