Next Generation IT Systems 2006 – An Enterprise Perspective

Dr. Donald F. Ferguson

IBM Fellow
Chief Architect, IBM Software Group
Chair, SWG Architecture Board
Co-Chair, IBM Asset Architecture Board
dff@us.ibm.com
Contents

• This is an *enterprise* perspective
  – What are a few trends in our industry?
  – What are the implications for IT in medium and large enterprises?

• Due to time, the presentation has a limited focus
  – What will happen *next*. Not general long-term predictions.
  – Selective, and will not cover many other trends.

• Two related, major trends that interest me
  – SOA and Web services evolution
  – Everyone can program → situational applications

• Some additional next generation IT system trends
Example: Complexity is Forcing Change
What is ……?

… a service?
A repeatable business task – e.g., check customer credit; open new account

… service orientation?
A way of integrating your business as linked services and the outcomes that they bring

… service oriented architecture (SOA)?
An IT architectural style that supports service orientation

… a composite application?
A set of related & integrated services that support a business process built on an SOA
What’s this “Service Thing?”

Message M1, M2, … …
Op1, inMsg1, outMsg1, faultMsg1
Op2, inMsg2, outMsg2, faultMsg2
 … … …

“Those lying IMS swine. That’s CICS.”

“Dude, that’s an EJB”

“.NET. I like it.”

“Thats my simul. package!”

“You whipper snapper, we Invented that in IMS in 1923.”

Narr! Das ist eine IDOC
SOA and Web Services

• **SOA** is an *architectural* style following some principles
  – Input and output messages completely describe behavior →
    No private signaling or shared data.
  – Coarse grain interactions, unlike OO.
  – Self-describing, cataloged, well-defined messages
  – etc.

• SOA “resonates” because many customers have been doing it for years. SOA codifies a set of *best practices* for
  – Message Driven Processing (MQ).
  – SessionBeans and the Transfer Object Pattern
  – CICS TP Systems

• Web Services are a set of standards for SOA
  – Enable interoperability between infrastructure (middleware), and between different development tools.
  – Eliminates the need to integrate the infrastructure before integrating business logic to form new, composite applications.
  – SOAP/HTTP
  – WSDL, WS-Policy, specific policy assertions, BPEL4WS
  – *Like “Diplomatic French” or “Esperanto”*
Implementing Services –
The Next Generation of SOA Standards

• Web services currently says little about how to implement services.
  – What are the standards?
  – What is the process guidance?
  – What are the best practices and patterns?

• There is a next generation of standards
  – Service Component Architecture
  – Service Data Objects

• The standards enable flexible IT solutions
  – Rapid, simple composition of new composite applications.
  – Configuration/customization of solutions
  – Portable services, modules and solutions
  – Role/skill/task/specific services
Simplifying Development

Daddy,
Mommy gave me these
documents to convert.

What type of EJB
do you want to build?

Um. I do not want to build
an EJB. You see, Mommy gave me these …

Maybe you didn’t understand the question.
Your choices are SLSB, SFSB,
CMP Entity, BMP Entity, MDB

You’re not very nice.

• This is crazy.
• Programmers want to build a “part” that implements a
  “basic building block” and then aggregate them together
Process Guidance and Service Impl. Kinds

• Each step in SOA development has an **intent**
  – Expose some data through SOA
  – Write simple new business logic for a simple task
  – Implement business logic using business rules
  – Process events using a (state, event, action, state) model
  – Choreograph a set of services into a new service

• There are “kinds” of service implementation
  – Support each intent
  – Concepts and tools specific to
    • Role
    • Skill
  – Process guidance for selecting specific kinds

• Answering “How do I …” with “It depends” → **Complex** → **Guide and simplify choice**
A Simple Example and Some Concepts

Something a DB dude recognizes

Author Tools or Text Editor

Deployment Tools

Deployment Package

SDOs

public class QuoteRequest {
    String ticker;
    Date when;
}

public class QuoteResponse {
    String ticker;
    float value;
    float sharesTraded;
}

Deployment Tools

Pragma This;
Pragma That;

/*
SELECT ticker, value, activity
FROM StockQuotes
WHERE ticker == QuoteRequest.ticker
INTO quoteResponse.ticker
quoteResponse.value,
quoteResponse.sharesTraded;
*/

Deployment Package

Interface(s)

Data Svc.

Generated Code

Interpreted Metadata

P0JO

04-Jul-2006
Service Components

“Web services describes the outsides. How do you implement a service? How do you compose services?”

- Encapsulate Components for Reuse; All look the same from outside
- Components may be wired together and aggregated via flow
- Business Objects are the data flowing on wires between Components
- **Enable type, role and skills specific tools.**
Assembling Services – Modules and Subsystems

UDDI, Service Registry Project, SCCS
ad hoc exchange

Build a module
or Subsystem

© 2006 Donald Ferguson

04-Jul-2006

NGITS 2006 -- http://mis.hevra.haifa.ac.il/~ngits/
Some Examples
Some Perspectives

• SOA
  – SOA is a mature, widely practiced architecture model
  – Web services are a set of standards for interoperability, tool federation and portability

• Business Value
  – Relatively dynamic assembly: Process, Structure
  – Relatively dynamic customization
    • Add new services
    • Version/content aware binding and routing
    • Dynamic modification of rules and policies
  – Simple model for declarative business events enable business insight
  – Virtual enterprise
Some More Perspective

• Haven’t we heard this before? OO, RPC, MDP, … …

• There are some differences
  – XML is language neutral; previous approaches implied a language model.
  – WSDL and XML are more forgiving of changes
  – Supports RPC and message/document approaches from beginning
  – Common type model for application servers, message systems and DBs
  – Builds on Internet protocols already deployed for “Web browsing.”
  – Uniform model for events/pub-sub, message routing and RPC
Everyone Can Program

• Some anecdotes
  – I study karate
    • I have a black belt
    • Don’t let the affect your appraisal of my presentation
    • There are a lot of high school students
  – One “kids” wrote
    • A Microsoft Access application
    • Tracks students class attendance
    • Uses a bar code scanner
    • Prints reports
  – Kids in the locker room know I work for IBM
    • Tell me about their cool PHP, Perl, … …
    • Tell me about their cool, dynamic Web sites
• Everyone can program the way we learned long division.
What Are the Implications?

• Current “IT” Model
  – The “data center”
    • Runs a relatively small number of applications
    • For a relatively large number of employees or customers
    • The applications change very slowly, to support QA
  – LOB, department or team have “immediate” need for some new functions
    • Respond to a complex RFP/RFP collaborating with new partner
    • Complex trouble ticket that needs a new team
    • Extend the managers workspace in in a store to integrate with local carpet installation, electricians, cabinet makers
  – What to do?
    • Submit a plea to the IT department or ISV for a change
    • Have the priests of programming make the change
    • Have the data center teams run QA

• Well,
  – The business need is no longer there
  – But, I have a 25 year old who can program “mashups” or LAMP
  – The new “teenagers” have the attention of a ferret on an espresso.
  – They will not wait for the priests
A Some Simple Scenarios

• Severe weather – Meet Dave, an insurance agent.
  – He sees a news report of a severe storm. What is the company’s risk?
  – Dave writes a simple script to retrieve affected zip codes and rainfall.
  – Uses an Excel control fronting Web services to retrieve insured properties;
    Filter and select only properties insured for more than $250,000.
  – Pass rainfall predictions to ESRI to compute floods level at properties.
  – Email a risk summary to executives.

• Emailing documents is frustrating
  – Changes upon changes upon changes upon comments
  – Multiple versions
  – I don’t have the right editor
  – Dave will make his “reports” a wiki page that refreshes the data
    • Sever side “mashup”
    • LAMP
    • Have the application refresh when the forecast changes, using feeds.
A Simple Scenario

- This may seem to require complex programming skills, but it does not.
  - Feeds, URIs and simple REST retrieval of form like data are not dissimilar to including Access data in a spreadsheet.
  - Passing information between the various “parts” is similar to spreadsheets, references and formulas.
  - Very simple scripting to sequence actions
  - Use of basic primitives, like print and e-mail the form.

- This is a new composite application
  - Composes public data
  - Composes data and behavior in operational, enterprise systems
  - Sequences steps
  - Supports user interaction
  - Derives new results
Application Wiki Example

Several composition models
• Wiki + controls
• Web Spreadsheet
• Merged formed/feeds
• Scripts + HTML

Micro-formats
REST, Feeds
Exemplars

Adaptors
New Logic
Feeds

A Web of …
• LOBs and casual application developers build and test applications, perhaps just in time applications.
• Pressures will drive application redeployment into IT center
  – Governance and compliance
  – Server management is a drag
  – Publication outside the enterprise
The “Data Center of the Future”

Is radically different from today

• Applications are like Word or email
  – Moving from 100 applications → 100,000 applications
  – 1000 new applications per day
  – 10,000 modified applications/data
  – An application is for three of four people

• Defines a completely new “application server model”
  – 100,000 “VMs”
  – Many, many programming models with
    • Integrated, open source frameworks
    • Certified for “virus free” and “IP clean”
  – Focus on governance tracking {app. Version, data in, data out} →
    “Why did I deny this claim?”
  – “Protect the mainframe! Protect SAP! From the Web crazies”
Some Additional Trends and Ideas

• SOA and Event Driven Architectures
• Recipes, Patterns and Templates
• What happens when
  – HW is free (multi-core, multi-HW thread, blades, …)?
  – Everyone can program?
• Standards evolve
  – Simple, portable, open source programming model
  – Domain formats and standards, and common processes
• High performance, distributed, real-time event driven architecture and SOA/Web services converge and run in the network
• Management of Web Services and Management Using Web services.
  – Application governance
  – IT management is a business process/composite application.
• Business Process Management includes business policy mgmt.