Illuminating Middleware Dark Matter

Steve Vinoski
Chief Architect, IONA Technologies
Middleware Dimensions

Synchronous

Simple

Embedded

Proprietary

Language-specific

 Procedure-oriented

Asynchronous

Document-oriented

Language-independent

Enterprise

Standards-based

Complex
Middleware Dimensions

Many approaches
Many trade-offs
No “one size fits all”
The State of Middleware

• Middleware is everywhere
  – From small embedded systems up to mainframes, and everything in between
  – Real-world business computing systems are always diverse and heterogeneous

• There is NO “one size fits all” solution
  – Over time, deploying solutions on different middleware has resulted in islands of integration
  – There are too many useful approaches for one middleware flavor to do it all
  – (By the way, who integrates the islands?)
Service-Oriented Architecture (SOA)
Service-Oriented Architecture (SOA)
Service-Oriented Architecture (SOA)

1. Advertise

Directory service

Server application
Service-Oriented Architecture (SOA)

1. Advertise

Client application

Server application

Directory service
Service-Oriented Architecture (SOA)
Service-Oriented Architecture (SOA)

1. Advertise
2. Lookup
3. Interact
Unfortunately, Reality != SOA
Unfortunately, Reality != SOA

- Many integration projects — perhaps even most — are very unsophisticated
Unfortunately, Reality != SOA

- Many integration projects — perhaps even most — are very unsophisticated
  - The big boss: “Do whatever works and get it done quickly”
Unfortunately, Reality != SOA

- Many integration projects — perhaps even most — are very unsophisticated
  - The big boss: “Do whatever works and get it done quickly”
  - The manager: “Services? We don’t need no steenkin’ services!”
Unfortunately, Reality ≠ SOA

- Many integration projects — perhaps even most — are very unsophisticated
  - The big boss: “Do whatever works and get it done quickly”
  - The manager: “Services? We don’t need no steenkin’ services!”
  - The developers and users: “What’s a service?”
Dark Matter (Astronomy)

- Dark matter is used to explain what we don't understand about kinematics at large radii in galaxies and galaxy clusters. The amount of dark matter is about 10 times that of ordinary matter.

- There are some scientists who think that our understanding of physics is incorrect and that is the cause of "dark matter." However, most believe in dark matter being an actual particle.

- Whatever it is, dark matter is an essential component of the Universe. It's remarkable that we have been making models of the Universe without knowing what 90% of it is made up of.

Source: http://hoku.as.utexas.edu/~gebhardt/a309s02/lect1dm.html
Middleware Dark Matter

- Traditional middleware such as CORBA, MOM, J2EE, EAI covers only a small portion of the middleware universe
- Lots of other “middleware” out there is
  - Not sexy or cool
  - Not covered by the technology media
  - Not the topic of significant R&D, or of theses
  - But it’s getting the job done
- This “dark matter” heavily influences the workings of the middleware universe
The Dark Matter Influence

- Traditional middleware requires sophisticated programming skills or heavy consulting
- Middleware dark matter is created and maintained by a category of users who are not programmers or consultants
- Middleware dark matter is driving the need to make integration middleware easier to understand, build on, deploy, and manage
Application Integration "Spaghetti"
Application Integration “Spaghetti”

- 40% of every project for integration.
- 18 months time to value for EAI.
- $500K cost of entry.
- Hugely expensive maintenance.
The Most Massive Dark Matter

• Most integration is done via text files
  – Comma-separated values (CSV) is a popular “marshaling” format
  – Requests are “sent” by writing a CSV file into a shared directory
  – Requests are “received” by watching the shared directory for file updates
  – FTP and email are also popular “transports” (point-to-point and hard-wired)
More Dark Matter

• Significant integration has been accomplished through the likes of Perl and Python
  – Scripting languages are much less scary than C++ or Java
  – Extensive libraries and modules do the hard parts for you

• Visual Basic
  – Just mentioning it usually draws giggles
  – But it’s everywhere
Middleware Challenges

• Integrate the dark matter
  – Traditional middleware should recognize the existence of the middleware dark matter and accommodate it

• Make traditional middleware appeal to the purveyors of dark matter
  – Traditional middleware needs to be much less scary
Is Web Services the “Light?”

- Dark matter influence is most obvious in the Web Services (WS) movement
- If done right, WS could illuminate all that middleware dark matter
- Unfortunately that’s a big “if” (even though there are already a number of WS success stories)
Web Services Challenges

- Avoid reinventing the wheel
  - WS should enhance, not replace, existing deployed middleware systems
- Avoid splintering the standardization efforts
  - Settle the REST debate
  - RPC-oriented vs. document-oriented
- Service-oriented architectures without service references?
- Avoid the wrapper/mapping problem
Mapping Components: Just Say No!

- Whenever a new technology (such as WS) comes along, some attempt to map it directly to their existing technologies
  - For example, when CORBA came along in 1991, C++ programmers wanted to map their C++ classes directly to CORBA interfaces
- This never works!
  - You can get it to work partially, for example the COM-CORBA mapping and the reverse Java-to-IDL mapping
  - But impedance mismatches between the systems cause pain and do not allow for full inter-working
Web Services at Multiple Levels

High Level

General Business Process Modeling
Exported Web Services incorporating business processes

Middle Level

XML Integration Layer
Web Services assembled via XML Integration layer, using process engine

Low Level

.NET Web Service
Web Service exported from component-based middleware

Web Service exported directly from database or packaged application
Web Services “Sweet Spots”

• Middleware for middleware
  – Composing services by combining components with process/document flows

• Standards-based messaging replacement for EAI
  – EAI products are based on proprietary asynchronous messaging, provide adapters into a variety of back-end sources, and require extensive consulting to get them running
Specific Middleware Challenges

- Ease of use
- Understandability
- Highly adaptable “multi-paradigm middleware”
- Minimize programming
  - e.g., on-the-fly code generation
- Service discovery
  - Jini, Apple Rendezvous
- Better abstractions, i.e., hide the complexity