



SOA Lessons Learned: Where the Proverbial Rubber Hits the Road

**San Diego NDIA 2008 SOA Symposium
Bahia Resort, Mission Bay San Diego, October 16th, 2008**

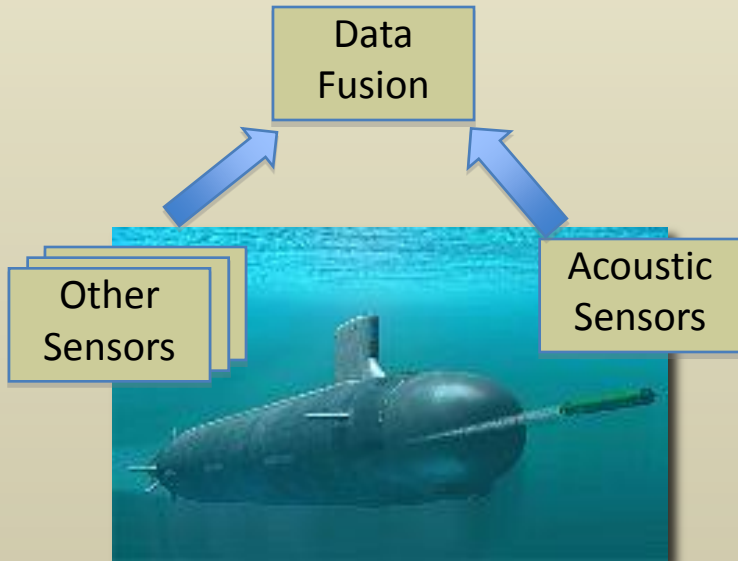
Michael Bilca

**ASSETT Executive Systems Engineer,
Combat System of the Future, PEO SUB**

- Mission: to depress Kevin Moore even further
- SBIR Project: Combat System of the Future, PEO SUB
 - Three companies building different sub-systems with minimal cooperation (restricted to agreement on the ONTOLOGY of the system)
 - ASSETT has architecture, integration, sub-system development
- Requirement: “Future-proof” next generation combat system using SOA
 - Expectation: treat *everything* that emits or consumes data on a submarine as a service
 - Expect some services to be consumed across platforms (ship-ship, ship-shore)
 - Be able to discover third party services “cold”

- Evolved from “Plug and Pray,” through IETF’s Zero-Conf, Apple’s Bonjour, to SOA Registry/Repository based auto-discovery
- Promise (best case scenario): solves integration once and for all
- Reality: distinguish two flavors:
 - WEAK Auto-discovery: SYNTAX
 - Example (WSDL field: “getLastStockPrice”)
 - STRONG Auto-discovery: SEMANTICS
- Example: Publisher’s Clearing House
- Strong Auto-discovery is needed for new/future (not-yet-invented) services
- NECESSARY: a machine-readable (no human-in-the-loop) ontology that fixes *semantics*, not only syntax
- **ONTOLOGY ties SYNTAX and SEMANTICS together**

Example: Interaction Between Sensors and Data Fusion



- At speed, acoustic sensors get noisy
- If implemented as “service” they could be asked (by SLA) to take themselves off line (decision is local)
- This takes some potentially useful data off the bus, and hides it from data fusion
- Alternative: when sensor service goes down, its data simply disappears from the bus topic (without de-registration), consuming services are free to interpret this based on local rules
- Whether implemented as service or simply a publisher to a topic, the logic that handles the “disappearance” of the data is handled in consumer
- Again, “strong” auto-discovery does not work

Result: functionality of Data Fusion STRONGLY COUPLED to SOA discovery

SOA is Not an All-or-Nothing Monolith

- Lesson: message-oriented middleware (MOM) is necessary *and* sufficient for some tasks
- Integration achieved by “wrapping” existing services in the ESB adapter layer is the first order solution, add SOA functionality/concepts *only* as needed
- Topics *can* be used as alternative to request/reply
- Take away: SOA concepts need to be GRANULARLY applied to systems/sub-systems
- **ONTOLOGY MUST COME FIRST** – *only* then can we talk of registries, repositories, auto-discovery, etc.
 - Corollary: there is no such thing as “cold” or “strong” auto-discovery, *some* human-assisted integration work is needed (and will be for a while...)
- Dave Bennett mentioned firewall crossing, testing difficulties
 - This becomes exponentially more difficult at the SEMANTIC layer