

# **Towards A Standard Design Markup Language**

## Bill Barnhill

- ◆ 14+ years in software engineering
- ◆ Associate, Booz Allen Hamilton
- ◆ Member, XRI and XDI TCs
- ◆ Specializing in XML, Semantic Technologies, and Social Software
- ◆ Contact via <http://xri.net/=Bill.Barnhill>

# Why are we here?

## ◆ The Problem:

Standards tend to be non-interoperable outside small communities of interest

## ◆ A Hypothesis:

If we apply software design patterns to standard's specification design we can significantly increase interoperability of the standards we create

...The standard's spec. as a software program

## What we'll cover

- ◆ What are some successful standards?
- ◆ What aspects of their design succeeded?
- ◆ What markup languages (ML) can we leverage?
- ◆ How does each ML adds value?
- ◆ How can we leverage each ML for interoperability?
- ◆ Where to go to get more information?

## Some successful standards

- ◆ SOAP, XMPP, XML, HTML
- ◆ By no means a comprehensive list
- ◆ But...
  - We can see a pattern:
    - Standard is highly modular
    - Standard is highly extensible
    - Standard is (mostly) unambiguous
    - Standard started simple with a minimal core
    - Standard's growth modeled incremental spiral

## Open standards are key in shaping today's SOA landscape

- A standard is a set of detailed technical guidelines that establishes uniformity
- A standard defines shared meaning that can drive commoditization of services
  - Reflects horizontal requirements from a generic problem domain
  - Considers broader applications and process models
  - Encourages levels of interoperability
- Characteristics of open standards
  - Publicly available
  - Developed by a process which sought a high level of consensus from a wide variety of sources
  - Supported by a range of readily available products

“Open standards are important to help **create interoperable and affordable solutions** for everybody. They also promote competition by setting up a **technical playing field that is level** to all market players.” (Source: Erkki Liikanen World Standards Day, 14 October 2003 )

## An Analogy

- Electrical Service Specification =
  - Source (The high pt, the line outside your house) +
  - Destination (The low pt., your toaster) +
  - Conduits (What carries the electricity, your wiring) +
  - Payload (The electricity)
- Essentially, a set of Electricity Exchange Patterns
- Software Service Specification =
  - Source (WS endpoint that initiates the exchange) +
  - Destination (WS endpoint(s) triggered during exchange) +
  - Conduits (The wiring of endpoints and composited services, in WSDL and BPEL, with WSIF or its like to connect APIs) +
  - Payload (The message in XML)
- ... A set of Message Exchange Patterns (MEPs)

History shows us ( via electrical, plumbing, internet, and WWW standards ) the need to be interoperable with platform, language, and vendor independence yields a reduced set of options and is achieved via an interface specification

# How can we learn from this analogy for interoperability?

- Current OASIS standard specifications are natural language amplified by XML interface definitions and examples
- We propose turning that around:
  - *Design each standard specification as a set of interface definitions [WSDL, WSIF], and Message Exchange Patterns (MEPs) [BPEL], with some amplifying natural language.*
  - *These parts would be DITA topics, tied together with a DITA topic map, published through XSL-FO*
  - *Topics would be shared and secured through XRI and XDI*
  - *XDI and RDF would be used to enable semantic searching*

If you can convert the MEPs from one standard's specification to the MEPs in another standard's specification, using an ontology and XSLT, then those two standards can be made technically interoperable through a BPEL wiring.

## What are the drivers for standards interoperability?

- Technology interoperability solves business problems (see next slide)
- Achieving interoperability helps prevent reinventing the wheel, and so frees us to work on new ideas, benefiting OASIS sponsors, vendors, implementers, and the TCs.

# Properly applied, standards can provide tangible business benefits

Business Problem	Standards-Based Interoperability Benefit
<p>EPA needs states to be able to post water quality data to a central EPA system in a standardized fashion</p>	<p>By using open SOA standards, EPA avoids specialized information exchange solutions for each state and reduces operating costs = <b>interoperability &amp; cost</b></p>
<p>Fidelity needs to exchange information between disparate applications without a separate middleware layer</p>	<p>Fidelity avoids the use of a proprietary and costly middleware layer and extends the use of existing technology applications = <b>agility, cost &amp; interoperability</b></p>
<p>To rapidly enter new markets, Google needs to expose its search engine to external users and applications</p>	<p>Google enables greater flexibility and reuse of the search engine by enabling searching via non-browser interfaces, such as cell phones, pagers, or visualization applications = <b>agility &amp; interoperability</b></p>
<p>Motorola needs to enable multiple incompatible business partner systems to validate customer warranty information in real-time to determine possible charges for repair</p>	<p>Motorola avoids data latency issues and allows multiple systems to pull information on demand for decision making = <b>interoperability &amp; process</b></p>
<p>Dollar Rent A Car needs to integrate its online-booking system with Southwest Airlines' website, without changing the Southwest website (which is outside of their control)</p>	<p>Southwest Airlines can leverage software capabilities of partner organizations without replicating systems = <b>agility &amp; interoperability</b></p>

## Why should we make standards more interoperable?

- The demand for interoperability by vendors and implementers will continue to increase as the number of standards, and the number of standards bodies increases.
- If we do not address it now, we will see a backlash against the use of standards. We are starting to see first signs of that now, with increased prevalence of microformats.

# Which markup languages can help?

- **XSLT** : Conversion of document descriptive markup (DITA+) to XSL-FO for publishing
- **XSL-FO** : Creation of reader friendly styled documents from markup
- **DITA** : Capture document content in re-usable and modular way
- **BPEL** : Precisely describing message exchange patterns, more machine readable than DITA's appstep
- **RDF/OWL/...** : Machine readable knowledge about standard within the standard
- **XRI** : Persistent, hierarchical identifiers of greater expressiveness than URIs
- **XDI** : Distributed, secured, and versioned re-usability of topics

# What can we do?

# Be Extensible!

- ◆ Be adaptable, especially in your XML Schema
- ◆ A good pattern:
  - ◆ eXtreme eXtensibility by Roger Costello
    - <http://www.xfront.com/eXtreme-eXtensibility.html>
- ◆ Start with a core spec, as small as possible,
- ◆ Freeze it as much as possible, as soon as possible
- ◆ Add small modular extensions, make them as optional as possible
- ◆ Specification development through evolution not revolution

# Be Modular!

- ◆ Specify how modules that extend spec will be published/found/negotiated in your core spec
- ◆ Make your core standard as much a microstandard as possible
- ◆ Everything else becomes a module, aka extension
- ◆ Publish the topics from your specs so they are re-usable
- ◆ Use XSLT to publish to multiple formats, multiple languages
- ◆ Don't tie yourself to one transport (HTTP)
  - ◆ Other possibilities: JMS, XMPP, SIP, ...

# Be Precise!

- ◆ Use DITA to write your standard
- ◆ Add in BPEL to describe your MEPs from the MEP initiator's POV and from triggered participant's POV
- ◆ Use XRI for robust and precise identifiers
- ◆ Add semantic markup to your topics and published docs
- ◆ Wrap published docs, topics in XDI for precise distribution controls

# Practice Safe Standards!

- ◆ Design for the semantic web (LC)
- ◆ Get your standard into a standards body as early as possible
- ◆ Have an open process and solicit feedback as soon as possible
- ◆ Re-use, don't re-invent

# But there are some problems...

- ◆ Biggest is tool support
  - There is none!
  - Well, some for individual markup languages
  - But none for the envisioned standards markup that combines them
- ◆ If you like these ideas help write tools to support them
- ◆ I'll be maintaining a list of related projects
- ◆ For more details contact me via:
  - <http://xri.net/=Bill.Barnhill>

# Next Steps

- If sufficient interest...
  - Publish example of standard design using this method, with examples of achieved interoperability
  - Promote creation of tool-chain to support this design method
  - Create new TC or subcommittee under existing TC (SOA RM?, WS-BPEL?) to standardize this design method

# For more information...

## DITA

- <http://www.ibm.com/developerworks/xml/library/x-dita1/>
- <http://www.oasis-open.org/committees/dita/>

## OASIS Extensible Resource Identifier (XRI) TC

- <http://www.oasis-open.org/committees/xri/>

## XDI.ORG

- <http://xdi.org/>

## Why XRI?

- <http://xml.gov/presentations/onename2/xri.ppt>

## ■ XSL

- <http://www.w3.org/TR/xslt>
- <http://www.antennahouse.com/XSL/sample/XSLsample.htm>
- <http://www.ibm.com/developerworks/xml/library/x-xslfo/>

## ■ RDF

- <http://www.w3.org/TR/rdf-primer/>
- <http://www.w3.org/RDF/>
- <http://www.ontopia.net/topicmaps/materials/tmrdf.html>

## ■ OASIS WSBPEL TC

- <http://www.oasis-open.org/committees/wsbpel>

# Auxiliary Slides

# How can XSL help?

- DITA+ to XSL:FO, XSL:FO gateway to any publishing medium
- Extension to XSL that allows match reasoning based on RDF
- RDF Example:
  - `*[rdf:match_asserted("?r rdfs:subTypeOf foaf:Person, ?r foaf:Organization http://communitivity.com")]`
- An extension that allows matching XRI type of topic resources
- XRI Example:
  - `*[xri:match_type("xri://@communitivity/(+person)"]`

# How can DITA help?

- ◆ Ambiguity is bane of interoperability
- ◆ Not to be confused with extensibility
- ◆ DITA enables writing with less regard for formatting
- ◆ Faster doc creation means more detail, less turn around time
- ◆ Documents created as a map of relevant re-usable topic modules, making editing and spec revision easier
- ◆ Extend DITA to include Message Exchange Pattern(MEP) BPEL topic modules
- ◆ Means each specification contains reference implementation in BPEL
- ◆ One suggestion:
  - Add a tie-in construct within DITA topic maps to ease narrative flow

# How can Semantic Web and Data Web technologies help?

- ◆ XDI links distributed DITA topics for re-use in a secure, versioned way
- ◆ RDF or XDI describe metadata about topics and published documents
- ◆ Re-usable topic sharing controlled via XDI link contracts and policy language
  - ◆ XACML?
  - ◆ WSPL?

# How can BPEL help?

- ◆ Each MEP in spec defined with two BPEL DITA topics:
  - One representing triggered party
  - One representing triggering party
- ◆ Syntax defined with XML Schema (W3C or RNG)
- ◆ BPEL captures message exchanges and high level wiring
- ◆ WSIF or something like it describing back end interface that needs to be implemented to support exchanges

# How can Booz Allen Hamilton Help?

- What can we do to help?
  - Assist clients to apply standards rapidly and efficiently to existing information sharing (e.g. interoperability) problems
- Who can we help?
  - Organizations who want to acquire systems that are extensible and interoperable
  - Everyone that can recognize the role of standards
  - Anyone who are mandated to comply with standards
- Why we can help?
  - We are honest brokers of standards and their applicability