Finding The 80/20 Point for XML Performance

David Orchard, BEA Systems

dorchard@bea.com

Questions

- Is there an 80/20 point?
- Can we get there by consensus?

Reaching the Right Solution

- Many candidates
- Few criteria for making an **well-engineered** decision.
- To get there, we need to:
  1. Find and define **real-world use cases**
     Why do we want to do it?
  2. Explore and agree upon **properties**
     Provide metrics to judge XML and candidates against
  3. Gather **Candidate Solutions**
     What do we have to work with? What tradeoffs do they make?
  4. **Select** and **Standardize**
     Measuring **candidates** against **established properties** and judge the results against **real-world use cases**.

Finding Real-World Use Cases

- We hear lots of demand for better XML performance, but is it really a problem with XML or…
  - Application architecture
  - Suboptimal tools
  - Verbose data (encouraged, but not caused, by XML)
  - Subjective measurement
- What problems require something new?
  - Remember that processors get faster, pipes get fatter and disk gets cheaper…
  - Any standard solution must be competitive in five years, not today
- How prevalent are the use cases we find?
- What's left are our **real-world use cases**

Relating Constraints & Properties

- XML's constraints yield **properties** (n:n)
  - content model+ns -> **extensibility, evolvability**
  - simplicity -> **interoperability**
  - human readability -> **modifiability, visibility**
  - human editability -> **modifiability**
  - self-description -> **Robustness, reusability**
  - streamability -> **application performance**
  - typing -> **robustness**
  - **…**
- **Each property is a metric** to judge use cases, candidates and XML itself against.
- Ubiquity because right set of Property values

Performance Metrics

- If the goal is to optimize performance, **shared, well-understood and objective performance metrics** are critical.
- Solution measurement should encompass as many aspects as possible;
  - Document size
  - Element/content mix
  - Diversity of elements
  - Use of XML Namespaces
  - Availability, complexity of Schema
  - …

Selecting and Standardizing

- Selecting a candidate should be approached as an **engineering problem**
- Find the **right 80/20 balance**, don't invent the perfect solution.
- For each candidate, consider:
  1. What properties does it have?
  2. What use cases does it optimize for?
- Standardize the candidate that:
  - Offers the **greatest gain in performance property**,
  - Loses the **least number of other properties**, and
  - Satisfies the **greatest proportion of use cases.**

What Shouldn't Happen?

- **An unquantified standard**
  - How do users know when it's appropriate to use?
- **A domain-specific standard**
  - Should be developed in specialized application domains (e.g., MTOM for Web services)
  - If we can't address a sizeable and diverse set of use cases, we shouldn't standardize anything
- **Multiple standards**
  - Risk of fragmentation, great reduction in XML's value
  - No standard is better than more than one!
- **A standard that doesn't make tradeoffs**
  - If it's better without sacrifice, it should be incorporated into XML
- **Standards body as R&D lab**
  - Complex, unusable outcome that's the worst of all worlds

Suggested Breakout Discussions

1. Agreement on Constraints and Properties
2. Gather use cases and contextualize them against the greater use of XML - do they justify a new encoding?
3. Consider whether its prudent to standardize anything.
4. Discuss what abstraction it should be based upon (e.g., characters, Infoset, Query Data Model, etc.).
5. Discuss what property tradeoffs are acceptable and necessary to satisfy the use cases.

Closing Thought

We currently exist in a state of grace - there is no:

- XML serialization choice to be made
- Negotiation mechanism necessary
- Need for parsers to support multiple serializations
- Need for special XML viewers
- Need for XML-based specifications to consider multiple serializations

Which properties are we willing to trade off, and what properties are we getting in return?