XML Issues and Recommendations for Best Practices within 3GPP SA-5

A Discussion of XML Related Issues in 3GPP SA-5 and best practice recommendations for dealing with said issues.

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issues

- Namespaces
  - **To default or not to default**
  - **Multiplicity**
  - **Component Namespace Localization**
- Global vs. Local Components
- Element vs. Type
- Variable Content Containers
- Composition vs. Inheritance
- Creating Extensible Content Models
  - **and extending XML schemas**
- Schema Versioning

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Default Namespaces

- What schema do I use as the default schema?
  - **XMLSchema**
  - **targetNamespace**
  - **No default schema**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Approaches to Namespaces

- With the exception of no namespace schema, every schema has 2 namespaces:
  - http://www.w3c.org/2001/XMLSchema
  - **targetNamespace**
- 3 approaches
  - **XMLSchema == default namespace, and qualify all references to components in the targetNamespace**
  - **targetNamespace == default namespace, and qualify all references to other schemas**
  - **Do not use a default namespace, and qualify all references to all schema entities**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Default XMLSchema, qualify targetNamespace

- Advantages (in using this approach)
  - **When the schema references components from a number of namespaces, this approach provides a consistent way of referring to the components**
- Disadvantages (in using this approach)
  - **Schemas which have no targetNamespace will be required to qualify references to XMLSchema**
  - **Schemas with a targetNamespace will not have to qualify references to XMLSchema.**
  - **This is confusing …**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Qualify XMLSchema, default targetNamespace

- Advantages (in using this approach)
  - **Schemas with no targetNamespace will need to qualify XMLSchema**
  - **Schemas with a targetNamespace will also need to qualify XMLSchema (and all others).**
  - **Consistency is maintain, this is good**
- Disadvantages (in using this approach)
  - **Some references will be qualified**
  - **Some will not, i.e., components in targetNamespace**
  - **This can be confusing …**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

No default, qualify everything

- Advantages (in using this approach)
  - **Everything is qualified**
    - **Makes it very easy to understand where components and elements originate**
  - **Simple, clean, and consistent**
- Disadvantages (in using this approach)
  - **Size of schema documents increases over previous two approaches**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practice

- No default namespace …qualify everything
  - **Consistency**
  - **Clarity**
  - **Specificity**

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Multiplicity of Namespaces (MoN)

- In a project that defines multiple schemas (such as 3GPP SA-5)
  - **Should all the schemas be within a single targetNamespace?**
  - **Should each schema be in its own (or the default) targetNamespace?**
  - **Should one of the schemas have a targetNamespace and the other schemas have no targetNamespace?**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Approaches to MoN

- Heterogeneous namespace design
  - **Each schema exists in within a unique namespace**
- Homogeneous namespace design
  - **All schemas exist within a single, umbrella namespace**
- “Chameleon-effect” namespace design
  - **Give the “managerial” schema a targetNamespace**
  - **Give the “supporting” schemas no targetNamespace**
  - **The supporting schemas will take on the namespace of the managerial schema at <include>**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Impact of MoN: Heterogeneous

- Each schema must have a namespace
  - **Corollary: there needs to be a names declaration for each namespace**
- The components must all be uniquely qualified
  - **Explicitly**
  - **via default namespace (bad)**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Impact of MoN: Homogeneous

- Namespaces get cluttered
- Management of dependencies becomes an issue
- Schema fragments include other schema fragments via <include/>

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Impact of MoN: Chameleon Effect

- Access is allowed from the “managerial” schema only to components that do not exist within a targetNamespace
  - **via <include/>**
- Name collision becomes an issue
  - **Wrapper schemas that <include/> and are then in turn <import/>’d by the managerial schema**
  - **Wrapper approach allows use of <redefine/>**
- The components with no targetNamespace get namespace-coerced [to the namespace defined by the managerial schema]

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

<redefine/>

- Applicable to
  - **Homogenous approach to MoN**
  - **“Chameleon-Effect” approach to MoN**
- Enables access to components in another schema
- Allows for the modification of zero or more of the components being imported

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practice: MoN, Chameleon

- With schemas that contain components that have no inherent semantics in isolation
- With schemas that contain components that have semantics only in the context established by an <include>ing schema
- As a rule of thumb, if the schema contains only type definitions, don't define a namespace

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practice: MoN, Homogeneous

- When all of the schemas in question are conceptually related
- When there is no need to [visually] identify [in instance documents] the origin of element or attribute

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practice: MoN, Heterogeneous

- When namespace isolation is required
  - **That is, when there are multiple components that are content orthogonal yet have the same name**
- When there is a need to [visually] identify the origin of each element/attribute

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practice: MoN, General

- Make use of the “id” attributed define by XML Schema
  - **Enables a schema internal identifier that provides a higher degree of granularity of identification than namespace alone**
  - **The combination of namespace plus the schema id attribute provides a powerful tool for visual and programmatic identification of components**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Component Namespace Localization

- When should a schema be designed to hide (localize) the namespaces of the elements and attribute it is using, versus when should it be designed to expose the namespaces in instance documents

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Schema Layering

- Schema paradigms
  - **“Mixin”**
    - **Not particularly useful on its own**
    - **Used to build composite schema**
  - **Composite**
    - **Built from “Mixin” schemas**
    - **Adds additional elements**
  - **Self-Contained**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Component Namespace Localization

- Applies only to the namespace in which elementFormDefault is defined
  - **That is, if you want to be able to hide component namespace complexities from mixin components, you must use "unqualified"**
- Component Namespace Localization requires
  - **The value of** elementFormDefault **must be** unqualified
  - **The element(s) may not be globally declared**
    - **"global elements" are immediate children of <xsd:schema/>**

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

CNL – Namespaces Hidden

- **Instance documents are**
  - Simple
  - Easy to read
  - Easy to understand
- **The mixin namespaces are irrelevant to the author of instance documents**
- **Use when**
  - simplicity, readability, and understandability are of the utmost importance
  - Need the flexibility of being able to change the aggregate schema, without changing instance documents

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

CNL – Namespaces Visible

- Instance documents are
  - **More complex**
  - **More difficult to read**
  - **More difficult to understand**
- Use when
  - **The lineage/ownership of the elements is important**
  - **Namespace qualification is needed to determine how to process an element within an instance document**
    - **Same name, different namespaces where context is insufficient to determine the correct course of action**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practice Recommendation

- Need for localizing is determined on a case by case basis
- Keep two versions of all schemas
  - **Identical except for the value of** elementFormDefault
    - **qualified**
    - **unqualified**
- Minimize the use of global elements and attributes
  - **This enables** elementFormDefault **to be used as an "exposure switch"**

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Global vs. Local Components

- When should an element type be declared globally versus locally ?

-- A component (*element, simpleType, and complexType*) instances are global if they are an immediate child of <xsd:schema/>

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Thinking about component design

- Three fundamental approaches to component design
  - **Russian Doll approach**
  - **Puzzle Piece Approach**
  - **Snap Fit Building Block approach**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Component Design Approach: Russian Doll

- A single top level component that contains all other components
- Can make use of both <element/> and < …Type/>
- Characteristics
  - **Opaque content**
  - **Localized scope (enables CNL)**
  - **Bundled**
  - **Instance documents are hierarchical and cohesive**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Component Design Approach: Puzzle Piece Approach

- A number of components, each with a fixed purpose
- Relies Heavily on the use of <element/>
- Characteristics
  - **Transparent content (components are visible)**
  - **Global Scope**
  - **Verbose**
  - **Coupled**
  - **Components are hierarchical and cohesive**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Component Design Approach: Snap Fit Building Blocks

- A number of modular components, each with a specific purpose
- All components are declared using <…Type/>
- Characteristics
  - **Maximizes reuse**
  - **Maximizes namespace hiding**
    - **Easy exposure switching**
  - **Related**
  - **Components are cohesive**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practice: Component Design Approach

- Use Snap Fit Building Block when
  - **Schemas must be flexible enough to turn namespace exposure on and off**
  - **Component reuse is of the utmost importance**
    - **ad-hoc reuse is desirable**
- Use Puzzle Piece when
  - **Component reuse is important, but not paramount**
    - **ad-hoc reuse is not desirable**
- Use Russian Doll when
  - **Schema size must be minimized**
  - **Component reuse is not desirable**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Element vs. Type

- When should a component be declared as an <element/> versus when should a component be declared as a <…Type/>?

-- This is a difficult topic to discuss, purely in terms of “practical XML”. The issues relevant in this area are not XML issues, but rather “pure information modeling issues”. As such discussion of the issues is not within the scope of a modest slide set; and for this topic, a set of recommendations for best practices are simply provided.

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practices (1)

- When in doubt, declare types
  - **You can always create an element from the type**
  - **<…Types>s and <element/>s exist in separate symbol spaces;**
  - **if you want it an element later, it is a simple matter to create one**
- If the component is not intended to be an element in an instance document, it should be a type
- If the content of a component is to be reused by other components, define it as a type

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practices (2)

- If a component is intended to be used in instance documents, and it is required that sometimes it can be nillable, and other times it cannot; it must be declared as a type
- If a component is intended to be used in instance documents and other components are allowed to substitute for the component; it must be declared as an element

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Variable Content Containers

- What is the best practice for implementing a container element that is to aggregate variable content?
  - **Two issues to consider are:**
    - **Is there a requirement that the “variable content” be allowed to originate from disjoint sources?**
    - **Is there a requirement that the allowable types of “variable content” grow over time?**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Variable Content Container Mechanisms

- Abstract Substitution
  - **Abstract element and element substitution**
  - **Abstract type and type substitution**
- <xsd:choice/>
- Dangling types
  - **VCC for simple content**
- Hermaphrodite types
  - **Bertrand Poisson; CHAMTS DSI BE CCE**
  - **Not covered in this slide pack, referenced for completeness**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Variable Content Container: Abstract Element Substitution (VCC-AES)

- Five key XML concepts
- Advantages
- Disadvantages

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-AES: Five Key XML Concepts

- An <element/> may be declared as abstract
- Abstract elements cannot be instantiated in instance documents
- Concrete elements must be declared within a <substitutionGroup/> with the abstract element for which they will be substituted.
- Membership in a substitutionGroup is restricted to elements that are of the same type as, or derived from, the type of the abstract element
- The abstract element and all other elements within the substitutionGroup must be declared globally

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-AES: Advantages

- Extensible
  - **Allows the extension of the set of elements that may be used in the VCC**
- Semantic Cohesion
  - **The AES approach to VCC requires inheritance of type, providing**
    - **Structural Coherence**
    - **Semantic Coherence**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-AES: Disadvantages

- No independent elements
  - **inheritance of type required**
  - **membership in substitution group**
    - **Adding a new element, requires an update to the schema wherein the substitution group is defined**
- Limited Structural Variability
  - **Change over time may give rise to elements that are conceptually similar, yet structurally very different.**
  - **Improper inheritance**
- Non-scalable XSLT processing
- No control over namespace exposure (CNL)

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Variable Content Container: Abstract Type Substitution (VCC-ATS)

- Three key XML concepts
- Advantages
- Disadvantages

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-ATS: Three Key XML Concepts

- A <complexType/> can be declared abstract
- An element declared in terms of an abstract type cannot have its type instantiated in instance documents
  - **That is, the element can be instantiated, but its abstract content may not**
- Within an instance document an element with an abstract type must have its content replaced from a concrete type that derives from the abstract type
  - **Classic "Type Substitution" as described by Liskov**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-ATS: Advantages

- Extensible, as for VCC via AES
- Minimal dependencies
  - **Extension of the set of content components for the VCC only requires access to the abstract type**
- Scalable XSLT process
  - **<xsl:for-each select=“ …”> </xsl:for-each>**
  - **No code changes as new subtypes are added**
- Semantic Cohesion
- Control over namespace exposure (CNL)

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-ATS: Disadvantages

- No independent elements
  - **As per VCC via AES**
- Limited Structural Variability
  - **As per VCC via AES**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Variable Content Container: <choice/> (VCC-<choice/>)

- Declare an instance of <xsd:choice/>
- An explicit list of all the elements that may be used within the VCC
- Advantages
  - **Elements are independent, i.e., no inheritance of type required**
- Disadvantages
  - **Not extensible**
  - **No semantic coherence**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Variable Content Container: Dangling Type (VCC-DT)

- Provides a mechanism to create Variable Content Containers for <simpleType/>s
- Key Concept
  - **With an <xsd:import/>, the schemaLocation attribute is optional**
- Design Pattern
- Advantages
- Disadvantages

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-DT: Design Pattern

- When declaring the VCC element, provide a type that is in another namespace
- When <import/>ing that namespace, do not provide a schemaLocation
- Create any number of implementations of the dangling type
- Instance documents identify the schema to be used to implement the dangling types

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC-DT: Advantages and Disadvantages

- Advantages
  - **While the definition is restricted to <simpleType/>s the implementation of the DT can be a <complexType/>**
  - **Extremely Dynamic**
- Disadvantages
  - **REQUIRES multiple namespaces (see Issue: MoN)**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC: Best Practices, VCC-AES

- Use when
  - **It is acceptable for all elements to inherit from a common type**
  - **It is required to be able to extend the collection of elements without modify the schema**
  - **There is not a requirement that requires the container elements to be namespace localized**
    - **That is, you can live with namespace exposure on**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC: Best Practices, VCC-ATS

- Use when
  - **All the elements in the VCC are of the same type; or are derived from the same type**
  - **It is acceptable that all elements in the VCC have a uniform name**
    - **That is, variable content nested within an element that contains the variable content**
  - **The collection of elements may grow, independent of the container schema**
  - **There is a requirement for namespace localization**
  - **There is a need to support scalable XSLT processing**

MOTOROLA
intelligence everywhere™
MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC: Best Practices, VCC-<choice/>

- Use when
  - **There is a requirement to contain a collection of independent elements**
  - **It is acceptable to allow an external authority (typically a human being) to select and verify the legal elements for the collection**
  - **Growth of the collection of elements is tightly determined by the external authority that controls the schema**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

VCC: Best Practices, VCC-DT

- Use when
  - **A VCC for <simpleType/>s is required**
  - **There is a requirement to allow the extension of a <simpleType/>**
  - **There are requirements that define a need for very dynamic, customizable content**

-- For schema validators that do not implement dynamic type support, the type="anyType" provides a work around that enables VCC-DT, with a loss of type safety

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Composition vs. Inheritance

- Should schemas be designed as
  - **Type hierarchies**
  - **Aggregate components**
  - **Some combination thereof**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

XML Design by Inheritance

- Complexity
  - **Deep hierarchies of “super-classes” requires the instance author to have a great deal of knowledge of the the things that come before**
- Tight Coupling
  - **If the root should change (or in fact any element within the hierarchy), the children will be impacted**
  - **This is not C++ or Java™ programming, data is not hidden from the instance document author**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

XML Design by composition

- Acknowledge that requirements
  - **expand**
  - **Change**
- Determine where the changes are likely to occur
  - **Points of Change (PoC)**
- Encapsulates the Points of Variation
- Favoring composition over inheritance is a fundamental precept of information modeling
  - **Type hierarchies remain small and are very unlikely to grow into unmanageable monsters**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practices: Composition vs. Inheritance

- It is unquestionable that type hierarchies are required in XML
  - **Avoid using derive-by-restriction (DbR)**
  - **Limit the depth of type hierarchies**
    - **There is no hard and fast rule here**
    - **If you go past 3 levels deep, there should be good reason for doing so**
- Favor composition, do not ignore inheritance

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Creating Extensible Content Models

- What is/are the best practices for creating extensible content models?

-- *A component (or schema) has an extensible content model if an instance document can extend that content model with additional elements without changing the schema.*

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Extensibility via Substitution (EvS)

- Abstract Element Substitution
- Abstract Type Substitution
- Disadvantages
  - **Location Restricted Extensibility**
    - **New content can only be added at the end**
  - **Unexpected Extensibility**
    - **Ad-hoc extension happens;**
    - **Often overlooked when creating XSLT**
      - **It would be nice to be able to flag extension points**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Extensibility via <any/> (EvA)

- An <any/> element allows a content model to indicate points where user [instance document author] defined content is expected to be inserted
  - **Use of <any/> removes the issue of location restricted extensibility**
  - **Explicitly flags the points of extension that are "recognized" by the schema author(s).**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Issue: Schema Versioning

- What is the best practice for versioning XML schemas?

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Versioning Scenarios

- The new schema changes existing components within the content model
- The new schema extends the content model without changing existing components of the content model

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Approaches to Schema Versioning

- Change the an [internal] schema version attribute
- Create a schema version attribute on the root element of the schema
- Change the schema targetNamespace
- Change the name location of the schema

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Change the [Internal] Schema Version Attribute

- Advantages
  - **Easy**
    - **its part of the schema specification**
  - **Transparent to instance documents**
  - **The schema contains an indication that a change has been made**
- Disadvantages
  - **Validators ignore the “version” attribute of <schema/>**
  - **Therefore, the constraint is not enforceable**

MOTOROLA
intelligence everywhere™
MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Schema Version attribute at root <element/>

- Advantages
  - **A “fixed” value for a mandatory attribute at the root element of a schema is an enforceable constraint**
  - **Instance documents will not validate unless they contain the matching value**
- Disadvantages
  - **The version number in the instance much match exactly**
  - **Prevents a single instance document from “validating” against multiple versions of the schema**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Change the Schema targetNamespace

- Advantages
  - **The schema contains an indication that a change has been made**
  - **Schemas and instance documents that make use of the new schema must be explicitly changed**
- Disadvantages
  - **Schemas and instance documents that make use of the new schema must be explicitly changed, if the the new features of the schema are needed**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Change the Name/Location of the Schema

- Advantages
  - **Schemas and instance documents that make use of the new schema must be explicitly changed**
- Disadvantages
  - **Schemas and instance documents that make use of the new schema must be explicitly changed, if the the new features of the schema are needed**
  - **The schema contains no indication of the change**
  - **schemaLocation is optional**
    - **Even when present, schemaLocation is not normative**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practices: Schema Versioning

- Capture the schema version somewhere in the XML Schema
- Identify in the instance document the version(s) of the schema to which the instance document is compatible
- Keep the previous versions of the schema available
- When the content model is extended without changing existing components, keep the extensions backward compatible (if at all possible)
- If backwards compatibility cannot be maintained, change the value of targetNamespace

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Best Practices in a Nutshell

- Make the targetNamespace the default namespace
- Uniquely identify all schema components via the “id” attribute
- Two versions of every schema
  - **elementFormDefault=“qualified|unqualified”**
- Postpone decisions as long as possible
  - **Postpone binding schema components to a namespace**
  - **Don’t give schema’s a targetNamespace, let schema’s that <include> your schema provide a targetNamespace**
  - **Postpone binding a type reference to an implementation,**
- Create Extensible Schemas
  - **You can’t see the future; that is you can not anticipate all the varieties of data that an instance document author might need to use**
  - **XML schemas are not going to be able to express all your business rules; use XSLT and/or Schematron**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

References

- http://www.xfront.com
- http://www.jenitennison.com

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

Additional Slides

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.

<include/> vs. <import/>

- <xs:include/>
  - **must occur before any other declarations/definitions**
  - **Included schema must have**
    - **the same targetNamespace or**
    - **no targetNamespace (chameleon-effect)**
- <xs:import/>
  - **must be used if multiple schemas are used**
    - **namespace attribute specifies namespace**

MOTOROLA

intelligence everywhere™

MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. © Motorola, Inc. 2002.