XML attribute vs XML element

Asked 16 years, 3 months ago Modified 1 year, 10 months ago Viewed 87k times

273

At work we are being asked to create XML files to pass data to another offline application that will then create a second XML file to pass back in order to update some of our data. During the process we have been discussing with the team of the other application about the structure of the XML file.

The sample I came up with is essentially something like:

```
<INVENTORY>
   <ITEM serialNumber="something" location="something" barcode="something">
      <TYPE modelNumber="something" vendor="something"/>
   </ITEM>
</INVENTORY>
```

The other team said that this was not industry standard and that attributes should only be used for meta data. They suggested:

```
<INVENTORY>
   <ITEM>
      <SERIALNUMBER>something</SERIALNUMBER>
      <LOCATION>something</LOCATION>
      <BARCODE>something</BARCODE>
      <TYPE>
         <MODELNUMBER>something</MODELNUMBER>
         <VENDOR>something</VENDOR>
      </TYPE>
   </ITEM>
</INVENTORY>
```

The reason I suggested the first is that the size of the file created is much smaller. There will be roughly 80000 items that will be in the file during transfer. Their suggestion in reality turns out to be three times larger than the one I suggested. I searched for the mysterious "Industry Standard" that was mentioned, but the closest I could find was that XML attributes should only be used for meta data, but said the debate was about what was actually meta data.

After the long winded explanation (sorry) how do you determine what is meta data, and when designing the structure of an XML document how should you decide when to use an attribute or an element?

xml xsd

Share

Improve this question

Follow

edited Feb 14, 2018 at 17:07

fredpi
8,942 ● 5 ● 44 ● 62

asked Aug 29, 2008 at 1:15

Jacob Schoen
14.2k ● 15 ● 86 ● 105

4 I found this really good resource: ibm.com/developerworks/xml/library/x-eleatt.html
– Laurens Holst Sep 27, 2010 at 10:59

10 +1 for *"...the debate was about **what** was actually meta data."* – Withheld Jul 24, 2013 at 14:00

Please note lowercase tag names with hyphens : stackoverflow.com/questions/1074447/…
– Ben Mar 8, 2015 at 17:04

Consider using s-expressions : defmacro.org/ramblings/lisp.html#part_6 :) – Alexey Dec 28, 2022 at 15:23

I'd keep the serial number as an attribute and make the rest elements. – Adam D. Sep 2, 2023 at 12:23

21 Answers

Sorted by: Highest score (default)

154

I use this rule of thumb:

1. An Attribute is something that is self-contained, i.e., a color, an ID, a name.
2. An Element is something that does or could have attributes of its own or contain other elements.

So yours is close. I would have done something like:

EDIT: Updated the original example based on feedback below.

```
<ITEM serialNumber="something">
    <BARCODE encoding="Code39">something</BARCODE>
    <LOCATION>XYX</LOCATION>
    <TYPE modelNumber="something">
       <VENDOR>YYZ</VENDOR>
    </TYPE>
 </ITEM>
```

Share

Improve this answer

Follow

edited Aug 29, 2011 at 13:14

CMircea
3,558 ● 2 ● 37 ● 58

answered Aug 29, 2008 at 1:28

user1921

23 I read through some of the answers and something that wasn't stressed enough form my experience is that if you data in an "attribute" and suddenly has a > or < you XML document will break I think there are five ascii chars (>, <, &, ?,") that will kill it. If this special character was in an Element you can simply add some CDATA tags around this data. I would say, only

use attributes when you 100% know what values are going to put in there, eg, a integer or a date, probably anything that is computer generated. If the BarCode was generated by a human then it should not be an attribute. – John Ballinger Jul 5, 2009 at 12:06

47 Really late to the party, but the special ASCII char argument is wrong -- that's what escaping is for, both for attributes and text data. – micahtan Nov 25, 2009 at 4:11

2 @donroby - Sorry, that would be my mistake in communicating. By escaping, I mean XML encoding. '<' = &lt; etc. It seems odd to me to decide between an attribute or element based on the characters that make up the content instead of the meaning of the content. – micahtan May 17, 2010 at 21:29

5 @donroby: it's incorrect. The replacement text of `&lt;` is `&#60;`, which is a character reference, not an entity reference. `&lt;` is OK in attributes. See: w3.org/TR/REC-xml/#sec-predefined-ent – porges Jun 24, 2010 at 4:10

17 @John: if this is a problem then there's something in your toolchain which isn't producing valid XML. I don't think this is a reason to choose between attributes or elements. (Furthermore, you can't "just add CDATA tags" around user-input because it might contain `]]>` !) – porges Jun 24, 2010 at 4:13

51

Some of the problems with attributes are:

- attributes cannot contain multiple values (child elements can)
- attributes are not easily expandable (for future changes)
- attributes cannot describe structures (child elements can)
- attributes are more difficult to manipulate by program code
- attribute values are not easy to test against a DTD

If you use attributes as containers for data, you end up with documents that are difficult to read and maintain. Try to use elements to describe data. Use attributes only to provide information that is not relevant to the data.

Don't end up like this (this is not how XML should be used):

```
<note day="12" month="11" year="2002"
      to="Tove" to2="John" from="Jani" heading="Reminder"
      body="Don't forget me this weekend!">
</note>
```

Source: http://www.w3schools.com/xml/xml_dtd_el_vs_attr.asp

Share
Improve this answer
Follow

edited Aug 28, 2018 at 8:05
gorn
5,310 ●7 ●32 ●47

answered Jan 7, 2009 at 12:49
user44350
529 ●5 ●5

2 First point is incorrect, see: w3.org/TR/xmlschema-2/#derivation-by-list – porges Jun 27, 2010 at 23:31

7 I'd say that first point is correct and `list` is a partial workaround to this problem. There can't be multiple attributes with same name. With `list` attribute still has only one value, which is a whitespace separated list of some datatypes. Separation characters are fixed so you cannot have multiple values if a single value of the wanted datatype can contain whitespace. This rules out the chances for having for example multiple addresses in one "address" attribute. – jasso Sep 5, 2010 at 1:49

9 'attributes are more difficult to manipulate by program code' - Can't agree with that one. In fact I've found the opposite to be true. It's not enough of a difference to really state either way. – Paul Alexander Mar 8, 2012 at 23:19

5 I'd also add that validation against a DTD isn't really relevant anymore, with XML-Schema, Schematron and Relax, et. al. all providing vastly more powerful and in some cases more intuitive ways of validating XML documents. Also, W3Schools is a really poor reference for anything – user764357 May 16, 2013 at 1:34

43

"XML" stands for "eXtensible *Markup* Language". A markup language implies that the data is text, *marked up* with metadata about structure or formatting.

XHTML is an example of XML used the way it was intended:

```
<p><span lang="es">El Jefe</span> insists that you
    <em class="urgent">MUST</em> complete your project by Friday.</p>
```

Here, the distinction between elements and attributes is clear. Text elements are displayed in the browser, and attributes are instructions about *how* to display them (although there are a few tags that don't work that way).

Confusion arises when XML is used not as a markup language, but as a *data serialization* language, in which the distinction between "data" and "metadata" is more vague. So the choice between elements and attributes is more-or-less arbitrary except for things that *can't* be represented with attributes (see feenster's answer).

Share Improve this answer Follow

answered Jun 24, 2010 at 4:02

dan04
90.9k ● 23 ● 168 ● 204

For me the distinction between an attribute and an element is very clear: the attribute allows to perform a processing while the element brings a semantic value to the data objects generated from the XML document. – gilaro Apr 13, 2022 at 4:51

1 @gilaro, could you give an example of an attribute then? Is `lang` or `locale` an attribute? Probably not, according to your distinction, as they affect the semantic value. – Alexey Dec

27, 2022 at 12:17

@Alexey: “Some informaticians maintain that attributes are for metadata about the element while elements are for the information itself. Other points out that’s not always obvious what’s the data and what’s metadata. Indeed, the answer may depend on where the information is put to use. [...] An element-based structure is a lot more flexible and extensible. Nonetheless, attributes are certainly more convenient in some applications. Ultimately, if you’re designing your own XML vocabulary, it’s up to you to decide when to use which.” — XML in a Nutshell (A Desktop Quick Reference). – gilaro Dec 28, 2022 at 14:28

@Alexey: So, I was partially wrong. – gilaro Dec 28, 2022 at 14:28

35

XML Element vs XML Attribute

XML is all about agreement. ***First defer to any existing XML schemas or established conventions within your community or industry.***

If you are truly in a situation to define your schema from the ground up, here are some general considerations that should inform the **element vs attribute decision**:

```
<versus>
  <element attribute="Meta content">
    Content
  </element>
  <element attribute="Flat">
    <parent>
      <child>Hierarchical</child>
    </parent>
  </element>
  <element attribute="Unordered">
    <ol>
      <li>Has</li>
      <li>order</li>
    </ol>
  </element>
  <element attribute="Must copy to reuse">
    Can reference to re-use
  </element>
  <element attribute="For software">
    For humans
  </element>
  <element attribute="Extreme use leads to micro-parsing">
    Extreme use leads to document bloat
  </element>
  <element attribute="Unique names">
    Unique or non-unique names
  </element>
  <element attribute="SAX parse: read first">
    SAX parse: read later
  </element>
  <element attribute="DTD: default value">
    DTD: no default value
  </element>
</versus>
```

Share Improve this answer Follow

answered Apr 17, 2014 at 12:56
kjhughes
111k ● 31 ● 194 ● 263

23

It may depend on your usage. XML that is used to represent stuctured data generated from a database may work well with ultimately field values being placed as attributes.

However XML used as a message transport would often be better using more elements.

For example lets say we had this XML as proposed in the answer:-

```
<INVENTORY>
   <ITEM serialNumber="something" barcode="something">
      <Location>XYX</LOCATION>
      <TYPE modelNumber="something">
         <VENDOR>YYZ</VENDOR>
      </TYPE>
    </ITEM>
</INVENTORY>
```

Now we want to send the ITEM element to a device to print he barcode however there is a choice of encoding types. How do we represent the encoding type required? Suddenly we realise, somewhat belatedly, that the barcode wasn't a single automic value but rather it may be qualified with the encoding required when printed.

```
   <ITEM serialNumber="something">
      <barcode encoding="Code39">something</barcode>
      <Location>XYX</LOCATION>
      <TYPE modelNumber="something">
         <VENDOR>YYZ</VENDOR>
      </TYPE>
   </ITEM>
```

The point is unless you building some kind of XSD or DTD along with a namespace to fix the structure in stone, you may be best served leaving your options open.

IMO XML is at its most useful when it can be flexed without breaking existing code using it.

Share Improve this answer Follow

answered Sep 30, 2008 at 10:24

AnthonyWJones
189k ● 35 ● 235 ● 307

Good point on the "barcode", I rushed my example and would have definitely broken that out into its own element. Also good point on the XSD/DTD. – user1921 Feb 23, 2009 at 21:26

12

I use the following guidelines in my schema design with regards to attributes vs. elements:

- Use elements for long running text (usually those of string or normalizedString types)
- Do not use an attribute if there is grouping of two values (e.g. eventStartDate and eventEndDate) for an element. In the previous example, there should be a new element for "event" which may contain the startDate and endDate attributes.
- Business Date, DateTime and numbers (e.g. counts, amount and rate) should be elements.
- Non-business time elements such as last updated, expires on should be attributes.
- Non-business numbers such as hash codes and indices should be attributes.* Use elements if the type will be complex.
- Use attributes if the value is a simple type and does not repeat.
- xml:id and xml:lang must be attributes referencing the XML schema
- Prefer attributes when technically possible.

The preference for attributes is it provides the following:

- unique (the attribute cannot appear multiple times)
- order does not matter
- the above properties are inheritable (this is something that the "all" content model does not support in the current schema language)
- bonus is they are less verbose and use up less bandwidth, but that's not really a reason to prefer attributes over elements.

I added *when technically possible* because there are times where the use of attributes are not possible. For example, attribute set choices. For example use (startDate and endDate) xor (startTS and endTS) is not possible with the current schema language

If XML Schema starts allowing the "all" content model to be restricted or extended then I would probably drop it

Share

Improve this answer

Follow

edited Dec 16, 2010 at 8:06

answered Dec 16, 2010 at 7:33

Archimedes Trajano

40.9k ● 24 ● 202 ● 336

10

There is no universal answer to this question (I was heavily involved in the creation of the W3C spec). XML can be used for many purposes - text-like documents, data and declarative code are three of the most common. I also use it a lot as a data model. There are aspects of these applications where attributes are more common and others where child elements are more natural. There are also features of various tools that make it easier or harder to use them.

XHTML is one area where attributes have a natural use (e.g. in class='foo'). Attributes have no order and this may make it easier for some people to develop tools. OTOH attributes are harder to type without a schema. I also find namespaced attributes (foo:bar="zork") are often harder to manage in various toolsets. But have a look at some of the W3C languages to see the mixture that is common. SVG, XSLT, XSD, MathML are some examples of well-known languages and all have a rich supply of attributes and elements. Some languages even allow more-than-one-way to do it, e.g.

```
<foo title="bar"/>;
```

or

```
<foo>
  <title>bar</title>;
</foo>;
```

Note that these are NOT equivalent syntactically and require explicit support in processing tools)

My advice would be to have a look at common practice in the area closest to your application and also consider what toolsets you may wish to apply.

Finally make sure that you differentiate namespaces from attributes. Some XML systems (e.g. Linq) represent namespaces as attributes in the API. IMO this is ugly and potentially confusing.

Share

Improve this answer

Follow

edited Jul 25, 2009 at 10:19

answered Jul 5, 2009 at 11:58

peter.murray.rust

38k ● 46 ● 159 ● 224

9

When in doubt, KISS -- why mix attributes and elements when you don't have a clear reason to use attributes. If you later decide to define an XSD, that will end up being cleaner as well. Then if you even later decide to generate a class structure from your XSD, that will be simpler as well.

Share Improve this answer Follow

answered Aug 29, 2008 at 1:27

Luke
19k ● 24 ● 88 ● 112

6

Others have covered how to differentiate between attributes from elements but from a more general perspective putting everything in attributes because it makes the resulting XML smaller is wrong.

XML is not designed to be compact but to be portable and human readable. If you want to decrease the size of the data in transit then use something else (such as google's protocol buffers).

Share
Improve this answer
Follow

edited Nov 10, 2009 at 17:11

answered Nov 10, 2009 at 16:43

Patrick
8,290 ● 7 ● 58 ● 72

Smaller XML text is more human-readable just because it's smaller! – Nashev Mar 13, 2018 at 10:51

5

the million dollar question!

first off, don't worry too much about performance now. you will be amazed at how quickly an optimized xml parser will rip through your xml. more importantly, what is your design for the future: as the XML evolves, how will you maintain loose coupling and interoperability?

more concretely, you can make the content model of an element more complex but it's harder to extend an attribute.

Share Improve this answer Follow

answered Aug 29, 2008 at 1:24

Adam
478 ● 1 ● 4 ● 7

5

Both methods for storing object's properties are perfectly valid. You should depart from pragmatic considerations. Try answering following question:

1. Which representation leads to faster data parsing\generation?
2. Which representation leads to faster data transfer?
3. Does readability matter?

...

Share
Improve this answer
Follow

edited Jun 20, 2020 at 9:12
Community Bot
1 ● 1

answered Aug 29, 2008 at 1:23
aku
124k ● 33 ● 176 ● 203

5

It's largely a matter of preference. I use Elements for grouping and attributes for data where possible as I see this as more compact than the alternative.

For example I prefer.....

```
<?xml version="1.0" encoding="utf-8"?>
<data>
    <people>
         <person name="Rory" surname="Becker" age="30" />
        <person name="Travis" surname="Illig" age="32" />
        <person name="Scott" surname="Hanselman" age="34" />
    </people>
</data>
```

...Instead of....

```
<?xml version="1.0" encoding="utf-8"?>
<data>
    <people>
        <person>
            <name>Rory</name>
            <surname>Becker</surname>
            <age>30</age>
        </person>
        <person>
            <name>Travis</name>
            <surname>Illig</surname>
            <age>32</age>
        </person>
        <person>
            <name>Scott</name>
            <surname>Hanselman</surname>
            <age>34</age>
        </person>
    </people>
</data>
```

However if I have data which does not represent easily inside of say 20-30 characters or contains many quotes or other characters that need escaping then I'd say it's time to break out the elements... possibly with CData blocks.

```
<?xml version="1.0" encoding="utf-8"?>
<data>
    <people>
        <person name="Rory" surname="Becker" age="30" >
```

```
            <comment>A programmer whose interested in all sorts of misc stuff.
His Blog can be found at http://rorybecker.blogspot.com and he's on twitter as
@RoryBecker</comment>
        </person>
        <person name="Travis" surname="Illig" age="32" >
            <comment>A cool guy for who has helped me out with all sorts of SVn
information</comment>
        </person>
        <person name="Scott" surname="Hanselman" age="34" >
            <comment>Scott works for MS and has a great podcast available at
http://www.hanselminutes.com </comment>
        </person>
    </people>
</data>
```

Share Improve this answer Follow

answered Sep 30, 2008 at 9:23

Rory Becker
15.7k ● 17 ● 72 ● 94

2 This is flat wrong I'm afraid - you should follow W3C guidelines: w3schools.com/DTD/dtd_el_vs_attr.asp - XML should not be formed on readability or on making it "compact" - but rather using elements or attributes correctly for the purpose which they were designed for. – Vidar Jan 7, 2009 at 12:59

27 I'm sorry, but this is misleading. The W3schools page is not W3C guidleines. The W3C XML recommendation (in which I was a participant) allows elements and attributes to be used according to the needs and styles of the users. – peter.murray.rust Jul 5, 2009 at 11:45

5

Use elements for data and attributes for meta data (data about the element's data).

If an element is showing up as a predicate in your select strings, you have a good sign that it should be an attribute. Likewise if an attribute never is used as a predicate, then maybe it is not useful meta data.

Remember that XML is supposed to be machine readable not human readable and for large documents XML compresses very well.

Share Improve this answer Follow

answered Jan 15, 2009 at 19:22

Michael J
2,523 ● 2 ● 23 ● 24

4

It is arguable either way, but your colleagues are right in the sense that the XML should be used for "markup" or meta-data around the actual data. For your part, you are right in that it's sometimes hard to decide where the line between meta-data and data is when modeling your domain in XML. In practice, what I do is pretend that anything in the markup is hidden, and only the data outside the markup is readable. Does the document make some sense in that way?

XML is notoriously bulky. For transport and storage, compression is highly recommended if you can afford the processing power. XML compresses well, sometimes phenomenally well, because of its repetitiveness. I've had large files compress to less than 5% of their original size.

Another point to bolster your position is that while the other team is arguing about style (in that most XML tools will handle an all-attribute document just as easily as an all-#PCDATA document) you are arguing practicalities. While style can't be totally ignored, technical merits should carry more weight.

Share Improve this answer Follow

answered Aug 29, 2008 at 1:26

erickson
269k ● 59 ● 401 ● 497

4

How about taking advantage of our hard earned object orientation intuition? I usually find it is straight forward to think which is an object and which is an attribute of the object or which object it is referring to.

Whichever intuitively make sense as objects shall fit in as elements. Its attributes (or properties) would be attributes for these elements in xml or child element with attribute.

I think for simpler cases like in the example object orientation analogy works okay to figure out which is element and which is attribute of an element.

Share Improve this answer Follow

answered Feb 9, 2011 at 13:08

rpattabi
10.2k ● 5 ● 48 ● 55

2

Just a couple of corrections to some bad info:

@John Ballinger: Attributies can contain any character data. < > & " ' need to be escaped to &lt; &gt; &amp; &quot; and &apos; , respectively. If you use an XML library, it will take care of that for you.

Hell, an attribute can contain binary data such as an image, if you really want, just by base64-encoding it and making it a data: URL.

@feenster: Attributes can contain space-separated multiple items in the case of IDS or NAMES, which would include numbers. Nitpicky, but this can end up saving space.

Using attributes can keep XML competitive with JSON. See *Fat Markup: Trimming the Fat Markup Myth one calorie at a time*.

Share
Improve this answer
Follow

edited Oct 7, 2013 at 23:28

answered Jul 23, 2009 at 21:38
brianary
9,332 ● 2 ● 38 ● 30

Not just ids or names. They can contain space-separated lists of just about anything. – John Saunders Jul 25, 2009 at 11:02

@JohnSaunders IDS or NAMES are specific DTD types (XML Schema too, I think), supported at a low level by most XML processors. If handled by the application layer instead of the XML libraries, any kind of character data works (separated values or whatever). – brianary Jul 24, 2013 at 20:12

Personally, just because you can doesn't mean you should. – user692942 Oct 2, 2013 at 9:14

1 @Lankymart As I said, I was just correcting some incorrect info (that was scoring high for some reason). Binary data doesn't usually belong in XML at all. – brianary Oct 7, 2013 at 23:43

1

This is very clear in HTML where the differences of attributes and markup can be clearly seen:

1. All data is between markup
2. Attributes are used to characterize this data (e.g. formats)

If you just have pure data as XML, there is a less clear difference. Data could stand between markup or as attributes.

=> Most data should stand between markup.

If you want to use attributes here: You could divide data into two categories: Data and "meta data", where meta data is not part of the record, you want to present, but things like "format version", "created date", etc.

```
<customer format="">
     <name></name>
     ...
</customer>
```

One could also say: "Use attributes to characterize the tag, use tags to provide data itself."

Share Improve this answer Follow

answered Aug 31, 2011 at 2:44

Walter A. Jablonowski
349 ● 1 ● 4 ● 13

1

I am always surprised by the results of these kinds of discussions. To me there is a very simple rule for deciding whether data belongs in an attribute or as content and that is whether the data has navigable sub-structure.

So for example, non-markup text always belongs in attributes. Always.

Lists belong in sub-structure or content. Text which may over time include embedded structured sub-content belong in content. (In my experience there is relatively little of this - text with markup - when using XML for data storage or exchange.)

XML schema written this way is concise.

Whenever I see cases like `<car><make>Ford</make><color>Red</color></car>`, I think to myself "gee did the author think that there were going to be sub-elements within the make element?" `<car make="Ford" color="Red" />` is significantly more readable, there's no question about how whitespace would be handled etc.

Given just but the whitespace handling rules, I believe this was the clear intent of the XML designers.

Share Improve this answer Follow

answered May 15, 2015 at 17:12
MGrier
19 ● 2

one of the few explanations I can read. have no idea whether or not it's a good idea...but at least I understand the point ;) – Thufir Jan 10, 2019 at 0:37

1 `<color roof="black" doors="red"/>` – Michael Kay Jan 26, 2022 at 11:23

0

The clear and unambiguous definition of an XML element is everything from (including) the element's start tag to (including) the element's end tag.

The below example is an element with text and a child element. The element's name is This_Is_An_Element. It's contents are the open and close tag and all the stuff in between, including any attributes, child elements, etc. And sub_element is also an element, but it has no contents aside from it's tag.

```
<This_Is_An_Element>and this is clear text <sub_element/> etc.
</This_Is_An_Element>
```

And, an attribute is member of an element. Here, This_Is_An_Element has an attribute, WithAnAttribute. And that attribute's value is, Attribute's Value. This attribute is part of the element, This_Is_An_Element.

```
<This_Is_An_Element WithAnAttribute="Attribute's Value">and this is clear text
<sub_element> etc. </This_Is_An_Element>
```

Share

Improve this answer

Follow

edited Feb 20, 2023 at 0:43

answered Feb 20, 2023 at 0:33

Display name

1,522 ● 1 ● 21 ● 32

-1

I agree with feenster. Stay away from attributes if you can. Elements are evolution friendly and more interoperable between web service toolkits. You'd never find these toolkits serializing your request/response messages using attributes. This also makes sense since our messages are data (not metadata) for a web service toolkit.

Share Improve this answer Follow

answered Apr 7, 2009 at 6:22

ottodidakt

3,741 ● 4 ● 30 ● 34

-1

Attributes can easily become difficult to manage over time trust me. i always stay away from them personally. Elements are far more explicit and readable/usable by both parsers and users.

Only time i've ever used them was to define the file extension of an asset url:

```
<image type="gif">wank.jpg</image> ...etc etc
```

i guess if you know 100% the attribute will not need to be expanded you could use them, but how many times do you know that.

```
<image>
  <url>wank.jpg</url>
  <fileType>gif</fileType>
</image>
```

Share Improve this answer Follow

answered Apr 8, 2011 at 19:51

oh pot

7 ● 1