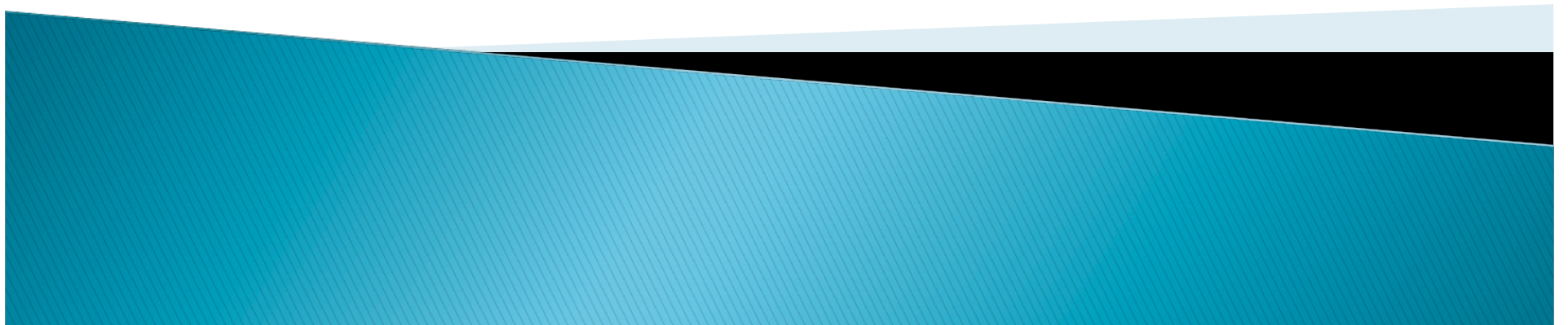


Justifying XML

“That software which is flexible, simple, sloppy, tolerant, and altogether forgiving of human foibles and weaknesses turns out to be actually the most steel cored, able to survive and grow while that software which is demanding, abstract, rich but systematized, turns out to collapse in on itself in a slow and grim implosion.”
– Adam Bosworth



Some words on HTML

- ▶ “When HTML first came out it was unbelievably sloppy and forgiving, permissive and ambiguous.”
- ▶ Microsoft Office Team, 1995: “HTML would never succeed because it was so primitive and that Word would win because Word documents were rich and controlled in their layout.”



The Great War

“There is an unacknowledged war that goes on every day in the world of programming. It is a war between the humans and the computer scientists...



Pragmatism

Html
success

Xml
success

...It is a war between those who want simple, sloppy, flexible, human ways to write code and those who want clean, crisp, clear, correct ways to write code.” – Adam Bosworth

Extensible Markup Language (XML):

- ▶ Extends HTML's ability to decouple data from content
- ▶ Self-Describing
- ▶ Malleable
- ▶ Been around since 1986 (as SGML)

```
<recipe name="bread" prep_time="5 mins" cook_time="3 hours">
  <title>Basic bread</title>
  <ingredient amount="8" unit="dL">Flour</ingredient>
  <ingredient amount="10" unit="grams">Yeast</ingredient>
  <ingredient amount="4" unit="dL" state="warm">Water</ingredient>
  <ingredient amount="1" unit="teaspoon">Salt</ingredient>
  <instructions>
    <step>Mix all ingredients together.</step>
    <step>Knead thoroughly.</step>
    <step>Cover with a cloth, and leave for one hour in warm room.</step>
    <step>Knead again.</step>
    <step>Place in a bread baking tin.</step>
    <step>Cover with a cloth, and leave for one hour in warm room.</step>
    <step>Bake in the oven at 180(degrees)C for 30 minutes.</step>
  </instructions>
</recipe>
```

Image from <http://en.wikipedia.org/wiki/XML>

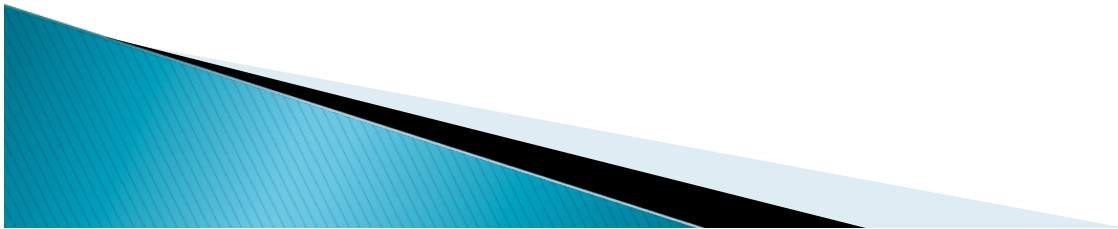
The Good

- ▶ Everywhere
- ▶ Easy to parse
- ▶ Enables AJAX via XMLHttpRequest
- ▶ Platform-independent
- ▶ Can represent most of the basic CS Data Structures (thus, xml serialization)



The Bad

- ▶ Everywhere
- ▶ Syntax can be unnecessarily verbose
- ▶ Files can be unnecessarily large
- ▶ Encourages abuse of XML in favor of creating custom, optimal layouts.
- ▶ Often completely unnecessary, but used anyway.



The Ugly

```
<attachments xmlns = "http://webservices..." >  
  <bytes>37</bytes>  
  <bytes>80</bytes>  
  <bytes>68</bytes>  
  <bytes>70</bytes>  
  <bytes>45</bytes>  
  <bytes>49</bytes>  
  <bytes>46</bytes>  
  <bytes>52</bytes>  
  <bytes>10</bytes>  
  <bytes>37</bytes>  
  <bytes>-30</bytes>  
  <bytes>-29</bytes>  
  <bytes>-49</bytes>  
  <bytes>-45</bytes>  
  <bytes>10</bytes>  
  <bytes>52</bytes>  
  <bytes>32</bytes>  
  <bytes>48</bytes>  
  <bytes>32</bytes>  
  <bytes>111</bytes>  
  ...
```

Source: <http://thedailywtf.com/Articles/Oh,-XML.aspx>



Thoughts?



See:

- ▶ **Adam Bosworth at International Conference on Service Oriented Computing (ICSOC) '04**
 - <http://adambosworth.net/2004/11/18/iscoc04-talk/>
- ▶ <http://en.wikipedia.org/wiki/Xml>

