

IN DEFENCE OF STYLE GUIDES

BALISAGE 2018

Ari Nordström | ari.nordstrom@gmail.com





The semantics should be obvious



The semantics are, um, less than clear

Let's have a look at some tags with similar semantics...

A common way to write a list is

```
<para>Here are my favourite films:  
  <list>  
    <item>Close Encounters of the Third Kind</item>  
    <item>2001</item>  
    <item>Amadeus</item>  
  </list>  
</para>
```


How I read it

Here are my favourite films: Close Encounters of the Third Kind, 2001, and Amadeus.

What they (usually) mean

Here are my favourite films:

- Close Encounters of the Third Kind
- 2001
- Amadeus

Just as many write

```
<para>Here are my favourite films:</para>
<list>
  <item>Close Encounters of the Third Kind</item>
  <item>2001</item>
  <item>Amadeus</item>
</list>
```

...but this is poor semantics

...see, we actually want this

```
<!ELEMENT list (para, item+)>
```


But what we get is frequently more like this

```
<!ELEMENT list (title?, (note | warning | graphic |  
table | group | titled-block)*,  
(item | bridge | graphic | table | group |  
titled-block)+)>  
<!ATTLIST list type (bullet | number | alpha) #REQUIRED  
wall-size (small | medium | large) #IMPLIED  
weird-number (1 | 2 | 3) #IMPLIED  
random-useless-attr CDATA #IMPLIED  
usage (yes | maybe | no) #REQUIRED  
qtulhu-index CDATA #IMPLIED  
fake-news CDATA #REQUIRED  
did-you-read-this-far CDATA #IMPLIED >
```

they put them in because somebody somewhere did,
for some reason

or because it looked pretty

But sometimes it is simply about looking pretty...



...or is it?

```
<core:para-grp>
  <core:desig value="17">17.</core:desig>
  <core:title>General financial arrangements.</core:title>

  <core:para>The following are to be paid out ...:</core:para>
  <core:list type="number">
    <core:listitem>...</core:listitem>
    ...
  </core:list>
  ...
  <core:para>The following are to be paid out ...:</core:para>
  <core:list type="lower-alpha">
    <core:listitem>...</core:listitem>
    ...
  </core:list>
  ...
</core:para-grp>
```

I looked up the DTD...

```
<!ATTLIST core:list
    type ( bullet | check-box | lower-alpha |
           lower-roman | mdash | ndash | number |
           plain | upper-alpha | upper-roman |
           upper-alpha-alpha | lower-alpha-alpha |
           smallcaps-alpha-alpha )
    #REQUIRED
    restart (yes | no) 'yes'
    source-pnum CDATA #IMPLIED
    %display-atts;
    lni CDATA #IMPLIED
>
```

...this is actually as it should be

This, however, is not

Follow these steps:

- *Do this.*
- *Then do this.*
- *And finally do this.*

Why is one list OK but the other is not?

Follow these steps:

- *Do this.*
- *Then do this.*
- *And finally do this.*

It's not an unordered list

It's not an ordered list

It's a **procedure**

The bullets are a misguided attempt at looking pretty.

How did this come about?

We analysed docs

We created a DTD that describes them

We (hopefully) identified the various list types

*But I'm betting nobody described how or when to use
any of it.*

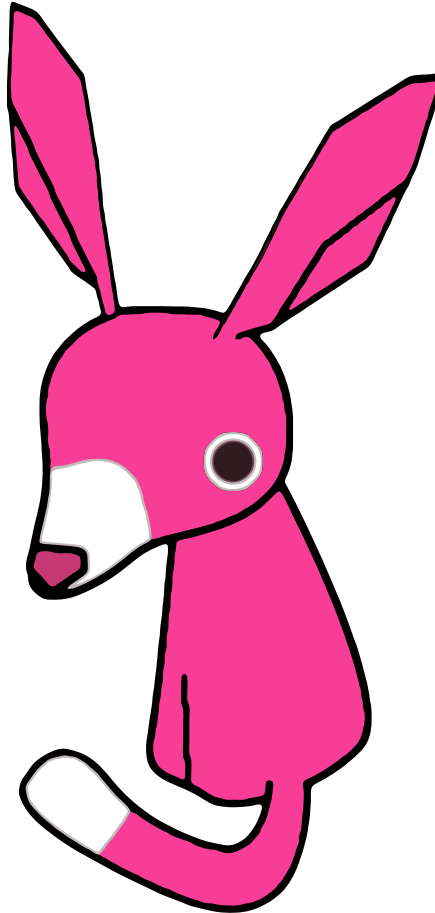
It could be that it really is about looking pretty.

It could be that the analysis didn't discover the why.

Or it could be that different writers do the same thing
differently.

Or, conversely, that this writer simply did what the
previous writer did.

Schematron? Schematron Quick Fixes?



Where do the schematron patterns come from?

A house style

In the olden days, a house style was described by a style guide



What is a style guide?

Think of it as a poet's schema

Think of it as the basis for Schematron rules

For example, remember those list variants?

Always introduce a list with a paragraph that explains the list's purpose. The introductory paragraph is not a title; rather, it is a qualifier, giving the list its proper context.

Never use an ordered list when you are writing a procedure (and don't even consider writing it using an unordered list).

Never insert a list or its introductory text inside a paragraph unless you intend to present your list inline.

(etc)

nce
upon
a
time...



THE ELEMENTS
OF STYLE

(illustrated)



STRUNK / WHITE / KALMAN

The **Chicago**
Manual of
Style
16

SIXTEENTH EDITION

The Essential Guide for
Writers, Editors, and Publishers

Oxford
Style
Manual

The
essential
handbook for
all writers
and editors

2.6.3 Numbered (Ordered) Lists

Use a numbered (ordered) list to present items that must be considered in a certain sequence or that have a definite priority. Such a list must be presented in the order of its performance or priority (first step or highest priority first). Avoid sublevels within a numbered list. If sublists must be used, the levels are the same as for bullet lists.

Example 6 A Numbered List with Unordered Sublists

1. Appearance of numbered lists
 - Second level
 - Introduced with a dash

Paragraphs in a System Administration Manual should be kept short.

Third person must be used in descriptive chapters.

Contractions must not be used.

Second person imperative is required in step-by-step descriptions in procedural chapters. Note, however, that step responses and comments are written in third person, as they depict system responses or clarify steps or responses.

Procedure steps must be kept short and include only one action (for example, one command to be typed or one mouse action) and one system response. More than one comment for each step is allowed, although discouraged. Uniform division of a procedure into steps is a difficult matter and requires the use of test results and editorial advice.

Note: Procedure names should use gerund, that is, the “-ing” form of verbs. For example, a procedure for recovering from a disc crash should be named “Recovering From a Disc Crash”, not “Disc Crash Recovery”.

6.7.1.1 Section 8.n <Procedure>

Use a separate level two section for each procedure listed in the introduction to the System Integration Procedures chapter. The procedure name in the section heading must be identical to the procedure name in the listing in the introduction.

Note: The level two section heading must be forced to the top of a page to make the beginning of the procedures easy to locate.

Introduce every procedure with a short paragraph that describes what the procedure will accomplish.

Then split the level two section into two level three sections, Requirements and Procedure.

Section 8.n.1, Requirements

Begin the Requirements section by describing the conditions that apply before the integration procedure at hand. If some special hardware or software is required, then mention this. If the procedure at hand requires another procedure to be performed first, then state this and include a cross-reference to that procedure.

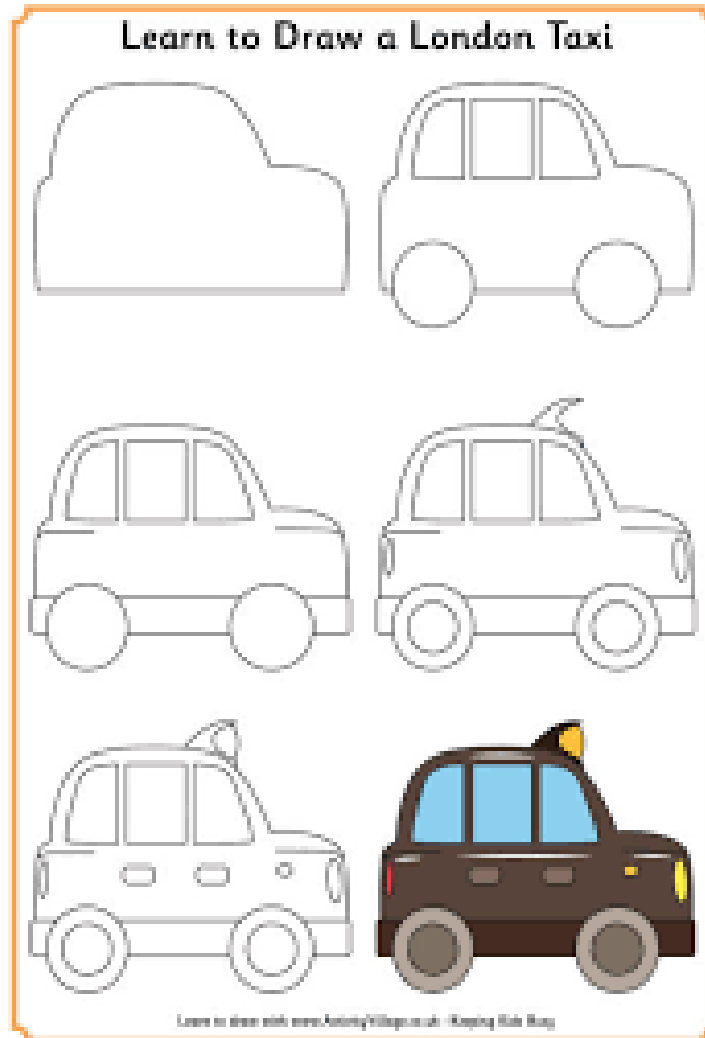
Section 8.n.2, Procedure

In the other level three section, Procedure, describe the procedure itself, step by step, using one action and one response per step.

If the procedure section is lengthy, divide it into level four sections. Name these sections according to their contents. The numbering of procedure steps should be continuous throughout the entire level three section, that is, throughout the entire procedure.

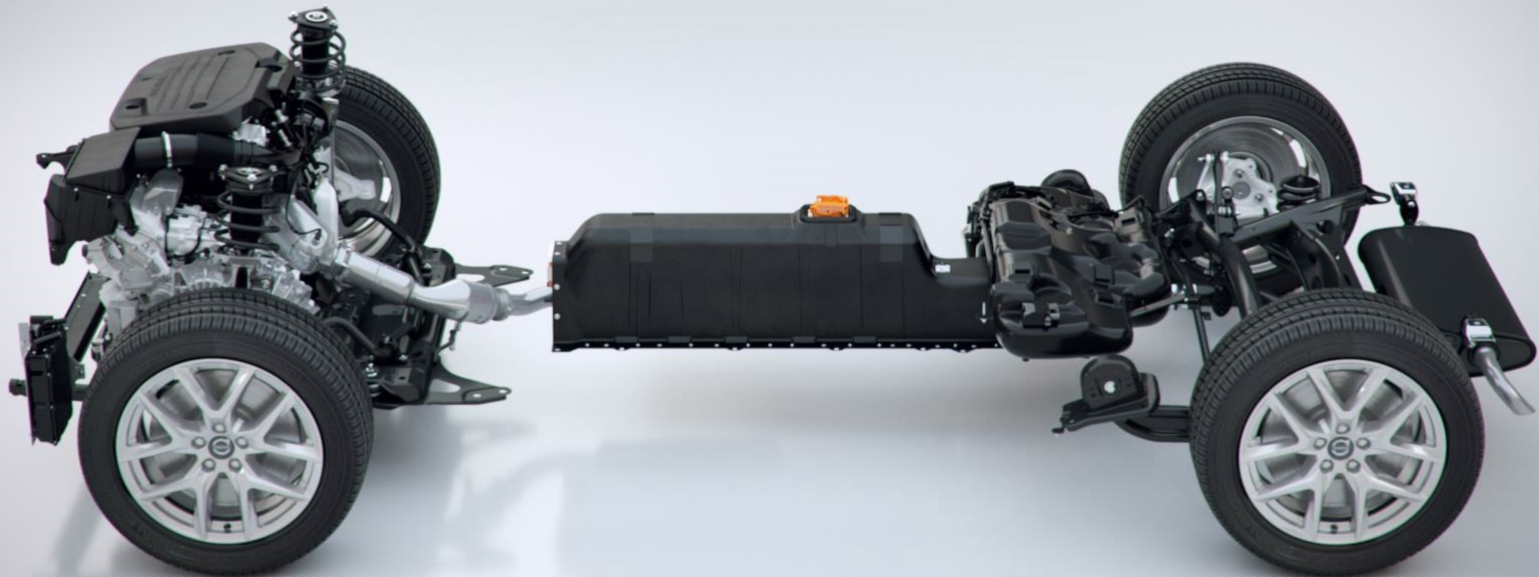
OK, so my lists may not be perfect, but do I actually
need all THAT?

A Cautionary Tale



Sharing documentation

THE NEXT GENERATION COMPACT PLATFORM



OK, you've convinced me, how do I get one?

Ideally...

Info analysis -> Style guide

Info analysis -> DTD

Start writing



In practice we're already writing...

New info analysis

Look at existing docs, schemas...

Look at common Schematron errors...

Define new house style (and kill your darlings)



What does that give us?

Separation of concerns

Schema checks structure, Schematron the styles
defined in a style guide

...allowing the schema to remain small

About roles

Tell me, do you employ editors?



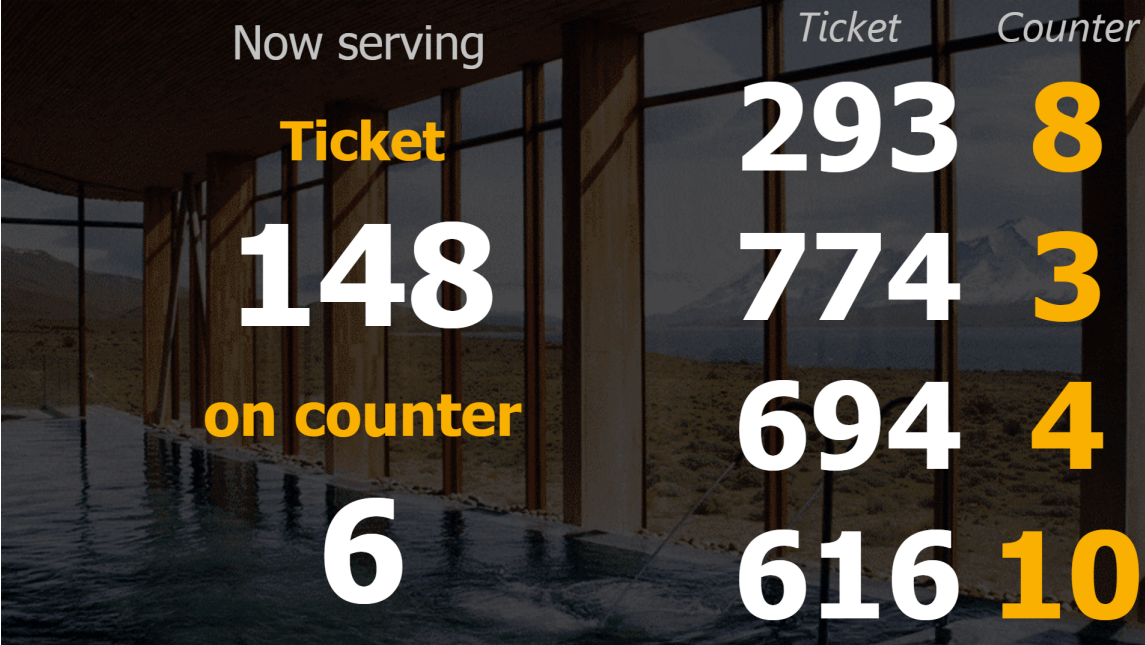
What does an editor do?

An editor, of course, is the person who makes sure that everyone follows the style guide, is the final arbiter of all things style, and frequently the one who edits the style guide.

The Schematron should check rules defined in a style guide, not define it.

A Schematron checks content after the fact; the style guide allows content to be created that *avoids* the errors to begin with.

Queues reinvented



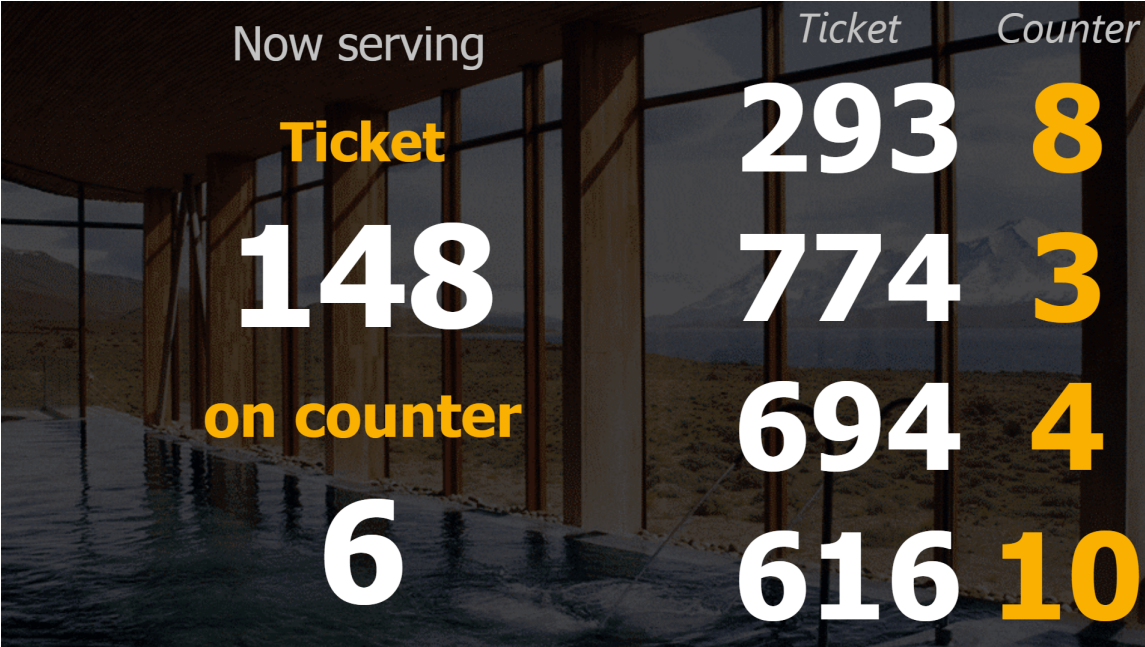
Now serving	<i>Ticket</i>	<i>Counter</i>
Ticket	293	8
148	774	3
on counter	694	4
6	616	10

Simplifies the model

Separation of concerns

Not buying this. Where's the style guide in that thing?

Compare this...



Now serving	<i>Ticket</i>	<i>Counter</i>
Ticket	293	8
148	774	3
on counter	694	4
6	616	10

...with this:

```
148 (6)
293 (8)
774 (3)
694 (4)
616 (10)
102 (9)
X (5)
602 (2)
X (7)
X (1)
```


- *Don't display any unmanned counters.*
- *Show the latest update in a larger font.*
- *Limit the number of counters shown to five.*
- ...

*An Englishman, even if he is alone,
forms an orderly queue of one*

- George Mikes

Thank you