MSc in Official Statistics
Statistical Computing:
XML and Design

Andrew Westlake

Survey & Statistical Computing
63 Ridge Road, London N8 9NP, UK
+44 (0) 20 8374 4723
AJW@SaSC.co.uk (E-Mail)
www.SaSC.co.uk

XML – eXtensible Markup Language

- Markup Language
  - » Text with Tags (<**Field**> field contents </**Field**>)
    - Identifies an Element of type Field with content field contents
  - » Content of an element can be simple or complex
    - Numbers, strings, etc., or combinations of other elements
  - » Nested Tags (elements) => multiple hierarchies
- Generic syntax for languages
  - » Tags not defined, only the language structure
- XML is a Standard from W3C
  - » Generic tools to read and write XML
    - Interface tools for application developers
    - Presentation tools, style sheets

An XML Fragment

```
<variable ident = "5" type = "quantity“>
  <name>Q5</name>
  <label>Miles travelled</label>
  <position start = "43" finish = "45"/>
  <values>
     <range from = "1" to = "499"/>
     <value code = "500">500 or more</value>
     <value code = "999">Not stated</value>
  </values>
</variable>
```

XML and Abstraction

- Level 2 – the XML specification
  - » Generic rules for XML document instances
- Level 1 – structures for specific applications
  - » DDI, SDMX, triple-S, defined through a Schema
- Level 0 – XML documents
  - » Actual instances of information
  - » Can be displayed and manipulated using generic tools based on level 2 specifications
  - » Needs level 1 specification to understand the information and display in context

Why is XML Important

- XML is plain text
- An XML document can represent a complex information structure
- Software (APIs) is readily available to read an XML document into an internal object structure (and to write to an XML document) and to check validity
- ➢ XML documents are an ideal **medium** for the exchange of complex information structures between systems
  - Solves the plumbing problem of transmission
- Example from Statmodel

XML as a Statistical Interchange Format

- Use XML to exchange Meta-Data, eg DDI
  - » Can include the description of actual data files
- Probably don't use XML for case (micro) data
  - » Existing methods such as CDF, ODBC adequate
  - » Triple-s includes Data
- Might be useful for aggregate (macro) data
  - » SDMX
- Exchange of XML document **files** adequate in many situations
- Can use message protocols containing XML where dynamic interchange is needed
  - » SOAP, WSDL, UDDI, etc, as used for Web Services

Defining XML Structure

- **Well-formed** XML obeys syntax rules, but can contain any structure
- **Valid** XML obeys rules about the specific tags and structures allowed in a specific context
    - XSD – XML Schema Definition
        - Strong data typing for simple elements
        - Clear declarations for complex structures
        - Limited to strict hierarchies
        - An XSD is an XML document – uses Namespaces
    - DTD - Document Type Definition
        - Traditional declaration, from SGML
        - Similar capabilities to XSD, but less data typing
        - Not an XML document

Related Technologies

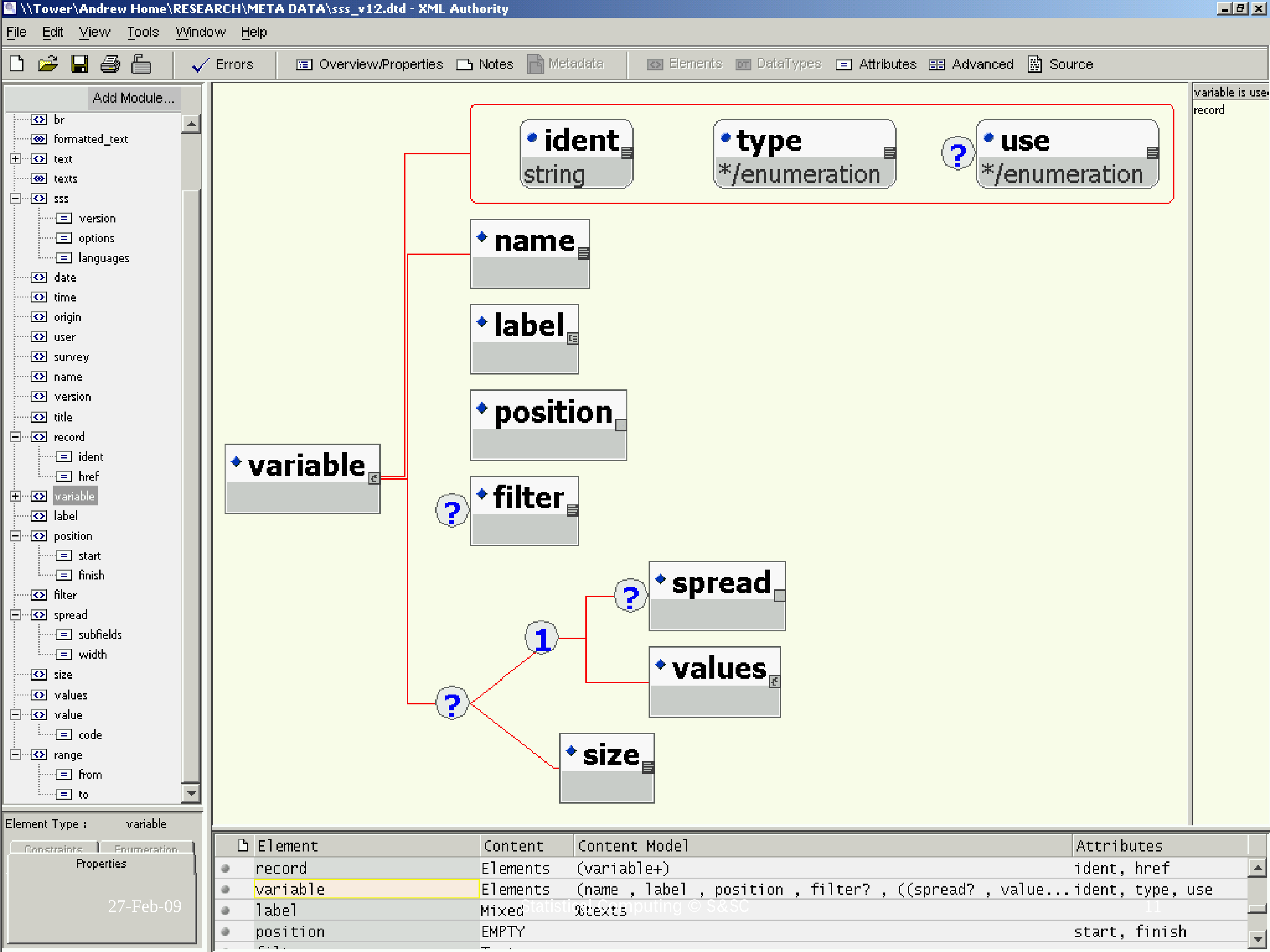
- All at level 2
- Namespace
  - » Mechanism for referring to standard XML definitions
  - » Avoids name duplication problems
- XSL – Extensible Style sheet Language
  - » Transformation and Processing system for XML documents, widely supported
  - » Provides views of selected components from structure
  - » Can produce reformatted listings (eg Text, or HTML)
  - » Can convert one XML structure to another
- XLink, XPath, XQuery, XPointer
  - » Systems for navigating within XML structures

XSL Transformations

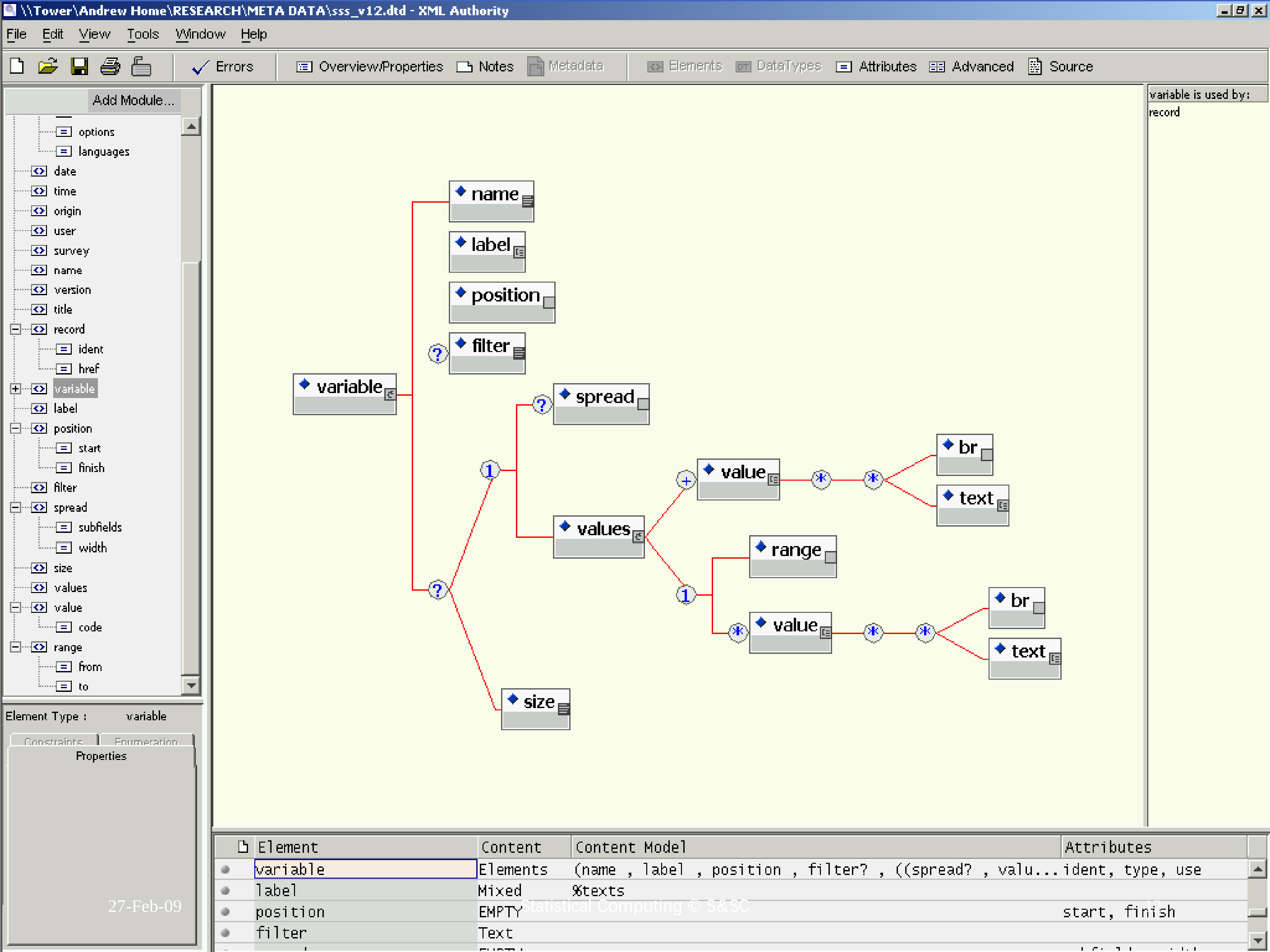
- Important because it gives us a quick way to view information from an XML document in different ways, according to the requirements of the context
- Rather like Views in a relational database
- Generally an application can do better, by knowing more about the ideal way to present selected information (level 1, semantics)
- An XSL file is an XML document

Designing XML structures

- Can use text editor to write XSD or DTD
- Various XML editors that check for well-formedness and validity
- Some systems build the structure graphically, then generate DTD or XSD
- Best approach is to model the information system, then convert the necessary parts of this to an XSD or DTD to support interchange



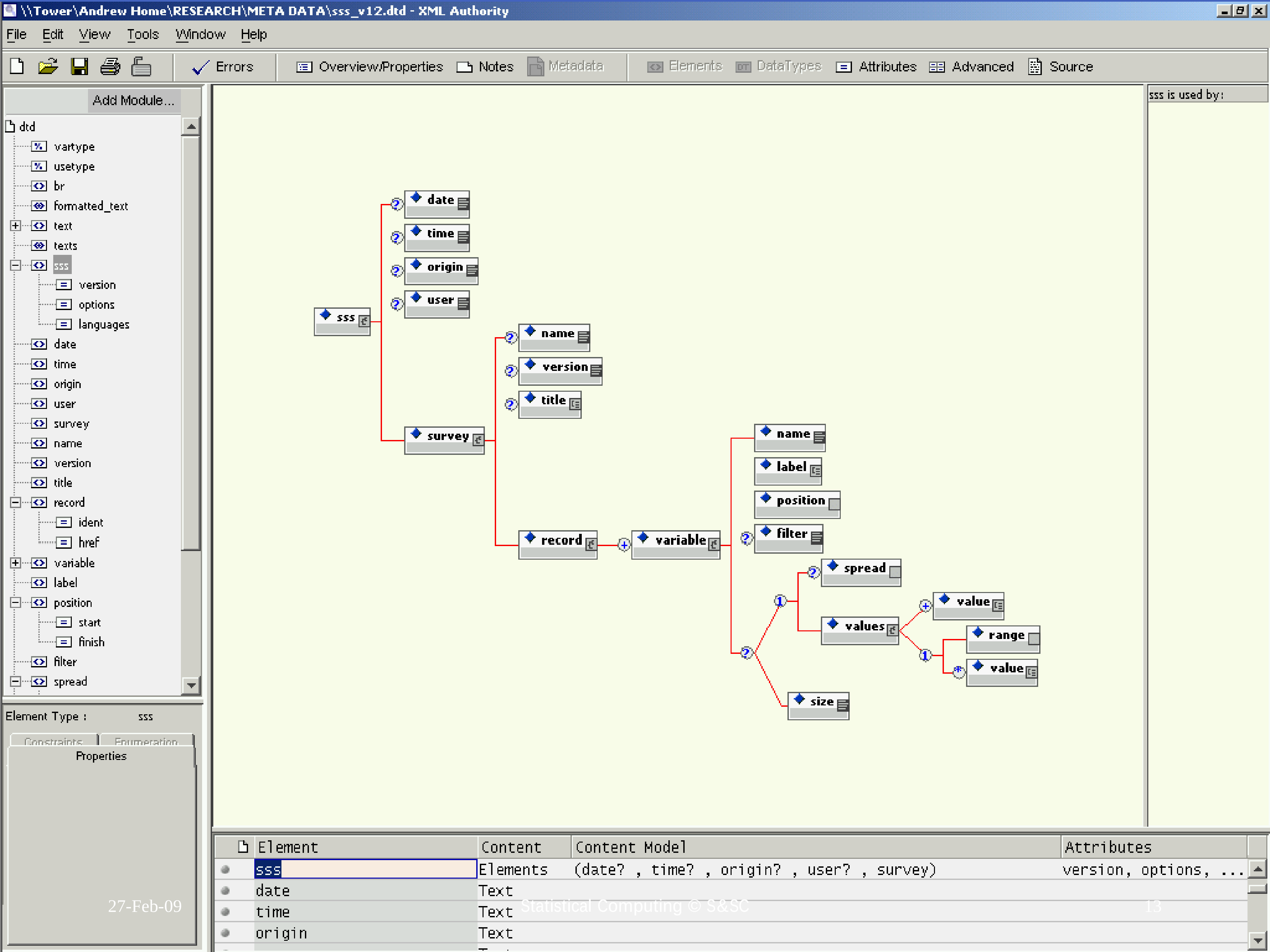
	Element	Content	Content Model	Attributes
●	record	Elements	(variable+)	ident, href
●	variable	Elements	(name , label , position , filter? , ((spread? , value...	ident, type, use
●	label	Mixed	%texts	
●	position	EMPTY		start, finish



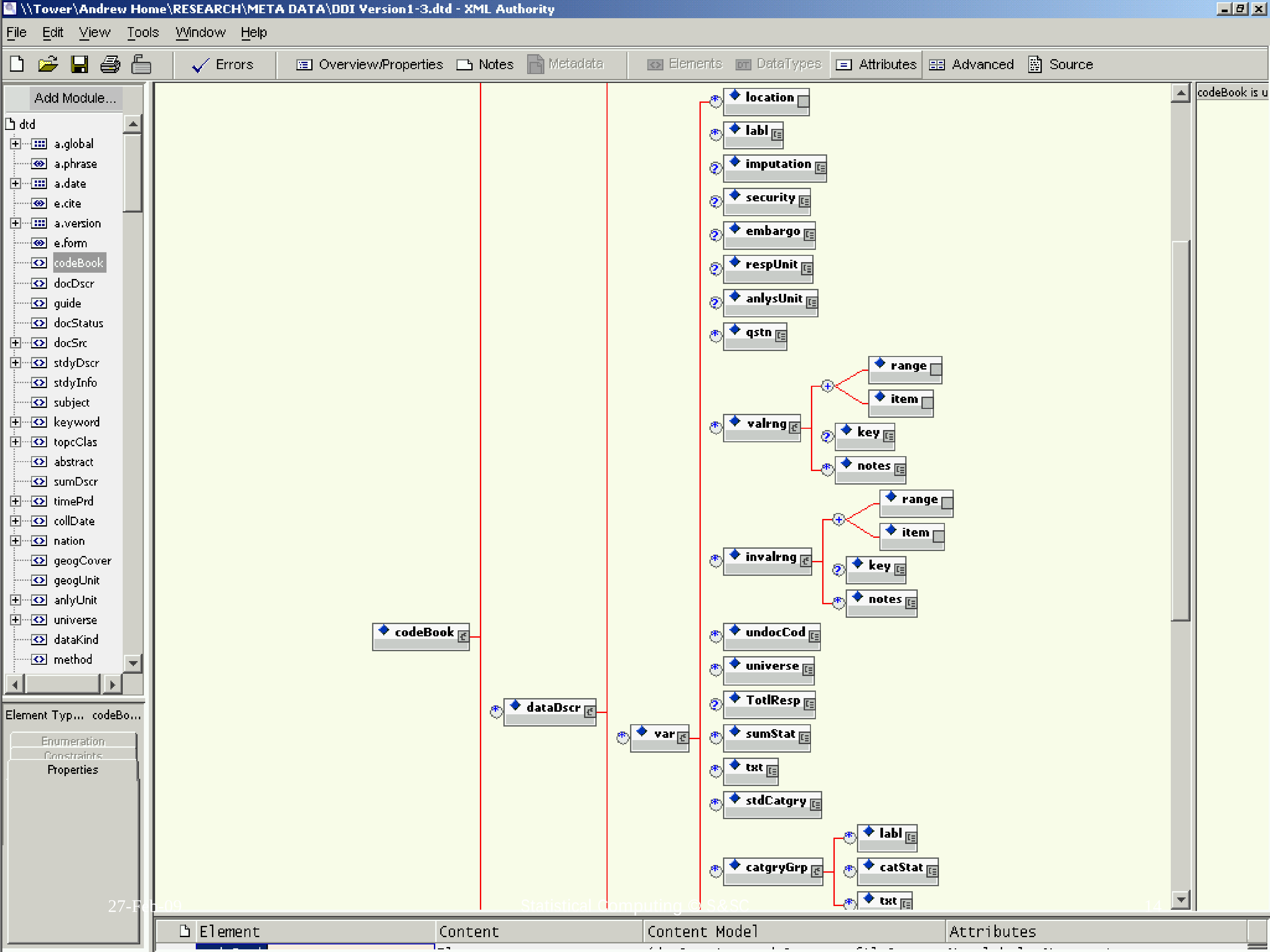
variable is used by:
record

Element Type : variable

Element	Content	Content Model	Attributes
variable	Elements	(name , label , position , filter? , ((spread? , valu...	ident, type, use
label	Mixed	%texts	
position	EMPTY		start, finish
filter	Text		



Element	Content	Content Model	Attributes
sss	Elements	(date? , time? , origin? , user? , survey)	version, options, ...
date	Text		
time	Text		
origin	Text		


\\Tower\Andrew Home\RESEARCH\META DATA\DDI Version1-3.dtd - XML Authority
File
Edit
View
Tools
Window
Help
Errors
Overview/Properties
Notes
Metadata
Elements
DataTypes
Attributes
Advanced
Source
Add Module...
dtd
a.global
a.phrase
a.date
e.cite
a.version
e.form
codeBook
docDscr
guide
docStatus
docSrc
stdyDscr
stdyInfo
subject
keyword
topcClas
abstract
sumDscr
timePrd
collDate
nation
geogCover
geogUnit
anlyUnit
universe
dataKind
method
Element Typ...
codeBo...
Enumeration
Constraints
Properties
codeBook is u
location
labl
imputation
security
embargo
respUnit
anlysUnit
qstn
range
item
valrng
key
notes
range
item
invalrng
key
notes
codeBook
undocCod
universe
TotlResp
dataDscr
var
sumStat
txt
stdCatgry
labl
catgryGrp
catStat
txt
Element
Content
Content Model
Attributes

Limitations of XML

- Cannot express semantics, only structure
  - Can have Comments in DTD, or Annotations in XSD, but these have to be read by the implementer or user, they cannot be enforced directly
- Limited to hierarchical structures
  - Adequate for simple structures
  - Need many-to-many links in many contexts
  - Can be overcome by using references, but the semantics have to be enforced by the applications, not generic tools
  - XLink proposal (generalised hyperlinks) may solve this

Recommendations for Standards

- Use XML as exchange format for information structures (MetaData)
- XSD (or DTD) is a necessary but not sufficient specification of the model for information structures
- Create a model for the information structure in UML
  - » Include all the semantics
  - » Generate the XML interchange specification (XSD or DTD) from the model
  - » Use the model to build interchange functionality into application software