What About XML?

Marc Balcer
ThoughtWorks

ThoughtWorks

Why this talk?

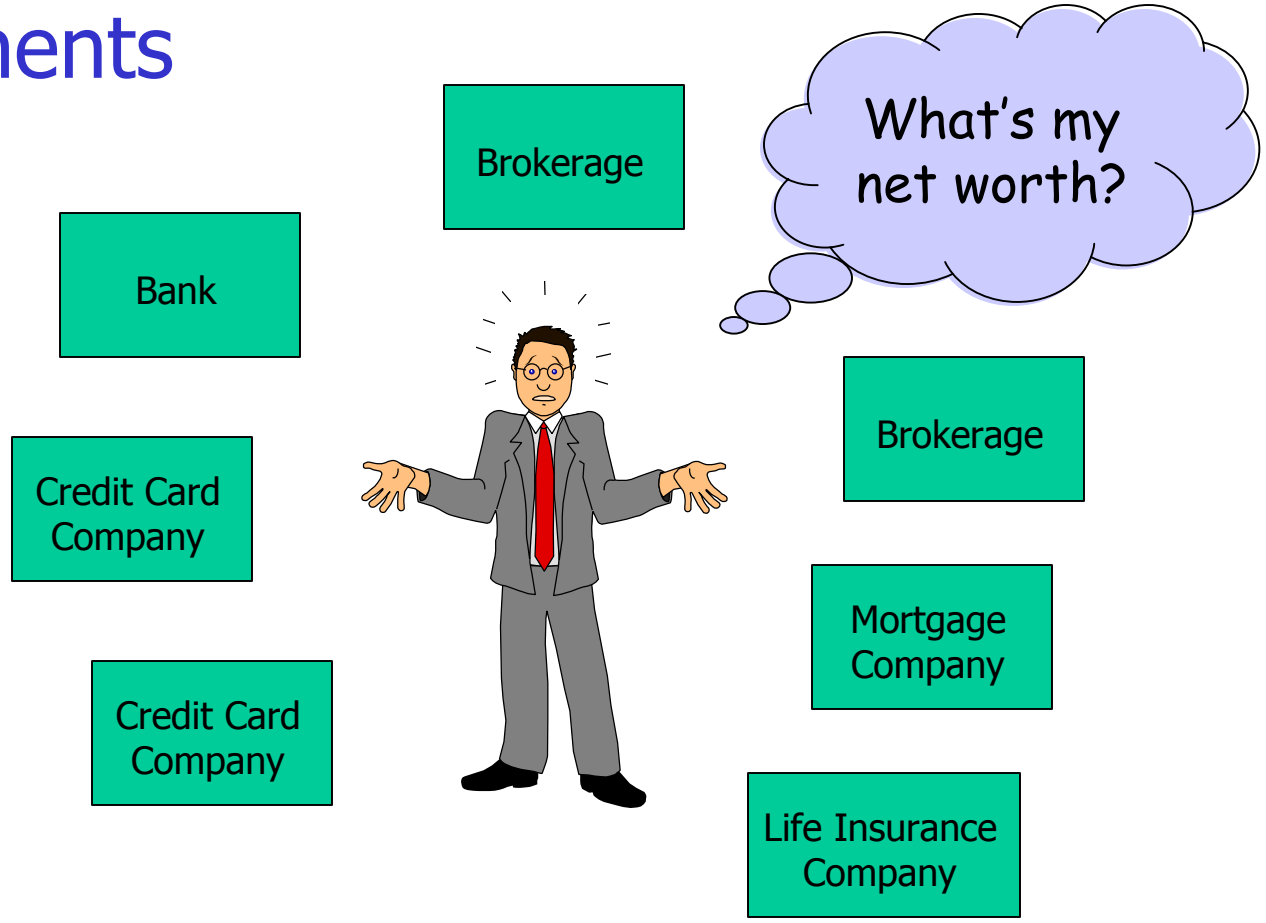
- XML is all over the news
- Be fully buzzword compliant
- Another reason to be SMUG —
  we've had "the answers" all along
- See how our sense of RD can help

Overview

- Why XML?
- Domains and XML
- Creating an XML Document
- Transforming XML
- Case Study

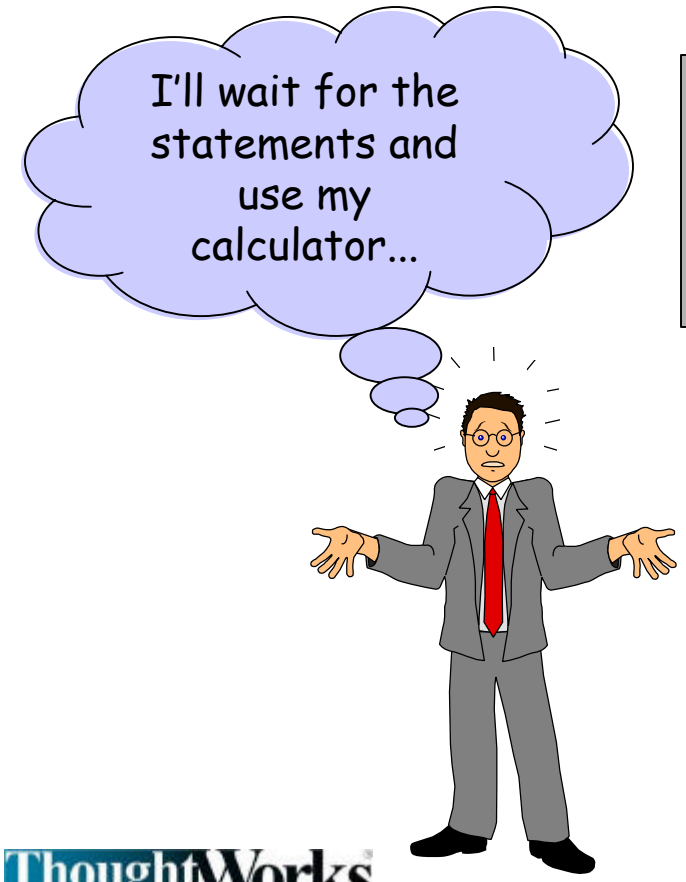
Why XML?

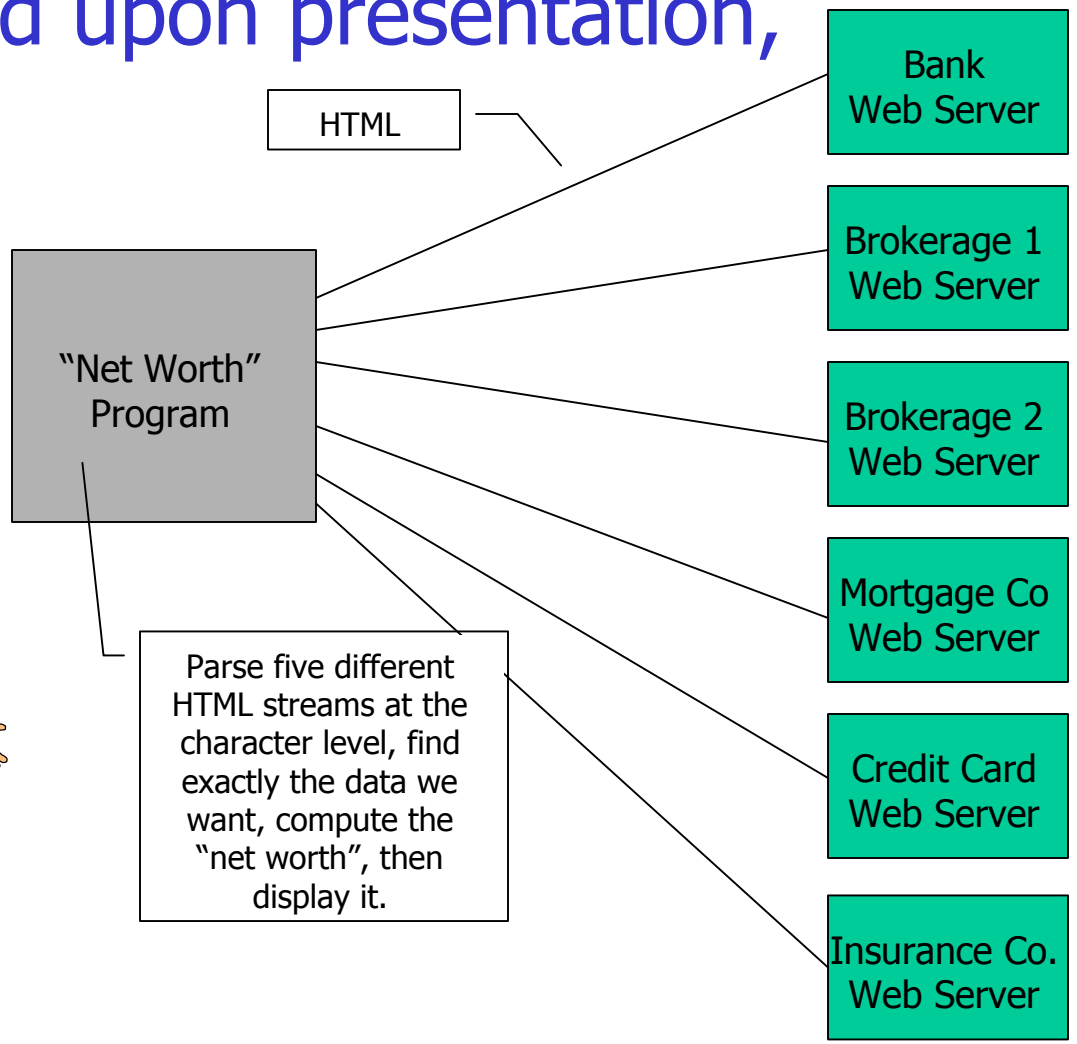
- Data exchange between heterogeneous components



What a contraption!

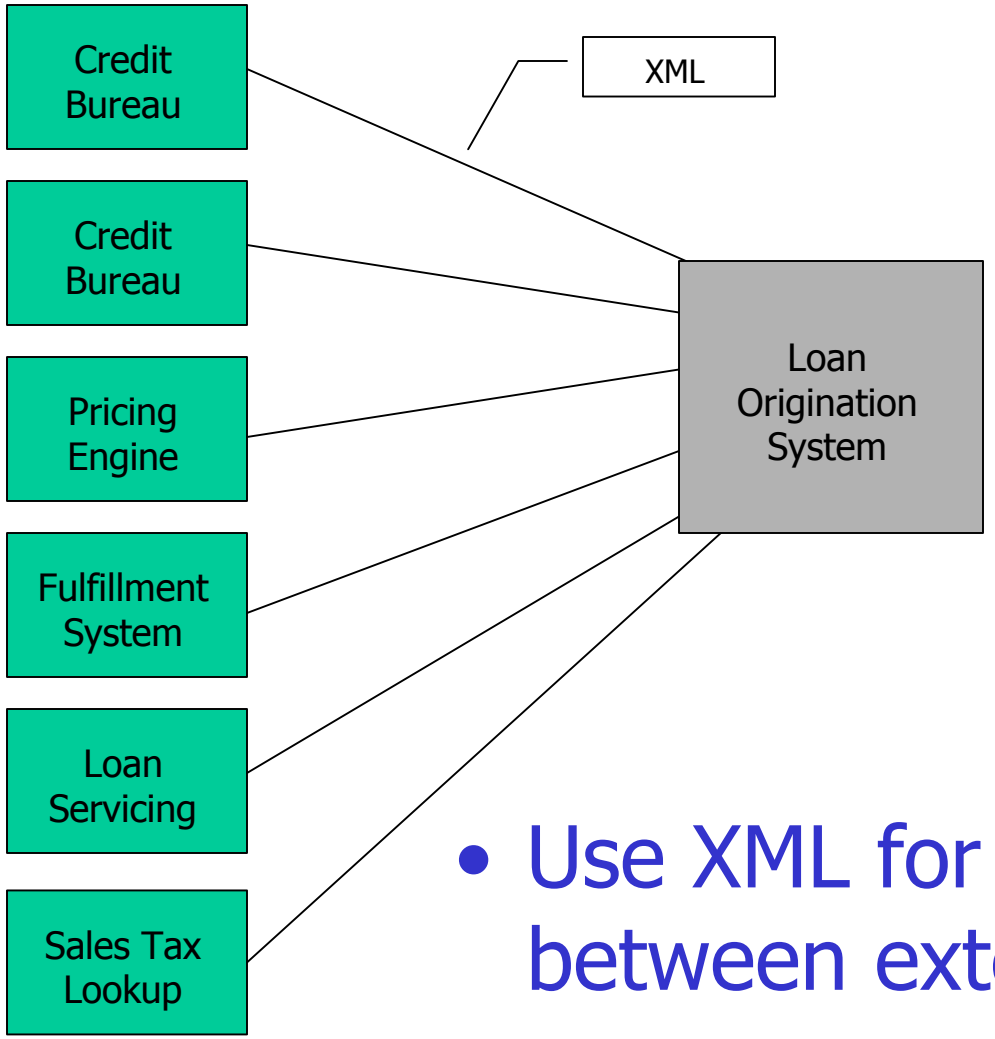
- HTML is focused upon presentation, not content





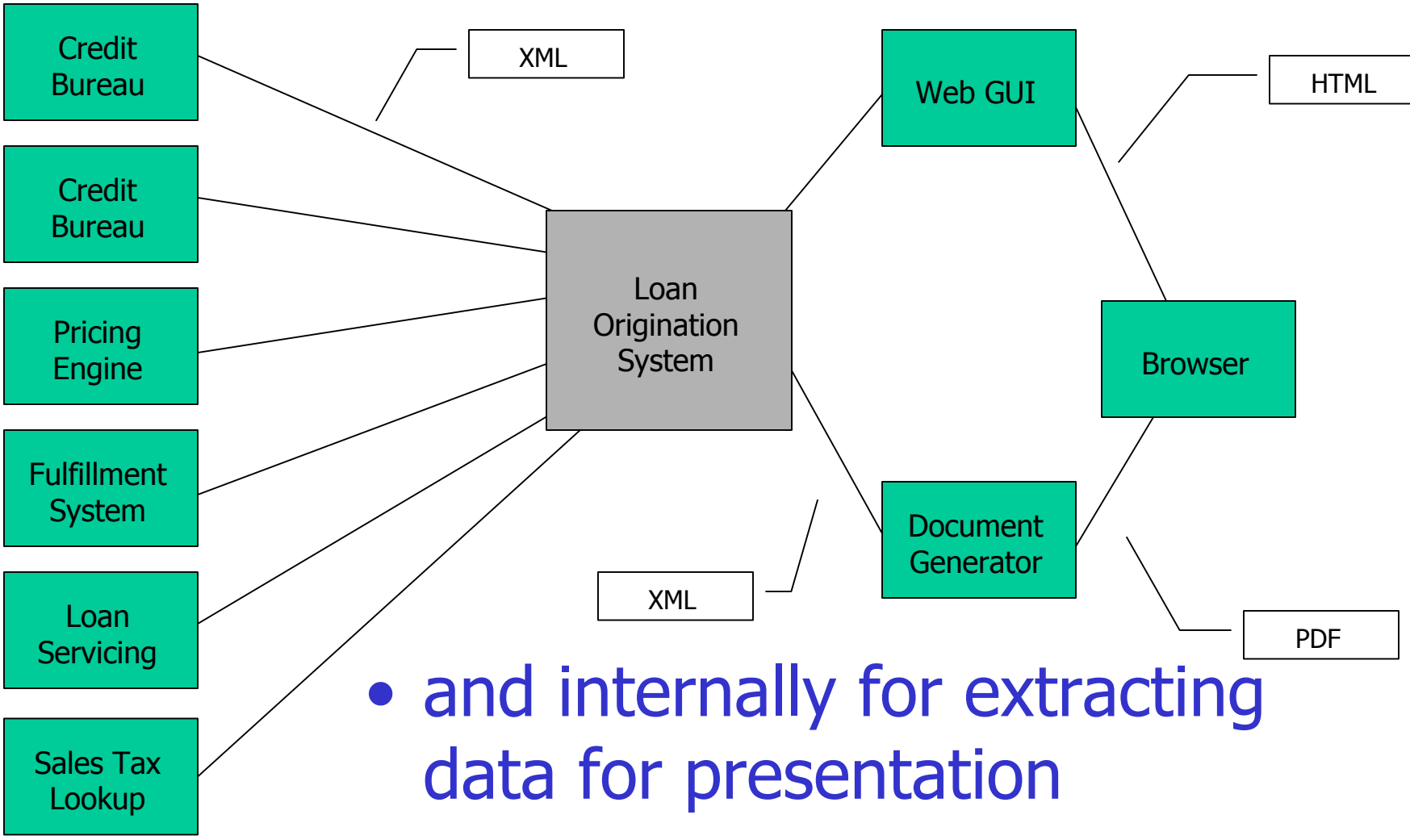
ThoughtWorks

XML in Practice



- Use XML for exchanging data between external systems

XML in Practice



- and internally for extracting data for presentation

HTML - Presentation

- HTML tags - presentation
  - set is fixed
  - goal: direct display in a web browser

Student Schedule

Student	Brown, Charlie
Term	Fall 2000

Department	Course#	Course Name	Days	Time
Comp Sci	202	Object-Oriented Analysis	MWF	10-11 a.m.
PE	316	Baseball	TTh	3-6 p.m.

```
<HTML>
<HEAD>
<TITLE>Student Schedule</TITLE>
<BODY>
<H3>Student Schedule</H3>
<TABLE border=1 cellPadding=1 cellSpacing=1>
  <TR>
     <TH align="left">Student</TH>
     <TD>Brown, Charlie</TD></TR>
  <TR>
     <TH align="left">Term</TH>
     <TD>Fall 2000</TD></TR>
</TABLE>
<TABLE border=1 cellPadding=1 cellSpacing=1>
  <TR>
    <TH>Department</TH>
    <TH>Course#</TH>
    <TH>Course Name</TH>
    <TH>Days</TH>
    <TH>Time</TH></TR>
  <TR>
    <TD>Comp Sci</TD>
    <TD>202</TD>
    <TD>Object-Oriented Analysis</TD>
    <TD>MWF</TD>
    <TD>10-11 a.m.</TD></TR>
```

XML - Content

- XML tags
  - set defined by the type of data being displayed
  - goal: describe structured data

```
<?xml version="1.0">
<Schedule Term="Spring 2000"
          Student="Brown, Charlie" >
  <Enrollment Department="Comp Sci"
              CourseNum="202"
              Title="Object-Oriented Analysis"
              Days="MWF"
              Times="10-11 a.m."/>
  <Enrollment Department="PE"
              CourseNum="316"
              Title="Baseball"
              Days="TTh"
              Times="3-6 p.m."/>
</Schedule>
```

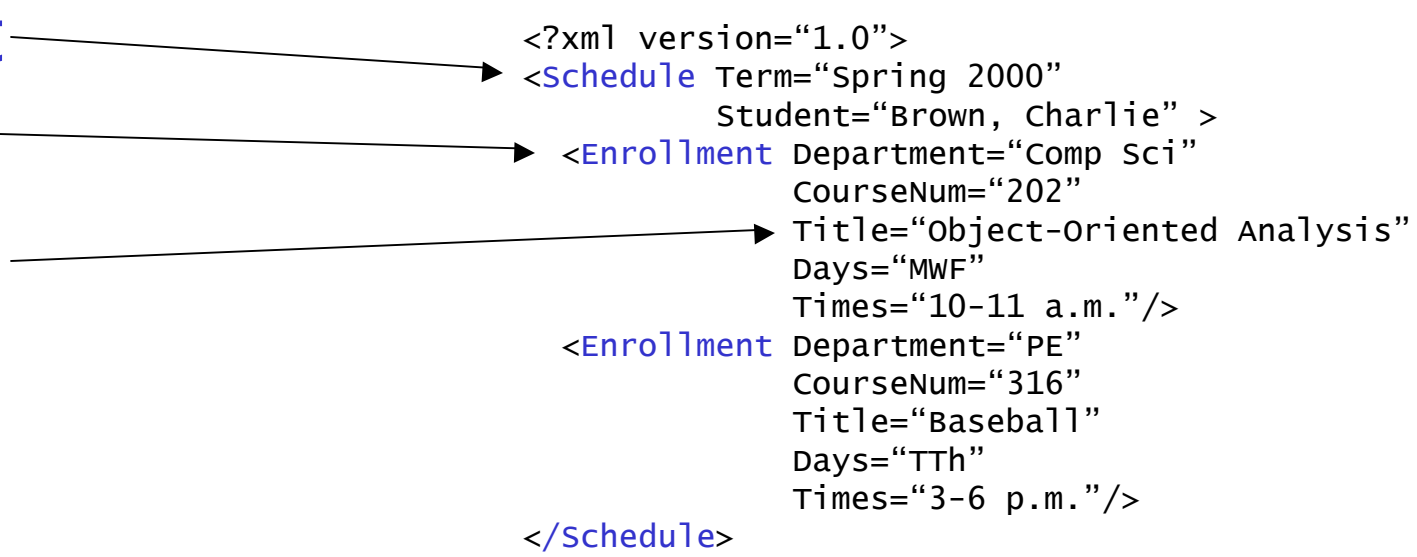
Student Schedule

Student	Brown, Charlie
Term	Fall 2000

Department	Course#	Course Name	Days	Time
Comp Sci	202	Object-Oriented Analysis	MWF	10-11 a.m.
PE	316	Baseball	TTh	3-6 p.m.

XML concepts

- Document
- Element
- Attributes



```
<?xml version="1.0">
<Schedule Term="Spring 2000"
          Student="Brown, Charlie" >
  <Enrollment Department="Comp Sci"
              CourseNum="202"
              Title="Object-Oriented Analysis"
              Days="MWF"
              Times="10-11 a.m."/>
  <Enrollment Department="PE"
              CourseNum="316"
              Title="Baseball"
              Days="TTh"
              Times="3-6 p.m."/>
</Schedule>
```

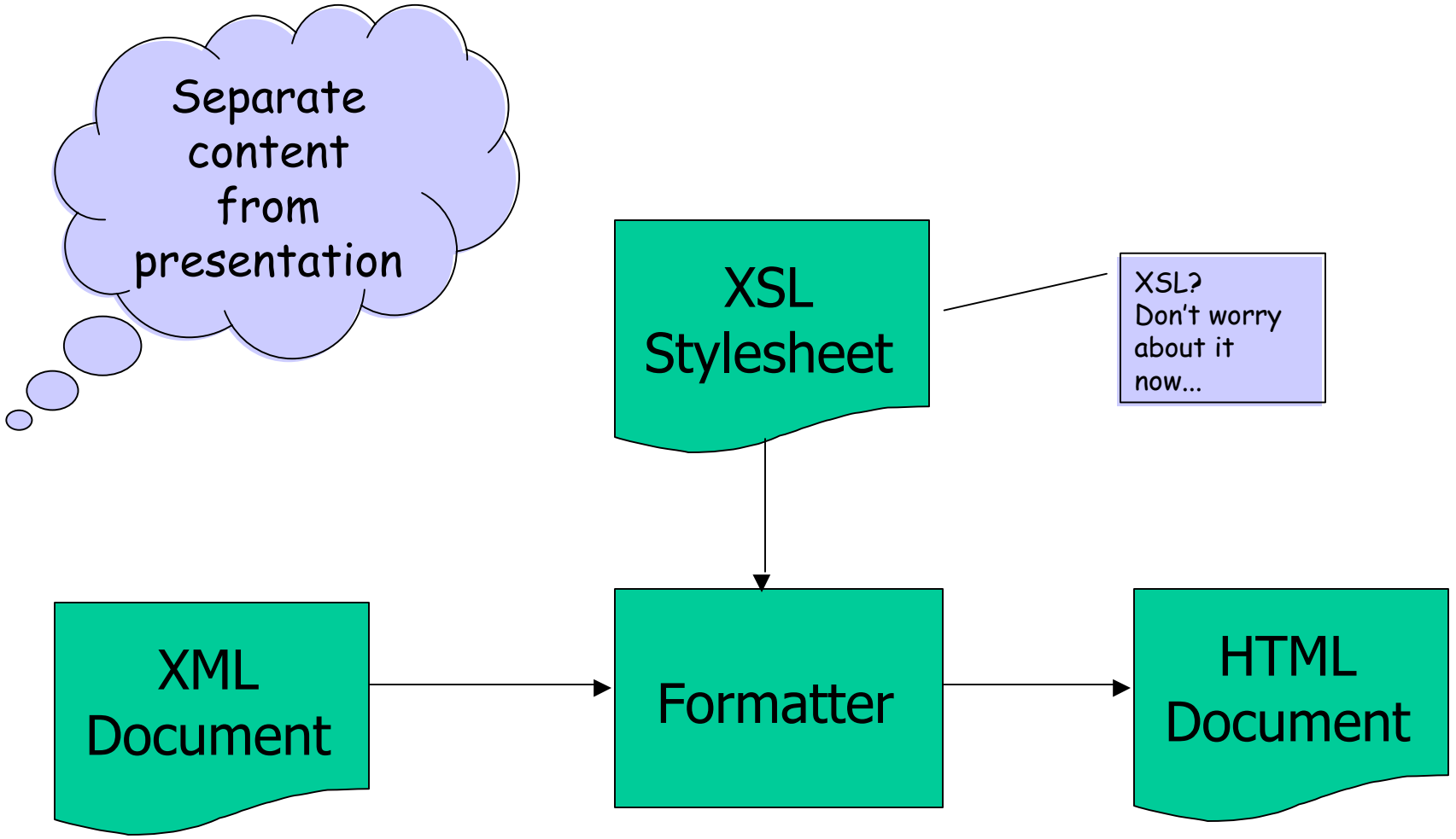
HTML and XML

- Properly formed HTML is XML (remember this…!)

```
<HTML>
<HEAD>
<TITLE>Student Schedule</TITLE>
<BODY>
<H3>Student Schedule</H3>
<TABLE border=1 cellPadding=1 cellSpacing=1>
  <TR>
     <TH align="left">Student</TH>
     <TD>Brown, Charlie</TD></TR>
  <TR>
     <TH align="left">Term</TH>
     <TD>Fall 2000</TD></TR>
</TABLE>
<TABLE border=1 cellPadding=1 cellSpacing=1>
  <TR>
    <TH>Department</TH>
    <TH>Course#</TH>
    <TH>Course Name</TH>
    <TH>Days</TH>
    <TH>Time</TH></TR>
  <TR>
    <TD>Comp Sci</TD>
    <TD>202</TD>
```

```
<?xml version="1.0">
<Schedule Term="Spring 2000"
          Student="Brown, Charlie" >
  <Enrollment Department="Comp Sci"
              CourseNum="202"
              Title="Object-Oriented Analysis"
              Days="MWF"
              Times="10-11 a.m."/>
  <Enrollment Department="PE"
              CourseNum="316"
              Title="Baseball"
              Days="TTh"
              Times="3-6 p.m."/>
</Schedule>
```

Classic XML example

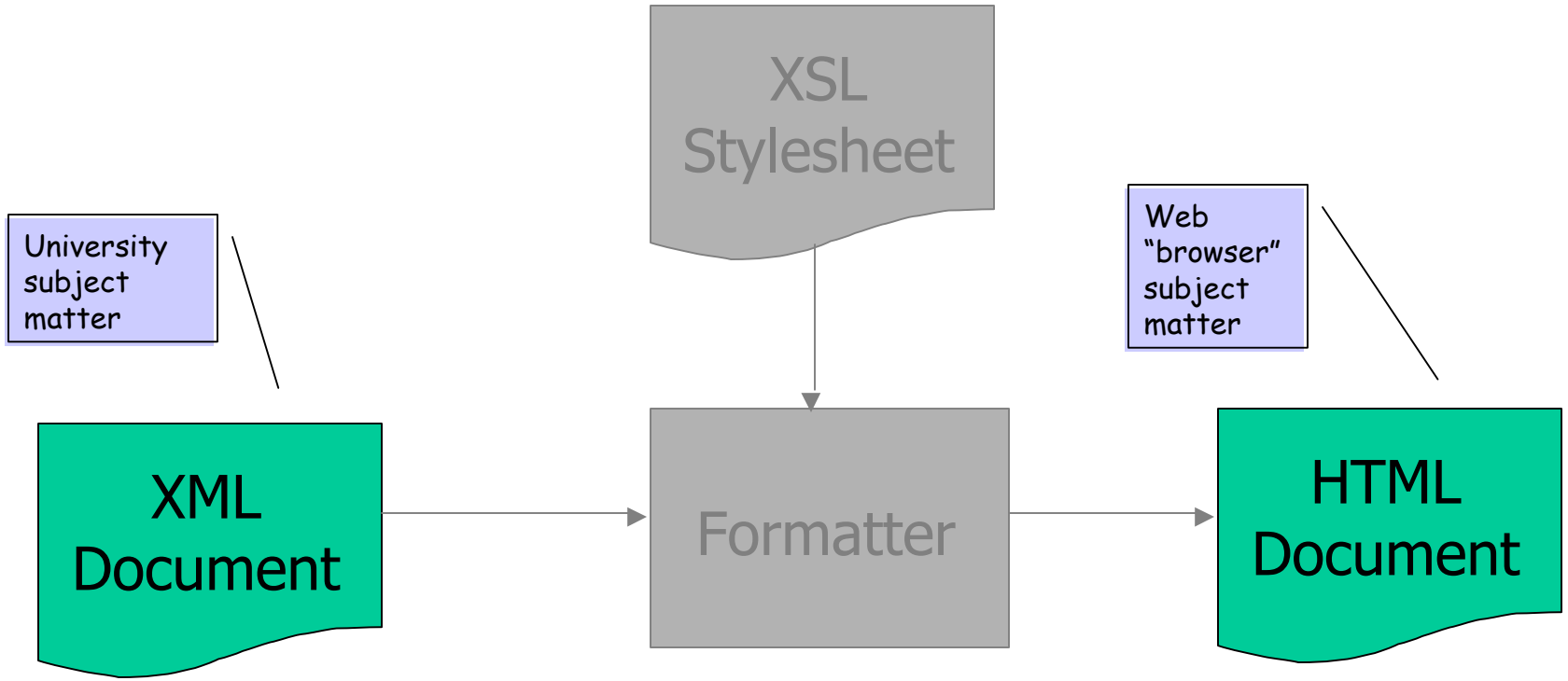


Classic XML Example

- XML document tags (elements & attributes)
  - Student, Course, Enrollment
- HTML document tags
  - H1, H2, Table, TR, TH, TD (header, subhead, table, row, table header, table data)

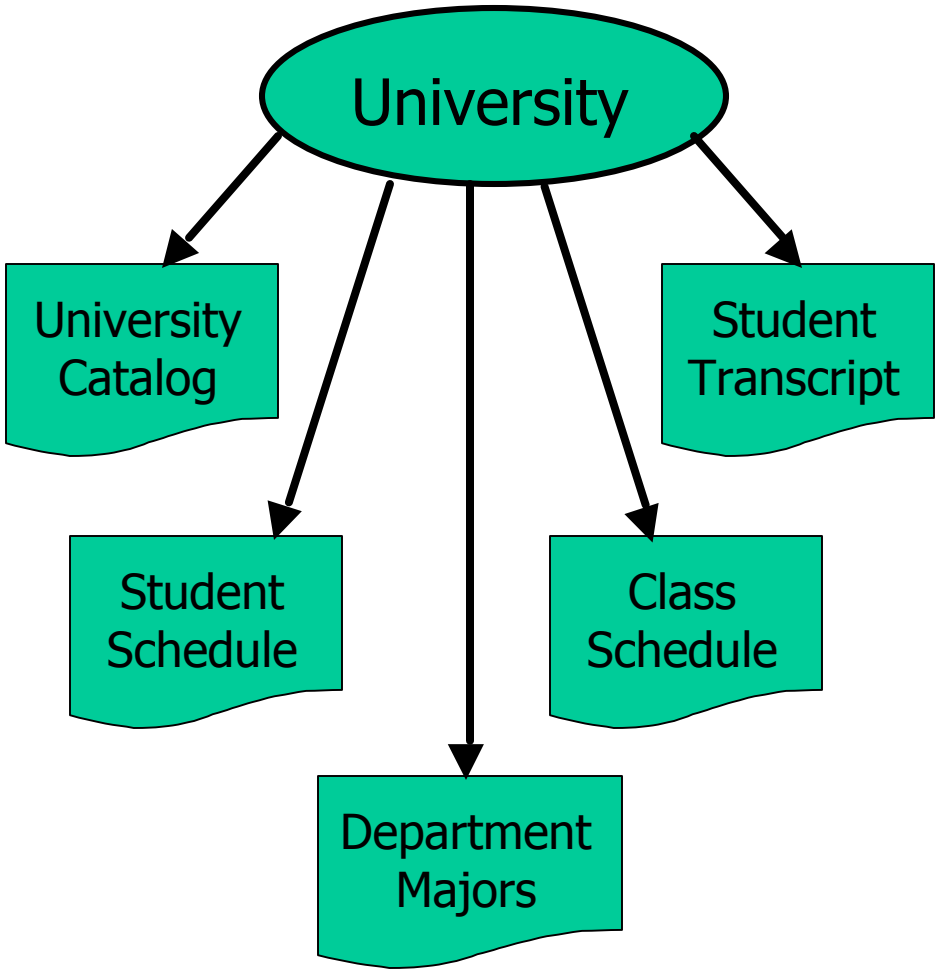
Domains and XML

- Each XML document describes a single subject matter



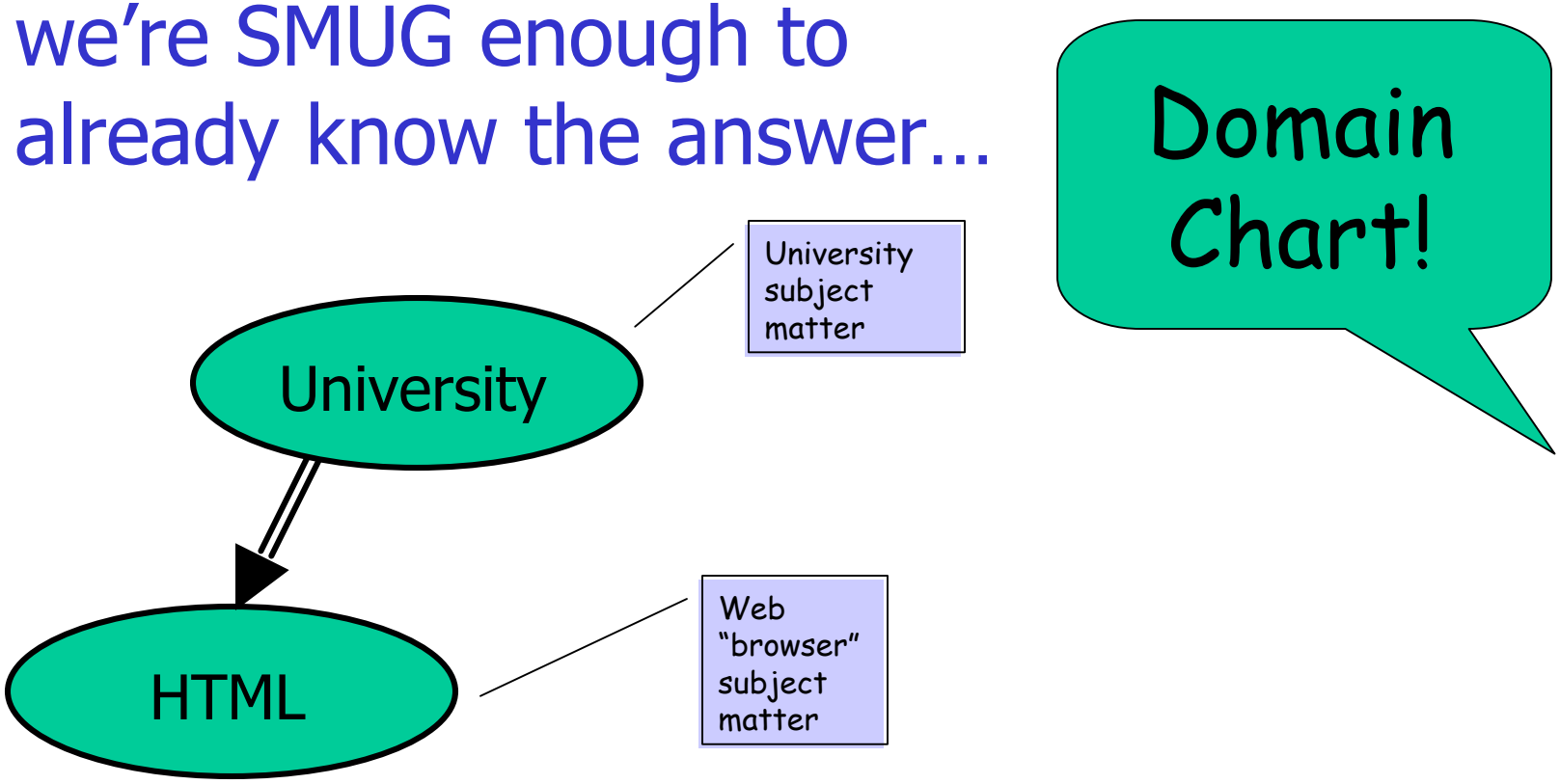
Domains and XML

- Each XML document is based upon a single domain
- The elements are based upon the objects in that domain
- There may be many documents per domain



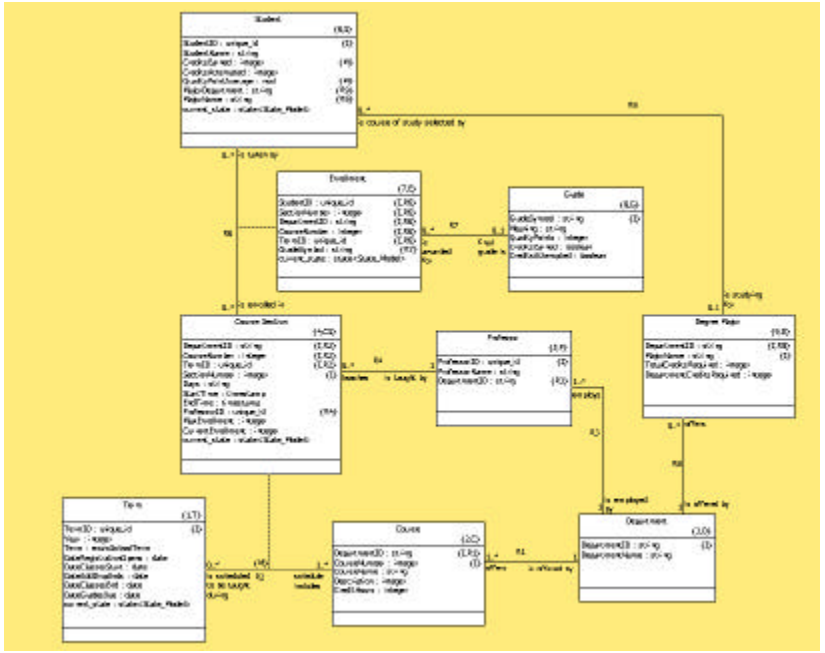
Commentary...

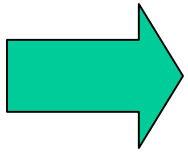
- While the rest of the world is struggling to figure out how to organize an XML project, we're SMUG enough to already know the answer...



Creating an XML Document

- Start with a root object
- Select attributes and related objects
- Traverse relationships,
  maintain multiplicity & conditionality

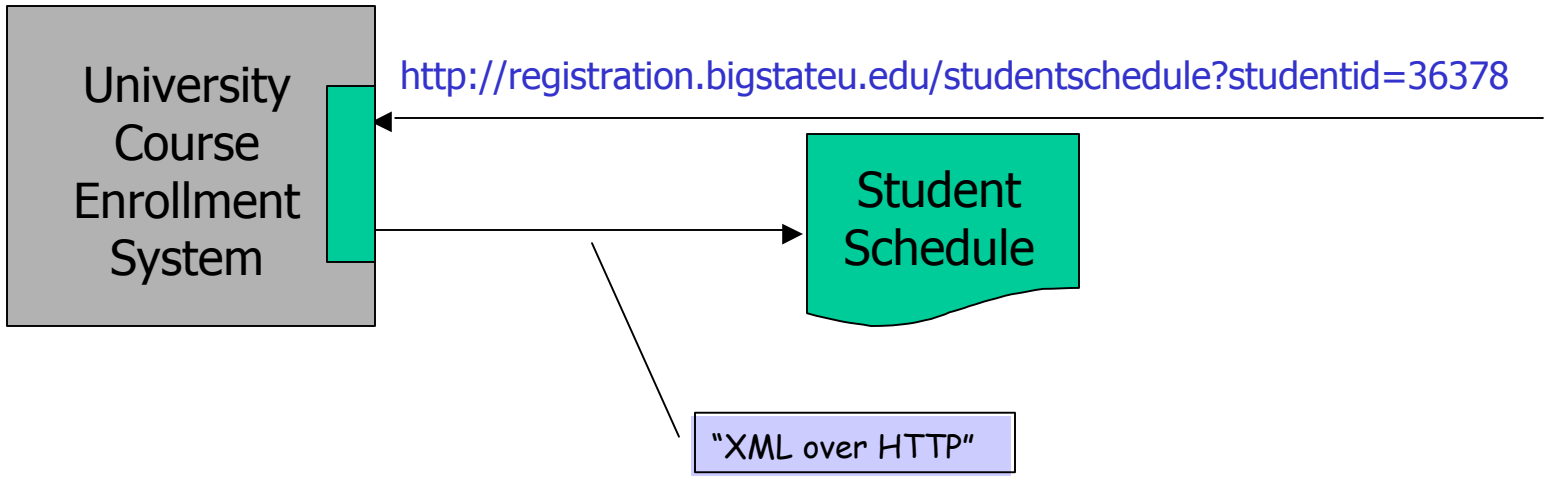




Course Schedule document

"Data Server"

- Applications provide interfaces for producing their XML documents



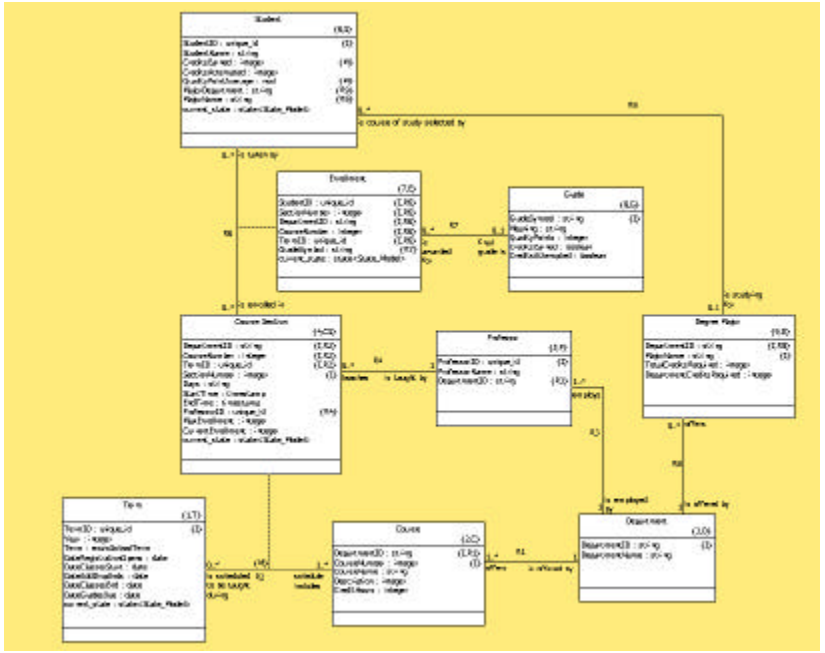
Data Type Definition (DTD)

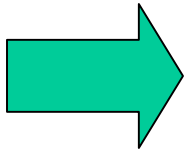
- Official way to specify and to validate an XML document[1]
  - What are the elements
  - What are the elements' attributes
  - Which elements include other elements and the multiplicity of those inclusions

[1] XML "schemas" are a new all-XML means of specifying an XML document, but they do not validate XML documents in all parsers yet.

Deriving a DTD

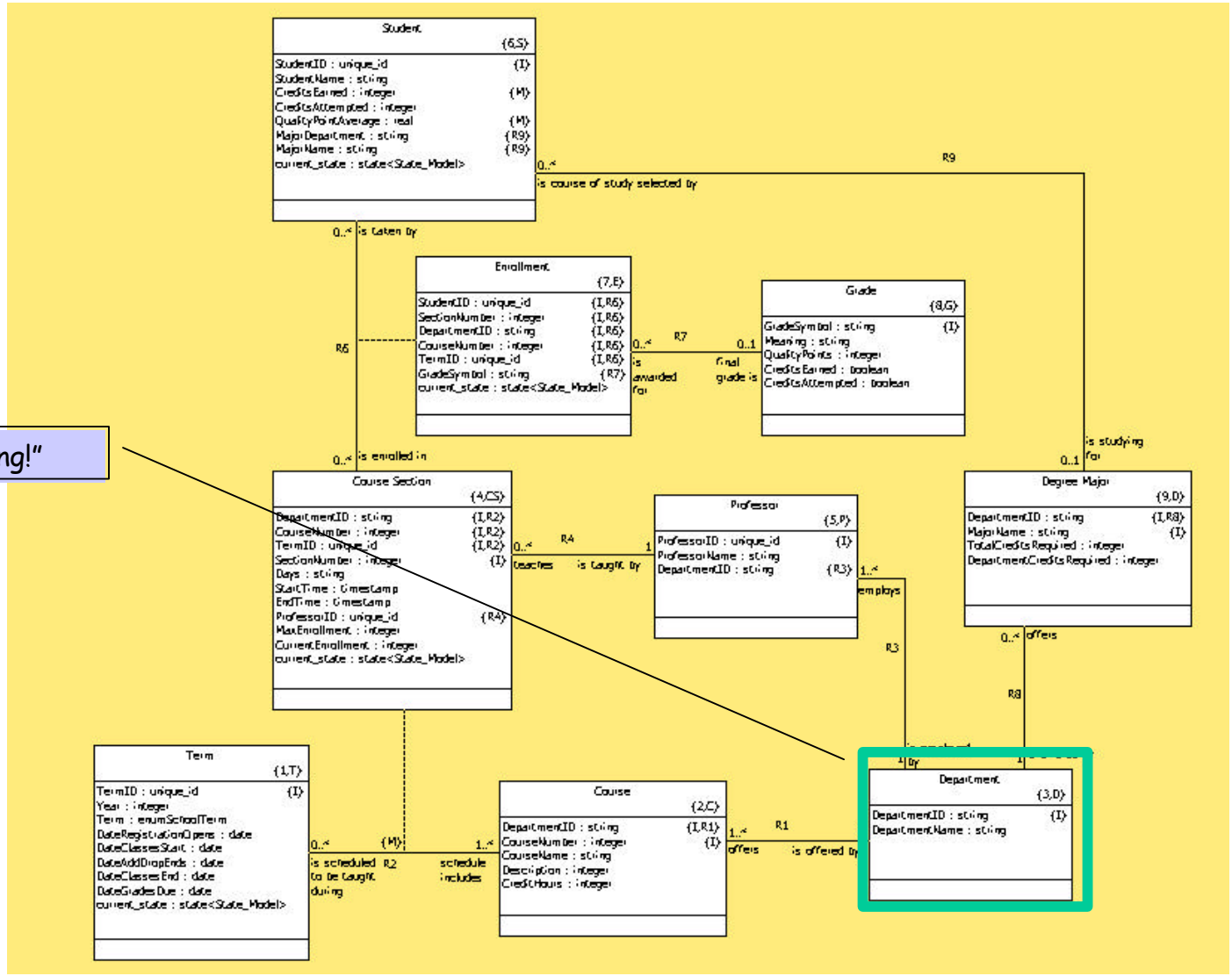
- Start with a root object
- Select attributes and related objects
- Traverse relationships,
  maintain multiplicity & conditionality





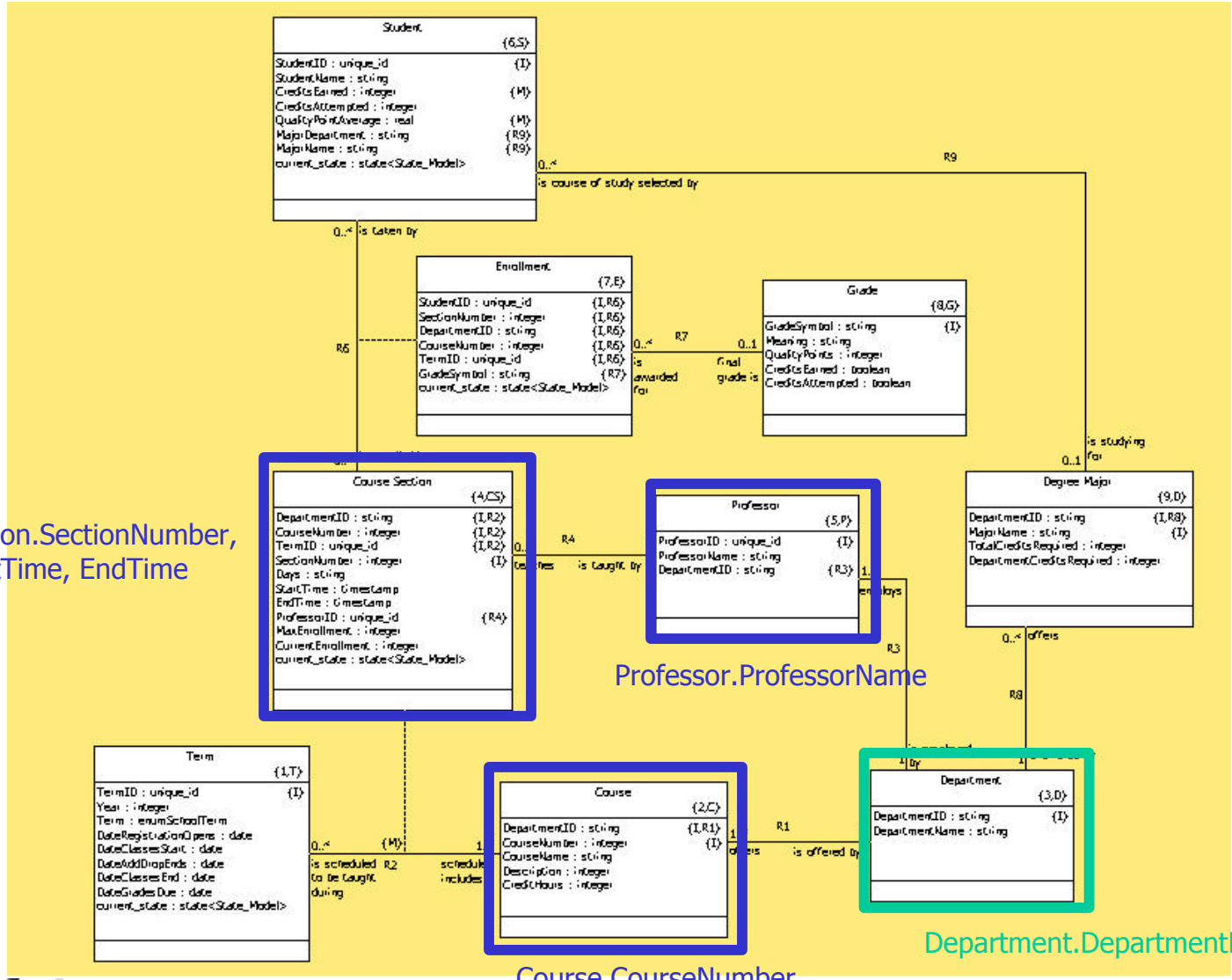
Course Schedule DTD

Root Object



It's "Coloring!"

Attributes & Related Objects



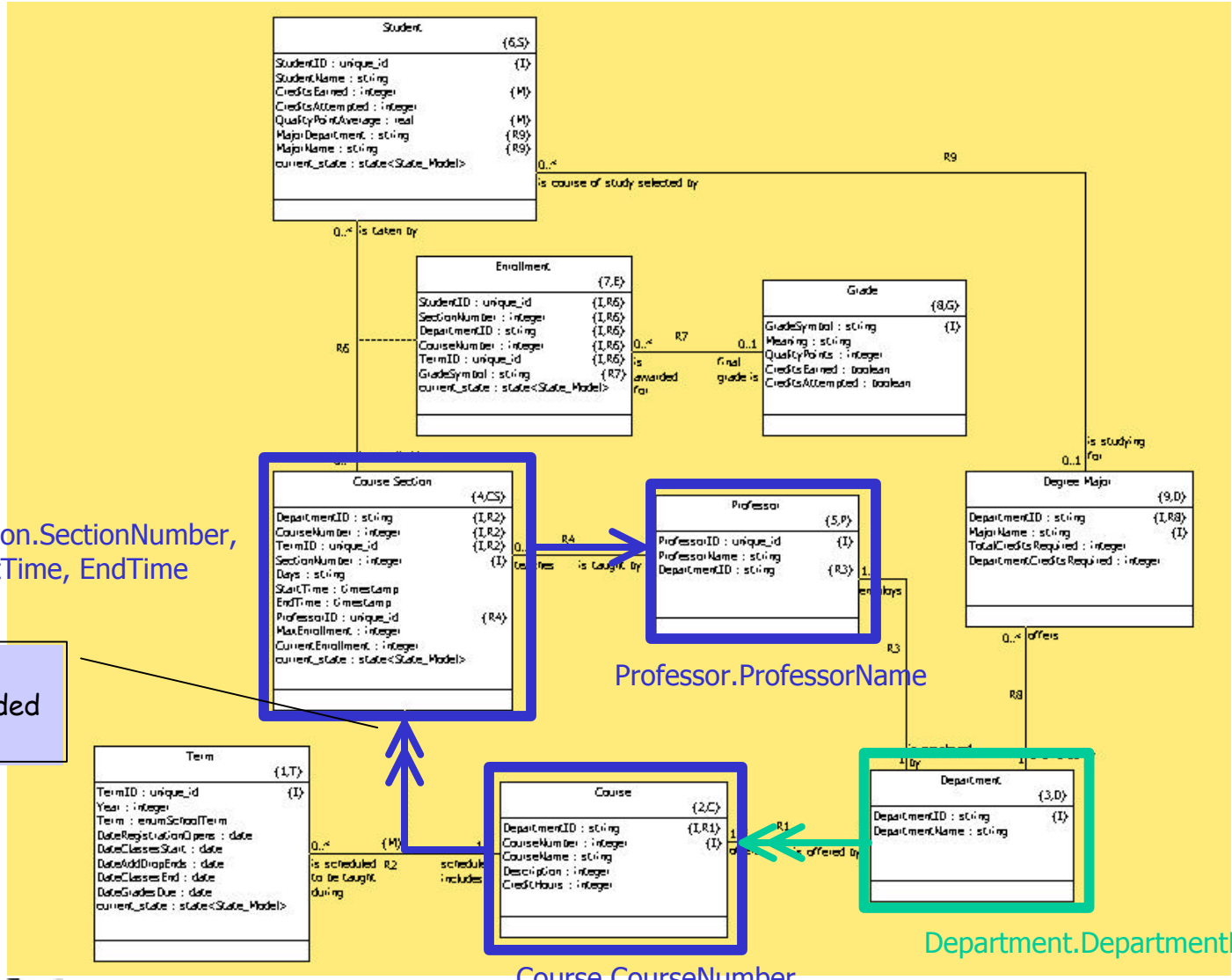
CourseSection.SectionNumber, Days, StartTime, EndTime

Professor.ProfessorName

Department.DepartmentName

Course.CourseNumber, CourseName, Description

Traverse Relationships



CourseSection.SectionNumber, Days, StartTime, EndTime

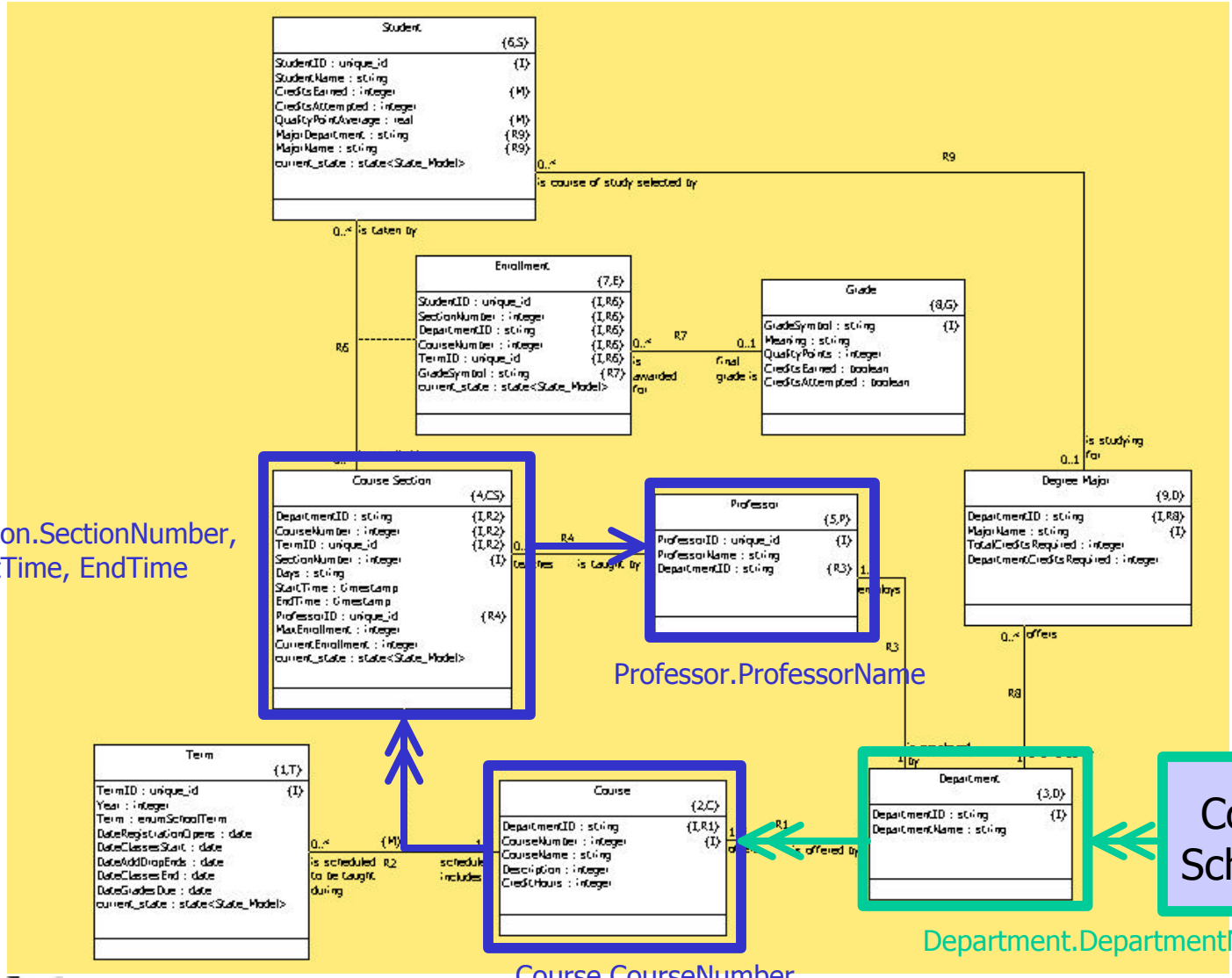
UML or no UML… those double-headed arrows are back!

Professor.ProfessorName

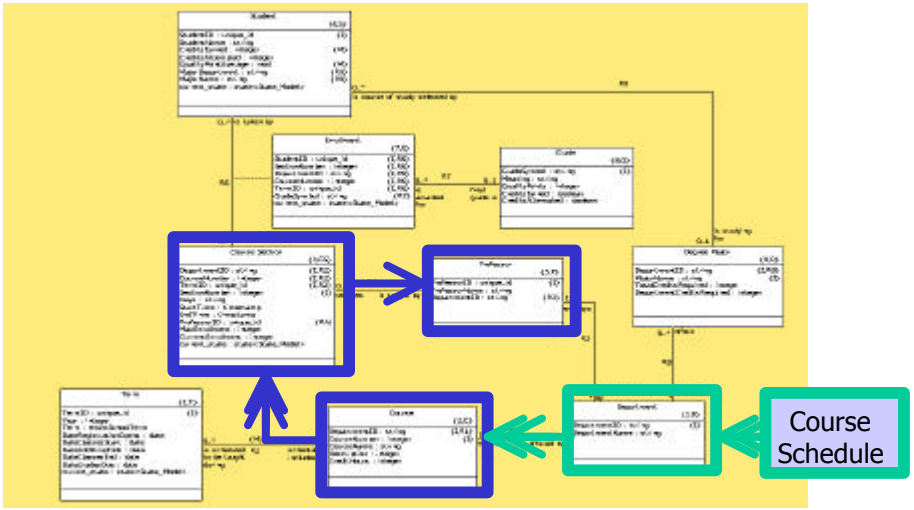
Department.DepartmentName

Course.CourseNumber, CourseName, Description

Single-Instance Object

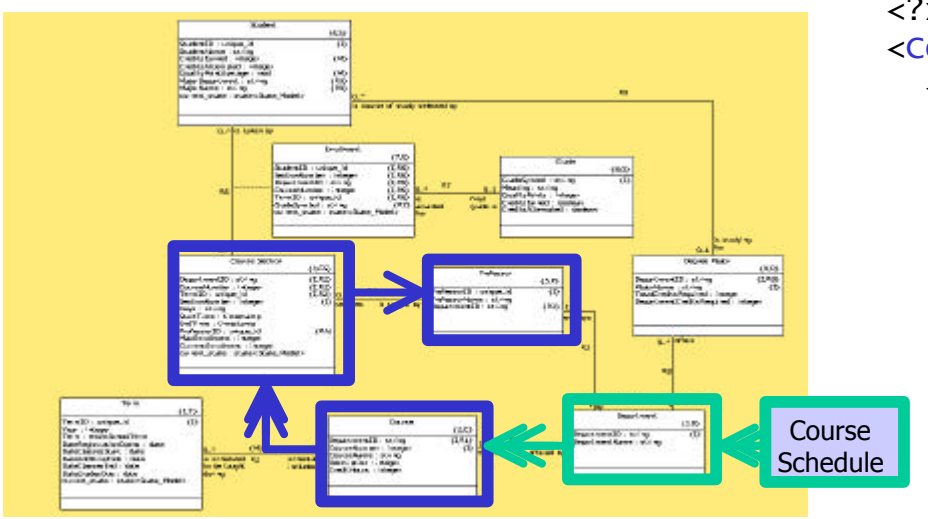


Course Schedule DTD



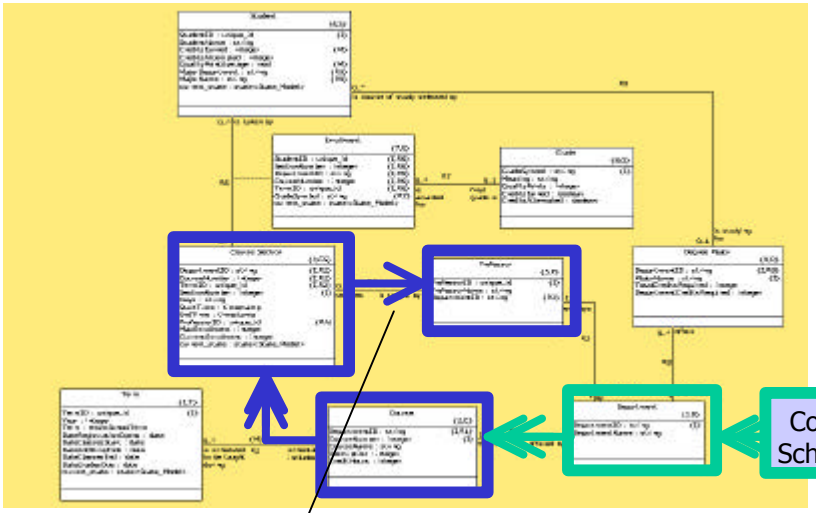

```
<!ELEMENT CourseSchedule (Department)+>
<!ATTLIST CourseSchedule
      Term CDATA #IMPLIED
>
<!ELEMENT Department (Course+)>
<!ATTLIST Department
      Name CDATA #IMPLIED
>
<!ELEMENT Course (CourseSection)+>
<!ATTLIST Course
      CourseName CDATA #IMPLIED
      CourseNumber CDATA #IMPLIED
      CreditHours CDATA #IMPLIED
>
<!ELEMENT CourseSection EMPTY>
<!ATTLIST CourseSection
      SectionNumber CDATA #IMPLIED
      EndTime CDATA #IMPLIED
      ProfessorName CDATA #IMPLIED
      StartTime CDATA #IMPLIED
      Days CDATA #IMPLIED
>
```

XML Document




```
<?xml version="1.0"?>
<CourseSchedule Term="Spring 2000">
  <Department Name="Comp Sci">
    <Course CourseNumber="101"
            CourseName="Java Programming"
            CreditHours="3">
      <CourseSection SectionNumber="1"
                     Days="MWF"
                     StartTime="10 a.m."
                     EndTime="11 a.m."
                     ProfessorName="Gosling"/>
      <CourseSection SectionNumber="2"
                     Days="MWF"
                     StartTime="3 p.m."
                     EndTime="4 p.m."
                     ProfessorName="Gates"/>
    </Course>
    <Course CourseNumber="202"
            CourseName="Object-Oriented Analysis"
            CreditHours="3">
      <CourseSection SectionNumber="1"
                     Days="MWF"
                     StartTime="11 a.m."
                     EndTime="12 a.m."
                     ProfessorName="Mellor"/>
```

Combining Attributes



With a :1 relationship…

```
<?xml version="1.0"?>
<CourseSchedule Term="Spring 2000">
  <Department Name="Comp Sci">
    <Course CourseNumber="101"
            CourseName="Java Programming"
            CreditHours="3">
      <CourseSection SectionNumber="1"
                     Days="MWF"
                     StartTime="10 a.m."
                     EndTime="11 a.m."
                     ProfessorName="Gosling"/>
      <CourseSection SectionNumber="2"
                     Days="MWF"
                     StartTime="3 p.m."
                     EndTime="4 p.m."
                     ProfessorName="Gates"/>
    </Course>
    <Course CourseNumber="202"
            CourseName="Object-Oriented Analysis"
            CreditHours="3">
      <CourseSection SectionNumber="1"
                     Days="MWF"
                     StartTime="11 a.m."
                     EndTime="12 a.m."
                     ProfessorName="Mellor"/>
```

…the related object's attributes can be combined into the element.

Combining Attributes

- Can combine attributes for
  - :1 relationship
  - subtype - supertype
    - Include elements for leaf subtypes
- Put this into the coloring

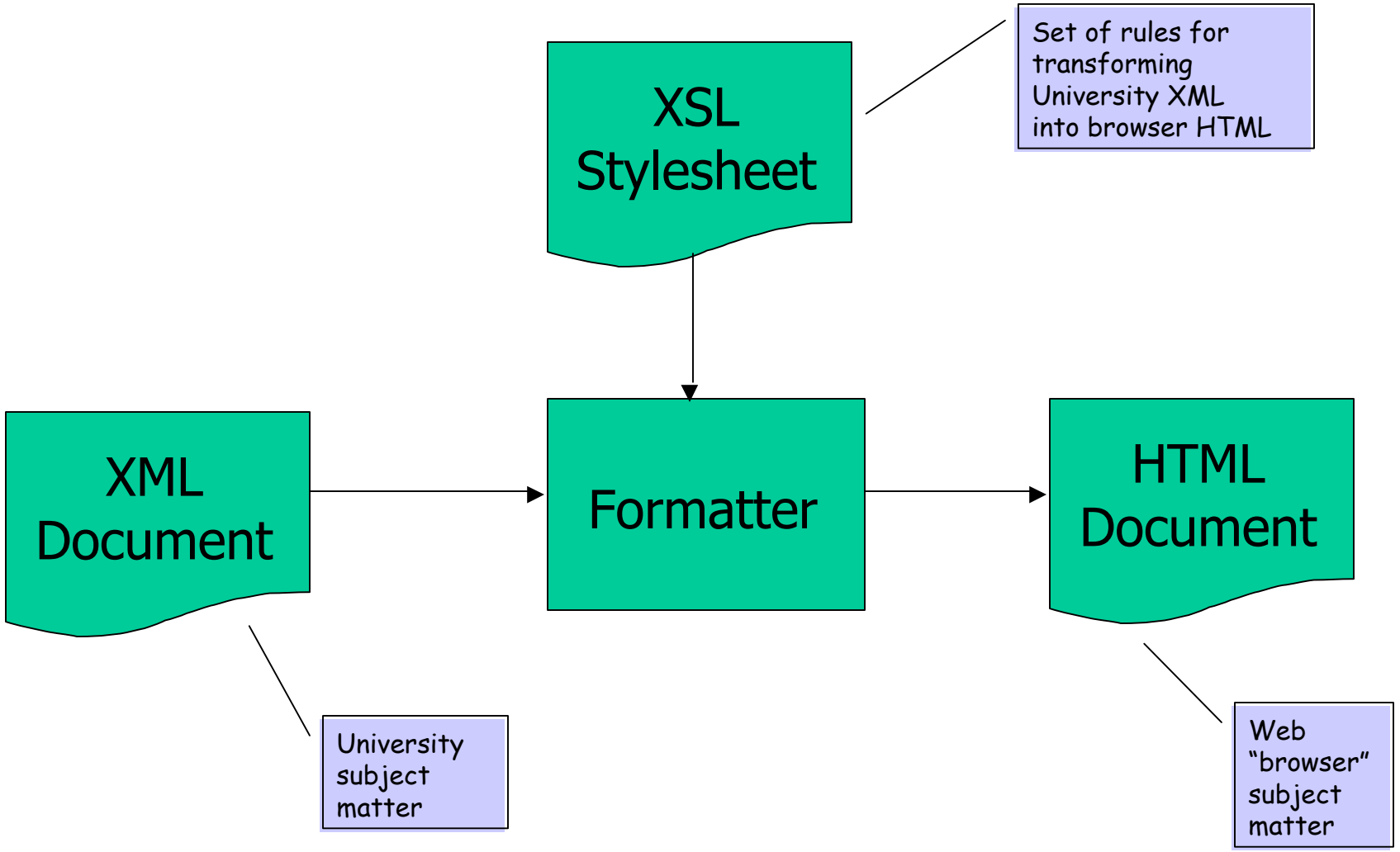
Special Considerations

- Relationship Loops
  - Careful!
- Referential Attributes
  - Generally don't include
- Structured types
  - Include as sub-elements

XML Archetype

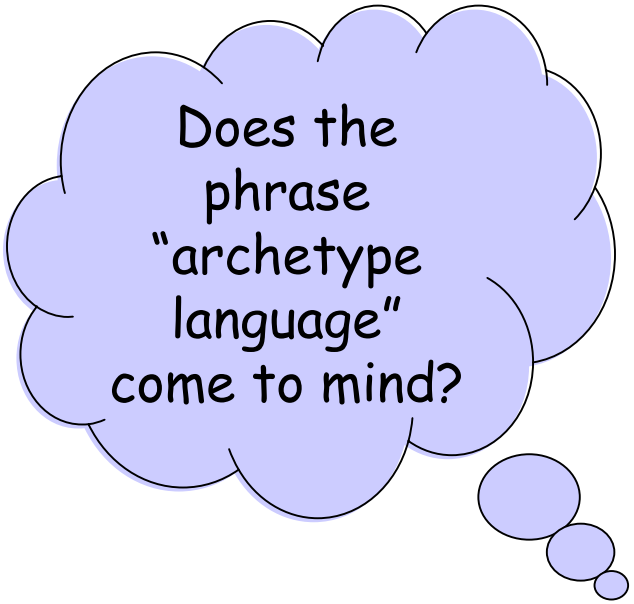
- Coloring tables for
  - which objects
  - which relationships
- Colorings + OOA IM can be used to create
  - the DTD or XML schema
  - code for the "data server"
    (as part of an overall architecture)

Transforming XML



XSL Stylesheets

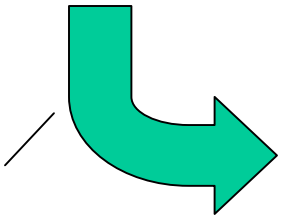
- XSL (eXtensible Stylesheet Language) is a means for representing a transformation of XML documents into any other representation.
  - XSL doesn't just create HTML!
  - XSL can create any other XML
  - XSL can create more than XML



XSL Example

```
<?xml version="1.0">
<Schedule Term="Spring 2000"
          Student="Brown, Charlie" >
  <Enrollment Department="Comp Sci"
              CourseNum="202"
              Title="Object-Oriented Analysis"
              Days="MWF"
              Times="10-11 a.m."/>
  <Enrollment Department="PE"
              CourseNum="316"
              Title="Baseball"
              Days="TTh"
              Times="3-6 p.m."/>
</Schedule>
```

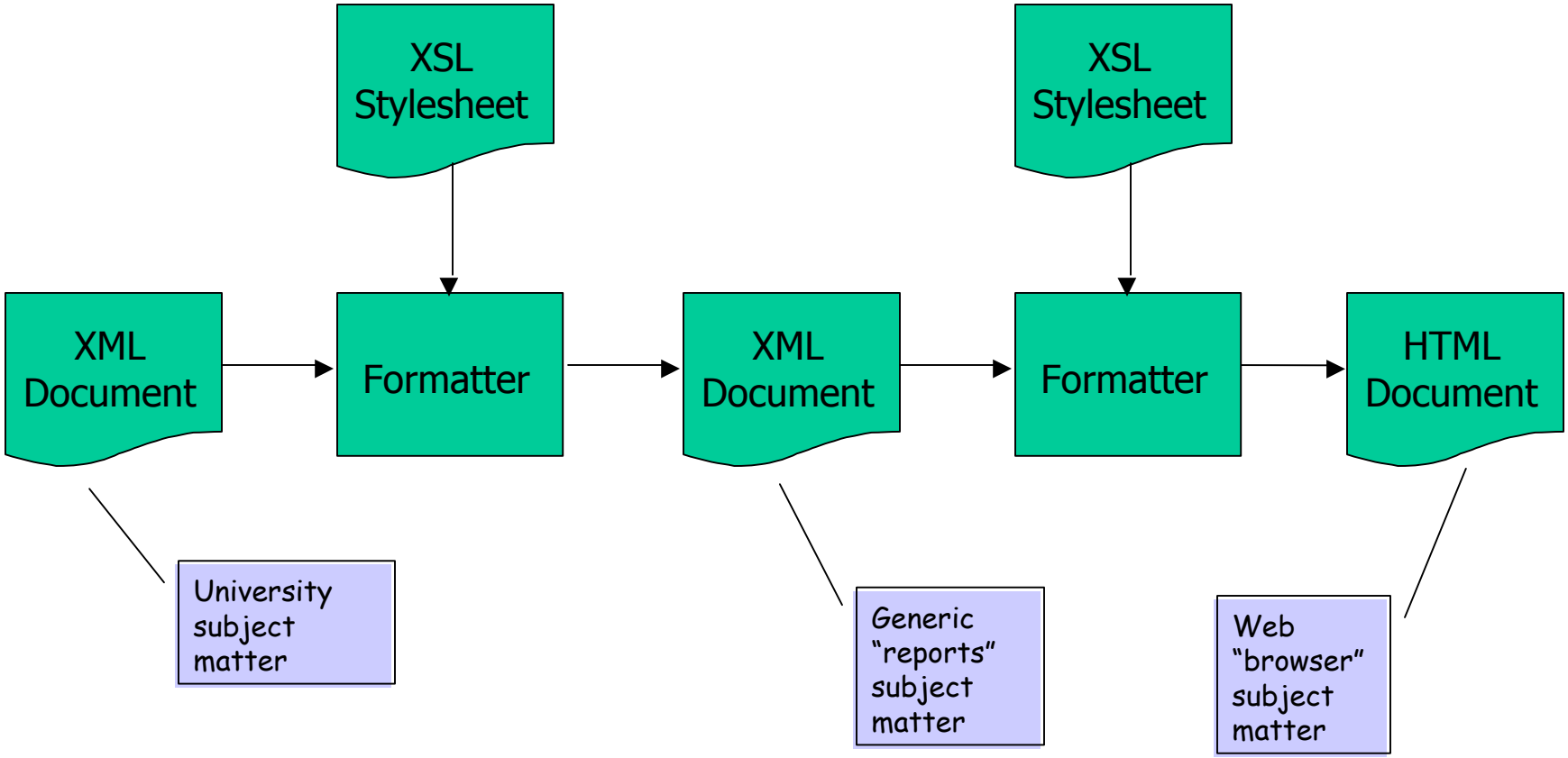
Can also apply different stylesheets to produce different forms of the same information, e.g. a display view and an editable form



```
<HTML>
<HEAD>
<TITLE>Student Schedule</TITLE>
<BODY>
<H3>Student Schedule</H3>
<TABLE border=1 cellPadding=1 cellSpacing=1>
  <TR>
     <TH align="left">Student</TH>
     <TD>Brown, Charlie</TD></TR>
  <TR>
     <TH align="left">Term</TH>
     <TD>Fall 2000</TD></TR>
</TABLE>
<TABLE border=1 cellPadding=1 cellSpacing=1>
  <TR>
    <TH>Department</TH>
    <TH>Course#</TH>
    <TH>Course Name</TH>
    <TH>Days</TH>
    <TH>Time</TH></TR>
  <TR>
    <TD>Comp Sci</TD>
    <TD>202</TD>
```

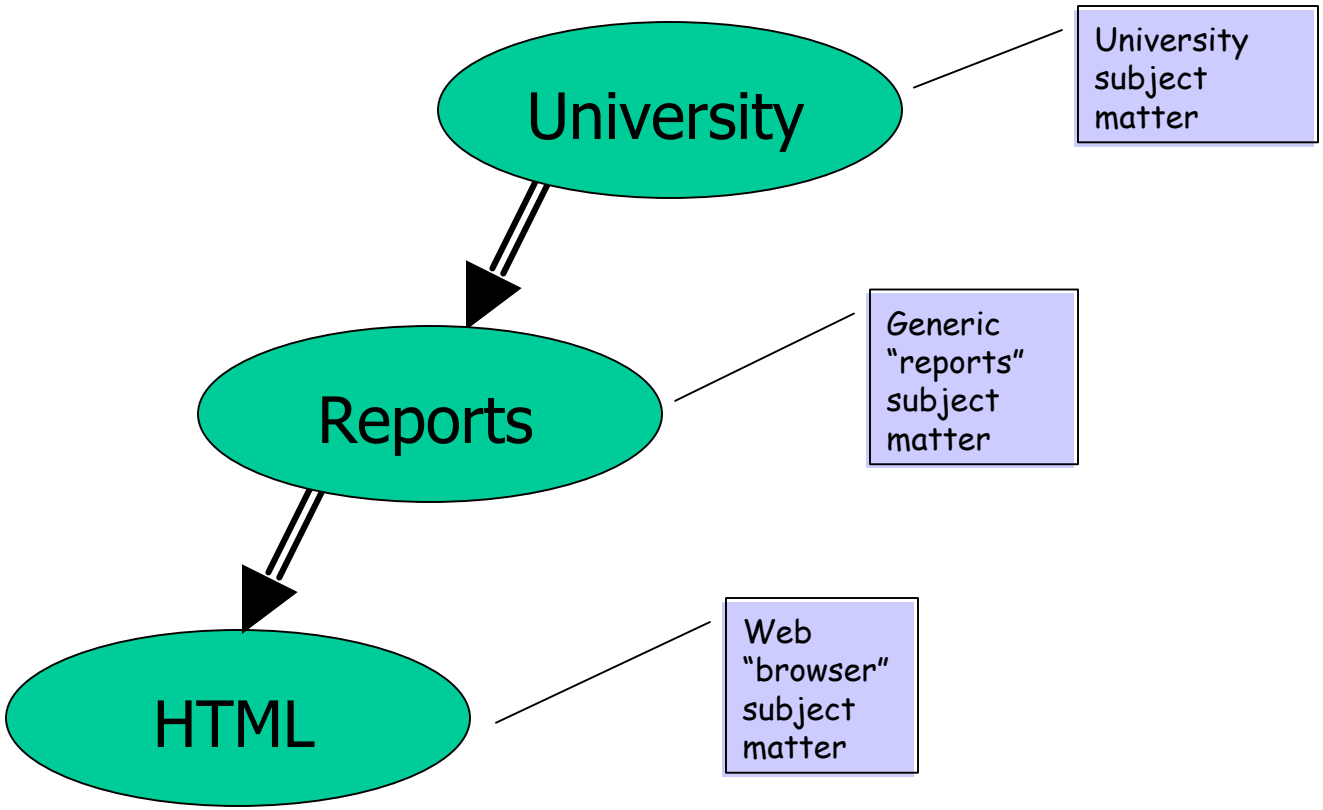
Multiple Transformations

- So we could also do this...



Multiple Transformations

- Or in a more familiar representation...

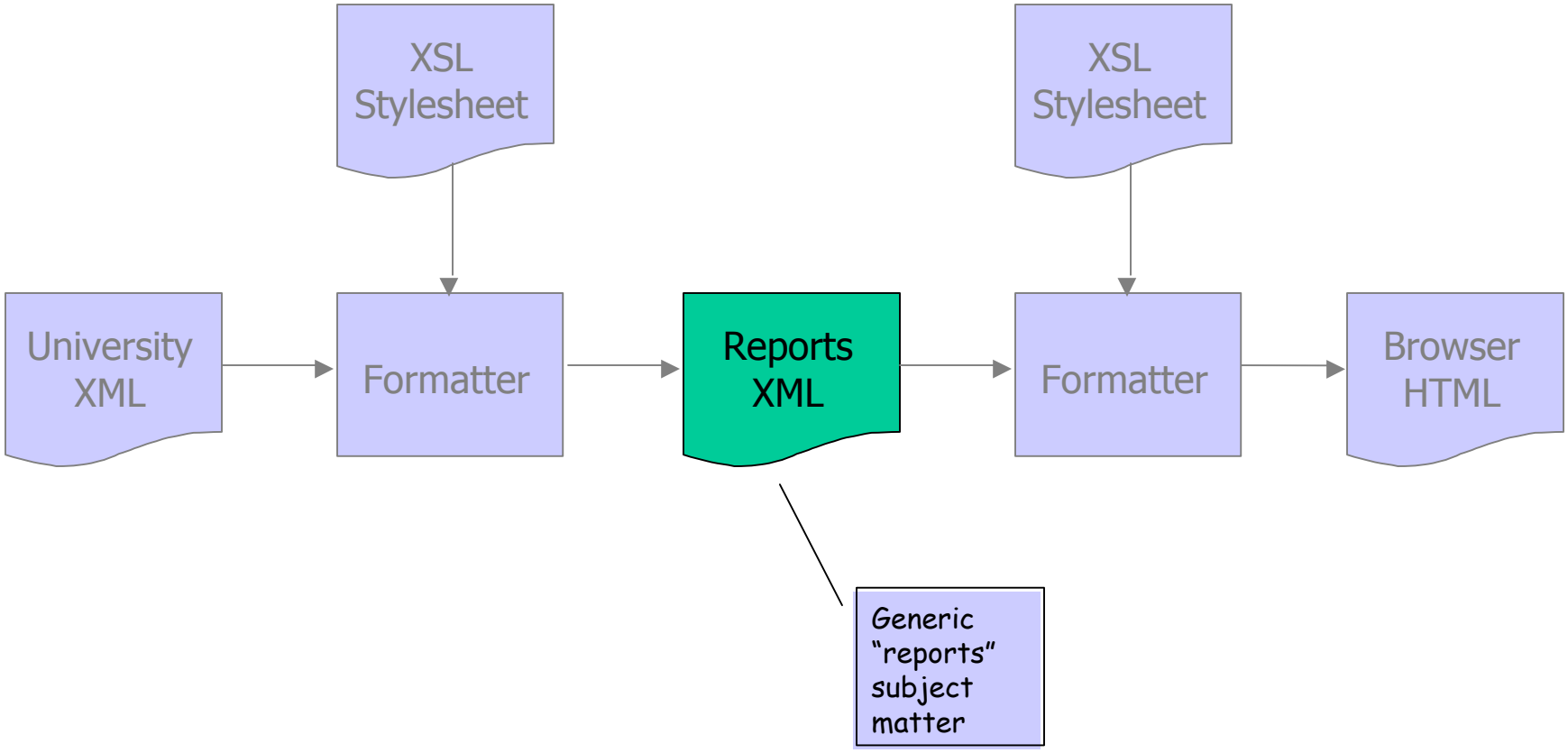


Generic "Reports" Domain

- Same reasons for creating service domains can justify intermediate steps in XML apps
  - division of responsibilities
    (e.g. client / consultant)
  - consistency
    (many reports, all use same stylesheet)
  - reuse
    (many things need reports)
  - simplification
    (two small steps easier than one)

Intermediate XML

- What does this intermediate XML look like?



Intermediate XML

```
<Report Title="Semester Schedule">
    <Table Title="Spring 2000">
        <HeaderRow>
            <Column Name="Days" Title="Days"/>
            <Column Name="Times" Title="Times"/>
            <Column Name="Department" Title="Dept"/>
            <Column Name="CourseNum" Title="Course No."/>
            <Column Name="Course" Title="Course Name"/>
        </HeaderRow>
        <Row>
            <Cell Name="Days" Value="MWF"/>
            <Cell Name="Times" Value="10-11 a.m."/>
            <Cell Name="Department" Value="Comp Sci"/>
            <Cell Name="CourseNum" Value="202"/>
            <Cell Name="Course" Value="Object-Oriented Analysis"/>
        </Row>
            . . .
    </Table>
</Report>
```

Look at the tags...

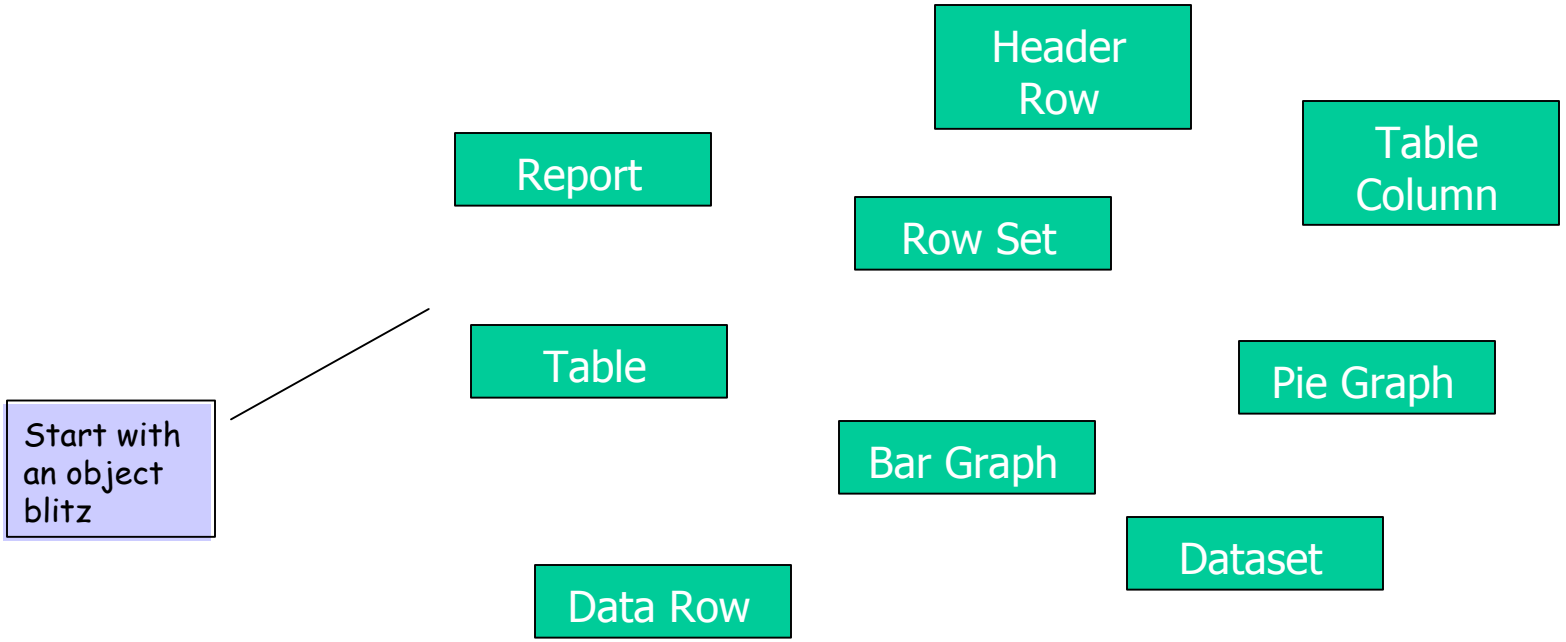
Intermediate XML

```
<Report Title="Semester Schedule">
    <Table Title="Spring 2000">
        <HeaderRow>
            <Column Name="Days" Title="Days"/>
            <Column Name="Times" Title="Times"/>
            <Column Name="Department" Title="Dept"/>
            <Column Name="CourseNum" Title="Course No."/>
            <Column Name="Course" Title="Course Name"/>
        </HeaderRow>
        <Row>
            <Cell Name="Days" Value="MWF"/>
            <Cell Name="Times" Value="10-11 a.m."/>
            <Cell Name="Department" Value="Comp Sci"/>
            <Cell Name="CourseNum" Value="202"/>
            <Cell Name="Course" Value="Object-Oriented Analysis"/>
        </Row>
            . . .
    </Table>
</Report>
```

...they're objects in the Reports service domain

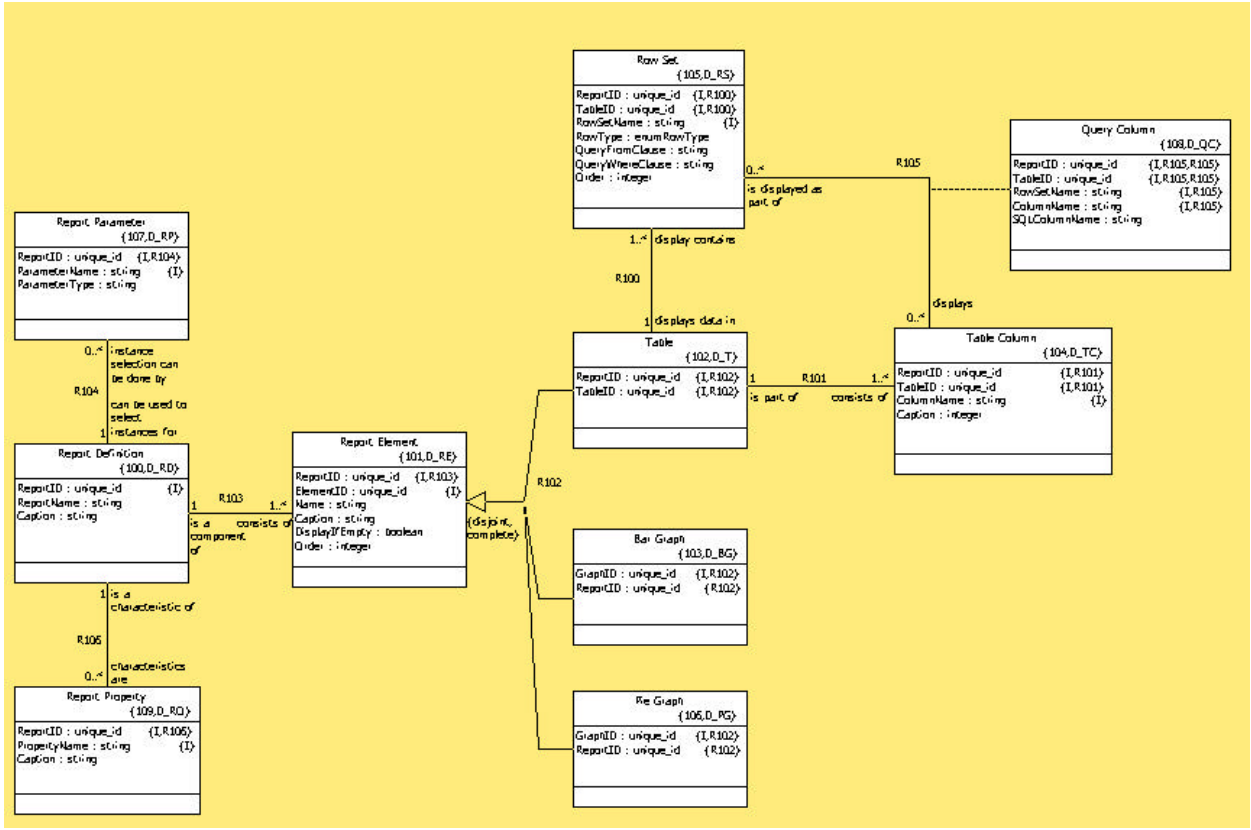
Do an OOA Model

- So we should start figuring out an intermediate XML by doing a model of that service domain!



Reports Domain

- Define the Reports XML as an XML based upon the Reports domain IM



App -> Reports Mapping

- Define the instances of the reports (populate the instances of the Reports domain)
- Map things in Reports to things in University

- These mappings can be formed as
  - XSL stylesheets
  - database entries
  - code

Commentary…

- While the rest of the world is struggling to figure out how to organize an XML project, we're SMUG enough to already know the answer…

