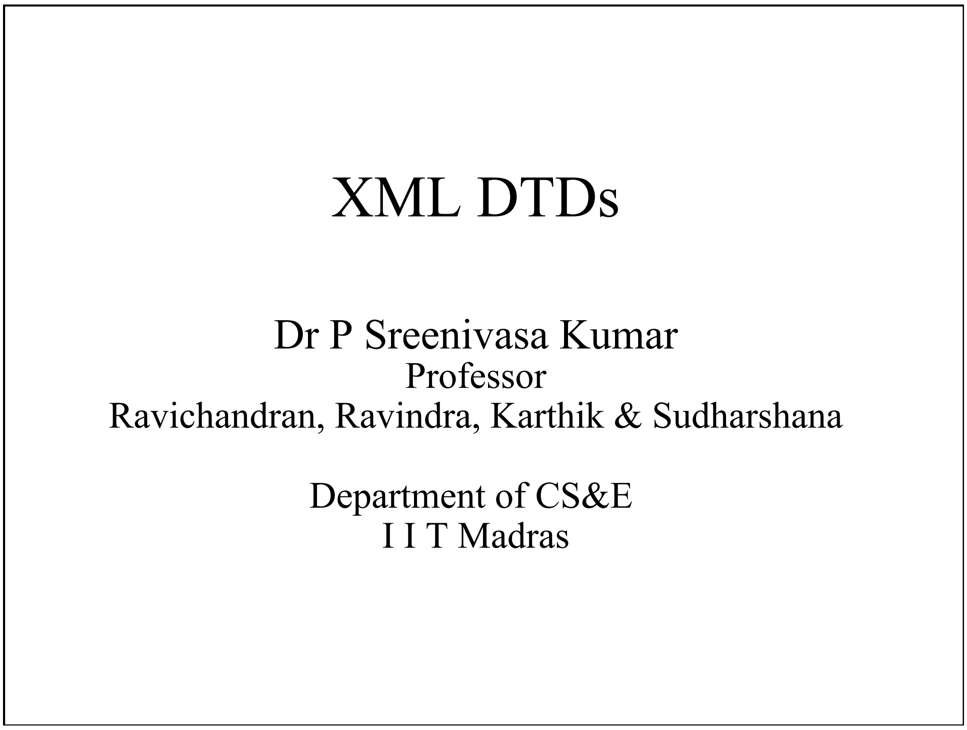
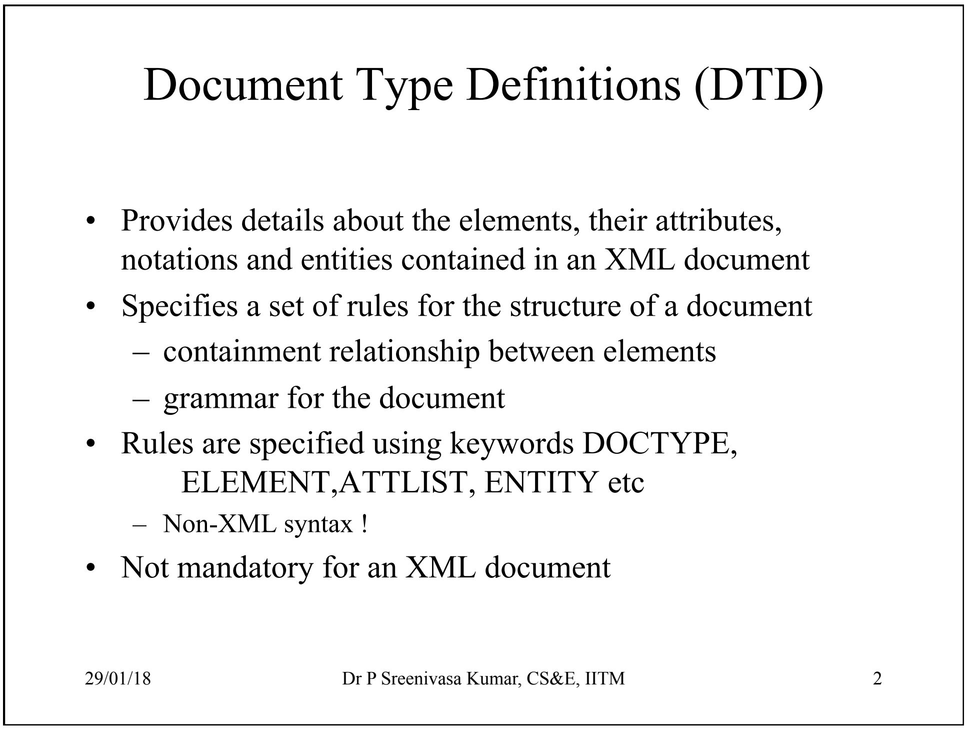


XML DTDs

Dr P Sreenivasa Kumar
Professor
Ravichandran, Ravindra, Karthik & Sudharshana

Department of CS&E
I I T Madras



Document Type Definitions (DTD)

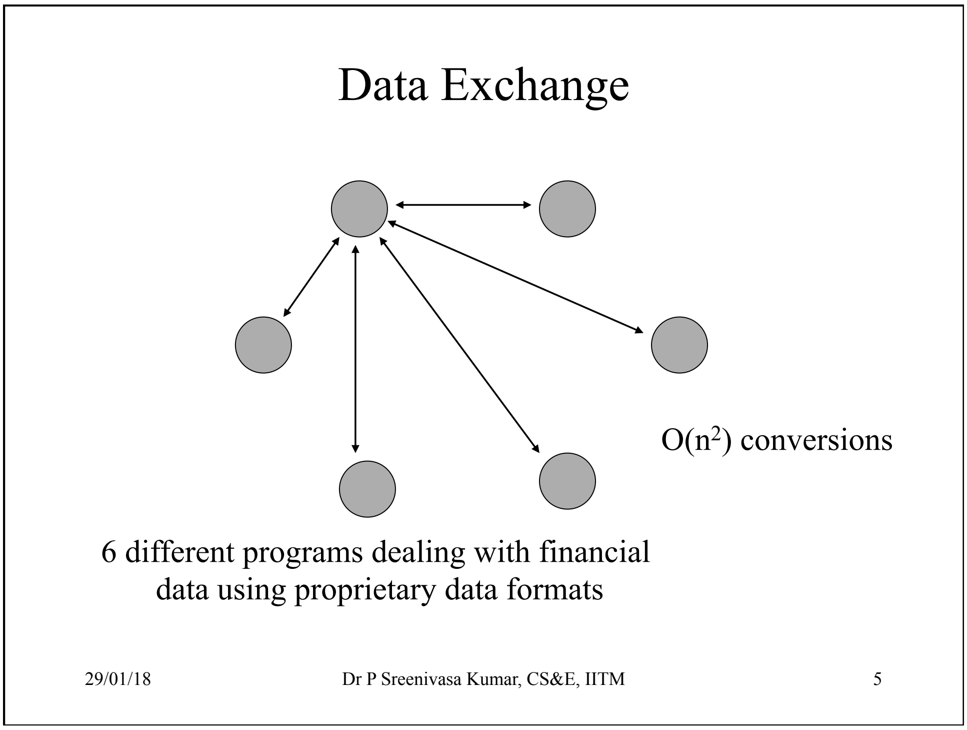
- Provides details about the elements, their attributes, notations and entities contained in an XML document
- Specifies a set of rules for the structure of a document
  - containment relationship between elements
  - grammar for the document
- Rules are specified using keywords DOCTYPE, ELEMENT,ATTLIST, ENTITY etc
  - Non-XML syntax !
- Not mandatory for an XML document

Significance of a DTD

- Tool for organizational standardizing
  - Repository of publicly available DTDs for common documents used in business
  - businesses produce data using public DTDs
  - easy interoperability between business partners
- Consortium of businesses in a domain
  - co-operate to create DTD repositories

Document Type Declaration

- Specifies the DTD a document uses
- Appears in a document’s prolog, after the XML declaration but before the root element
- A document type declaration might either contain a DTD or refer to a DTD
- May contain
  - DTD
  - URL giving where DTD is stored - PUBLIC /SYSTEM
  - both DTD as well as URL
- XML documents without document type declarations can not be validated.

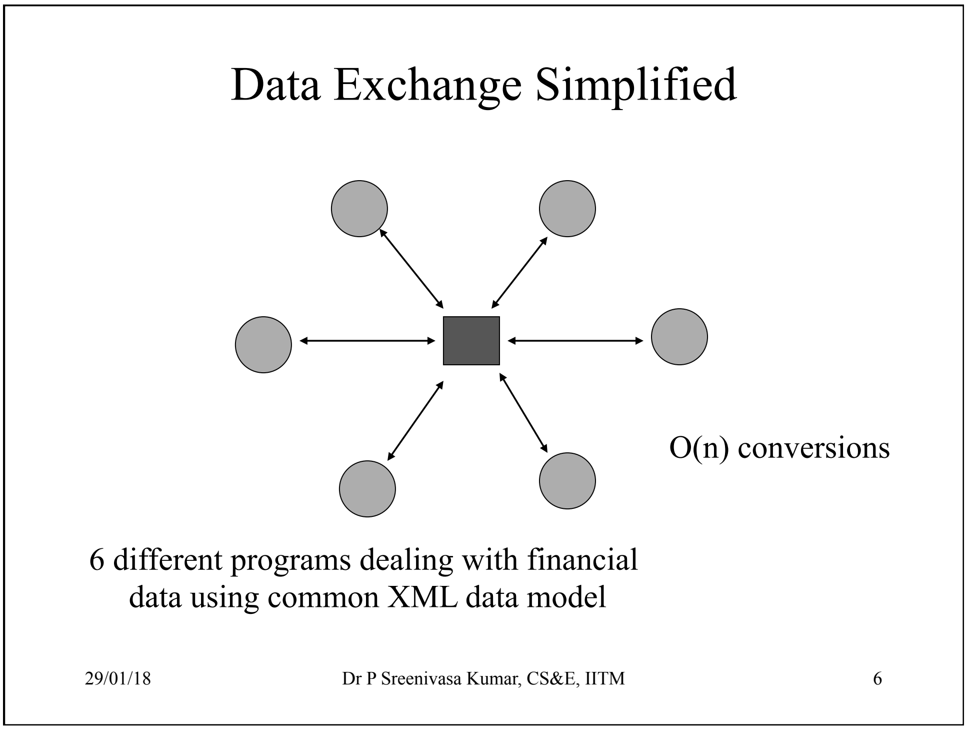


Data Exchange

O(n^2) conversions

6 different programs dealing with financial data using proprietary data formats

29/01/18 Dr P Sreenivasa Kumar, CS&E, IITM 5



Data Exchange Simplified

O(n) conversions

6 different programs dealing with financial data using common XML data model

29/01/18 Dr P Sreenivasa Kumar, CS&E, IITM 6

addresses.xml

An example XML file containing the DTD internally.

```
<?xml version = “1.0”, standalone = “yes”?>
<!DOCTYPE addresses[
<!ELEMENT addresses(address*)
<!ELEMENT address (name, street, city, pin, state, country)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT street(#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT  pin (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT country (#PCDATA)>]>
```

addresses.xml (contd.)

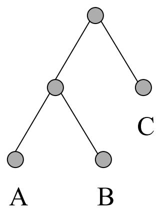
```
<addresses>
   <address><name> Jayalalitha </name>
        <street> Poes Garden </street>
        <city> Chennai</city> <pin> 600 041</pin>
        <state> Tamil Nadu</state><country>India</country>
  </ address >
  < address ><name> Karunanidhi</name>
        <street>Gopalapuram</street>
        <city>Chennai</city><pin>600 042</pin>
        <state> Tamil Nadu</state><country>India</country>
  </ address >
 </ addresses >
```

An Example DTD for books DB

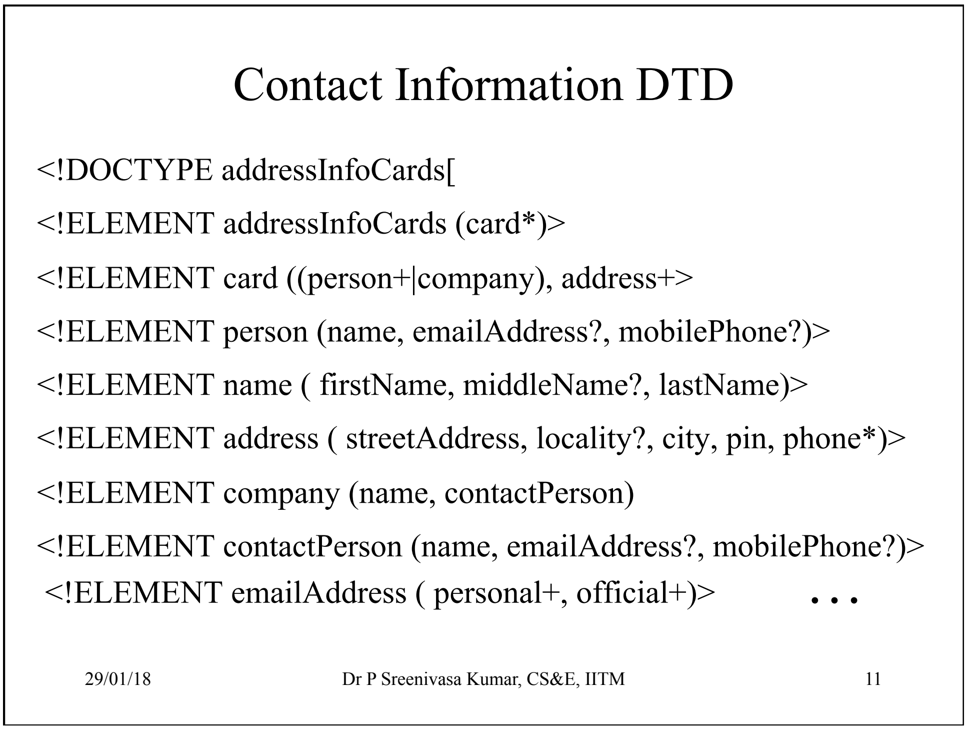
```
<!DOCTYPE library[
<!ELEMENT library ( book+)>
<!ELEMENT book ( title, author+, publisher, year)>
<!ELEMENT author ( firstName, midInitial, lastName)>
<!ELEMENT publisher ( name, city)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT city (#PCDATA)>
]
```

DTD’s can be recursive

```
<!DOCTYPE binaryTree [
<!ELEMENT binaryTree ( node )>
<!ELEMENT node ( leaf | ( node, node) )>
<!ELEMENT leaf (#PCDATA)> ]
```

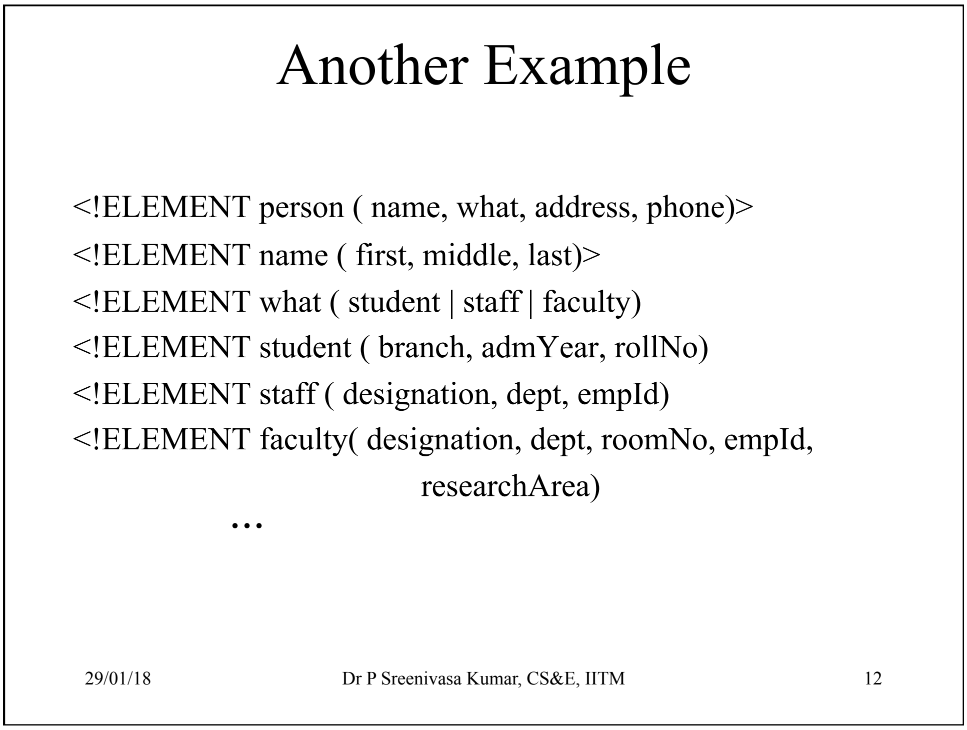


```
<binaryTree>
    <node>
       <node>
            <node> <leaf> A </leaf>  </node>
            <node> <leaf> B </leaf>  </node>
        </node>
        <node> <leaf> C </leaf> </node>
     </node>
</binaryTree>
```



Contact Information DTD

```
<!DOCTYPE addressInfoCards[
<!ELEMENT addressInfoCards (card*)>
<!ELEMENT card ((person+|company), address+>
<!ELEMENT person (name, emailAddress?, mobilePhone?)>
<!ELEMENT name ( firstName, middleName?, lastName)>
<!ELEMENT address ( streetAddress, locality?, city, pin, phone*)>
<!ELEMENT company (name, contactPerson)
<!ELEMENT contactPerson (name, emailAddress?, mobilePhone?)>
 <!ELEMENT emailAddress ( personal+, official+)>          . . .
```



Another Example

```
<!ELEMENT person ( name, what, address, phone)>
<!ELEMENT name ( first, middle, last)>
<!ELEMENT what ( student | staff | faculty)
<!ELEMENT student ( branch, admYear, rollNo)
<!ELEMENT staff ( designation, dept, empId)
<!ELEMENT faculty( designation, dept, roomNo, empId,
                                researchArea)
              …
```

Order among sub elements

```
<!ELEMENT x ( a, b, c)>

<!ELEMENT x ( a | b| c)+>

        …
```

Report DTD

Richard Erlander (http://pdbeam.uwaterloo.ca/~rlander/)

```
<!DOCTYPE REPORT [
<!ELEMENT REPORT (TITLE, (SECTION|SHORTSECT)+)>
<!ELEMENT SECTION (TITLE, %BODY;, SUBSECTION*)>
<!ELEMENT SUBSECTION (TITLE,%BODY;, SUBSECTION*)>
<!ELEMENT SHORTSECT (TITLE, %BODY; )>
<!ELEMENT TITLE %TEXT;>
<!ELEMENT PARA %TEXT;>
<!ELEMENT LIST (ITEM)+>
<!ELEMENT ITEM (%BLOCK;)>
<!ENTITY % TEXT "(#PCDATA|CODE|KEYWORD|QUOTATION)*">
<!ENTITY % BLOCK "(PARA|LIST)+">
<!ENTITY % BODY "(%BLOCK;|EXAMPLE|NOTE)+">
<!ELEMENT EXAMPLE (TITLE?,%BLOCK;)>
```

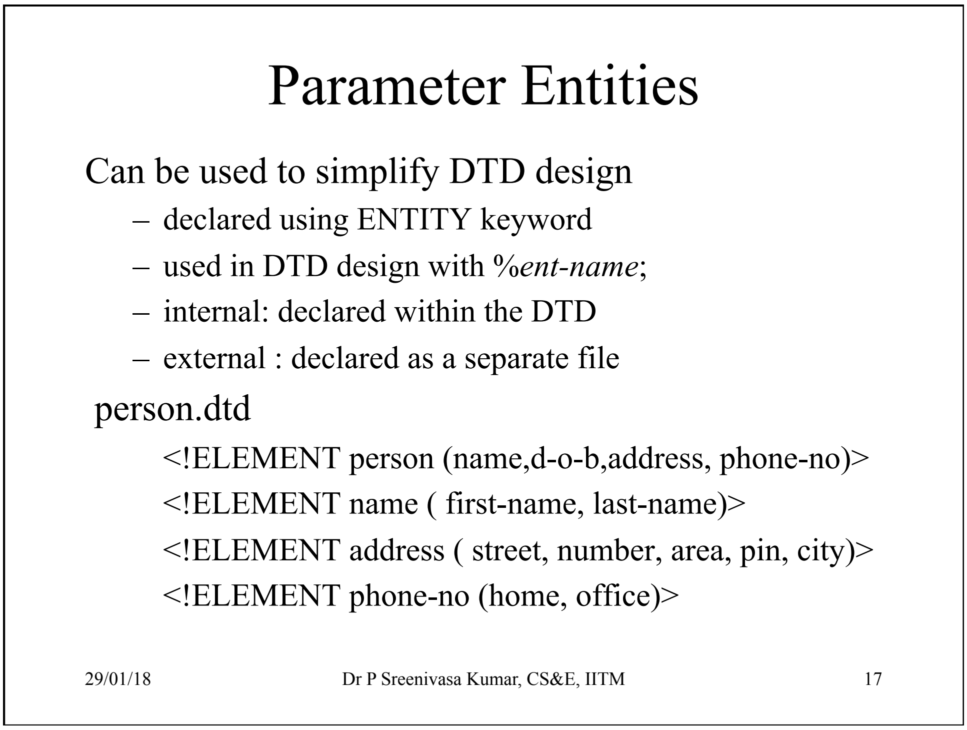
Report DTD

Richard Erlander (http://pdbeam.uwaterloo.ca/~rlander/)

```
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT KEYWORD (#PCDATA)>
<!ELEMENT GRAPHIC EMPTY>
<!ATTLIST REPORT security (high | medium | low ) "low">
<!ATTLIST CODE type CDATA #IMPLIED>
<!ATTLIST GRAPHIC file ENTITY #REQUIRED>
<!ENTITY xml "Extensible Markup Language">
<!ENTITY sgml "Standard Generalized Markup Language">
<!ENTITY pxa "Professional XML Authoring">
<!NOTATION GIF SYSTEM "">
<!NOTATION JPG SYSTEM "">
<!NOTATION BMP SYSTEM "">
]>
```

Validating an XML document

- Must meet the constraints specified by the DTD
- The main constraints specified by the DTD for validity:
  - the parent - child relationships between the various elements,
  - the sequence in which the subelements of an element occur
- Root element must be the one specified in the document type declaration
- XML documents - checked for validity using validating parsers



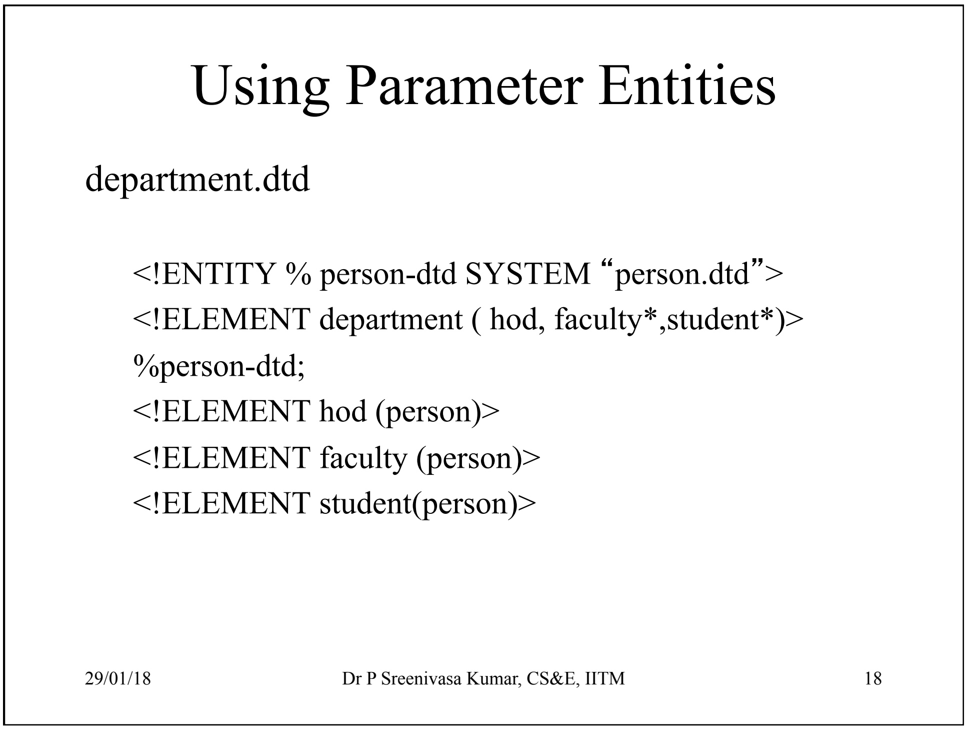
Parameter Entities

Can be used to simplify DTD design

- declared using ENTITY keyword
- used in DTD design with %*ent-name*;
- internal: declared within the DTD
- external : declared as a separate file

person.dtd

```
<!ELEMENT person (name,d-o-b,address, phone-no)>
<!ELEMENT name ( first-name, last-name)>
<!ELEMENT address ( street, number, area, pin, city)>
<!ELEMENT phone-no (home, office)>
```



Using Parameter Entities

department.dtd

```
<!ENTITY % person-dtd SYSTEM “person.dtd”>
<!ELEMENT department ( hod, faculty*,student*)>
%person-dtd;
<!ELEMENT hod (person)>
<!ELEMENT faculty (person)>
<!ELEMENT student(person)>
```

Mixing Tag Sets

Each XML application - own tag set

- describing books
- describing persons

Using tags from two or more such sets

- sometimes required
  - books and their authors
- "title" is a useful tag name in both
  - book - name of the book
  - person - "Mr", "Ms", "Dr", "Mrs"
- causes confusion when both tag sets are mixed

Namespaces

Disambiguate the tag names

- Each application's tag set placed in its own namespace

Definition

- using xmlns:*prefix* attribute on the root element
- *prefix* is to be replaced with the name of the namespace
- value of the attribute is the URI of the namespace

  <xsl:stylesheet xmlns:xsl = "http://www.w3.org/XSL/
  Transform/1.0">
- defines XSL namespace
- all tag names to be prefixed with "xsl:"