XML and NNF

- A Standard Form for XML Documents (XNF)
- Properties
  - As few hierarchical trees as possible
  - No redundant data values in any tree
- Method
  - Model the application as an ORM.
  - Make the ORM canonical.
  - Transform the canonical ORM into an NNF scheme-tree forest with as few trees as possible.
  - Cast the NNF scheme trees into XML DTD.

XNF Example

NNF with as few schemes as possible:

FD Closures: NrBeds: 4, Cost: 4, RoomNr: 3, RoomName: 3, GuestName: 2
Back off by one.
Result: RoomNr, RoomName, Cost, NrBeds, (View)*, (GuestNr, GuestName)*
DTD
(Document Type Definition)

- Defines a structure for XML documents
- Declarations
  - DOCTYPE
  - ELEMENT
    - Structural elements in terms of regular expressions (, *, +, ? |)
    - Elementary elements as Parsable Character Data (#PCDATA)
  - ATTLIST

Straightforward Translation

- Main Idea: Use the scheme-tree forest directly as the DTD specification.
- Details:
  - Introduce a root name (e.g. <!DOCTYPE BandB [...]>).
  - Rename object-set names appearing in multiple trees.
  - Add the scheme trees as structural element (e.g. 
    <!ELEMENT BandB ((RoomNr, RoomName, ...)*)>).
  - Declare all object sets as elements with character data (e.g. 
    <!ELEMENT RoomNr (#PCDATA)>).
Example

RoomNr, RoomName, Cost, NrBeds (View)* (GuestNr, GuestName)*

```xml
<!DOCTYPE BandB [ 
  <!ELEMENT BandB ( 
    ( RoomNr, RoomName, Cost, NrBeds, (View)*, (GuestNr, GuestName)* )*)> 
  <!ELEMENT RoomNr (#PCDATA)> 
  <!ELEMENT RoomName (#PCDATA)> 
  ... 
  <!ELEMENT GuestName (#PCDATA)> ]>

<BandB>
  <RoomNr>1</RoomNr>
  <RoomName>Kennedy</RoomName>
  <Cost>90</Cost>
  <NrBeds>2</NrBeds>
  <View>Forest</View>
  <View>Sea</View>
  <GuestNr>101</GuestNr>
  <GuestName>Smith</GuestName>
  <RoomNr>2</RoomNr>
  ... 
</BandB>
```

A More Sophisticated Translation

- **Main Idea:** Nest the XML according to the scheme-tree structure
- **Details**
  - Create concept names for each node (e.g. Rooms).
  - Identify key elements for objects (e.g. RoomNr).
  - Nest according to node names and key elements (e.g. `<!ELEMENT Rooms (RoomNr)*>`).
  - Use XML attributes for key elements (e.g. `<!ATTLIST RoomNr value CDATA #REQUIRED>`).
Example

Rooms: RoomNr, RoomName, Cost, NrBeds
Views: View
Guests: GuestNr, GuestName

```xml
<rooms>
  <roomnr value="1">
    <roomname>Kennedy</roomname>
    <cost>90</cost>
    <nrbeds>2</nrbeds>
    <views>
      <view>Forest</view>
      <view>Sea</view>
    </views>
    <guests>
      <guestnr value="101">
        <gUESTNAME>Smith</gUESTNAME>
      </guestnr>
    </guests>
  </roomnr>
  ...
</rooms>
```

Additional Translations

- No unique solution
- Correspondences for additional ORM features
  - Roles: role attributes (e.g. role="Current Guest")
  - Optionals: zero or one occurrence (e.g. Room (Occupant?))
  - Partitions: $\cup$ (e.g. (OccupiedRoom | UnoccupiedRoom))
- Each translation maintains the properties of XNF
  - As few scheme trees as possible
  - No potential redundancy