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: NfBeds: 4, Cost: 4, RoomNr: 3, RoomName: 3, GuestName: 2
Back off by one.
Result: RoomNr, RoomName, Cost, NfBeds, (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 [...] >).
  - Declare all object sets as elements with character data (e.g. <!ELEMENT RoomNr (#PCDATA)>).

Example

```
<!DOCTYPE BandB [
  <!ELEMENT BandB (RoomNr, RoomName, Cost, NfBeds (View)*, (GuestNr, GuestName)* )>*
  <!ELEMENT RoomNr (#PCDATA)>
  <!ELEMENT RoomName (#PCDATA)>
  ...]
]
<BandB>
  <RoomNr>1</RoomNr>
  <RoomName>Kennedy</RoomName>
  <Cost>90</Cost>
  <NfBeds>2</NfBeds>
  <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

```xml
<!DOCTYPE Rooms [ 
<!ELEMENT Rooms (RoomNr)*>
<!ELEMENT RoomNr (RoomName, Cost, NrBeds,
Views, Guests)> 
<!ATTLIST RoomNr value CDATA #REQUIRED>
<!ELEMENT RoomName (#PCDATA)> 
<!ELEMENT Cost (#PCDATA)> 
<!ELEMENT NrBeds (#PCDATA)> 
<!ELEMENT Views (View)*>
<!ELEMENT View (#PCDATA)> 
<!ELEMENT Guests (GuestNr)*>
<!ELEMENT GuestNr (GuestName)> 
<!ATTLIST GuestNr value CDATA #REQUIRED>
<!ELEMENT GuestName (#PCDATA)> ]>

<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>
  <RoomNr value="2">
    ...
  </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: U (e.g. (OccupiedRoom | UnoccupiedRoom))
- Each translation maintains the properties of XNF
  - As few scheme trees as possible
  - No potential redundancy