## Patient records exercise

A clinic uses XML to transmit data about patients it has treated to its local hospital.

1. Write a single **well-formed** XML document to represent the information given below. You will need to spend some time deciding what **tags** to use and what information, if any, to represent using **attributes**.

   **Mrs Alison Avocado** (patient id 120372) had a consultation with Dr Claudio on the 20/11/2001 and then had her in-growing toenails treated on the 23/11/2001.

   **Miss Beatrice Beetroot** (patient id 234562) had consultations with Dr Duke on the 21/11/2001 and the 23/11/2001 and with Dr Claudio on the 28/11/2001.

   Data security and integrity is very important and so the clinic goes through a process of double checking the data to confirm it. At the current time the data for Mrs Avocado has been confirmed whereas the data for Miss Beetroot hasn't yet. It is important that the hospital knows which data has been confirmed and which hasn't.

2. Draw a hierarchical tree to represent your XML document

## Patient records exercise

There are many correct answers. Here's one:

```
<patient_details>
   <patient data_status="confirmed">
      <name>
         <title>Mrs</title>
         <firstname>Alison</firstname>
         <lastname>Advocado</lastname>
      </name>
      <patient_id>120372</patient_id>
      <consultation>
         <doctor_name>
            <title>Dr</title>
            <lastname>Claudio</lastname>
         </doctor_name>
         <date>22/11/2001</date>
      </consultation>
      <treatment>
         <type>ingrowing toenails</type>
         <date>23/11/2001</date>
      </treatment>
   </patient>
```

## Patient records exercise (continued)

```
  <patient data_status="unconfirmed">
    <name>
      <title>Miss</title>
      <firstname>Beatrice</firstname>
      <lastname>Beetroot</lastname>
    </name>
    <patient_id>234562</patient_id>
    <consultation>
      <doctor_name>
        <title>Dr</title>
        <lastname>Duke</lastname>
      </doctor_name>
      <date>21/11/2001</date>
    </consultation>
    <consultation>
      <doctor_name>
        <title>Dr</title>
        <lastname>Duke</lastname>
      </doctor_name>
      <date>23/11/2001</date>
    </consultation>
    <consultation>
      <doctor_name>
        <title>Dr</title>
        <lastname>Claudio</lastname>
      </doctor_name>
      <date>28/11/2001</date>
    </consultation>
  </patient>
</patient_details>
```

## Patient records exercise (continued)

Some good things about the given answer

- All the data is represented including which data is confirmed and which unconfirmed

- It is well-formed (e.g. single root element, correct nesting, all elements have end tags, attribute values quoted)

- Attributes have been used sensibly i.e. not to represent the data itself so much as something about the data (i.e. whether or not it has been confirmed)

Other observations about the answer

- Everyone doing this exercise will come up with a slightly different answer. Even if the structure is the same it is unlikely all the element and tag names will be. This means that organisations exchanging data using XML need to **agree** to use a comment set of elements and tags i.e. to design an XML based language. The rules for that language can be embodied in a Schema or a DTD

- The date hasn't been broken down into its constituent parts which may make it difficult to process. Perhaps it would be better to be explicit e.g. code the dates as:

```
<date>
    <dd>28</dd>
    <mm>11</mm>
    <yyyy>2001</yyyy>
</date>
```

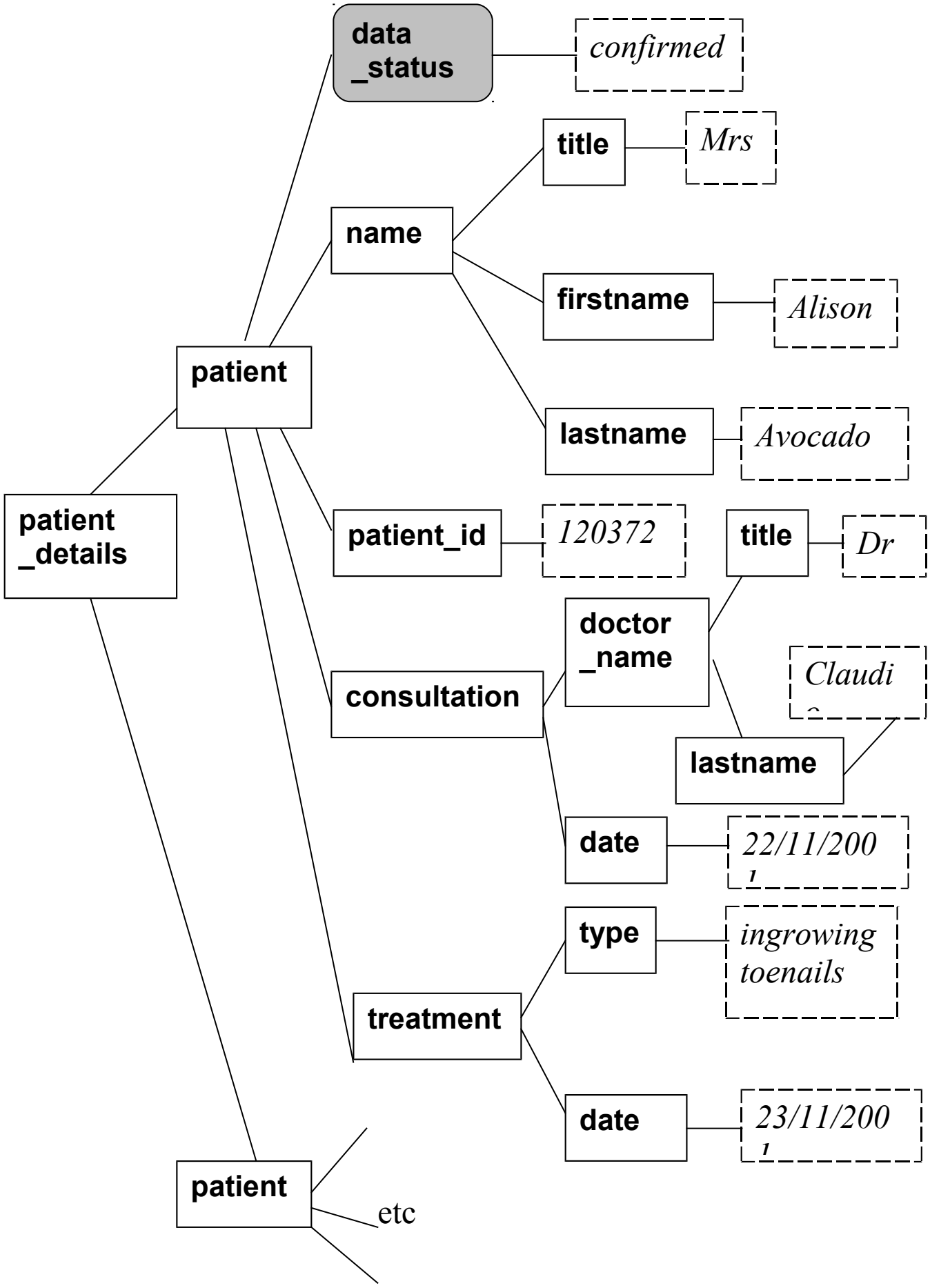
## Patient records exercise (continued)

Other observations about the answer (continued)

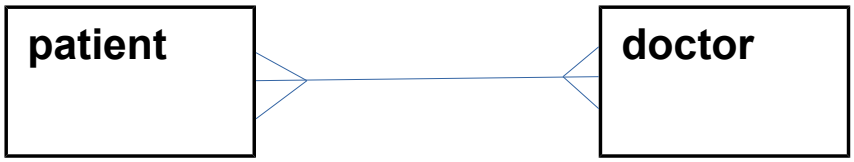
- Two elements represent names (patient and doctor) but have been given different tag names (name and doctor_name). Perhaps it would be better to represent the doctor for a consultation as:

```
<doctor >
   <name>
      <title>Dr</title>
      <lastname>Claudio</lastname>
   </name>
</doctor >
```

## Patient records exercise (continued)



- Could we refine this schema further?




- Doctors clearly treat many patients and patients are treated by one or more doctors. If this was a database we would want to normalise the schema to remove this N-to-N relationship.

- Before we fully normalise this perhaps we could take the doctors out of the consultation and put them somewhere else.

```
<clinician>
        <doctor_name did=”D123”>
          <title>Dr</title>
          <lastname>Duke</lastname>
        </doctor_name>
        <doctor_name did=”D456”>
          <title>Dr</title>
          <lastname>Claudio</lastname>
        </doctor_name>
</clinician>
```

- This can be used to reduce redundancy in the records

```
<patient_details>
  <patient data_status="confirmed">
    <name>
      <title>Mrs</title>
      <firstname>Alison</firstname>
      <lastname>Advocado</lastname>
    </name>
    <patient_id>120372</patient_id>
    <consultation cid=”D456”>
      <date>22/11/2001</date>
    </consultation>
    <treatment>
      <type>ingrowing toenails</type>
      <date>23/11/2001</date>
    </treatment>
  </patient>
```