Birkbeck
(University of London)

School of Computer Science and Information Systems

Development of Internet Applications (COIY032P)

Exam Solutions 2008

1. (a) The problem is that we want different content models for new and used cars. With a DTD, it is not possible to have a content model definition dependent on the value of an attribute. (4 marks)

   (b) Change the content model of `used-car` to exclude `condition` and add the following attribute declaration:

```
<!ATTLIST used-car
condition (as new | good) #IMPLIED >
```

   (6 marks)

   (c) `count(/stock/used-car[mileage<20000]/model)` (5 marks)

   (d)
```
<xsl:template match="/stock">
  <stock>
    <xsl:for-each select="new-car[price &lt; 10000]">
      <xsl:copy-of select="."/>
    </xsl:for-each>
    <xsl:for-each select="used-car[condition = 'as new']">
      <xsl:copy-of select="."/>
    </xsl:for-each>
  </stock>
</xsl:template>
```

   (10 marks)

2. (a)
```
<xsd:element name="book">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="title" type="xsd:string/>
      <xsd:element name="year" type="xsd:gYear"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="person">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="title">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Mr"/>
            <xsd:enumeration value="Ms"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="name" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Such a definition would not be possible using a DTD because there can be only one content model defined for `title`, i.e., element declarations are global. (18 marks)

(b) The hash symbol is used to delimit a resource reference from a fragment identifier. The question mark is used to delimit a resource reference from a query to the resource. (4 marks)

(c) The end of a resource is indicated either by using the HTTP `Content-length` header, or, if the length is not known in advance, by using chunked transfer encoding. (3 marks)

3. (a) XHTML is an XML vocabulary and hence all tags are required. HTML is defined using an SGML DTD where certain tags are identified as being optional. An HTML parser knows this DTD and can, if necessary, infer the presence of the missing tags because the DTD has been designed to ensure this can be done unambiguously. (4 marks)

(b) A namespace is declared by using a special attribute name, starting with `xmlns`, in the start tag of an XML element. If the default namespace is being declared, then the attribute name comprises simply `xmlns`. Otherwise this is followed by a *prefix* which will be used on the names of all elements to be considered part of the corresponding namespace. The value of the attribute gives the URI of the namespace. (8 marks)

(c) The sets of nodes denoted by each of the 5 axes do not overlap, and their union includes every node in the tree. (4 marks)

(d) Both are used as application-layer protocols running on top of IP. UDP provides a connectionless, unreliable service, whereas as TCP provides a connection-oriented, reliable service by detecting missing, corrupted and duplicate packets. (9 marks)

4. (a) MIME stands for Multipurpose Internet Mail Extensions. It was introduced to overcome the limitation of the 7-bit ASCII format used by SMTP which was inadequate for non-English and non-textual data. MIME allowed non-US-ASCII message bodies, an extensible set of different formats for non-textual message bodies, multi-part message bodies, and non-US-ASCII textual header information. (9 marks)

(b)
- i. the client is configured to use a proxy `P` as a cache
- ii. when the client requests a URI such as `http://S/doc.html` from server `S`, the request is sent to `P`, (not `S`)
- iii. if `P` has a copy of `doc.html`, and the copy has not expired, `P` returns it to client
- iv. else if `doc.html` has expired, `P` validates its copy with the original on `S`. `S` returns either
  - A. the HTTP status `304 not modified`, indicating `P`'s copy is up to date, or
  - B. a new version of `doc.html`
- v. else `P` doesn't have a copy of `doc.html`; `P` issues a `GET` request to `S`

Items 4(b)iii and 4(b)ivA correspond to a cache hit, while items 4(b)ivB and 4(b)v correspond to a cache miss. (16 marks)

5. The first 7 lines check whether the browser supports the `XMLHttpRequest` object: the first 2 lines for Mozilla-based browsers, the next two for Internet Explorer. `XMLHttpRequest` retrieves an XML file, given its URL, from the same domain as the loaded page. The general form of the `open` method is `open(method, URL, async)`, where the `method` parameter is any HTTP method (`GET` in the given fragment) and `async` specifies whether the script executes asynchronously with the request or not. The `escape` function escapes any characters that cannot usually appear in a URL to their ASCII hexadecimal values.

   Because the file must come from the same domain as the loaded page, the PHP script is used as a proxy just to retrieve the contents of the `url` and return it. The `false` argument means the call is synchronous: the Javascript interpreter waits for the response before continuing. The `send` method sends the request to the server; the content is `null` for the HTTP `GET` method—the message body is always empty. The `responseXML` method makes the HTTP response available as XML.

   In the PHP code, `header()` is used to send an HTTP header back as part of the response. The `Content-type` header is set with a MIME type of `text/xml` and a character set of `utf-8`. `$_GET['url']` gets the value of the `url` parameter from the query string. `file_get_contents` returns the file contents as a string which is echoed to the server which in turn returns it to the browser. (25 marks)

6. (a)

```
<site>
  <people>
    <person id="m1"><name>Jack</name></person>
    <person id="f1"><name>Jill</name></person>
  </people>
  <open_auctions>
    <open_auction>
      <initial>20</initial>
      <bidder>
        <personref person="f1"/>
        <increase>10</increase>
      </bidder>
      <seller person="m1"/>
    </open_auction>
    <open_auction>
      <initial>10</initial>
      <seller person="f1"/>
    </open_auction>
  </open_auctions>
</site>
```

(12 marks)

(b) The solution could use 1, 2 or 3 rules. The following is a sample solution using 2 rules.

```
<xsl:template match="/site">
  <html>
    <body>
      <table border="1">
        <tr>
          <th>Seller</th><th>Initial price</th>
          <th>Bidder</th><th>Increase</th>
        </tr>
        <xsl:apply-templates select="open_auctions/open_auction"/>
      </table>
    </body>
  </html>
</xsl:template>

<xsl:template match="open_auction">
  <tr>
    <td><xsl:value-of select="id(seller/@person)/name"/></td>
    <td><xsl:value-of select="initial"/></td>
  </tr>
  <xsl:for-each select="bidder">
    <tr>
      <td/><td/>
      <td><xsl:value-of select="id(personref/@person)/name"/></td>
      <td><xsl:value-of select="increase"/></td>
    </tr>
  </xsl:for-each>
</xsl:template>
```

(13 marks)