Mobile Internet Applications—XML-based Languages

Lesson 02

XML

A markup language

- For presentation of the marked (tagged) textual content
- An encapsulated text– processed, displayed, or printed as per the tag
- A browser– for the presentation

XML or XML-based language

- Not only encapsulates the data and metadata but can represent a behaviour or set of actions
- XML document– a text with the tags
- The XML document has an extension .xml
- A tag in the document specifies the meaning of the text encapsulated within the start and corresponding end tag

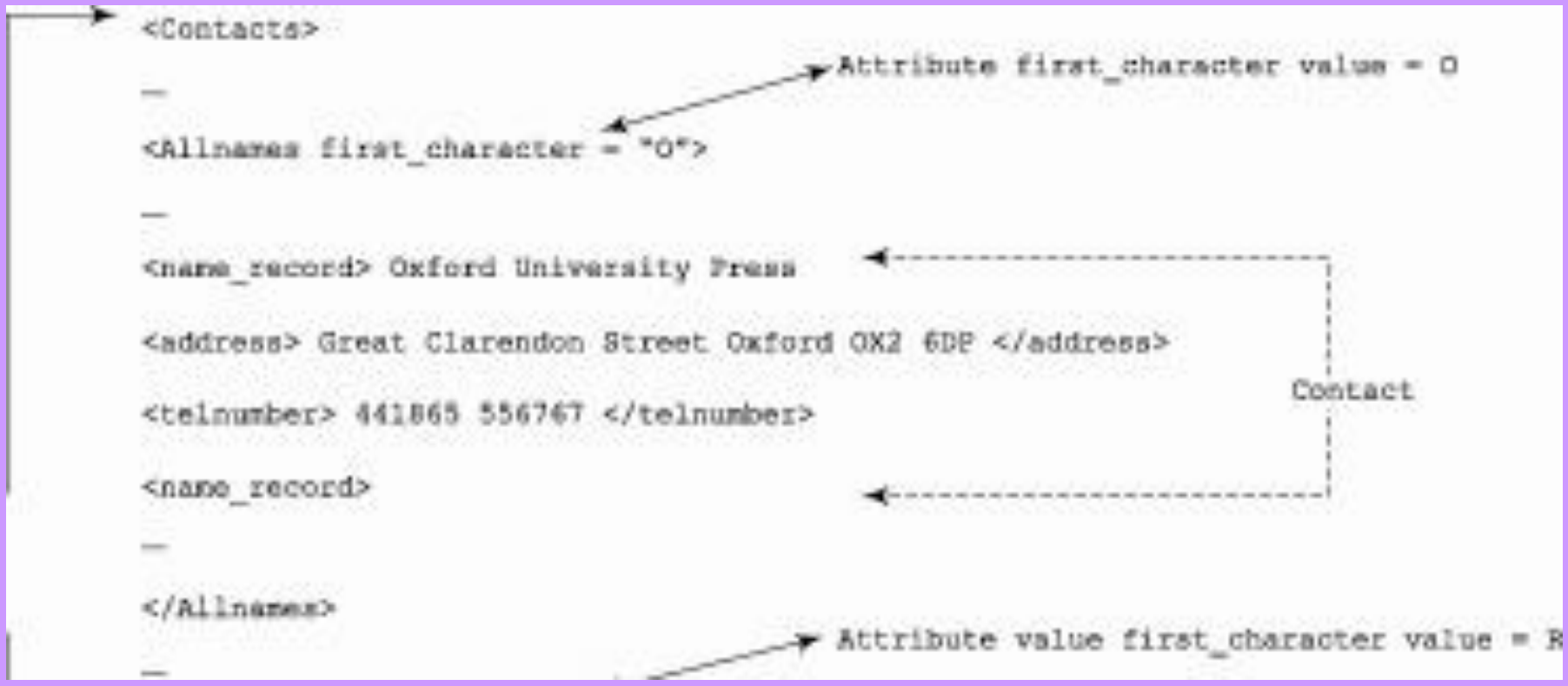
XML (EXTENSIBLE MARKUP LANGUAGE)

- A derivative of SGML (standard generalized markup language)
- Extensible– special instances of the tag-based languages can be defined such that each instance observes the fundamental rules of representing and structuring the XML document

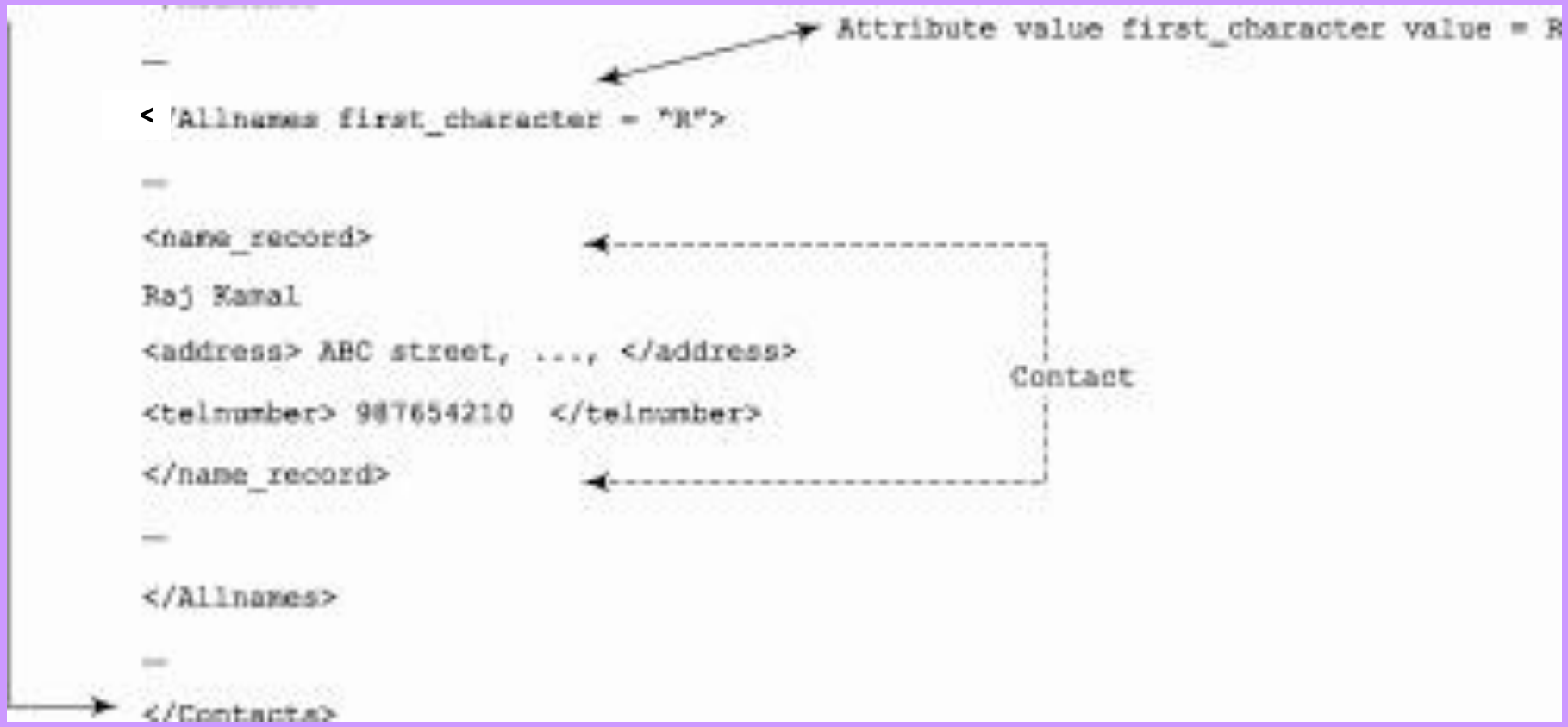
An XML-based language

- Uses the extensible property of XML to define the standardized sets of instances of the tags, attributes and their representation and behaviour and other characteristics
- Examples– SyncML

Example of XML document which can be used as contacts in a mobile smart phone



Example of XML document which can be used as contacts in a mobile smart phone



XML DOCUMENT REPRESENTATION OF A DATABASE

- Using tags and a pair of start and end tag in the document specifies the start and end of a record in the database
- A database– used to retrieve the specific record or set of records by querying the database or by business logic transaction

Example of a hierarchical structure in a mobile application – Contacts

- The elements name_record, address, and telnumber
- name_record has a textual content (Raj Kamal) and two elements address (ABC Street, ….) and telnumber (9876543210) within it
- A document formed by XML document creating a database named Contacts

Example of XML for textual document to represent a database

- <name_record> Raj Kamal <address> ABC Street, …. </address> <telnumber> 9876543210 </telnumber> </name_record>

XML DOCUMENT MAKING NON-TEXTUAL USE OF TEXT— command

- A tag can represent a command
- For processing the data using the command name within the pair of start and end tags in the document text
- A tag along with its attributes can specify the command, source file(s), and data to process the command

Example of Tags: Command, Command reference and message reference

- **<Cmd>Alert</Cmd>**
  <CmdRef>1</CmdRef>
  <MsgRef>1</MsgRef>
- When data is modified (for example, new email or newly downloaded music file) at the server (PC or remote mail server), the server alerts the client using message in XML

Example– <Cmd>Alert</Cmd> <CmdRef>1</CmdRef> <MsgRef>1</MsgRef>

- Command– Alert (means a server- or client-initiated notification)
- A number within CmdRef refers to a notification
- In place of transmitting full text of the server-notification for alerting the client, only a number is sent

Example– <Cmd>Alert</Cmd> <CmdRef>1</CmdRef> <MsgRef>1</MsgRef>

- The client at the other end interprets the notification from the referred number
- Command refers to a command referred by 1 within <CmdRef> and </CmdRef>
- Server message which is used to alert is identified by 1 as there is reference to 1 within the start and end tags <MsgRef> and </MsgRef>

SMIL Sample Code 9.5—<audio id = "my_id" src = "myaudio.wav" begin = "0s" dur = "10s"/>

- The tag and attributes together represent the system behaviour
- Command is audio
- Attribute id specifies the user id is my_id

SMIL Sample Code 9.5—<audio id = "my_id" src = "myaudio.wav" begin = "0s" dur = "10s"/>

- Attribute src specifies the source file, myaudio.wav
- Attribute begin specifies 0s time for start of playing the audio file

SMIL Sample Code 9.5—<audio id = "my_id" src = "myaudio.wav" begin = "0s" dur = "10s"/>

- Attribute dur specifies 10 s time for duration of playing the audio file,
- Command is to play a .wav file "myaudio.wav" and begins it from 0 second mark for a duration of 10 seconds

Document Type Defintions (DTDs) file or within the document

- Gives the specification of the role of text elements in a model document
- Specifies the attributes associated with an element as well as their valid values
- An external DTD document has file extension .dtd

Functions of DTD

- To enable validation of a document
- To specify which document structures can be used for authoring of the document
- To specify which structures a parser must handle

Parser

- A parser first validates an XML document and then handles the specified document structures
- Needs the DTDs for validation of the document

Validation

- Investigating–

(i) whether the document contains a root element with the same name as of DTD

(ii) whether it contains the header information for the version, encoding, and reference to other files

Validation

(iii) whether it contains the DTD with declaration of the markups in the document or in a linked external DTD document

Parsers

- kXML parser is a parser which parses XML document using J2ME and KVM (Kilo-byte virtual machine) in mobile devices
- .NET XML parsers in Microsoft Windows Mobile- and Windows CE-based devices
- WAP 1.x supports XML parser
- Enables processing of WML (wireless markup language)

Two models of an XML document

- One model is called SAX (simple API for XML) model
- Other DOM (document object model)
- Two types of parsers

SAX (simple API for XML)

- Each set of elements within the tags is independent and need not be considered as in a tree-like structure

SAX model APIs

- SAX provides the API a serial access and event-driven mechanism for reading and parsing data from an XML document
- SAX model document such that its parser can generate the series of events which are created as processing proceeds from beginning to end

Example of XML file contacts.xml having many contacts

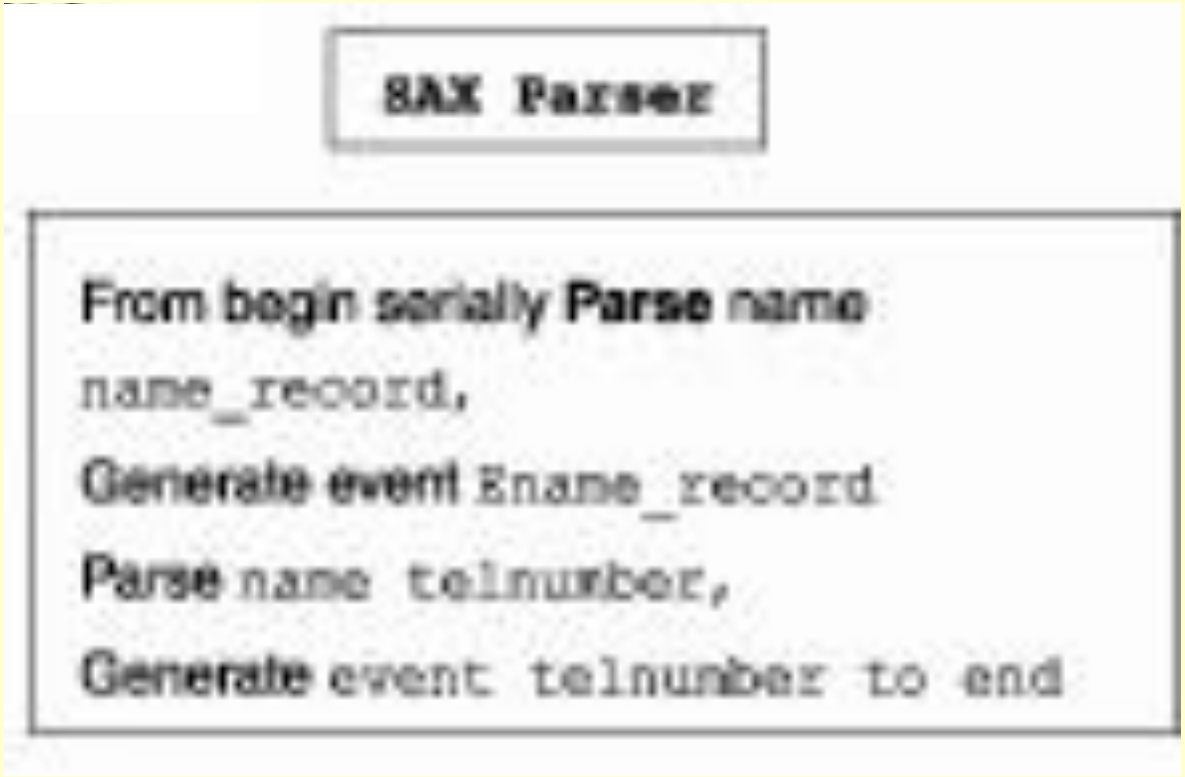
- Application needs to dial a contact with name record Raj Kamal and dial the corresponding telnumber

Example of SAX model of an XML document corresponding to Contacts

```
<Contacts>
...
<Allnames first_character = "O">
...
<name_record> Oxford University Press </name_record>
<address> Great Clarendon Street Oxford OX2 6DP </address>
<telnumber> 441865 556767 </telnumber>
```

Example of SAX model of an XML document corresponding to Contacts

Example of Parser for SAX model Contacts



Example

- Demonstrates a SAX parser of XML document and an alternative arrangement of elements for each contact in the Contacts which helps in fast processing by the parser

SAX parser

- Generates an event on name_record as it parses through the record and on event accepts the value ‘Raj Kamal’
- Then it generates another event on telnumber as it parses through the document and on event accepts the value ‘9876543210’

Advantages of XML SAX parser

- Entire document need not be first parsed thoroughly
- All the needed data need not be extracted

Document object model (DOM)

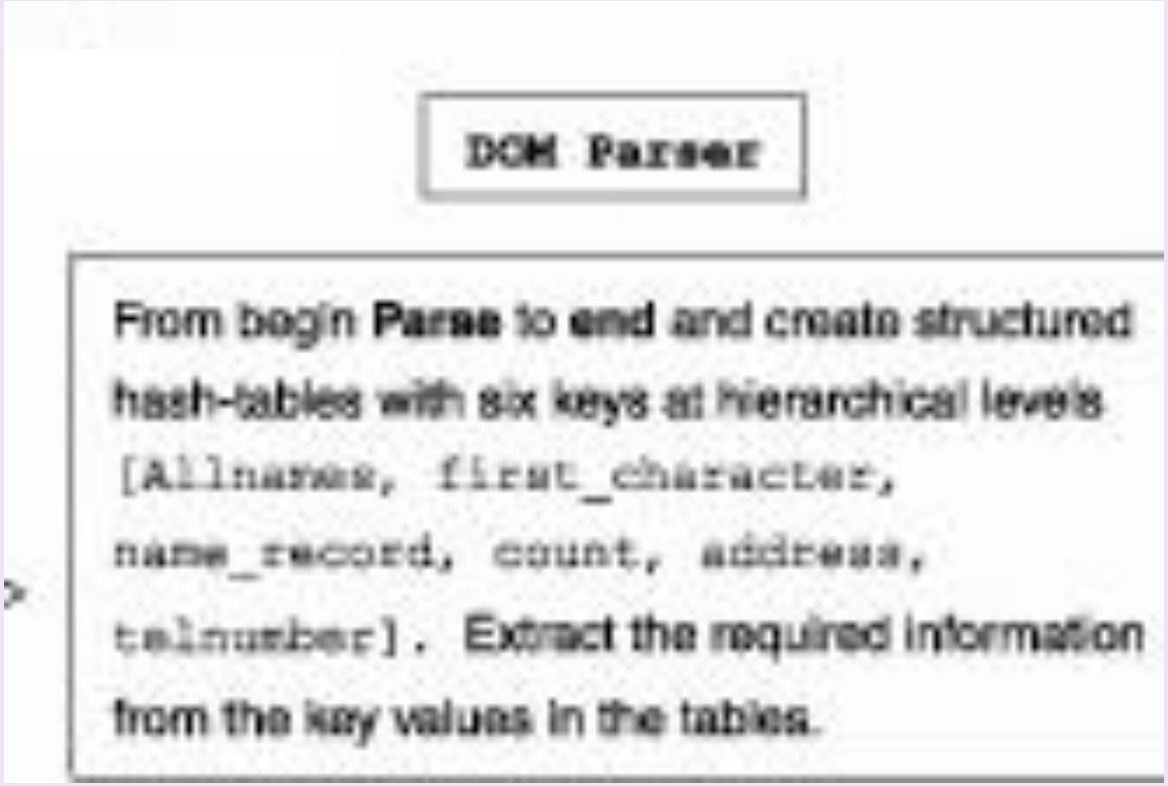
- In which each set of elements is dependent and derives from a root element
- Whole document forms a tree-like structure

DOM model of an XML document corresponding to Contacts

```
<Contacts>
...
<Allnames first_character = "O">
...
<name_record count = "22"> Oxford University Press
<address> Great Clarendon Street Oxford OX2 6DP </address>
<telnumber> 441865 556767 </telnumber>
```

DOM model of an XML document corresponding to Contacts

Parsing of DOM model document Contacts



DOM model document hierarchically arranged

- The whole document must be parsed initially to create a hash table of keys and corresponding values for each key

Advantage of DOM

- The structure is well-defined and thus same parser can be used for parsing all XML documents
- Later the interpreter or processing program is able to extract the desired information by simply using the keys

Disadvantage of DOM

- In an intermittently connected wireless environment or in case of a long document with many levels of hierarchy, it could take a long time before whole document is received at the parsing end

Example of the applications

1. Create ascending or descending order tabular data as per selection by the user
2. Count the number of contacts
3. Display the desired contact using appropriate GUI by deploying up-down menu keys in the keypad

Use of parsed inforamtion and data by an application

- Extracted output inforamtion and data from the parser further processing using a programming language
- The parsed data interprets at the application

XML TEXTUAL DOCUMENT FOR PLATFORM-INDEPENDENT APPLICATION DATA

1. Database
2. Data objects for synchronization, device configuration, user interface
3. Forms for processing of data at the server
4. Web applications at the server
5. Web services

XML TEXTUAL DOCUMENT FOR PLATFORM-INDEPENDENT APPLICATION DATA

6. Representing behaviour of or action by a tag and its attributes
7. An integrator of two diverse platforms for running an application
8. As a client application when sending a request to server for data or result of executing a method (routine) or search of database record at the associated backend server
9. Used by server to push the data to devices

XML

10. Used by WAP protocol for presentation of data to client using an XML browser
11. For multimodal user interfaces.
12. Internally for specifying the information in an application or framework

XML

13. Used by HTML browser after translating XML information into HTML information using a technology called XSLT (extensible style-sheet language transformation)
14. XHTML-MP format of XML used as HTML web pages with portability and extensibility in mobile devices

Use of Metadata at the XML document

- To represent the relationships among the data in the document
- To describe the structure and workable methods to facilitate an organized use of information
- To describe the method for manage that information

Use the metadata in XML document by an application

- To speed up and enrich searching for the resources
- Organising the information and managing the data

Summary

- Extensible markup language
- XML document– a text with the tags
- Attributes in the tag
- XML tag in the document specifies the meaning of the text encapsulated within the start and corresponding end tag
- XML Not only encapsulates the data and metadata but can represent a behaviour or set of actions or database records …

Summary

- Data type definition DTD internal or in separate file
- Parser for parsing XML document before running an application
- Two models of documents– SAX and DOM

Summary

- Each set of elements within the tags is independent and need not be considered as in a tree-like structure

Summary

- SAX provides the API a serial access for reading and parsing data
- SAX model document such that its parser can generate the series of events
- Entire document need not be first parsed thoroughly …

… Summary

- DOM model
- Hierarchical structure
- Entire document parsed first and then parsing creates hash table of keys and corresponding values for each key
- XML document data after parsing used in number of applications and databases
- Metadata also used in the applications

End of Lesson 02

XML