Extensible Markup Language (XML) – Revision Notes

What is XML?

- **XML** stands for *eXtensible Markup Language*.
- It's a **text-based syntax** for encoding structured data (words, phrases, numbers).
- Written using **Unicode characters** – no binary.
- **Extensible**: You can define your own tags.
- **Universal format**: All XML software must interpret the same XML identically.
- Commonly used for **data storage, messaging**, and **document structure**.

XML Parsers

- A **parser** reads XML and checks for syntactic correctness.
- Replaces **entity references** with their values (recursively).
- Two types:
  - **Validating Parser**:
    - Must read and validate **all entities** and **DTD content**.
    - Fails if it can't find any referenced entity.
  - **Non-validating Parser**:
    - Tries to retrieve all entities.
    - If it can't find some, it still proceeds and provides unresolved entity info to the application.

DTDs & Entities

- **DTD**: Document Type Definition – defines structure and legal elements/attributes.
- **Internal entities**: Defined within the XML file.
- **External entities**: Defined outside, may fail to load due to restrictions (e.g., firewall).
- Well-formed XML must follow DTD syntax rules if defined.

Entity Resolution

- Parser replaces all entity references with actual content.
- **Recursively resolved**, even inside other entities.

XML Processing APIs

1. **SAX (Simple API for XML)**
   - **Event-based** (fires events like startTag, textNode)
   - **Fast**, low memory
   - ❌ Not good for modifying XML in-memory
   - ☑ Good for streaming large files
2. **DOM (Document Object Model)**
   - **Tree-based**, object model of entire XML document
   - Allows full **access and modification**
   - ☑ Good for **dynamic modifications**, queries
   - ❌ Slower & memory-heavy
3. **JDOM**
   - Java-specific, **object-oriented** DOM
   - ☑ Easier to use than DOM, Java-only
   - ❌ Not part of core Java (yet), memory-heavy
4. **dom4j**
   - XML framework for Java
   - Supports **SAX, DOM, JDOM**, XSLT, JAXP
   - ☑ Mixed parsing: SAX + DOM
   - ☑ Open-source, Apache license

XSLT (eXtensible Stylesheet Language Transformations)

- Used to **transform XML documents** using **XSLT stylesheets**.
- Performs **tree transformations**, produces new XML/HTML.
- ☑ Best for **querying** and **restructuring XML**
- ❌ Can be slow and hard to debug

XML Messaging

- Use of XML for **message exchange between systems**.
- ☑ Self-describing, uses existing protocols (HTTP, SMTP, JMS, etc.)
- ❌ Asynchronous, large size, depends on shared contracts

Messaging Standards

1. **XML-RPC**
   - Simple XML encoding of function calls & parameters

- Easy remote invocation format

2. **SOAP (Simple Object Access Protocol)**
   - Complex message formatting
   - Supports **schemas, interfaces, proxies**
   - Key part of **.NET and WebSphere**

Quick Comparison Table

Feature	SAX	DOM	JDOM	XSLT
Type	Event-based	Tree-based	Tree-based	Stylesheet-based
Memory	Low	High	High	Medium
Speed	Fast	Slower	Slower	Medium
Editing	Manual	Easy	Easy	Not suitable
Language	All	All	Java only	All (XML-based)

Key Terms to Remember

- **DTD**: Defines structure (elements, attributes).
- **Entity**: Placeholder for data (internal/external).
- **Validating Parser**: Checks against DTD strictly.
- **SAX**: Fast, event-based parsing.
- **DOM**: Full in-memory tree representation.
- **XSLT**: Transform one XML structure to another.
- **SOAP/XML-RPC**: Messaging standards for remote communication.