Patent Application with XML at the USPTO:
The Electronic Legal Record

SPS

Software *Performance* Systems

Legal Records at the USPTO

- World's largest accumulation of intellectual property
  - 215,000 applications in FY1997
  - 275,000 applications in FY2002

- Legal record of an application is paper.

- The paper record is not manageable.

- Hence, USPTO's strategic plan:
  - Eliminate paper-based processing
  - Evolve the business to electronic commerce and electronic workplace
  - Reduce patent and trademark cycle time

Scope of Presentation

- Topics this presentation will *not* cover:
    - Data integrity
    - Authentication
    - Non-repudiatability
- Topics this presentation *will* cover:
    - Future-proofing legal records using XML
        - (eXtensible Markup Language)
    - Leveraging XML to implement PTO's strategic plan

SPS

Software *Performance* Systems

Future-Proofing: The Fundamental Requirement

- Future-proofing the *sine qua non* for the legal record at USPTO
  - Abandoned application: half-life of 20 years
  - Patent: half-life unknown–the life of the Republic
- Many electronic record formats are not future-proof
  - Data-format: half-life of 5 years
  - Corporations: half-life of 25 years
- Requirements for a future-proof electronic legal record:
  - Independent of software and storage medium
  - Self-descriptive to humans and machines

WHY XML?

- XML is a proper subset of SGML (ISO 8879).
    - Well respected international standard
    - Robust
    - Poorly marketed
- SGML met PTO requirements for electronic legal records:
    - Independent of software and storage medium
    - Self-descriptive to humans and machines
- XML is optimized for:
    - Web delivery
    - Tools development

SPS

Software *Performance* Systems

XML Syntax in Two Minutes

Content = data + markup

Element

Delimiter

Start tag

Attribute

End tag

```
<![DOCTYPE foo [●●●]><foo id="t1">Well-formed</foo>
```
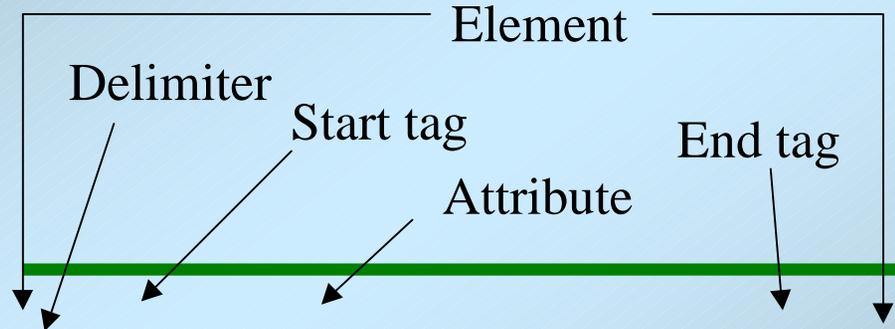
Valid

Content Model

```
D
T  <!ELEMENT foo (#PCDATA)>
D  <!ATTLIST foo id ID #REQUIRED>
```

Document = dtd? + instance

Generic identifier

Declared Value

Default value

SPS

Software *Performance* Systems

XML Enables Future-Proofing

- Documents are independent of software and medium.

  - XML content uses a standardized character set: Unicode.

- Documents are self-descriptive

  - Human-readable tag names...

  - … validated by a machine using a DTD (Document Type Definition)

SPS

Software *Performance* Systems

XML DTDs for a Future-Proof Electronic Legal Record

- Document analysis of all paper inputs.
  - Develop consistent English-like tag names.
  - Enforce business rules with DTDs.
    - "MPEP-centric"
- 25 DTDs for EFS and TEAM.
  - Non-bulk (forms-like)
  - Bulk
- All DTDs conform to a single architecture:
  - The Legal Instrument
  - Architectures are templates for DTDs (ISO 10744)

SPS

Software *Performance* Systems

Legal Instrument Architecture for PTO XML DTDs

- Pre-conditions
  - Is a filing correct and complete?
  - Has the filing been signed by an appropriate signer?

- Post-conditions
  - Unique to each instrument.

- Invariants
  - "Boilerplate"

- Parallel to notion of "programming by contract."

Quality Filings:
Leveraging the Legal Instrument Architecture (1)

- Filing correctly and completely.

- Pre-conditions are expressed as XML tags.

  - Example: `<paying-fees>`

- *Result*: Automated assistance to the applicant from the system.

  - Tag selection asserts the existence of a pre-condition.

  - System can verify assertion and offer assistance.

    - For example, that the correct fee has been paid, based on the preconditions for a document type.

SPS

Software *Performance* Systems

Reduced Cycle Time:
Leveraging the Legal Instrument Architecture (2)

- Enabling machine-assisted processing at USPTO.

- Post-conditions are expressed as XML tags.

    – Example: `<change-of-address>`

- *Result*: Filing system can now assist LIE in routing data and performing changes.

    – Tag selection indicates the appropriate post-condition.

    – System can then take appropriate action.

SPS

Software *Performance* Systems

Electronic Commerce:
Leveraging the Legal Instrument Architecture (3)

- Automating calculations

- Invariants are expressed as XML `#FIXED` attribute values
  - <additional-claim-fees> <surcharge-late-filing-fee-or-oath class="fee" amount="65">

- *Result*: Any XML document can be a storefront
  - DTD-driven
  - Using standard scripting languages (JavaScript, VBScript)

SPS

Software *Performance* Systems

Administrative Efficiency:
Leveraging the Legal Instrument Architecture (4)

- Simplifying administration
- Invariants
  - <subtotal-additional-fees class="subtotal">
  - CSS style sheet treats all elements of the same class in the same way
- *Result:* Stylesheet administration is simplified
  - 600+ elements over 26 DTDs
  - But less than 25 classes

Summary

- XML enables a future-proof electronic legal record
- Given an XML legal record, XML architectures yield:
    - Quality filings
    - Machine-assisted processing at PTO
    - Automated calculation
    - Simplified administration

SPS

Software *Performance* Systems