



Introducción a XML - Validación y Parseo

Huibert Aalbers, Senior Certified Software IT Architect



IT Insight podcast

- Este podcast pertenece a la serie IT Insight
- Pueden suscribirse al podcast a través de iTunes.
- El material adicional (presentación en formato PDF, white papers y otros) se obtienen directamente del sitio <http://www.huibert-aalbers.com> en la sección de IT Insight
- Me pueden enviar comentarios acerca del podcast o preguntas a mi correo personal, huibert_aalbers@mac.com



Introducción a XML

- XML proviene de (eXtensible Markup Language)
- XML surgió como una propuesta para simplificar SGML
- XML es un lenguaje pensado para describir datos
- La estructura de los datos de un archivo XML se valida a través de un DTD o de XML-Schema
- Los archivos XML junto con el DTD o el XML-Schema son no requieren de descripción adicional.
- XML se basa en recomendaciones de la W3C



Introducción a XML

- XML es gratis y extensible
- Cualquier persona puede crear un nuevo tipo de documento para definir un tipo de datos con sus propias reglas y tags.
- Existen ya muchos tipos de documentos XML estándar
 - MathML
 - SVG
 - TaxML
 - etc.
- Antes de crear un nuevo tipo de documento XML es recomendable verificar que no exista un estándar



Introducción a XML

- Reglas a seguir para crear un documento XML válido
 - Debe empezar por un encabezado estándar
 - `<?xml version="1.0" encoding="ISO-8859-1"?>`
 - Todos los tags deben cerrarse
 - `<persona><nombre>Huibert</nombre></persona>` es válido
 - `<persona><nombre>Huibert</persona>` no es válido
 - `<edad valor="15"/>` es válido
 - Los tags deben anidarse correctamente
 - `<persona><nombre>Huibert</persona></nombre>` no es válido



Introducción a XML

- Reglas a seguir para crear un documento XML válido
 - El documento debe tener un solo elemento raíz
 - Los atributos deben estar entre comillas (dobles o sencillas)
 - `<edad valor=15 />` no es válido
 - Los tags son sensibles al uso de mayúsculas/minúsculas
 - `<Persona><nombre>Huibert</nombre></persona>` no es válido
 - Los espacios en blanco se conservan
 - A diferencia de HTML en los que múltiples espacios se convierten a uno solo.
- Se pueden insertar comentarios en un archivo XML
 - `<!-- Esto es un comentario -->`



Introducción a XML

- XML es extensible
 - Si se cambia la estructura de un documento XML para que contenga más información, las aplicaciones que usen ese documento no deben verse afectadas.
- En los documentos XML existen relaciones de dependencia
 - Se habla de los hijos de un elemento o del padre.



Relación entre XML y HTML

- HTML no es una forma de XML
 - Porque se pueden crear documentos HTML válidos que no son documentos XML válidos.
 - Eso es un problema porque no hay una manera estándar de validar documentos HTML y cada browser los interpreta como puede
- xHTML es una nueva forma de describir documentos HTML basada en XML
 - Todos los browsers modernos soportan xHTML



Validación de documentos XML

- Un documento XML puede estar correctamente formado pero no ser válido si su estructura no es la esperada en base a su DTD o XML-Schema
- La validación de documentos XML se hace utilizando un parser
 - Los usuarios de Java normalmente utilizan Xerces, basado en XML4J, el primer parser Java donado a la comunidad por IBM
 - Contiene distintos tipos de parsers (SAX y DOM)
- El uso de DTDs (Document Type Definition) está cayendo en desuso
 - Porque no permite validar muchos tipos de datos comunes
 - Porque no está basado en XML



SAX vs DOM

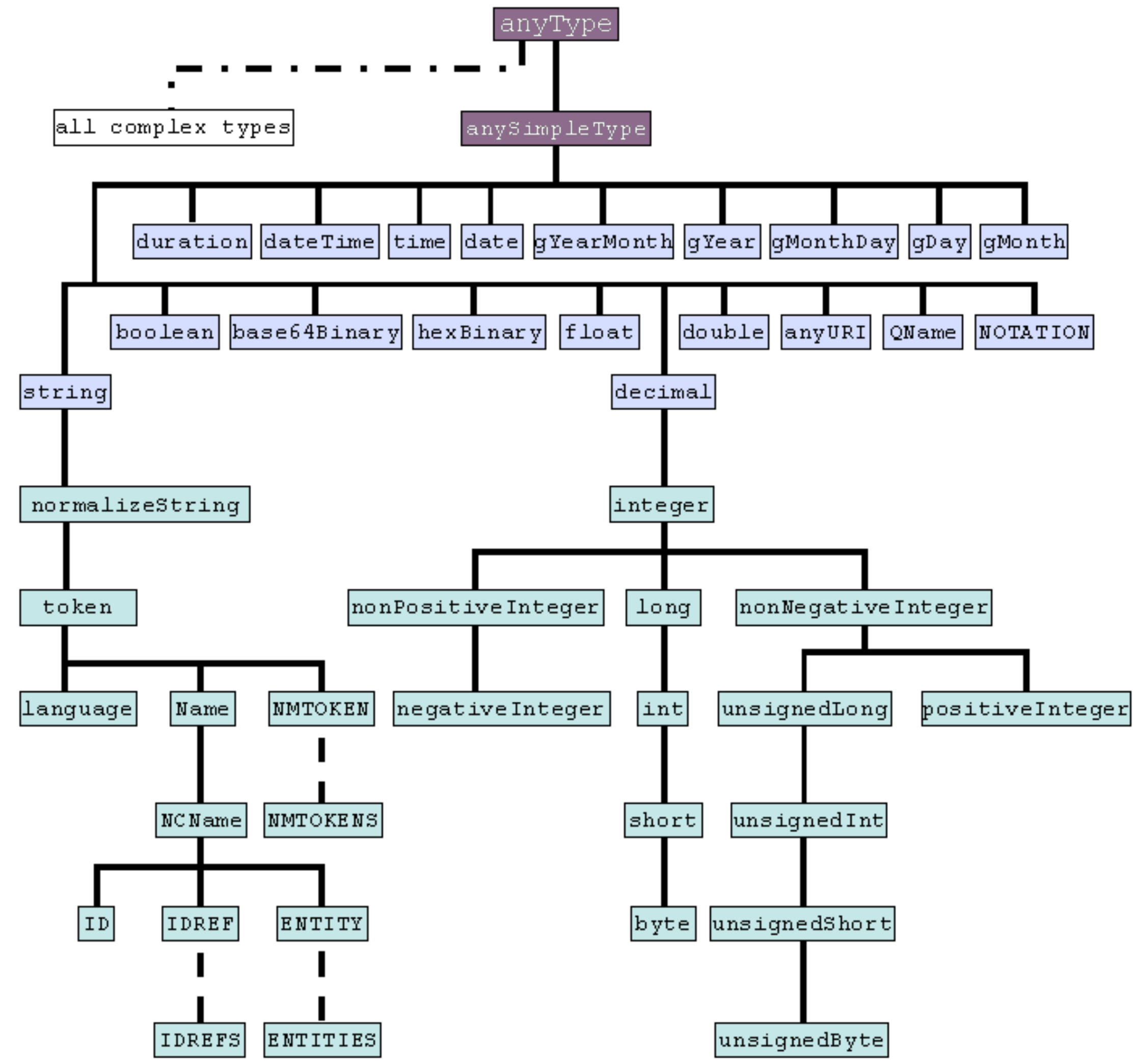
- SAX se refiere a Simple API for XML processing
 - El documento es leído en una sola pasada y diferentes eventos son generados durante el proceso
 - SAX se usa principalmente para procesar rápidamente un documento que no necesita ser modificado
- DOM son las siglas de Document Object Model
 - El documento XML completo es cargado en memoria
 - El API incluye métodos para añadir, borrar o modificar nodos y extremidades
 - A diferencia de SAX, DOM permite a los desarrolladores navegar el documento en cualquier dirección
 - El poder adicional que ofrece DOM se traduce en un mayor costo en términos de uso de memoria



Introducción a XML-Schema

- Se trata de una alternativa a DTD, basada en XML
- También se conoce como XML-Schema Definition (xsd)
- Con XML-Schema es más fácil la integración de XML con bases de datos relacionales
 - En general es sencillo convertir el esquema de una tabla, incluyendo constraints a un archivo XSD
- XML-Schema permite la creación de nuevos tipos de datos
 - A partir de tipos de datos existentes
 - Utilizando restricción, extensión y herencia
 - Tipos de datos totalmente nuevos

XML Schema



Introducción a XML-Schema

Imaginemos el siguiente archivo XML de ejemplo:

```
<?xml version="1.0"?>
<mensaje>
  <to>Tove</to>
  <from>Jani</from>
  <titulo>No olvides</titulo>
  <body>No olvides la junta del grupo de usuarios de WebSphere</body>
</mensaje>
```

Si lo validáramos con un DTD, usaríamos el siguiente:

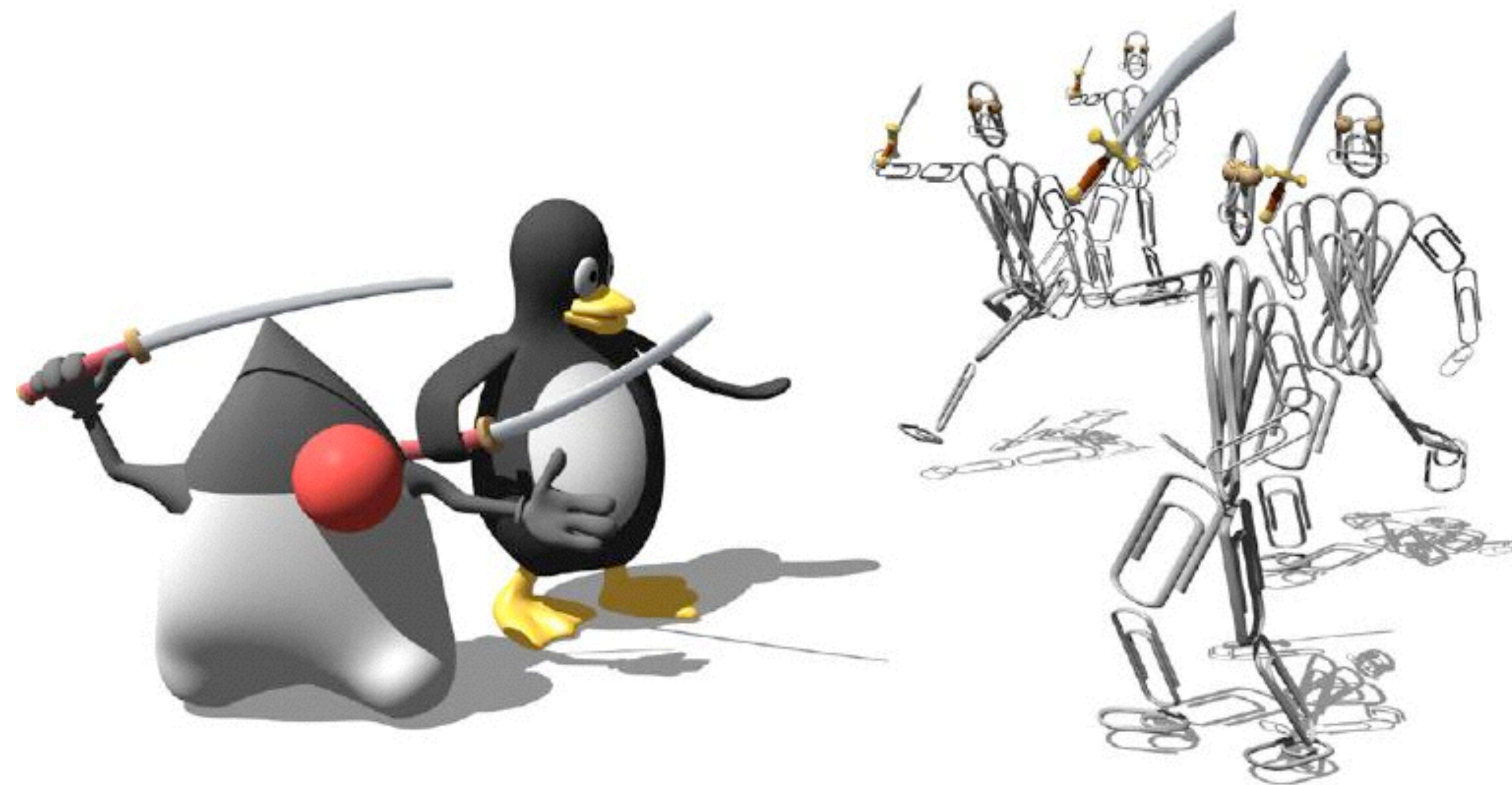
```
<!ELEMENT mensaje (to, from, titulo, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

Introducción a XML-Schema

Con XML-Schema usaríamos el siguiente archivo:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com"
xmlns="http://www.ibm.com" elementFormDefault="qualified">

  <xs:element name="mensaje">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="titulo" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



Para mayor información, contácteme directamente
huibert_aalbers@mac.com