

Xdefinition® 2.1

Václav Trojan

Synteia Software Group

email: vaclav.trojan@synteia.cz

<http://xdef.synteia.cz>

How the cat and dog baked a cake



Xdefinition 2.1

The cake baked for

- Analysts
- System designers
- Data and interface description
- Programmers
- Everybody (human readable)

Used for

- Validation (possibly external)
- Event handling (errors, data actions)
- Data processing (unlimited size)
- XML (valid) data generation
- Transformation XML → XML
- New way of programming XML
- Connection with external code (java, XPath functions, ...)

Concept of Xdefinition

Example of XML data:

```
<Book cover = "hard">  
  <Title>  
    Harry Potter  
  </Title>  
  <Author>  
    J. K. Rowling  
  </Author>  
  <ISBN>  
    9780439887458  
  </ISBN>  
</Book>
```

Concept of Xdefinition

Remove data values

```
<Book cover = "    ">  
  <Title>  
  
  </Title>  
  <Author>  
  
  </Author>  
  <ISBN>  
  
  </ISBN>  
</Book>
```

Concept of Xdefinition

Replace data values by description of data format (data type) and insert data model to Xdefinition envelope

```
<xd:def xmlns:xd = "http://cz.syntea.xdef/2.1"  
  name = "Example">
```

```
  <Book cover = "tokens ('hard|paperback') ">
```

```
    <Title>
```

```
      string
```

```
    </Title>
```

```
    <Author>
```

```
      string
```

```
    </Author>
```

```
    <ISBN>
```

```
      integer
```

```
    </ISBN>
```

```
  </Book>
```

```
</xd:def>
```

Concept of Xdefinition

Add specification data occurrence:

```
<xd:def xmlns:xd = "http://cz.syntea.xdef/2.1"
  name = "Example">

  <Book cover = "optional tokens('hard|paperback') ">
    <Title xd:script = "required">
      required string
    </Title>
    <Author xd:script = "unbound">
      required string
    </Author>
    <ISBN xd:script = "optional">
      required integer
    </ISBN>
  </Book>

</xd:def>
```


Concept of Xdefinition

„Required“ occurrence is default; it may be omitted. You can also specify occurrence of elements explicitly:

```
<xd:def xmlns:xd = "http://cz.syntea.xdef/2.1"
  name = "Example">

  <Book cover = "optional tokens('hard|paperback') ">
    <Title>
      string
    </Title>
    <Author xd:script = "occurs 0,*">
      string
    </Author>
    <ISBN xd:script = "occurs 0,1">
      integer
    </ISBN>
  </Book>

</xd:def>
```

Events and actions in Xdefinitions

Action connected to event “create”:

```
<Book cover = "optional tokens ('hard|paperback') ;
    create getColumn('cover') ">

<Title>
    string; create getColumn('title')
</Title>
<Author xd:script = "unlimited;
    create hasValue('title') ">
    string; create getValue('title')
</Author>
<ISBN xd:script = "optional;
    create hasValue('title') ">
    Integer; create getValue('title')
</ISBN>
</Book>
```

Events and actions in Xdefinitions

Example of “create” mode with XML source:

```
<Books>
  <Book xd:script = "unlimited; create from('//item') "
    cover = "optional tokens('hard|paperback')">
    <Title> string; create from('@title') </Title>
    <Author xd:script = "unlimited;
      create from('authors/author')">
      string
    </Author>
  </Book>
</Books>
```

```
<library>
  <item cover = "paperback"
    title = "Harry Potter" );
  <authors>
    <author>J.R. Rowling</author>
  </authors>
</item>
```

...

```
</library>
```

Events and actions in Xdefinitions

Example of other actions:

```
<Book cover = "optional tokens ('hard|paperback') ;
    onError putError('Incorrect cover') ;
    create getColumn('cover') ">
  <Title>
    string; create getColumn('title')
  </Title>
  <Author xd:script = "unlimited;
    create hasValue('title') ">
    string; create getValue('title')
  </Author>
  <ISBN xd:script = "optional;
    create getValue('title') ">
    integer; onError putError('Incorrect ISBN') ;
    create getValue('title')
  </ISBN>
</Book>
```

Events and actions in Xdefinitions

Example of other actions:

```
<Book cover = "optional tokens ('hard|paperback') ;
    parsed display(getValue()) ;
    onError putError('Incorrect cover') ;
    onAbsence display('cover unknown') ">
  <Title>
    string; parsed display(getValue()) ;
  </Title>
  <Author xd:script = "unlimited">
    string; parsed display(getValue())
  </Author>
  <ISBN xd:script = "optional">
    integer; parsed display(getValue()) ;
    onError putError('Incorrect ISBN') ;
  </ISBN>
</Book>
```

Alternative models - event “match”

```
<xd:choice>
```

```
  <shape xd:script="match xpath(@kind = 'rectangle') "  
    kind = "fixed 'rectangle' "  
    A = "float" B = "float"/>
```

```
  <shape xd:script="match xpath(@kind = 'circle') "  
    kind = "fixed 'circle' "  
    r = "float"/>
```

```
</xd:choice>
```

Alternative models - event “match”

```
<xd:choice>
```

```
  <shape xd:script="match (@kind = 'rectangle');  
    ref rectangle"/>
```

```
  <shape xd:script="match (@kind = 'circle');  
    Ref circle" />
```

```
</xd:choice>
```

```
<rectangle kind = "fixed 'rectangle'"  
  A = "float" B = "float"/>
```

```
<circle kind = "fixed 'circle'"  
  r = "float"/>
```

Alternative models - event “match”

```
<xd:choice>
```

```
  <shape xd:script="match xpath($sw = 'rectangle');  
    ref rectangle"/>
```

```
  <shape xd:script="match xpath($sw = 'circle');  
    Ref circle" />
```

```
</xd:choice>
```

```
<rectangle kind = "fixed 'rectangle'"  
  A = "float" B = "float"/>
```

```
<circle kind = "fixed 'circle'"  
  r = "float"/>
```


Event finally, forget

```
<Very_large_XML_Document
  xd:script="finally printResult(getCount());" >

  <item xd:text = "string()"
    xd:script="occurs *; ... ; forget" >
    ...
  </item>

  ...

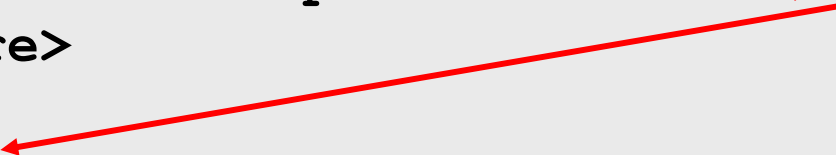
</Very_large_XML_Document>
```

References

Example of reference to model:

```
<Store>
  <Book xd:script = "unlimited; ref Book" />
</Store>

<Book cover = "optional tokens('hard|paperback')">
  <Title>
    string
  </Title>
  <Author xd:script = "unlimited">
    string
  </Author>
  <ISBN xd:script = "optional">
    integer
  </ISBN>
</Book>
```



Groups

Three groups: choice, sequence, mixed

```
<root>  
  <xd:choice>  
    <foo/>  
    <bar/>  
  </xd:choice>  
  <xd:sequence>  
    <foo/>  
    <bar/>  
  </xd:sequence>  
  <xd:mixed>  
    <foo/>  
    <bar/>  
  </xd:mixed>  
</root>
```

Groups

Script of group :

```
<root>
  <xd:choice script = "optional">
    <foo xd:script = "optional"/>
    <bar xd:script = "optional"/>
  </xd:choice>
  <xd:sequence script = "occurs 3">
    <foo/>
    <bar xd:script = "optional"/>
  </xd:sequence>
  <xd:mixed>
    <foo xd:script = "optional"/>
    <bar xd:script = "unbound"/>
  </xd:mixed>
</root>
```

Groups


Script in group items:

```
<root>
  <xd:choice script = "optional">
    <foo xd:script = "optional" />
    <bar xd:script = "unbound" />
  </xd:choice>
  <xd:sequence script = "occurs 3">
    <foo xd:script = "optional" />
    <bar xd:script = "unbound" />
  </xd:sequence>
  <xd:mixed>
    <foo xd:script = "optional" />
    <bar xd:script = "unbound" />
  </xd:mixed>
</root>
```

Groups

Reference to group:

```
<root>
  <xd:choice script = "optional; ref foobar"/>
    <foo/>
    <bar/>
  </xd:choice>
</root>
```



```
<xd:choice name = "foobars">
  <foo/>
  <bar/>
</xd:choice>
```

References between Xdefinitions

```
<xd:def xmlns:xd = "http://cz.syntea.xdef/2.1"
  name = "A">
  <root>
    <foo xd:script = "ref B#foo"/>
  </root>
  <bar/>
</xd:def>
```

```
<xd:def xmlns:xd = "http://cz.syntea.xdef/2.1"
  name = "B">
  <foo>
    <bar xd:script = "ref A#bar"/>
  </foo>
</xd:def>
```

Collection of Xdefinitions

```
<xd:collection xmlns:xd="http://cz.syntea.xdef/2.1">  
  
</xd:def name = "A">  
  <root>  
    <foo xd:script = "ref B#foo"/>  
    <root>  
    <bar/>  
</xd:def>  
  
<xd:def name = "B">  
  <foo>  
    <bar xd:script = "ref A#bar"/>  
  </foo>  
</xd:def>  
  
</xd:collection>
```


Anotation in Xdefinition

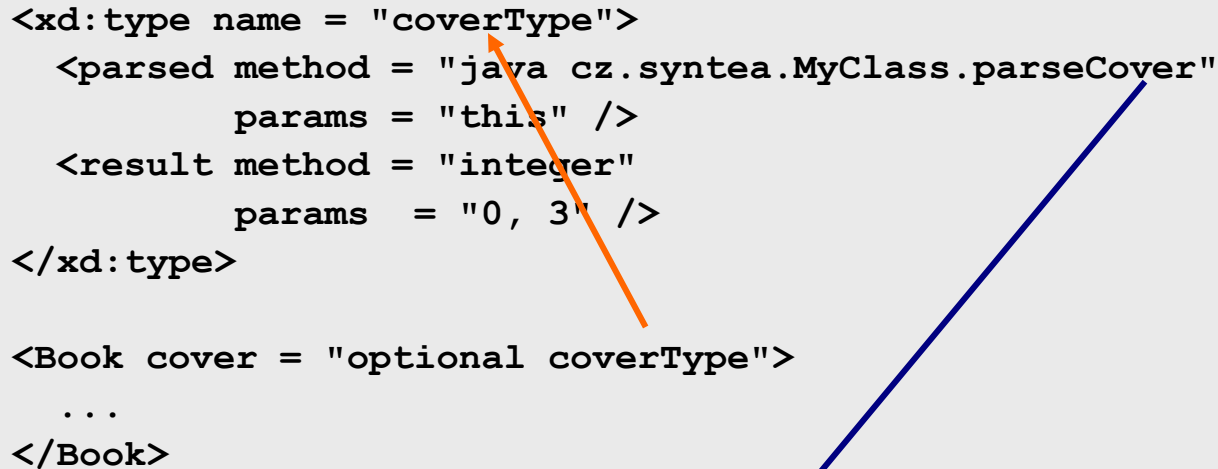
Anotation – special text comment in script

```
<Book cover = "optional tokens ('hard|paperback' ) ">
  /** anotation{ here can be any text } */
  <Title>
    string /** anotation{...} */
  </Title>
  <Author xd:script = "unlimited">
    string
  </Author>
  <ISBN xd:script = "optional
    /** anotation{...} */">
    integer
  </ISBN>
</Book>
```

User defined type

```
<xd:type name = "coverType">
  <parsed method = "java cz.syntea.MyClass.parseCover"
    params = "this" />
  <result method = "integer"
    params = "0, 3" />
</xd:type>

<Book cover = "optional coverType">
  ...
</Book>
```



External method in Java:

```
public static boolean parseCover(cz.syntea.xdef.XdNode xnode) {
    org.w3c.dom.Node n = xnode.getXMLNode();
    String s = n.getValue();
    if ("hard".equals(s)) {
        n.setValue("1");
    } else if ("paperback".equals(s)) {
        n.setValue("2");
    } else {
        n.setValue("0");
        xnode.setError("Incorrect value of book cover type");
        return false;
    }
    return true;
}
```

Public objects

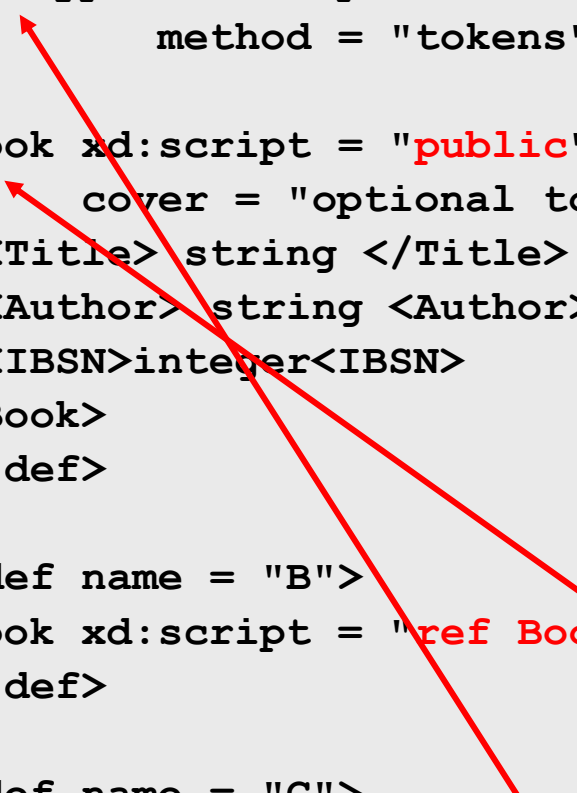
Element models and types can be public:

```
<xd:def name = "A">
  <xd:type name = "public coverType"
    method = "tokens" params = "'hard|paperback'" />

  <Book xd:script = "public"
    cover = "optional tokens('hard|paperback')">
    <Title> string </Title>
    <Author> string <Author>
    <IBSN>integer<IBSN>
  </Book>
</xd:def>

<xd:def name = "B">
  <Book xd:script = "ref Book"/>
</xd:def>

<xd:def name = "C">
  <MyBook cover = "coverType"/>
</xd:def>
```

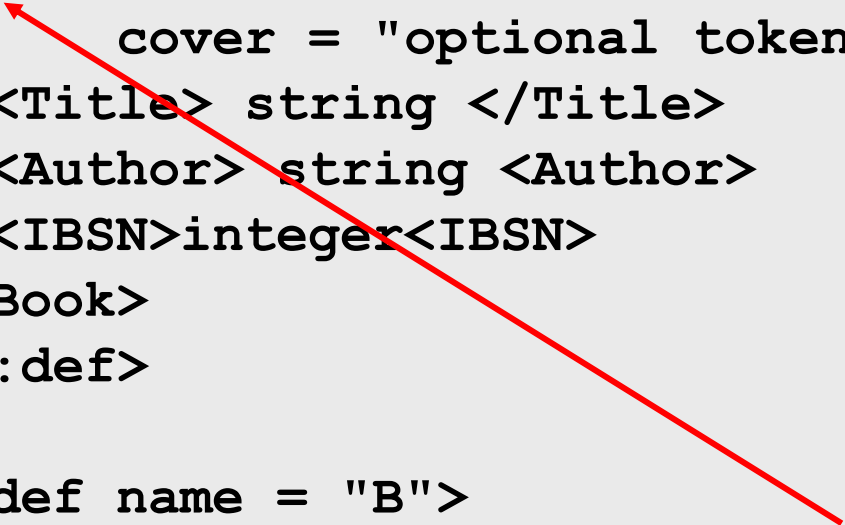


Abstract Xdefinition and extension

Extension of abstract model:

```
<xd:def name = "A">
  <Book xd:script = "abstract"
    cover = "optional tokens('hard|paperback')">
    <Title> string </Title>
    <Author> string <Author>
    <IBSN>integer<IBSN>
  </Book>
</xd:def>

<xd:def name = "B">
  <Book xd:script = "extends A#Book; finally save()">
    <Title> parsed outln(getValue) </Title>
    <Author> parsed outln(getValue) </Author>
  </Book>
</xd:def>
```



Final dezert: Xdefinition of Xdefinitions

```
<meta:def xmlns:meta="http://myURI"
  xmlns:xd="http://cz.syntea.xdef/2.1"
  xd:metaNamespace = "http://myURI">

  <xd:def name = "string">
    <root>
      </meta:choice>
      <foo>
      <bar/>
      </meta:choice>
    </root>
  </xd:def>

</meta:def>
```

Compatibility with validation tools

DTD --> Xdefinition (full)

XML schema --> Xdefinition (full?)

(RELAX NG --> Xdefinition) (full?)

Xdefinition --> DTD (subset)

Xdefinition --> XML Schema (subset)

(Xdefinition --> RELAX NG) (subset)

Implementation

- Java (1.3, ...)
- J#
- NetBeans

