NEW EXTENDED AND QUANTIFIED CONSTRAINTS IN XML SCHEMA

A. Duffoux
Groupe esaip - 18 rue du 8 mai 1945 49180 Saint Barthelemy d'Anjou, France
LERIA - 2 Boulevard Lavoisier 49045 Angers cedex 01

B. Duval, S. Loiseau
LERIA - 2 Boulevard Lavoisier 49045 Angers cedex 01, France

Keywords: XML Schema, Quantified constraint, Conditional constraint, Validation.

Abstract: In this paper, we present a new model to represent complex constraints in XML schema. Due to its flexibility and its capacity to describe all kinds of data, XML has become a widely used exchange format during the last years. Hence, such data have been integrated in several information systems. However these systems need strengthness and coherence that XML in its primary form can not provide. We thus propose to extend classical XML schema in order to integrate a quantification of constraints and to allow conditional constraints on several elements. Thanks to this extension, XML applications can have a richer and stronger framework. To illustrate the use of this new model, we present a case study concerning XML curriculum vitae treatment.

1 INTRODUCTION

eXtensible Markup Language format (Bray et al., 2004) has become the new data exchange standard. The success met by XML is due to its flexibility and its capacity to describe all kinds of data. Indeed, XML documents convey not only the data but also their description. This description is made thanks to the concept of XML schema[1] which is used as a grammar to validate the data description. This grammar can be specifically adapted to each application description. It defines the attended structure of processed documents and can specify some *simple constraints* on the data embedded in them. These *simple constraints* are essentially based on the document structure - presence or absence of elements[2], cardinality, imbrication. Unfortunately, most of the current schema formalisms are not powerful enough to deal efficiently with more complex constraints like constraints concerning several elements at any level of the document hierarchy. We thus need to develop specific methods to deal with such constraints and validate the data description.

A lot of studies have been lead in order to add complex constraints to XML schemas. We can classify them according to two different approaches : one that we call "integrated" (Thompson et al., 2001; Clark and Murata, 2001) and the second that we call "composed" (Jacinto et al., 2003; Jelliffe, 2001). The integrated solutions are the more used. They essentially specify *simple structure constraints*. Some of them (Thompson et al., 2001) also express some *content constraints* - enumeration, domain range checking, pattern matching. Their advantages are that they need only one specification document - the schema - and only one validation. However, they are essentially based on structure definition and take a little into account the content. The composed solutions are more complete. They allow to express a wide set of rich *content constraints* on XML documents, using for example aggregation functions. These constraints are expressed in XML compatible languages and use existing XML technologies such XPath (Clark and DeRose, 1999) or XSL (Berglund, 2006). Their drawback is that they can not affect the validation result. Furthermore, they need at least two specification documents, one for the schema and another one for the description of constraints, which implies at least two

[1] In this paper, we differenciate XML schema which is a generical term and W3C XML Schema which is the specific W3C definition

[2] In the rest of this paper, we will use *element* indifferently for elements and attributes

Duffoux A., Duval B. and Loiseau S. (2009).
NEW EXTENDED AND QUANTIFIED CONSTRAINTS IN XML SCHEMA.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 261-266
DOI: 10.5220/0001662202610266

validation steps. Moreover, the constraint definition is totally disconnected from the type definition.

In this paper, we present eqCXSD, for *extended quantified Constraints in XML Schema Definition*, a more complete solution capable to express *quantified and conditional constraints* on any element or set of elements. We extend a classical "integrated" XML solution, W3C XML schema format (Thompson et al., 2001), with *constraint quantification* and *conditional content constraints*. The constraint quantification is made thanks to a quantifier added to the element concerned by the constraint. It expresses how much occurences of the element is concerned by the constraint. The proposed extension is made in the same Schema to keep the coherence between the element structure definition and its constraints.

We first present the expressiveness of XML schemas. Second, we present extended constraints on XML elements. Third, we introduce our new extended formalism. Fourth, we present the XML translation in W3C XML Schema of our new formalism.

2 XML SCHEMAS AND THEIR EXPRESSIVENESS

An XML document can been seen as an unranked ordered and labelled tree composed of simple or complex elements. The general structure of the document and its elements definition are given in its schema. This schema essentially describes structural constraints on XML documents. There are several schemas formalisms, the most known and used are W3C XML Schema (Thompson et al., 2001) and Relax NG (Clark and Murata, 2001). Each of them have different constraint mechanisms and expressiveness. A detailed description and comparison of these schema languages can be found in (Murata et al., 2005; Lee and Chu, 2000). In (Mani and Lee, 2002), the authors have proposed a new formalism, called XSchema, which matches with XML Schemas.

Definition 1 (Mani and Lee, 2002). An XSchema is a 6-tuple $X = (E, A, M, P, r, \Sigma)$ where :

- E is a finite set of element names,
- A is a function from an element name $e \in E$ to a set of attribute names a,
- M is a function from an element name $e \in E$ to its element type definition α where α is
$$\alpha ::= \varepsilon \mid \tau \mid \alpha + \alpha \mid \alpha,\alpha \mid \alpha^* \mid \alpha^? \mid \alpha^+$$
where ε denotes the empty element, τ is an atomic data type (e.g., String, Integer, ...), "+" the union, "," the concatenation, α^* for the Kleene star, $\alpha^?$ for $(\alpha + \varepsilon)$ and α^+ for (α , α^*),
- P is a function from an attribute name a to its attribute definition $\beta = (\tau, n, d, f)$ where n is either nullable or not nullable, d is a finite set of valid domain values of a (that can be ε) and f a default value of a (that can be ε),
- $r \subseteq E$ is a finite set of root elements,
- Σ is a finite set of integrity constraints. These constraints concern ID and IDREF elements and represent XML keys and foreign keys.

To illustrate the notions presented in this article, we use a case study concerning the selection of candidates in a Master program. These candidates are represented by their XML Curriculum Vitae. The following example shows a XSchema corresponding to a part of our case study. We represent the candidate experience with its scholar experiences, its professional experiences and its competencies.

Example 1 : $G_1 = (E, A, M, P, r, \Sigma)$ is a XSchema.

- E ={Experience, Diploma, Degree, Year, Mark, Field, Professional, Competence, Type}
- M = {Experience $\longmapsto$ (Diploma+, Professional*, Competence*);
Diploma $\longmapsto$ (Degree, Year, Mark, Field);
Degree $\longmapsto$ String; Year $\longmapsto$ Integer;
Mark $\longmapsto$ float; Field $\longmapsto$ String;
Professional, Competence $\longmapsto$ (Type, Field);
Type $\longmapsto$ String}
- r ={Experience}
- $A, P, \Sigma = \emptyset$

G1 defines the expected structure of a CV. E is the set of all the element names. Their definition are expressed in M. A *complex type* definition is composed of subelements, like *Experience* which is composed by at least one *Diploma* and several or no *Professional* and *Competence*. A *simple type* definition is an atomic data type, like *Mark* which is a float. *Experience* is the root element (r). For the sake of simplicity, this example contains no attribute.

XSchema can express *integrity constraints* and *simple constraints* of different kinds that are listed in the taxonomy of figure 1.

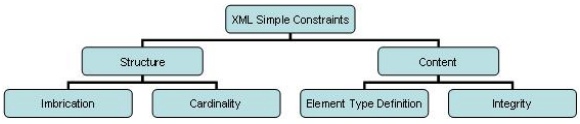

Figure 1: A classification of XML simple constraints

In addition to these constraints, W3C XML Schema manages content constraints as Enumeration, Domain range checking and Pattern matching. In this language, a constraint is applied to a simple type

element (leaf) and if this element has multiple occurences, the constraint will concern each occurence.

The definition of W3C XML Schema based on XSchema is given in Definition 2. For a complete formal definition, please refer to (Brown et al., 2001).

Definition 2. W3C XML Schema extends XSchema considering :

- (E,A,P,r,Σ) as in a XSchema.
- M is a function from an element name $e \in E$ to its element type definition α where α is

$$\alpha ::= \varepsilon \mid \varphi \mid \alpha + \alpha \mid \alpha,\alpha$$

where ε denotes the empty element, φ a 4-tuple (τ,d,o_i,o_s) where τ is an atomic data type (e.g., String, Integer, ...), d is a finite set of valid domain values of e (that can be ε), o_i and o_s are respectively the minimum and maximum cardinality of the element α, "+" the union and "," the concatenation,

3 EXTENDED CONSTRAINTS

We extend the classical expression of constraints with respect to three aspects. The first one is the use of a quantifier to define the application scope of the constraint. The second one is the expression of constraints on elements of complex type. The third one is the management of conditional constraints expressed by implication rules.

3.1 Quantification

Classical constraints in W3C XML Schema are implicitly *universally quantified*, i.e. the constraint concerns each occurence of the concerned element. Our formalism, eqCXSD, offers the possibility to use another quantifier, the *existential quantifier*. An existentially quantified constraint will be satisfied if at least one occurence of the concerned element satisfies it. For example, the constraint *"∀ Mark, Mark > 15"*, which is equivalent to *"Mark > 15"* in W3C XML Schema, means that each occurence of the element Mark in the document must have a value superior to 15. The constraint *"∃ Mark, Mark > 15"* means that one occurence of the element Mark in the document must be superior to 15.

3.2 Constraints on Complex Type Elements

The introduction of the existantial quantifier offers the possibility to deal with more complex situations that can be correctly expressed only if a constraint is applied to an internal node defining a complex element. This is illustrated by the following example. The constraint *"There exists one diploma the degree of which is a Bachelor and the field of which is in Computer Science"* does not have the same meaning as *"There exists a diploma.degree which is a Bachelor and there exists a diploma.field in Computer Science"*. The first constraint concerns the same diploma, which must satisfy two requirements: its degree is a Bachelor AND its field is "Computer Science". The second expression does not necessary concern the same Diploma. It will be verified if the document contains one diploma with a Bachelor degree and another diploma in the field of "Computer Science". The first requirement can be expressed by a constraint which is exitentially quantified and where the quantification concerns the complex type element *Diploma*. Consequently, our eqCXSD formalism allows to express constraints on every level of the tree. The semantic of the constraint will be then dependent of the quantifier previously defined and the node level on which we want to express the constraint. The section 4 will give the precise syntax for these *quantified constraints*.

3.3 Conditional Content Constraints

Conditional content constraints express relations, i.e. implication rules, between the contents of different elements. The figure 2 shows the different types of constraint that we can encounter.

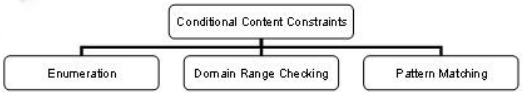

Figure 2: A classification of XML complex constraints

A conditional constraint can be a conditional enumeration, domain range checking or pattern matching where the enumeration, domain range checking or pattern matching depends on the value of an element. For example, *"If the diploma title is "professional bachelor", the mark has to be greater than 15."* is a conditional domain range checking.

4 EQCXSD: EXTENDED AND QUANTIFIED CONSTRAINTS IN XML SCHEMA DEFINITION

This section presents the formalism eqCXSD that we propose to deal with extended constraints. It is based on XSchema and manages quantified and conditional constraints.

Definition 3. The extended model EX_t is a 6-tuple $EX_t = (E_t, A_t, M_t, P_t, r_t, \Sigma_t)$ where

- E_t, A_t, P_t, r_t are defined as in XSchema definition presented in definition 1.
- M_t is defined as W3C XML Schema definition presented in definition 2.
- Σ_t is a finite set Σ of integrity constraints for XML model and a finite set Σ_{cc} of extended constraints. $\Sigma_t \equiv \Sigma \cup \Sigma_{cc}$.

To define which kind of expression can be found in Σ_{cc}, we give the definitions of a quantified XML formula and an extended quantified constraint.

Definition 4. A *XML formula* is defined by the following rules:
- $formula ::= atom, atom^*$
- $atom ::= element\ relation\ value$
- $relation ::= = \mid \equiv \mid \neq \mid < \mid > \mid \leq \mid \geq$
- $element \in E_t$
- *value* is a value of an atomic data type

The formula below is a conjunction of two atoms concerning two elements of a CV document.

Mark>12, Field≡"Computer Science"

As explained in section 3, a formula can be quantified in order to precise which element is concerned and how its occurences are concerned.

Definition 5. A *quantification* is a couple composed by a quantifier $\phi ::= \forall \mid \exists$ and an element $e \in E_t$.

Definition 6. A *quantified formula* is a quantification followed by a XML formula. The XML formula will be surrounded by square brackets, this brackets define the *scope* of the quantification.

As XML documents are hierarchically organised, if an element e of complex type is concerned by a *quantification*, the formula in its scope can contain any element of the subtree attached to e.

For example, the following quantified formula means that "there exists a diploma whose mark is strictly superior to 12 and whose field is equivalent to "Computer Science"".

$\exists$ *Diploma [Mark>12, Field≡"Computer Science"]*

In order to express the desired constraints, we also need to define quantified implications.

Definition 7. A *quantified implication* has the form: $\phi X_k[H \implies B]$ where

- ϕX_k is the quantification of the implication
- H is a XML formula
- B is an atom

We can give now the definition of an extended constraint.

Definition 8. An *extended quantified constraint* $C \in \Sigma_{cc}$ can be:

- a quantified implication $\phi X_k[H \implies B]$
- an expression $\phi X_{k1}[H_1] \wedge ... \wedge \phi X_{kn}[H_n] \implies \phi X_k[B]$ where
  - n can be equal to 0.
  - ϕX_{ki} is the quantification of the formula H_i
  - H_i is a *formula*
  - B is an atom

For each extended constraint, we have a set of Head expressions (H_i), which can be ε, and a Body expression (B). Head and Body expressions can be respectively considered as conditions and conclusion. A head expression is an XML *formula*. Each head and the body can have their specific *quantification*. In case of a quantified implication, the implication is under the scope of a sole quantifier.

For example, the following extended constraint means that *if there is no diploma whose field is equivalent to Computer Science, there must exist a competence in Computer science*.

$\forall$ *Diploma[Field* $\neq$ *"Computer Science"]* $\implies$
$\exists$ *Competence[Field="Computer Science"]*

This other constraint means that *"If the diploma title is a professional bachelor, the mark for this diploma must be superior to 15."*

$\forall$ *Diploma [Title = "BachelorP"* $\implies$ *Mark>* 15*]*

5 XML TRANSLATION OF EQCXSD

This section presents how our formalism can be expressed in XML representation. As eqCXSD is based on W3C XML Schema, we have translated eqCXSD in W3C XML Schema Definition.

As explained in the previous section, a constraint is composed of XML formulas that are themselves composed of atoms, and a atom contains only one element. To integrate our formalism into schema, we have chosen to split a constraint representation into atomic parts. Each atomic part concerns a single element and will be expressed inside this element definition. Likewise a quantification concerns a single element and will be expressed in the same way, inside the definition of the quantified element. To integrate this decomposition, eqCXSD extends the classical element type definition of W3C with additional information concerning the constraints. Moreover, to link these different parts of a constraint representation, a constraint is represented by a unique identifier.

For example, considering the XML formula *"there exists a long term contract professional expe-*

rience in Computer science", this formula is composed by 2 atoms. We then decompose the formula on each element concerned by each atom. We join to the definition of "Professional.Type" the atom $type \equiv$ *"Long term contract"* and to the definition of "Professional.Field" the atom $field \equiv$ *"Computer Science"*. In order to garantee the coherence of the formula, we assign a unique identifiant to the formula. Furthermore, we join the existential quantifier, any, to the definition of the concerned element, "Professional".

To summarize, we redefine classical XML element definitions wrt *quantification* of the formula and the atom definition as defined in definition 4 plus the information concerning the place ot this atom in the constraint (head or body). For each case, we propose a new XML Schema definition of "element".

5.1 Quantification Representation

Definition 9. The new definition of a *quantification* is the classical W3C XML Schema element definition extended by a 3-tuple (ID_{cc}, ID_f, Op) where

- ID_{cc} is the global identifiant of the constraint
- ID_f is the identifiant of the formula
- Op is the logical operator : "all" for $\forall$ and "any" for $\exists$.

The classical definition of *element* given in the W3C XML Schema specification is extended with a sub element called *quantification*. An *element* can be part of several or no *quantification*, depending on the different constraints expressed in the model. The *quantification* is an empty element with three attributes : IDcc, IDf and Operator, which can have two values : "all" or "any". The corresponding new XML Schema is given in table 1 and an example of use is given in table 2. We will then use this element definition in our schemas instead of W3C definition. In order to avoid any confusion between W3C element and our *element*, we will use a specific namespace for our extended schema ("exsdc").

5.2 Atom Representation

Definition 10. The new definition of an *atom element* is the classical W3C XML Schema element definition extended by a 5-tuple (ID_{cc}, ID_f, *place*, *relation*, *value*) where
- ID_{cc} is the global identifiant of the constraint
- ID_f is the identifiant of the formula
- *place* is the place in the constraint : "head", for condition, or "body", for conclusion
- *relation* and *value* correspond to the elements definition of the XML formula given in Def 4.

Table 1: Definition of the new element "quantification".

```
<xsd:complexType name="elementquantification">
  <xsd:complexContent>
   <xsd:extension base="xsd:element">
    <xsd:sequence>
     <xsd:element name="quantification" minOccurs="0"
        maxOccurs="unbounded">
     <xsd:complexType>
      <xsd:simpleContent>
       <xsd:extension base="xsd:empty">
        <xsd:attribute name="IDcc" type="xsd:int"/>
        <xsd:attribute name="IDf" type="xsd:int"/>
        <xsd:attribute name="operator" use="required">
         <xsd:simpleType>
          <xsd:restriction base="xsd:string">
           <xsd:enumeration value="all"/>
           <xsd:enumeration value="any"/>
          </xsd:restriction>
    </.../>
</xsd:complexType>
```

Table 2: Example of definition of a quantification : *The quantification of the formula 1 of the constraint 1 is* $\forall$ *Diploma.*

```
<exsdc:element name="diploma" type="diplomaType">
   <exsdc:quantification IDcc="1"
          IDf="1" operator="all"/>
</exsdc:element>
```

As for the element *quantification*, we extend the classical definition of element with a sub element *atom*. An *element* can be implied in several or no *atom*, depending on the different constraints expressed in the model. The *atom* is an empty element with five attributes : $IDcc$, IDf, *place* which can only have two values : "head" - in case of condition - or "body" - in case of conclusion -, *relation* which represents the comparison operator and *value* which is the information with which the element will be compared. These five information are defined as additional attributes of the W3C element. The corresponding new XML Schema is given in table 3 and an example of use is given in table 4.

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a new formalism of integrated XML Schema capable to manage quantified and conditional constraints on XML documents. This extended XML Schema is based on a classical schema definition. Our contribution is on three folds.

Table 3: Definition of the new element ”Atom”.

```
<xsd:complexType name="elementAtom">
 <xsd:complexContent>
  <xsd:extension base="xsd:element">
   <xsd:sequence>
    <xsd:element name="atom" minOccurs="0"
       maxOccurs="unbounded">
    <xsd:complexType>
     <xsd:simpleContent>
      <xsd:extension base="xsd:empty">
       <xsd:attribute name="IDcc" type="xsd:int"/>
       <xsd:attribute name="IDf" type="xsd:int"/>
       <xsd:attribute name="place">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:enumeration value="head"/>
          <xsd:enumeration value="body"/>
         </xsd:restriction>
        </xsd:simpleType>
       </xsd:attribute>
       <xsd:attribute name="relation">
        <xsd:simpleType>
         <xsd:restriction base="xsd:string">
          <xsd:enumeration value="="/>
          ...
       </.../>
       <xsd:attribute name="value" type="xsd:any"/>
      </xsd:extension>
    </.../>
</xsd:complexType>
```

Table 4: Example of definition of an atom : *An atom of the formula 1 of the constraint 1 is a condition (head) and concerns the element ”field”. It can be binded to the quantification defined previously. It means ”∀ Diploma[field ≠ ”Computer Science”...]”*

```
<exsdc:element name="field" type="xsd:string">
  <exsdc:atom IDcc="1" IDf="1" place="head"
     relation="≠" value="Computer Science"/>
</exsdc:element>
```

Firstly, we have added a quantifier to constraint parts. This quantifier can be universal or existential. On the one hand, the universal quantifier means that each occurence of the element concerned by a constraint has to verify this constraint. On the other hand, the existential quantifier means that only one occurence of the element concerned by the constraint needs to verify the constraint. W3C XML Schema, which is the current richest integrated solution, only manages implicitly universal quantifier.

Secondly, we express constraints at any level of the XML Schema hierarchy. Constraints can be expressed not only on leaves, as made in W3C WML Schema, but also on any node of the XML tree, even on complex type elements. We then enrich the classical definition of the element which will be concerned by the constraint. Thus, the element concerned by the constraint added with the quantifier defined previously are what we call the *quantification of the constraint*. They precise the scope of the constraint and then give a richer framework.

Thirdly, we have defined a new type of constraints, the *conditional content constraints*. Such constraints are implication rules between several elements, they express relations between the contents of these elements. This type of constraint does not exist in integrated approach. To have a richer semantic of constraints, these constraints are composed of quantified subconstraints that we have called *quantified formulas*. Hence, this formalism is more expressive than other integrated approaches and allows to express more complex constraints.

REFERENCES

Berglund, A. (2006). Extensible Stylesheet Language (XSL). W3C Recommendation.

Bray, T., Paoli, J., Sperberg-McQueen, C., and Maler, E. (2004). Extensible Markup Language(XML). W3C Recommendation.

Brown, A., Fuchs, M., Robie, J., and Wadler, P. (2001). MSL - a model for w3c XML schema. In *World Wide Web*, pages 191–200.

Clark, J. and DeRose, S. (1999). XML Path Language. W3C Recommendation.

Clark, J. and Murata, M. (2001). RELAX NG Specification. OASIS.

Jacinto, M. H., Librelotto, G. R., Ramalho, J. C., and Henriques, P. R. (2003). XCSL : XML Constraint Specifcation Language. *CLEI Electronic Journal*, 6,1.

Jelliffe, R. (2001). Schematron : specification 1.5. Web page.

Lee, D. and Chu, W. W. (2000). Comparative analysis of six XML schema languages. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 29(3):76–87.

Mani, M. and Lee, D. (2002). XML to Relational Conversion using Theory of Regular Tree Grammars. In *Efficiency and Effectiveness of XML Tools and Techniques and Data Integration over the web (EEXTT)*.

Murata, M., Lee, D., Mani, M., and Kawaguchi, K. (2005). Taxonomy of XML schema languages using formal language theory. In *ACM Trans. Internet Techn.*, volume 5(4), pages 660–704.

Thompson, H. S., Beech, D., Moloney, M., and Mendelsohn (Eds), N. (2001). XML Schema. W3C Recommendation.