社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2003－DPS－114　(1)
2003－EIP－ 20　(1)
2003／8／28

XMLデータの包含関係比較における高速化手法の提案

高田智規[†]　山本隆二[†]　西岡秀一[†]
阿部剛仁[†]　川村春美[†]

XML を用いた情報流通が盛んになっており，XML データ同士の包含関係を比較する機会が増大している．複雑な比較判断を実サービスにおいて利用する際にはレスポンスタイムが顧客満足度に重要な要素となるため，XML データの効率的な比較処理が重要となる．

本稿では，比較を行う XML データの一方があらかじめ定義されている場合において，XML データの内容と比較条件情報を組にした中間データをあらかじめ作成し，この中間データともう一方の XML データとを比較することで比較処理を高速化する手法を提案する．

次に，著作権管理のために用いられる，XML ベースの権利記述言語である XrML に提案手法を適用し，ライセンス管理システムにおいて実装した．また，比較処理について処理時間に関する測定を行った．この結果，提案手法が有効であることを確認した．

An Efficient Method of Comparing XML Data

Tomonori Takada,[†] Ryuji Yamamoto,[†] Shuichi Nishioka,[†]
Takehito Abe[†] and Harumi Kawamura [†]

In the last decade there has been a surge of XML data distribution, and data comparison is strongly needed in the data processing. When complex comparison is required in real service, its response time is distinctly critical from the viewpoint of customer satisfaction. An efficient comparison method is necessary for such services.

In this paper, we propose a fast comparison method, in which an intermediate data set combined with its condition is leveraged to compare XML data if the compared data is fixed.

Also, we applied this to data comparison of XrML which is rights expression language, and implemented to the license management system. The experimentation result of processing time measurement confirms that the proposed method is effective.

1. はじめに

近年，XML を用いた情報流通が盛んになっている．BtoB 電子商取引のフレームワークを規定する ebXML (Electronic Business using eXtensible Markup Language)[1] や，権利記述言語である ODRL (Open Digital Rights Language)[2]，XrML (eXtensible rights Markup Language)[3] などの，XML を利用した様々な仕様が策定されている．これらを用いることにより，これまでシステムや各社ごとに独自形式であったデータ構造や表現方法などを XML で統一的に表現にすることができ，運用性の向上・システム構築費の削減などを実現することができる．今後も XML は標準仕様としてますます普及するものとみられる．

† NTT サイバーソリューション研究所
NTT Cyber Solutions Laboratories.

このように，XML によるビジネスルールの記述が進めば，価格・注文数などの数量的条件判断が頻繁に発生することが想定される．さらに，単純な数量比較だけでなく，期間や使用条件などといったより複雑な比較判断を行う必要が生じてくる．実サービスにおいては，レスポンスタイムが顧客満足度に大きく影響するため，XML データの効率的な比較処理が重要となる．

本稿では，ある XML データで表現されている内容が，他の XML データに包含されるかどうか，という比較処理を行う際に，処理を高速に行う手法を提案する．次に，提案手法を XrML に適用し，ライセンス管理システムへ実装した例について報告する．また，実装した比較処理について，処理速度に関する測定実験を行い，本提案手法の有効性を確認した．

以降，2 節で XML データの比較方法の概要を述べ，効率的な比較手法を提案する．次に，3 節で提案手法をライセンス管理システムに適用した例を示す．4 節

では，実装した比較処理に関する実験およびその結果について述べる．最後に 5 節でまとめと今後の課題について述べる．

2. XML データの比較

2.1 XML データの包含関係

XML で記述された情報に関して，その包含関係 (ある XML データの内容が別の XML データの範囲内にあるかどうか) を判断する機会は多いと考えられる．例えば，物品購入の注文書を XML データで記述した場合に，注文数量が在庫数量を越えているかどうかを判断したり，あるコンテンツに関するライセンス発行を依頼する場合に，その条件が契約の範囲内であるかどうかを判断する場合には，包含関係の比較処理が必要になってくる．

包含関係を比較するためには，XML データ内の要素 (要素名と要素値から構成される) をそれぞれ比較していくことが必要となる．一般に，要素の種類によって比較条件は異なる．さらに，親要素の種類など，要素の構造によって比較条件が異なることもある．そのため，要素ごとに比較条件を定義する情報を用意し，比較処理を行う際にその条件を用いることが必要である．この比較情報はビジネスルールの変更や記述対象の追加などによって随時変更が加えられることが想定されるため，比較処理自体がこの情報を保持するのではなく，DB やファイルなどの形式で外部に保持することが重要である．

このような比較処理を行う場合には，比較する XML データ，比較される XML データ，比較条件の 3 つの情報を解析する必要があり，比較処理の負荷が大きくなる．そのため，リアルタイム処理が必要とされる場合にはふさわしくない．

これを解決するために，比較処理を高速化する手法を提案する．提案する手法では，比較を行う XML データのうち一方をあらかじめ比較条件と結びつけておくことで比較処理を高速化する．

2.2 サマリデータを用いた比較方式の提案

XML データの持つ要素をあらかじめ比較条件と結びつけておくことで比較処理時の処理を軽減することができるが，比較を行う XML データの構造が異なる場合，要素同士の関係を一致させることは困難である．そのため，提案手法では比較を行う XML データの構造が同一である場合に限定する．

なお，比較を行う元となる XML データ (比較条件との結びつけを行う XML データ) を比較元 XML と呼び，比較元 XML と比較を行う XML データを比較対象 XML と呼ぶ．また，要素に対応する比較条件を定義した情報を比較条件定義情報と呼ぶ．

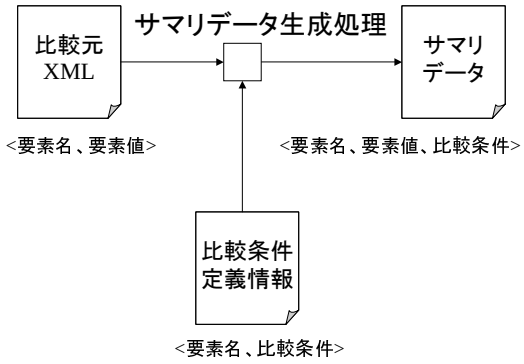

図 1 サマリデータ生成処理概要フロー
Fig. 1 A flow of generating a summary data.

以下，比較元 XML と比較条件を結びつける処理であるサマリデータ生成処理と，サマリデータを利用して比較対象 XML との比較を行う比較処理について述べる．

2.2.1 サマリデータ生成処理

提案手法では，あらかじめ比較元 XML と比較条件定義情報を結びつけ，中間データを生成する．この中間データをサマリデータと呼ぶ．サマリデータを生成する処理の概要を図 1 に示す．この動作について以下に示す．

(1) 比較元 XML の構文解析
構文解析を行い，XML データ中の要素を得る
(2) 比較条件定義情報の読み込み
要素名と比較条件の組を取得する
(3) サマリデータ生成
比較元 XML の要素に該当する要素名を比較条件定義情報から検索する．該当する要素が存在する場合，要素名・要素値・比較条件の組を作成し，これをサマリデータに書き出す．

2.2.2 比 較 処 理

実際の比較を行う際には，サマリデータと比較対象 XML を用いる．処理の概要を図 2 に示す．この動作について以下に示す．

(1) 比較対象 XML の構文解析
構文解析を行い，XML データ中の要素を得る
(2) サマリデータの読み出し
サマリデータから要素名・要素値・比較条件の組を読み込む
(3) 探索および比較
比較対象 XML 中の要素がサマリデータの要素

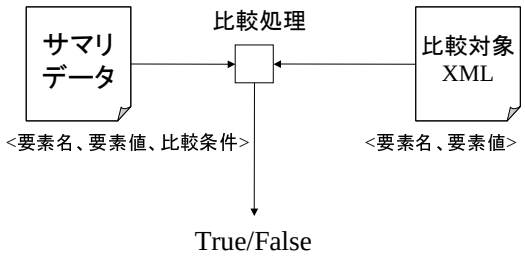

図 2 比較処理概要フロー
Fig. 2 A flow of comparison.

表 1 比較処理計算時間に用いられる記号一覧
Table 1 Symbols used to express comparison processing time.

記号	意味
XML_s	比較元 XML
XML_t	比較対象 XML
CCD	比較条件定義情報
SUM	サマリデータ
$p(x)$	x の構文解析処理に必要な時間
$m(x,y)$	x と y のマッチング処理に必要な時間

名と一致する場合，サマリデータの要素値と比較条件を用い，比較対象 XML の要素値と比較を行う．この結果が偽であった場合，比較対象 XML は比較元 XML の定義範囲を超えていると判断できるため，比較結果を偽とする．これを繰り返し，全ての要素について真であった場合，比較対象 XML は比較元 XML に包含されていると判断できるため，比較結果を真とする．

2.2.3 従来手法との比較

従来の比較手法では，比較元 XML・比較対象 XML・比較条件定義情報を解析し，比較する必要があった．

従来手法において比較処理時に必要となる計算時間を以下に示す．なお，記号の意味を表 1 に示す．

- 構文解析処理
  $p(XML_s) + p(XML_t) + p(CCD)$
- 比較元 XML と比較条件定義情報のマッチング処理
  $m(XML_s, CCD)$
- 比較元 XML と比較対象 XML の比較処理
  $m(XML_s, XML_t)$

ここで，比較元 XML と比較対象 XML は同一構造であるため，$p(XML_s) \quad p(XML_t)$ とみなせる．よって全体の処理時間は

$$2 \cdot p(XML_s) + p(CCD) + m(XML_s, CCD) + m(XML_s, XML_t) \quad (1)$$

となる．

一方，提案手法の比較処理に必要となる計算時間は以下の通りである．

- 構文解析時間
  $p(SUM) + p(XML_t)$
- サマリデータと比較対象 XML の比較処理
  $m(SUM, XML_t)$

ここで，サマリデータのサイズ・比較対象 XML のサイズともに比較元 XML のサイズとほぼ同じとみなせるため，全体の処理時間は，

$$2 \cdot p(XML_s) + m(XML_s, XML_t) \quad (2)$$

と近似できる．

(2) 式と (1) 式の差から，提案手法は従来手法と比べ，$p(CCD) + m(XML_s, CCD)$ だけ処理時間を軽減できることがわかる．なお，比較条件定義情報は考えうる全ての XML データの要素名と比較条件を記述する必要があるため，一般にサイズが大きくなると考えられる．よって，この値は大きくなり，提案手法を用いることで大幅に比較処理に要する時間を削減可能であることがわかる．

なお，従来手法は事前の処理は不要であるが，提案手法では，サマリデータ生成処理が必要となるため，$p(XML_s) + p(CCD) + m(XML_s, CCD)$ の処理を行う必要がある．

3. ライセンス管理システムにおける比較方式の検討

提案手法を，XML を利用した権利記述言語である XrML (eXtensible rights Markup Language)[3] に適用し，ライセンス管理システムにおいて実装した．

3.1 ライセンス管理システム

実装を行ったライセンス管理システムは複数の DRM (Digital Rights Management)[4] を統合的に扱うことができ，著作権管理システム[5] と連動してコンテンツ流通[6] の実現を目指すものである．本システムは Microsoft 社の WMRM (Windows Media Rights Manager)[7]，Real Networks 社の RSMCS (Real System Media Commerce Suite)[8] などの DRM に関する利用許諾条件を一元管理し，コンテンツ視聴のためのライセンスを発行する[9]．

利用許諾条件はコンテンツ再生回数や利用期限などの項目を持ち，各 DRM によってそれぞれ定義されている．本システムでは，これらの条件を XrML にて統一的に記述・管理している．

XrML は OASIS (Organization for the Advancement of Structured Information Standards) の権利

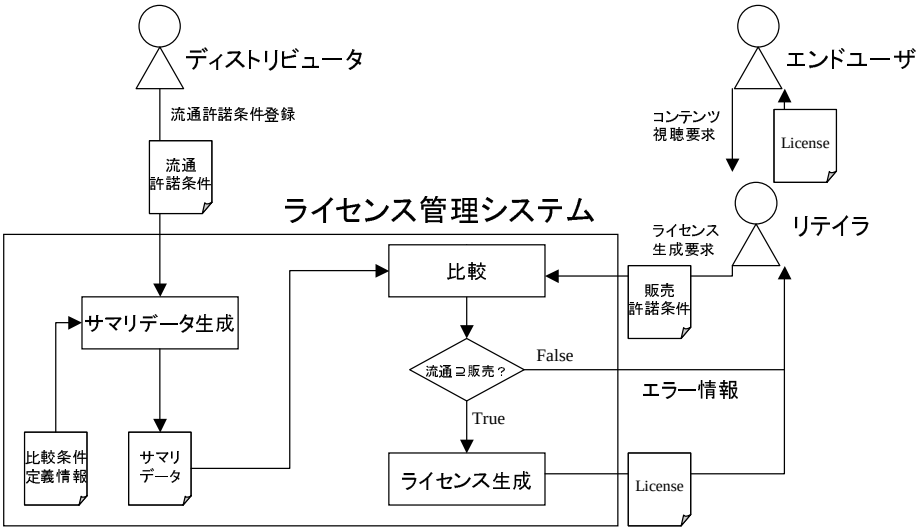

図 3 ライセンス管理システム概要図
Fig. 3 Overview of License management system.

言語技術委員会 (RLTC) にて標準化の検討が進められている仕様であり，Core Schema, Standard Extention (SX), Content Extension (CX) から構成される．XrML によって記述される権利は，許可される動作を表す Right，制約条件を表す Condition，Right や Condition のコンテナで許可を表す Grant などを持つ．

図 3 に，本システムの構成および処理の概要を示す．本システムでは，利用許諾条件の最大値をコンテンツ流通業者 (ディストリビュータ) が定義する．ディストリビュータの定義した利用許諾条件を流通許諾条件と呼ぶ．コンテンツ販売業者 (リテイラ) はディストリビュータと個別に契約を行い，流通許諾条件の範囲内であれば自由に条件を修正し，エンドユーザに販売することが可能である．リテイラがユーザに販売した際の条件を販売許諾条件と呼ぶ．このようなモデルを考える際には，リテイラがライセンスの生成を要求する際に送信する販売許諾条件が流通許諾条件の範囲内にあるかどうかを判定する必要がある．

流通許諾条件はコンテンツが流通される間，一貫して用いられる利用許諾条件であり，一度コンテンツが流通すれば変更される機会は多くないと考えられる．一方，販売許諾条件はリテイラがエンドユーザにコンテンツを販売するごとに変更される可能性がある．例えば，視聴 1 回あたりの料金を設定しておき，回数を変動できる形での販売や，期間ごとに料金を変動させるなど，ビジネスモデルに応じて多様な販売形態が考えられる．そこで，ディストリビュータがライセンス管理システムに流通許諾条件 を登録する際にサマリデータの生成を行い，ライセンスの生成を行う際 (エンドユーザがリテイラの作成した販売用 Web サイト等を経由してライセンス生成を要求する際) に販売許諾条件とサマリデータの比較を行う．比較の結果，販売許諾条件が流通許諾条件の範囲内であると判断された場合にライセンス生成を行う．

```
ヘッダ部
        データ型定義
        対象外パス定義
        部分一致対象パス定義
要素定義部
        要素名
        データ型
        比較条件
```

図 4 比較条件定義情報フォーマット
Fig. 4 Format of condition definition information.

3.2 提案手法の拡張

本システムに提案手法を適用するにあたり，幾つかの拡張を行った．これらを以下に示す．

- **比較条件定義情報への項目追加**
  実装にあたり，比較条件定義情報に型名を記述する項目を追加した．また，型ごとに比較演算の処

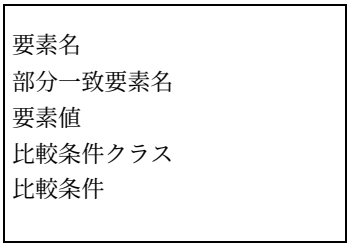

図 5 サマリデータのフォーマット
Fig. 5 Format of summary data

理が異なるため，比較処理を定義しているクラス名をヘッダ部に記述することとした．これにより，型判断を容易にするとともに，新たな型の出現にも対応可能とした．これに伴い，サマリデータに比較クラス名を追加した．

- **検索対象外要素**
  比較する必要のない要素名を比較条件定義情報に明記することで，不要な要素に関する処理を軽減させることとした．
- **部分一致要素**
  提案手法では，比較される 2 つの XML データは同一構造であることが必要条件であるとしていたが，これでは使用できる機会が多くない．そこで，基本構造が同一であることは制約としつつ，ある特定の要素以下の項目を削除可能とした．例えば，XrML においては，grant 要素は「許可」を表しているため，これを削除しても権利は縮小する方向へと変更されため，問題はない．このような要素を部分一致要素として比較条件定義情報内に記述することで，一部構造変更に対応した．これによって，同一構造の XML データ同士だけではなく，ある XML データとその部分集合の構造を持つ XML データの比較を可能とした．
  また，部分一致要素の解析を容易にするため，サマリデータに部分一致した要素名を追加した．

これらの拡張を行った上での，比較条件定義情報のフォーマットを図 4 に，サマリデータのフォーマットを図 5 にそれぞれ示す．比較条件定義情報では，上記変更により，ヘッダ部が増えているともに，要素定義部にデータ型を記述している．また，サマリデータには，部分一致要素名と比較条件クラスを記述できるよう項目を追加している．

3.3 提案方式の適用

本システムにおける比較条件定義情報の例を図 6 に示す．ヘッダ部に，データ型に関する定義 (DATA_TYPE_DEFINITION)，検索対象外要素名定義 (EXCLUDE_NODE)，部分一致対象要素名定義 (PARTIAL_COINCIDENCE) を持ち，データ部として要素名 (CHECK_PATH)，データ型 (DATA_TYPE)，比較条件 (CHECK_OPERATION) を持っている．実際には，型名に対するクラス名，要素に対する比較条件の全てを記述する必要があるが，

```
<!-- データ型の定義 -->
<DATA_TYPE_DEFINITION>
  <DATA_DEFINITION>
    <!-- 型名 -->
    <DATA_TYPE>Integer</DATA_TYPE>
    <!-- クラス名 -->
    <CHECKER_CLASS>
      jp.co.. ..checker.IntegerChecker
    </CHECKER_CLASS>
  </DATA_DEFINITION>
</DATA_TYPE_DEFINITION>

<STRUCTURE_VERIFICATION>
  <!-- 検索対象外対象外要素名 -->
  <EXCLUDE_NODE>
    <EXCLUDE_XPATH>
      //*[local-name()='issuer' and
        namespace-uri()='http://...']
    </EXCLUDE_XPATH>
  </EXCLUDE_NODE>

  <!-- 部分一致対象要素名-->
  <PARTIAL_COINCIDENCE>
    <PART_AGREES_XPATH>
      //*[local-name()='grant' and
        namespace-uri()='http://...']
    </PART_AGREES_XPATH>
  </PARTIAL_COINCIDENCE>
</STRUCTURE_VERIFICATION>

<VALUE_VERIFICATION>
  <!-- 再生回数 -->
  <VALUE_CHECK>
    <CHECK_XPATH>
      //*[local-name()='exerciseLimit' and
        namespace-uri()='http://...']
       /*[local-name()='stateReference'
        and namespace-uri()='http://...']
       /*[local-name()='count' and
        namespace-uri()='http://...']
    </CHECK_XPATH>
    <DATA_TYPE>Integer</DATA_TYPE>
    <CHECK_OPERATOR>
      ge
    </CHECK_OPERATOR>
  </VALUE_CHECK>
</VALUE_VERIFICATION>
```

図 6 比較条件定義情報の例
Fig. 6 An example of condition definition information

ここでは一部のみを示した．

次に，流通許諾条件と販売許諾条件を表現するのに用いられる XrML データの例を図 7 に示す．この例では，再生可能な権利を表す cx:play という Right に対し，以下の条件を満たす範囲内において権利の行使を認めている．

- 再生回数は最大 5 回まで
  (sx:exerciseLimit/sx:stateReference/sx:count)
- 利用可能期間は 2003 年 8 月 28 日 10:45:00 から 2003 年 8 月 29 日 17:30:00 まで
  (ValidityInterval/notBefore, ValidityInterval/notAfter)
- 利用可能期限は最初に再生を行った時点から 1 日と 3 時間 20 分まで
  (sx:validityIntervalFloating/sx:stateReference/sx:validFor)

図 6 の比較条件定義情報と図 7 の XrML データによって生成されるサマリデータの一部を図 8 に示す．本システムでは，各項目の区切りとして改行を用いている．そのため，5 行単位で，要素名・部分一致要素名・要素値・比較クラス名・比較条件が組となって記述されている．例えば，1 行目〜5 行目は 1 行目で示される要素 (再生回数) に関して 3 行目の値 (5 回) と比較し，5 行目の条件 (ge, 流通許諾条件の値が販売許諾条件の値より大きいか同じ) でなければいけないことを表している．同様に，6 行目〜10 行目は利用開始日時は 2003 年 8 月 28 日 10:45 以降でなければならないことを表している．

```
<license>
  <grant>
    <cx:play/>
    <allConditions>
      <sx:exerciseLimit>
        <sx:stateReference>
          <sx:count>5</sx:count>
        </sx:stateReference>
      </sx:exerciseLimit>
      <validityInterval>
        <notBefore>
          2003-08-28T10:45:00
        </notBefore>
        <notAfter>
          2003-08-29T17:30:00
        </notAfter>
      </validityInterval>
      <sx:validityIntervalFloating>
        <sx:stateReference>
          <sx:validFor>
            P1DT3H20M
          </sx:validFor>
        </sx:stateReference>
      </sx:validityIntervalFloating>
    </allConditions>
  </grant>
</license>
```

図 7 XrML による記述例
Fig. 7 An example described in XrML.

4. 実　　験

4.1 概　　要

本システムにおいて，XML データの比較に要する処理時間を測定した．

本実験では，3 種類の XrML データを用意し，流通許諾条件 DST_A 〜 DST_C とした．また，これらと同じ構造をもつ販売許諾条件 RET_A 〜 RET_C を用意した (表 2)．なお，DST_B は DST_C の grant 要素を 1 つ省略した構造になっている．

これらに対し，構造が同じ 3 種類の比較に要する処理時間，および構造の異なる DST_C と RET_B の比較に要する処理時間を測定した．この結果を，表 3 に示す．

なお，提案手法は Java によって実装され，Solaris 8(Sun Fire 280R, UltraSparc III 750MHz, 2Gbyte RAM) 上で動作している．

表 2 使用した XrML データ一覧
Table 2 XrML data used in the experiment.

名称	サイズ	総要素数	許諾条件
DST_A , RET_A	1.7kB	18	2
DST_B , RET_B	4.2kB	45	12
DST_C , RET_C	11.6kB	58	19

表 3 実験結果
Table 3 The result of the experiment.

流通許諾条件	販売許諾条件	処理時間 (秒)
DST_A	RET_A	0.044
DST_B	RET_B	0.065
DST_C	RET_C	0.074
DST_C	RET_B	0.066

4.2 考　　察

実験結果では，比較処理に要する時間と要素数，および許諾条件の数はほぼ線型になっており，比較する

```
 1: license[0]/grantGroup[0]/grant[0]
    /allConditions[0]/sx:exerciseLimit[0]
    /sx:stateReference[0]/sx:count[0]
 2: license[0]/grantGroup[0]/grant[0]
 3: 5
 4: jp.co.. ..checker.IntegerChecker
 5: ge

 6: license[0]/grantGroup[0]/grant[0]
    /allConditions[0]/validityInterval[0]
    /notBefore[0]
 7: license[0]/grantGroup[0]/grant[0]
 8: 2003-08-28T10:45:00
 9: jp.co.. ..checker.DateTimeChecker
10: le

11: license[0]/grantGroup[0]/grant[0]
    /allConditions[0]/validityInterval[0]
    /notAfter[0]
12: license[0]/grantGroup[0]/grant[0]
13: 2003-08-29T17:30:00
14: jp.co.. ..checker.DateTimeChecker
15: ge

16: license[0]/grantGroup[0]/grant[0]
    /allConditions[0]
    /sx:validityIn-tervalFloating[0]
    /sx:stateReference[0]/sx:validFor[0]
17: license[0]/grantGroup[0]/grant[0]
18: P1DT3H20M
19: jp.. ..checker.DurationChecker
20: ge
```

図 8 生成されるサマリデータの例
Fig. 8 An example of summary data.

XML データのサイズが大きくなった場合においても，実用上問題無いことがわかる．

また，サイズの大きい流通許諾条件 DST_C に対して，その一部が削除された構造である販売許諾条件 RET_B との比較を行った場合，処理時間はほぼ DST_B との比較と等しくなっている．これは，流通許諾条件をサマリデータとして既に解析してあるため，再度 DST_C を解析し直す必要がないためであると考えられる．

これらのことから，提案手法はあらかじめ定義されている XML データに対して，それと同一構造または部分的に変更された類似構造を持つ XML データを比較する際に有効であると考えられる．また，比較元 XML のサイズが大きい場合において提案手法が特に有効であることがわかった．

5. ま と め

XML データの包含関係を比較するための手法を提案した．提案手法は，XML データのうち一方をあらかじめ解析し比較条件と結びつけておくことで，比較時の処理を軽減させ高速化を行っている．

また，提案手法をライセンス管理システムに実装するとともに，処理速度の測定実験を行い，提案手法が有効であることを示した．実験結果から，高速な比較処理が実行可能であり，サイズが大きな比較元 XML と，その部分構造を持つ比較対象 XML とを比較する際に有効であることを確認した．

今後は，さらなる高速化を検討するとともに，構造が異なる XML データ同士の比較方式の検討や比較条件定義情報への標準技術の適用を進めていく予定である．

参 考 文 献

1) UN/CEFACT and OASIS: Electronic Business using eXtensible Markup Language (ebXML), http://www.ebxml.org/.
2) IPR Systems: Open Digital Rights Language (ODRL), http://www.odrl.net/.
3) ContentGuard: Extensible rights Markup Language (XrML), http://www.xrml.org/.
4) 櫻井 紀彦, 木俵 豊, 高嶋 洋一, 谷口 展郎, 難波 功次: コンテンツ流通における著作権技術の動向, 情報処理学会論文誌 データベース, Vol. 42, No. SIG15(TOD12), pp. 63–76 (2001).
5) 山田 智広, 松浦 由美子, 山本 奏, 萬本 正信, 川村 春美, 高嶋 洋一, 黒川 清, 大村 弘之: 権利流通プラットフォームの開発および評価, 情報処理学会研究報告 電子化知的財産・社会基盤研究会 2002-17, pp. 51–57 (2002).
6) 安田 浩, 安原 隆一: ポイント図解式 コンテンツ流通教科書, アスキー (2003).
7) Microsoft: Windows Media Rights Manager (WMRM), http://www.microsoft.com/.
8) RealNetworks: RealSystem Media Commerce Suite (RSMCS), http://www.realnetworks.com/.
9) 西岡 秀一, 高田 智規, 山本 隆二, 阿部 剛仁, 川村 春美, 大村 弘之, 有澤 博: ライセンス情報の統合管理方式に関する一手法, 情報処理学会研究報告 2003-DBS-131(II), pp. 33–40 (2003).