

UDDI レジストリの実装

2Z-04

小泉 清一[†] 富沢 伸行[†] 藤田 悟[†] 山之内 徹[†][†] NEC インターネットシステム研究所

e-mail: s-koizumi@ap.jp.nec.com

1. はじめに

インターネット上のサービス指向的なソフトウェア構築環境の1つとして、Web サービスが注目されている。Web サービスとは、インターネット標準の各種プロトコルを利用してアクセス可能なコンポーネント・アーキテクチャである。

Web サービスは、設計時ではなく、実行時に要求に合ったサービスを発見/統合することもできるという特徴を持ち、そのための検索機構を実現するための枠組みとして、UDDI (Universal Description, Discovery and Integration)¹⁾ 仕様が制定されている。

UDDI レジストリに登録されるデータは、Web サービスの内容や利用方法が記述された XML 文書である。また、データの登録、更新よりも検索が主要な操作であることが考えられる。そのため、UDDI レジストリの実装においては大量の XML 文書を効率良く格納して高速に検索することが必要となる。

XML 文書の主な格納手段としては、ファイルシステム、eXcelon²⁾ 等のネイティブ XML データベース、リレーショナルデータベース (RDB) 等が挙げられるが、データ連携の容易さや、実運用での信頼性などから RDB に格納する利点も少なくない。

RDB を用いた XML 文書格納手法の既存研究³⁾ は、任意のスキーマ構造を持つ XML 文書の格納を目的としており、検索/格納コストが大きい。

我々は、格納されるスキーマ構造や検索手段が仕様により定められているという UDDI レジストリの特徴を基に、RDB 上での検索コストを低減した UDDI レジストリの実装を目的とする。

2. UDDI レジストリにおける検索コスト低減方法

2.1 UDDI レジストリに格納されるデータ

UDDI レジストリに格納されるデータは XML 文書である。その XML 文書の内容としては、Web サービスを提

供する企業の情報 (businessEntity)、Web サービスの内容 (businessService)、接続情報 (bindingTemplate)、仕様の記述 (tModel) 等がある。

ここでは businessEntity を文書例として図 1 に示す。

```
<businessEntity businessKey="XXXXXXXX-XXXX" >
  <name>NEC Corporation </name>
  <description xml:lang="en" >NEC has emerged as a
    Internet Solution Provider. </description>
  <contacts >
    <contact useType="main" >
      <description xml:lang="en" >the address...</description>
      <personName >Corporate Headquarters </personName>
      <phone useType="main" >(+81)X-XXXX </phone>
      <address >Minato-ku, Tokyo .... </address>
    </contact>
  </contacts >
</businessEntity >
```

図 1 格納される企業情報 (businessEntity) の例

2.2 格納方法に関する分析

UDDI レジストリは、「格納されるデータの XML スキーマがある」、「検索 API で検索条件を指定できる要素は限られている」という特徴をもつ。

これらの特徴より、格納されるデータの要素を「検索対象となる要素」と「検索対象とならない要素」に分類することが出来る。例えば、図 1 において検索対象となる要素は点線で囲われた部分である。

また、格納されるデータの要素は最深でも 5 階層で、主に 2, 3 階層の位置に分布している*。

上記の分析結果より考察すると、UDDI レジストリのデータ構造は RDB のテーブルにマッピングしやすく、かつ、全ての要素をマッピングする必要がない。そこで、本研究では検索対象となる要素については値を取り出して RDB のテーブルに格納し、検索対象とならない要素については XML のまま格納する、というマッピング方法を採用した。

本研究のマッピング方法による図 1 の businessEntity の格納例を図 2 に示す。また、要素の階層毎にテーブルを対応付けた場合の格納例を図 3 に示す。本研究のマッピング手法を適用した場合、businessEntity の例は単一のテーブル

Design and implementation of a UDDI registry.
Seiichi Koizumi[†], Nobuyuki Tomizawa[†],
Satoru Fujita[†], Toru Yamanouchi[†]

[†]Internet Systems Research Laboratories,
NEC Corporation.

* UDDI レジストリに格納される XML 文書を businessEntity, businessService 等に切り分けた場合

に格納されるが、図3に示すマッピング方法の場合は、テーブルを2つ用意する必要がある。

本研究のマッピング方法では、検索対象となる要素の存在する階層毎にテーブルを用意する。図1の businessEntity のように、検索対象となる要素の存在する階層が、全階層より少ない場合は本マッピング方法の利用によりテーブル数が削減される。

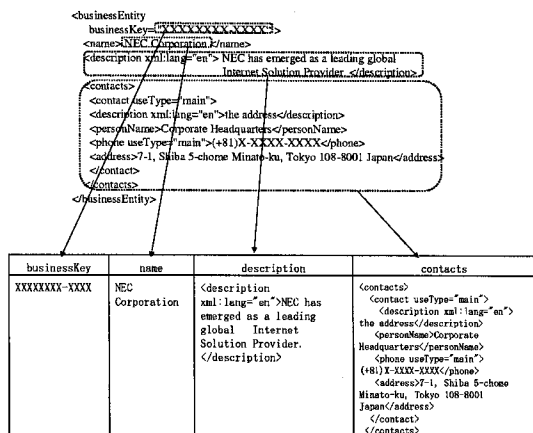


図2 本研究のマッピング方法

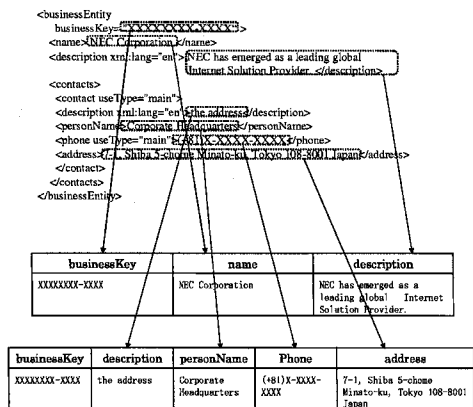


図3 要素の階層毎にテーブルを対応付けたマッピング方法

2.3 本マッピング方法の効果

このマッピング方法の場合、検索の結果集合を生成する際のテーブルの結合演算数が減るため、検索コストを減らすことができる。例として、格納されたXML文書(図1)の要約(図4)を結果とする検索を考える。

図3に示したマッピング方法の場合、結合演算操作を2回行わなくてはならないが、本マッピング方法の場合、結合演算操作を行う必要が無い。

```
<businessEntity businessKey="XXXXXXXX-XXXX" >
  <name>NEC Corporation</name>
  <description xml:lang="en">NEC has emerged as a global
    Internet Solution Provider.</description>
</businessEntity>
```

図4 検索結果

図3に示したマッピング方法の他に、検索対象となる要素のみからなるテーブルを作成し、格納されたXML文書を参照するというマッピング方法も考えられる。この場合、格納されたXML文書をそのまま結果として返すような検索に対しては高速な検索が可能である。しかし、UDDIレジストリの場合図4のように要約を検索結果とする必要があり、その際の検索コストが大きい。

また、本マッピング方法の格納時の効果として、格納時に検索対象部分のみにおいて値を抽出するため、値を抽出する回数とテーブルに格納する回数が減少する。これにより、格納コストも減らすことができる。

3. 実装

UDDI Version 2仕様準拠しつつ、2節にて述べた手法に基づいて実装を行った。

UDDIレジストリに格納される全データに対して図3に示すマッピング方法を適用した場合、テーブル数は31(属性数は84)である。これに対し、本研究のマッピング手法を適用した場合のテーブル数は14(属性数は59)である。実際のUDDIレジストリ運用時における検索コストの内訳として、RDBのアクセスコストが大きいことがわかっており、本方式のようにRDB構造を単純化した効果は大きい。

4. まとめ

本稿ではRDB上にUDDIレジストリを実装する際のXML-RDBマッピング方法を述べた。XML文書を格納する際、任意の構造を持つXML文書を格納するのではなく、UDDIレジストリのように、再利用性を高めるためにスキーマ構造を予め定義して格納するという用途が今後増加すると思われる。そのような用途においても本研究のXML-RDBマッピング方法は有用である。

今後の課題として、本研究のXML-RDBテーブルマッピングの有効性を示すための評価実験を行う予定である。

参考文献

- 1) "Universal Description, Discovery and Integration" homepage, <http://www.uddi.org>
- 2) eXcelon Corp., <http://www.exceloncorp.com/>
- 3) 吉川正俊, 志村壮是, 植村俊亮, "オブジェクト指向データベースを用いたXML文書の格納と検索", 情報処理学会論文誌データベース, Vol 40, No. SIG6(TOD3), 1999.