

# 携帯電話用 JAVA コンテンツ検索に向けた メタデータ構成方法に関する検討

角野 宏光<sup>†</sup> 鈴木 偉元<sup>†</sup> 加藤 剛志<sup>†</sup> 上野 英俊<sup>†</sup> 石川 憲洋<sup>†</sup>

(株) NTT ドコモ マルチメディア研究所

メタデータを用いて携帯電話用の JAVA コンテンツを検索する手法について検討した。具体的には、i-mode の JAVA コンテンツ検索に向けたメタデータ構成方法について提案し、既存の ADF ファイルを利用して自動的にメタデータを生成する手法について述べる。さらに、セマンティック Web 技術の実現に向けて、複数のデータベースにメタデータを分散して配置し、それぞれのメタデータの関連性をもとにした分散型の検索手法についても提案する。

## A Study of a Metadata Construction Method on Search of JAVA Contents for Mobile Phones

Hiroimitsu SUMINO, Hideharu SUZUKI, Takeshi KATO, Hidetoshi  
UENO, and Norihiro ISHIKAWA

Multimedia Laboratories, NTT DoCoMo, Inc.

This paper describes how to realize an automatic search of JAVA contents for mobile phones with metadata. We propose a metadata construction method of NTT DoCoMo's JAVA contents (i-appli) by using RDF. The main feature of this method is to generate RDF metadata automatically by extracting the information from ADF which is the description file of i-appli. An arrangement of the metadata for distributed databases is also presented in this paper as an example of metadata search service towards the realization of the semantic web.

### 1. はじめに

携帯電話による JAVA コンテンツの利用は、2001 年 1 月に NTT ドコモのサービス開始を皮切りに、J-フォン、KDDI と国内 3 大キャリアが JAVA コンテンツ対応のサービスを行っている。以来、携帯電話のマルチメディアサービスとして急速に普及しており、有効なコンテンツ検索手段が要望されている。

現在、WWW の世界において必要とするコンテンツを発見するためには、一般的に検索サイトが利用されている。検索サイトとしては、利用するデータベースの構築方法によりディレクトリ型とロボット型の大きく分けて 2 通りの

方法がある。Yahoo[1]に代表されるディレクトリ型は人手によって情報収集され、階層構造でデータが整理されるものであり、Google[2]に代表されるロボット型は Web ロボットと呼ばれる特殊なプログラムが自動的に収集してきたデータを元にキーワードを抽出するものである。それぞれ優劣があるが、急増する Web コンテンツをより多く検索対象とするためには、人手によるディレクトリ型よりロボット型の方が有効であるといえる。

しかし、一般的なロボット型の検索ではテキストから情報を収集するため、JAVA コンテンツ等のマルチメディアコンテンツを検索できないという問題がある。現状、携帯電話用の JAVA コンテンツの検索サイトとして GIGA アプリ[3]、アプリ Get[4]といった検索サイトが利用されているが、これらの検索サイトはいずれもコンテンツ作成者による登録型の検索サイトであり、ロボット型の検索は実現されていない。

<sup>†</sup> 株式会社 NTT ドコモ マルチメディア研究所  
NTT DoCoMo, Inc. Multimedia Laboratories  
3-5, Hikarinooka, Yokosuka, Kanagawa, 239-8536, Japan

ところで、近年 W3C(World Wide Web Consortium)により提案されたセマンティック Web が非常に注目を浴びており、次世代 Web 技術の候補として欧米を中心に活発に研究が進められている[5]。セマンティック Web は、Web サイト上の情報に意味情報を付加してソフトウェアで自動処理を行う技術であり、検索サービスへの応用が検討されている。このセマンティック Web に基づく検索サービスは、様々な形式の生データを説明するきめ細かなメタデータを登録することで、ユーザが要求する情報により近い回答を与えるものである。セマンティック Web の構想では、図 1 に示すように、現在の HTML で構成される Web とは別の空間にてマシン処理可能なメタデータを扱うものである。

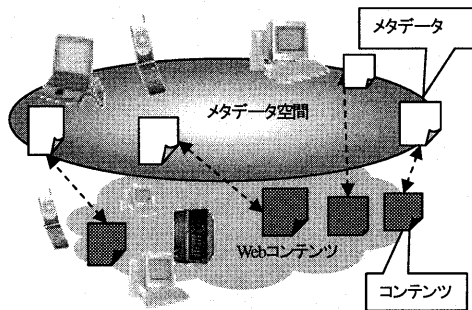


図 1：セマンティック Web におけるメタデータ空間

このように、メタデータを用いることで、テキスト以外のコンテンツの検索を可能とするだけでなく、検索の精度の向上が期待されている。

しかし、セマンティック Web に基づく検索サービスは現状研究段階であり、コンテンツにメタデータを付与する方法、メタデータの定義方法、メタデータの処理方法等を実用するためには様々な課題がある。特に、メタデータの構成に関しては、検索システムへ適用するための具体的な構成について十分検討されていない。

そこで、本研究ではメタデータを用いたマルチメディアコンテンツの検索として、携帯電話用 JAVA コンテンツの検索に関して検討を行い、具体的なメタデータ構成について提案する。また、セマンティック Web の構想に基づき、分散されたメタデータ利用方法について検討を行った。具体的には、提案のメタデータを複数の検索サイトで利用することを想定し、分散されたメタデータの関連性を考慮した分散型の検索手法について述べる。

## 2. コンテンツ検索におけるメタデータ

### 2.1. マルチメディアコンテンツの検索

マルチメディアコンテンツの検索を実現する方法として、

コンテンツに付属するメタデータにコンテンツの内容をテキストで記述し、自動的にマシン処理を行い検索することが検討されている。コンテンツの内容を適正に記述したメタデータを用いることで、テキスト及びマルチメディアコンテンツに対して精度の高い検索が実現可能となる。

このようなメタデータを利用するためには、それぞれのメタデータに対するメタデータの定義（スキーマ）を各マシンにおいて理解する必要がある。同一目的で利用されるメタデータの定義は統一されていないとマシン間で相互に利用することが困難であるため、できるだけ統一され汎用的なメタデータの定義が要望されている。

現在メタデータとしてよく知られているものとしては、図書情報で用いられる Dublin Core[6]、マルチメディアコンテンツの内容を記述する MPEG-7 等があり、それぞれの分野で利用が期待されている。また、Web の標準化を推進している W3C においては、セマンティック Web を実現するためのメタデータの記述形式として、RDF (Resource Description Framework) を推奨している[7]。RDF は、特定のアプリケーションやプラットフォームに依存せずに、相互運用可能な形で「リソースを記述する」ための標準的な枠組みを提供するものとして、1999 年に W3C において制定されたメタデータ記述方式である。

### 2.2. RDF メタデータ

RDF の基本データモデルは「リソース」「プロパティ」「値」の 3 つから構成され、それぞれ「主語 (Subject)」「目的語 (Object)」「述語 (Predicate)」の意味を持つ。例として、「Sumino (主語) はコンテンツ http://.../sample (目的語) の作者 (述語) である。」というステートメントをラベル付きグラフで表現すると図 2 のようになる。また、RDF は XML により記述され、XML の特徴であるマシンによる自動処理を可能とする。上記の RDF 構文を XML にて記述すると図 3 のように表すことができる。

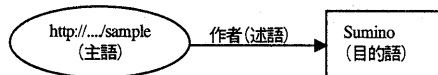


図 2：RDF 基本モデル

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description about="http://.../sample">
    <作者>sumino</作者>
  </rdf:Description>
</rdf:RDF>
```

図 3：RDF 記述例

RDFを用いることによって、コンピュータが扱う情報の分類や検索などの自動化を図ることができ、高度な検索エンジンによる Web 上のリソースの検索への応用が期待されている。

また、RDFの文書を記述する際に使うリソースやプロパティのデータ型(データ構造)の定義を行うものとしてW3CにおいてRDF Schema[8]が制定されている。RDF Schema仕様では、いくつかの特徴を持ったリソースを分類するための抽象的なリソース(クラス)を定義しており、クラス間の関係を記述するルールやプロパティの取りうる値やそのプロパティを持ちうるクラスの制約条件等を定義している。図4にスキーマの例を示す。クラス間の包含関係はsubClassOfであらわされ、プロパティ間の包含関係はsubPropertyOfであらわされる。図4において、リソースAがClassA2であり、ClassA2がClassAのsubClassと定義される場合、リソースAはClassAのインスタンス(実例)として扱うことが可能である。また、リソースAのProperty2がリソースBであり、Property2がProperty1のsubPropertyで定義される場合には、リソースAのProperty1はリソースBとして扱うことが可能である。このような処理は推論とよばれ、RDF Schemaの特徴の一つである。

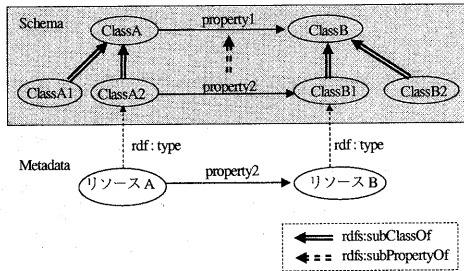


図4：スキーマ定義の例

このように、RDFはコンテンツをリソースとして扱い、それらの関連を述語表現で記述することで、柔軟性の高いメタデータ記述方法を提供する。また、メタデータの値はスキーマに基づいた意味付けができるため、共通のスキーマを用いて別々のメタデータ同士の関連を記述することが可能である。

### 2.3. 携帯電話用 JAVA コンテンツとメタデータ

現在、各キャリアの JAVA コンテンツは登録型の公式なサイトで公開されているが、NTTドコモでは、JAVA コンテンツの一般サイトでの公開が可能である。そこで、本研究では、このようなNTTドコモで提供される一般サイトの JAVA コンテンツの検索方法について検討を行う。

NTTドコモの JAVA コンテンツ(iアプリ)[9]は、クラスフ

ファイル(拡張子:jar)と ADF(拡張子:jam)で構成される。ADF(Application Descriptor File)は、アプリケーションのインストールや起動、ネットワークアクセスなどを制御するための情報が記述されたテキストファイルであり、クライアントが最初に ADF にアクセスして、端末に対応したコンテンツかどうかを判別するために用いる。ADFとして記述される内容、及び記述例をそれぞれ表1及び図5に示す。ADFはiアプリコンテンツの内容を記述しているため、メタデータとして利用することが考えられる。そこで、このADFを利用してコンテンツ検索に用いるRDFメタデータを作成する方法を提案する。また、次章にてメタデータ構成方法の検討を行う。

表1：ADF ファイル

ADF 項目	内容
AppName (必須)	アプリケーション名
AppVer	アプリケーションのバージョン
PackageURL (必須)	アプリケーション(JAR ファイル)の URL
AppSize (必須)	JAR ファイルのサイズ
AppClass (必須)	アプリケーションの起動に使われるメインクラス名
AppParam	メインクラスの起動パラメータ
KvmVer	KVM のバージョン
SpSize	ScratchPad のサイズ
LastModified (必須)	アプリケーションの最終変更日時(JST)
UseNetwork	ネットワーク機能を使用する場合の通信方式
TargetDevice	ターゲットとする携帯電話の機種名
LaunchAt	アプリケーションに自動起動を行わせる場合の起動タイミング

```

AppName=オセロ
AppVer=1.0
PackageURL=othellogame.jar
AppSize=7556
AppClass=othellogame
KvmVer=CLDC-1.0
LastModified=Thu, 01 Feb 2001 05:15:52
  
```

図5：ADF ファイルの記述例

ここで、現状の ADF の記述状況について把握するために、実際に現在 Web 上に公開されている ADF を収集して記述内容の調査を行った結果について紹介する。サンプルとして収集した 87 個の ADF に関する調査結果を表 2 に示す。ADFとして必須の項目については全て記述されているが、オプションの項目については記述がない場合も多く、正しい記述形式で記述されていないものも多く存在した。ADF を検索に用いる場合には、主に ADF の必須項目、特にアプリケーションの名称である AppName を基にした検索が有効であると考えられる。

表 2: ADF の現状調査

ADF 項目	利用頻度	調査結果
AppName (必須)	100%	全てのサンプルにほぼ適正な値が記載
AppVer	84%	ほぼ適正とみられる値が付与されている
PackageURL (必須)	100%	全てのサンプルに適正な値が記載
AppSize (必須)	100%	全てのサンプルに適正な値が記載
AppClass (必須)	100%	全てのサンプルに適正な値が記載
AppParam	0%	今回のサンプルでは設定例なし
KvmVer	30%	設定例のうち、設定の形式が守られていないものが約半数
Spsize	69%	ほぼ適正とみられる値が付与されている
LastModified (必須)	100%	全てのサンプルに適正な値が記載
UseNetwork	39%	現状では http のみ設定可能であり、本来は全てに http が設定されるべき。
TargetDevice	1%	設定されていたものは1つのみ
LaunchAt	0%	今回のサンプルでは設定例なし

### 3. メタデータ構成の提案

#### 3.1. iアプリ用メタデータ構成

前章で述べたように、ADF の内容のみではメタデータの特徴である精度の高い検索を実現するには不十分であると考えられる。そこで、ADF に記述された内容以外に、検索を行う上で有効であると思われる項目を追加する検討を行った。表 3 にメタデータに用いる項目案を示す。表 3 の項目は、できるだけ標準的に利用可能な項目として、現在標準として最も有名な DublinCore の項目を一部、ADF の項目より検索に有効と考えられる項目を一部、さらに課金の項目を一つ用いた。

表 3: メタデータ項目

項目 (Property)	値の例	設定値の説明
Title (アプリケーション名)	オセロ	Literal (ADF の AppName)
Size (サイズ)	7556	Literal (ADF の AppSize)
KvmVer (KVM のバージョン)	CLDC-1.0	Literal (ADF の KvmVer)
Date (最終変更日時)	01 Feb 2001	Literal (ADF の LastModified)
TargetDevice (対応機種)	X503	Literal (ADF の TargetDevice)
Subject (キーワード)	ゲーム、パズル、対戦型	Literal
Description (摘要)	ネットワーク対応対戦型オセロゲーム。	Literal
PriceInfo (課金情報)	100 円	Literal
Creator (作者)	http://www.docomo.ne.jp/Creator/s/001	Resource
Right (著作権)	http://www.docomo.ne.jp/Right/class-a	Resource

本メタデータ構成では各項目の値について、それぞれの項目の特徴を考慮し、リテラル (文字列) 又はリソースを用いる。リソースは URI で指定され唯一のものとして扱うことが可能であるため、プロパティ値をリソースにて記述した場合には、複数のメタデータ間においても同一の意味として利用することが可能である。本提案では、相互に利用可能な項目として、作者、著作権の項目のプロパティ値にリソースを用いた。

メタデータ及びそのスキーマについて、図 6 に RDF モデルのグラフを示す。図 6 ではメタデータのスキーマの一部として、コンテンツのリソースが属するクラスの構成について示している。本スキーマでは、コンテンツのリソースを複数のカテゴリーで階層化されたクラスに割り当てた。このクラス分けを利用することによりクラス間での推論を実現する。たとえば、Game クラスに対応するコンテンツの検索要求が生じた場合に、Game クラスの配下の 3 つのクラス Puzzle, RPG, Action に対応するコンテンツに対して検索が可能である。つまり、スキーマにてあらかじめ定義されたクラスの分類に応じてカテゴリー検索が可能である。上記のメタデータは XML 形式で記述すると図 7 のようになる。

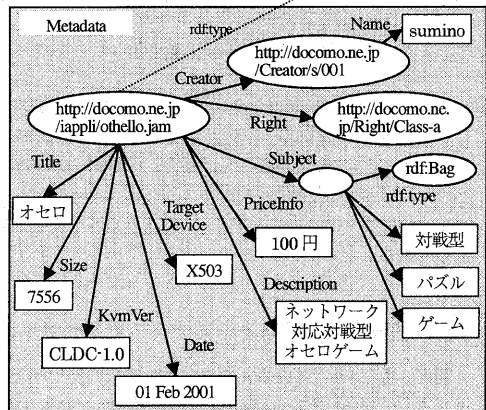
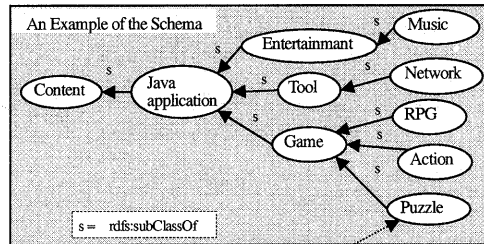


図 6: メタデータ及びスキーマ例 (RDF モデル)

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ia="http://www.nttdocomo.co.jp/2001/02/02/content-schema#"
  >
  <rdf:Description about="http://.../othello.jam">
    <dc:Title>オセロ</dc:Title>
    <i:Size>7556</i:Size>
    <i:KvmVer>CLDC-1.0</i:KvmVer>
    <dc:Date>Thu,01Feb2001 05:15:52</dc:Date>
    <i:TargetDevice>X503</i:TargetDevice>
    <dc:Subject>
      <rdf:bag>
        <rdf:li>ゲーム</rdf:li>
        <rdf:li>パズル</rdf:li>
        <rdf:li>対戦型</rdf:li>
      </rdf:bag>
    </dc:Subject>
    <dc:Description>ネットワーク.....ゲーム</dc:Description>
    <i:PriceInfo>100 円</i:PriceInfo>
    <dc:Creator rdf:resource="http://.../Creator/s/001"/>
    <dc:Right rdf:resource="http://.../Right/class-a"/>
  </rdf:Description>
</rdf:RDF>

```

図 7: アプリのメタデータ記述

### 3.2. メタデータの生成方法

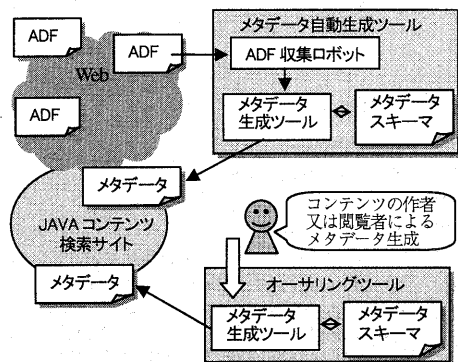


図 8: メタデータ生成方法

メタデータ生成方法の概要図を図 8 示す。前項で提案したメタデータ構成において、ADF に記述される内容については、既存 ADF より内容を抽出し自動的にメタデータを生成する。まず、メタデータの自動生成を行う場合、ADF 収集用のロボットを用いて Web 上に存在する ADF を収集し、次にメタデータ生成ツールを用いて、ADF を検索に用いる RDF 形式のメタデータへ変換してメタデータを生成する。ここで、メタデータに変換する際にはメタデータのスキーマを参照し、それに対応させて変換を行う。このよう ADF を用いることで、ロボットを用いたメタデータの自動生成が可能である。

一方、提案したメタデータのうち、ADF がない項目を

めてメタデータを生成する場合には、コンテンツ作成者、又は閲覧者がオーサリングツール等を利用し、メタデータを作成する必要がある。この場合、メタデータの項目はメタデータのスキーマを参照して入力することになる。

### 3.3. JAVA コンテンツの検索手法

以下に RDF のメタデータを利用した場合の検索方法について述べる。RDF のメタデータを利用した検索サーバに対する検索要求としては、「Title= オセロ」のように、プロパティとプロパティの値を指定することでより正確な検索を行うことが可能である。また、提案したメタデータの例 (Game クラスのコンテンツ) のように、クラスを指定して検索を行うことも有効である。

ここで、実際にこのようなメタデータを検索するためには、検索サーバにおいて RDF を処理するためのデータベースを用いる必要がある。RDF を扱うデータベースは現在研究段階であるが、RSSDB[10]、rdfDB[11]等いくつかの研究機関から既に実装例が報告されている。このような実装例では、RQL、Algae 等、独自に開発された Query 言語を用いているが、データベースの実装に依存しない標準的な問合せ言語の確立は今後の課題である。

### 3.4. メタデータの分散配置

メタデータはコンテンツを提供するサイトにて公開する場合や検索サイトにて保持する場合等が考えられるが、RDF 形式で統一されたメタデータは一般サイトや複数の検索サイト間で相互に利用可能である。そこで、サーバ間でメタデータの情報を交換する場合のメタデータの分散配置方法について述べる。図 9 にメタデータの分散構成についての概要を示す。サーバ毎に独立なプロパティやリソースに関する情報を分散して配置する。

例として、あるクラスのリソースを一括して管理するサーバについて説明する。作者情報 (Creator) に関するリソースを全てサーバ C で一括して管理することとする。検索サイトであるサーバ A やサーバ B で管理するメタデータは作者のリソースに関する情報は保持せず、検索要求として作者情報に関する問合せがあった場合にのみ、サーバ C へ問い合わせを行う。このような配置とすることで、多量なメタデータ全てに重複して作者情報を付与する必要がなく、さらにデータを更新する際には一つのメタデータを更新するだけで、更新が可能となる。

同様に別のサーバで異なるプロパティ情報を保持する場合について述べる。あるプロパティ情報を信頼できるサーバにて一括で管理したい場合、例えばサーバ D で様々なコンテンツの料金、及び著作権について一括して管理している場合には、サーバ A やサーバ B でそれらの情報について問い合わせが生じた場合のみサーバ D へ問合せを行う。図の例では検索ユーザより、サーバ A に対して「Title=オ

ゼロ”かつ PriceInfo=”100 円”のコンテンツ」という検索要求が生じた場合、サーバ A は B に対してプロパティの情報「PriceInfo=”100 円”」に合致するリソースを問合せ、その回答とサーバ A で保持する「Title=”オセロ”」に合致するリソースとを統合してユーザに応答を返す。

このようなメタデータの分散型の構成は、複数のサイトで管理する異なるメタデータの情報を統合することで膨大なメタデータ情報を利用した巨大なサイバライブラリを形成することが可能である。

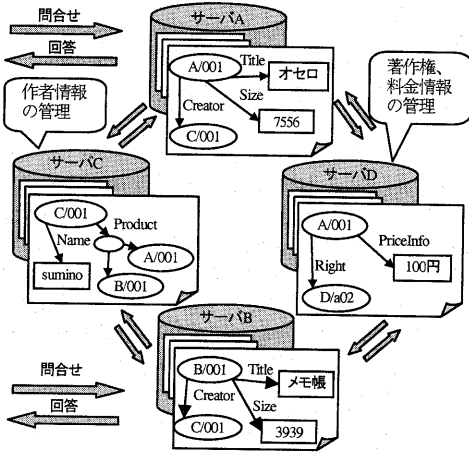


図 9：分散型メタデータ構成

#### 4. 考察および課題

本検討にて紹介したように、メタデータを用いることでマルチメディアコンテンツの検索が可能となり、さらに、コンテンツの内容を詳細にメタデータに記述することで、精度の高い検索を実現することが可能である。

また、このようなメタデータは、複数のサーバ間で情報を流通させ、相互に処理可能な形式で提供されることが期待されている。そのため、汎用的かつ標準的なメタデータの確立が課題である。今回は携帯電話用 JAVA コンテンツの検索を実現するためのメタデータを提案したが、他の様々な種類のコンテンツに対応したより汎用的なメタデータを定義することが重要であると考え。それにより、異なる種類のコンテンツ間で共有できる項目（作者情報など）を相互に利用が可能となる。

現状、今回メタデータ記述に用いた RDF は最も汎用的と考えられているが、RDF はまだ研究段階であり、実用レベルに達していない。主な課題としては、メタデータ記述、意味及び知識の記述方式、意味及び知識の記述方式、実装等があり、特に実装については、現状では実用レベルの RDF のデータベース及び推論エンジンの実装は存在しないため、それらの開発は今後の大きな課題である。

検索に用いるメタデータとしては、様々なものを適用することが考えられる。意味及び知識の記述を行うことにより、プロパティ、クラスの種類や関係付けを補うオントロジに基づく記述方法についてはさらなる検討が必要である。

また、複数のサーバに情報を分散させた分散型の検索について検討を行ったが、分散されたデータを利用したセマンティック Web の実現形態については、様々な効果が得られると考えられるため、今後の検討が期待される。

#### 5. おわりに

本研究では、携帯電話用 JAVA コンテンツを検索するためのメタデータについて検討を行い、RDF を用いたメタデータ構成を提案した。また、本提案のメタデータ生成方法として、JAVA コンテンツに対して ADF を利用しメタデータを自動的に生成する方法を示した。このメタデータを用いた検索方法により、自動的に JAVA コンテンツの検索を実現するとともに、精度の高い検索を実現することが可能となる。また、セマンティック Web に関する技術の一例としてメタデータの分散配置方法に関して提案を行い、メタデータの効率的な利用方法について述べた。今後は RDF のメタデータを利用した検索システムを実装し、本検討で述べた JAVA コンテンツの検索手法について実証する予定である。

#### 参考文献

- [1] <http://www.yahoo.co.jp>
- [2] <http://www.google.com>
- [3] <http://www.appget.com>
- [4] <http://g-appli.net>
- [5] W3C, Semantic Web Activity, <http://www.w3.org>
- [6] DublinCore Metadata Initiative, <http://dublincore.org>
- [7] Ora Lassila, Ralph R. Swick, "Resource Description Framework(RDF) Model and Syntax Specification": W3C Recommendation 22 February 1999.
- [8] Dan Brickley, R.V. Guha, "Resource Description Framework (RDF) Schema Specification 1.0": W3C Candidate Recommendation 27 March 2000.
- [9] [http://www.nttdocomo.co.jp/p\\_s/imode/](http://www.nttdocomo.co.jp/p_s/imode/)
- [10] Sofia Alexaki, Karsten Tolle, "The ICS-FORTH RDF Suite: Managing Volumes RDF Description Bases", WWW conference 10, 2001
- [11] R.V.Guha, "rdfDB", <http://web1.guha.com/rdfdb/>