

# 柔軟なシステム連携のための意味情報に基づくメッセージマッピング手法の提案と評価

## Verification of Message Mapping Technique based on Semantics for Flexible System Cooperation

中辻 真<sup>▼</sup> 三好 優<sup>◆</sup> 木村 辰幸

Makoto NAKATSUJI Yu MIYOSHI  
Tatsuyuki KIMURA

近年の企業活動は、ネットワーク上の多様なシステムで働くソフトウェアコンポーネントの分散協調に基づき実行される事が多い。しかし、複数システム間のメッセージやプロセスを規定するインタフェース定義が業務部門ごとに個別最適で設計されているため、インタフェース整合に関する種々の問題が生じている。本稿では急速に変化するビジネス環境でシステム設計者の目的に応じ自動的にシステム連携を実現するため、オントロジ言語 OWL を用いメッセージのフォーマットとフォーマットの持つ意味情報の関係を機械処理可能な知識ベースとしてモデル化し、様々なシステムの知識ベースをマッピングする事でメッセージの差異を半自動的に吸収するメッセージマッピング手法を提案し、実運用ネットワーク管理システムを対象とした評価実験を行う。

These days, many companies are executing their business aims based on decentralized cooperation of software components which work on various systems over a network. However, messages and processes between systems are designed individually in each operations division. Therefore, the system development for adjusting interfaces is expensive, so the companies cannot introduce their services in a dynamic business environment. To resolve such problems, we propose message mapping technique which models the relationship between the message formats and semantics on the formats by using Web Ontology Language and executes the mapping between the message formats by using semantics. We evaluated our proposed methods based on the interface specifications of real network management systems.

### 1. はじめに

e ビジネスなどの企業活動は、ネットワーク上の多様なシステム上で働くソフトウェアコンポーネントの分散協調に基づいて実行される事が多くなっている。しかし、複数シ

▼ 正会員 日本電信電話株式会社NTTネットワークサービスシステム研究所 nakatsuji.makoto@lab.ntt.co.jp

◆ 日本電信電話株式会社NTTネットワークサービスシステム研究所 {miyoshi.yu,kimura.tatsuyuki}@lab.ntt.co.jp

ステム間のメッセージやプロセスを規定するインタフェース定義が業務部門ごとに個別最適で設計されているため、インタフェース整合にかかるコストや導入までの開発期間が長くビジネスチャンスに即したサービス導入に問題がある[1]。これに対し著者らは、急速に変化するビジネス環境でシステム設計者の目的に応じ、迅速かつ自動的にシステム連携を実現することを目的に研究を進めて来た[2]。本稿では、オントロジ言語OWL[3]を用い、メッセージフォーマットとフォーマットの持つ意味情報の関係を知識ベースとしてモデル化するインタフェースモデリング手法、様々なシステムの意味情報を半自動でマッピングする事でメッセージの差異を吸収するメッセージマッピング手法を提案し、実運用NW管理システムのインタフェース仕様を題材とした検証を実施する。

### 2. 既存システム連携技術と関連研究

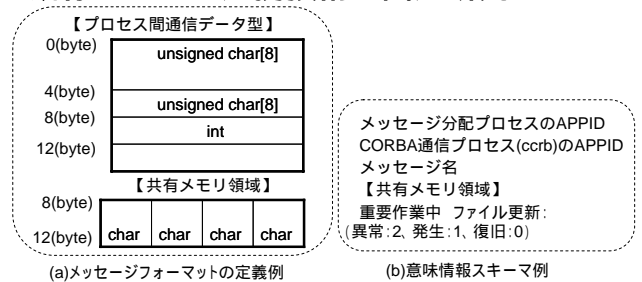


図1 メッセージフォーマットの意味情報スキーマ欠如

Fig.1 Semantic schemas lacking in message formats.

まず、既存システム連携技術の特徴を自動化の観点より整理する。従来型技術やCORBA[4]はメッセージ処理コストやサイズが小さいためネットワーク管理など緊密な情報交換を行う領域に適す。一方Web Services[5]は、メッセージをXMLベースで記述するためWeb上のエンドユーザ向けサービスなどテキスト形式の情報交換を行う領域に適す[1]。このように各技術の適用領域は異なるため、Web ServicesのみでなくCORBAや従来型技術によるシステム構築も今後継続して行われると考えられる。そのため著者らは、各技術に対する連携自動化が必要と考えるが、現状、各技術ともメッセージ記述形式が固定であり、連携を行うには予めシステム間で交換されるメッセージフォーマットを人手整合する必要がある。例えば、従来型技術におけるメッセージフォーマットは図1-(a)に示すように各メッセージ要素の識別IDはメッセージフォーマット内の格納メモリ位置で定義され、データタイプは各メモリ位置に対するデータタイプ定義により別途表される。しかし、図1-(b)に示すようなメッセージフォーマットに割り当てられる意味情報スキーマとの関係が機械識別可能に記述されていない為、メッセージフォーマットが異なるシステム間では、インタフェース仕様書等に記載される情報を基に人手で意味を比較し整合しなければ連携できない。

これに対しSemantic Web Services[6]では、Web上の情報に対しその背景知識を機械処理可能な意味情報とし記述し、様々なソフトウェアが自動で意味情報による処理を実行する事を目指すSemantic Web技術[7]をWeb Servicesにおけるサービスの自動選択・実行・連携に適用する事を試みている。そのためWeb Servicesで公開されるインタフェースに対し、OWLを拡張したOWL-S[6]を用い入出力メッセージ、サービスを行うための前提条件、サービスにより得られる効果の意味記述を行う。他にもWSMO(Web Services Modeling

表 1 OWL DL による意味情報記述  
Table 1 OWL description used in our technique.

クラス公理			プロパティ公理		インスタンスによる事実の記述	
rdfs:subClassOf	owl:equivalentClass	owl:oneOf	プロパティの制約		owl:sameAs	
			owl:allValuesFrom	rdfs:range		rdfs:domain
参照クラスのサブクラスとして、主語クラスの必要条件を形成	参照クラスと同じインスタンスを持つクラスという、主語クラスの必要十分条件を形成	主語クラスのインスタンスを過不足なく列挙する	主語クラスの全てのインスタンスについて、プロパティの全ての値が目的語クラスのインスタンスであることを示す	プロパティの目的語は、参照クラスのインスタンスである	プロパティの主語は、参照クラスのインスタンスである	2つのインスタンスが同一であることを示す

Ontology)[8]では、Semantic Web Servicesにおけるメディアエータをオントロジによりモデリングし、自動的なサービス連携を実現するためのフレームワークの構築を目指している。しかし、上記研究におけるサービス連携はコンセプトを規定する段階であり、連携に必要なメッセージフォーマット整合手法の提案・検証が行われておらず、また従来型技術で構築されたシステムへの適用も考慮していない。

### 3. インタフェースモデリング手法の提案

本章ではメッセージフォーマットと意味情報の関係を機械で自動識別可能な知識ベースとして記述するインタフェースモデリング手法の提案を行う。

#### 3.1 知識ベースの構築

システム設計者が知識ベースを記述するには記述ルールの確立が必要である。そのため、NW 管理システム(NMS)等のインタフェース仕様書を基に実運用システムインタフェースの調査・分析を行い、従来型技術のインタフェースに必要な記述ルールを抽出した。なお Web Services(WS)の記述ルールは、OWL 記述された意味情報スキーマと WSDL[5]で定義するメッセージフォーマットとの関係を記述できるため、OWL-S のグラウンディング[6]を利用する。

##### 3.1.1 知識ベース記述の要求条件

システム連携を実現するには、システム間で交換されるデータであるメッセージ要素、各メッセージ要素の識別 ID とデータタイプを識別できる事、フォーマットスキーマと意味情報スキーマの関係を機械識別可能とするため両者を分離記述した上で対応関係を記述する事が必要である。そのため知識ベースの記述には以下の要求条件を満たす必要がある。

- 意味情報スキーマとフォーマットスキーマ間の対応関係を機械識別可能とするため、両者を分離記述する事
- スキーマ記述ができる事
- 過去のマッピング結果の再利用のため、各クラスが所属するシステムやメッセージを識別できる事
- 連携の際、メッセージ要素を整合する必要があるためメッセージ要素を識別できる事
- 意味情報・フォーマットスキーマの関係を識別できる事
- 各システムのインタフェースの持つ特性を表現するため、メッセージ要素のシステム間の交換形式を識別できる事
- データタイプの整合も実現するため、メッセージ要素の持つデータタイプを識別できる事

なお、意味情報スキーマとフォーマットスキーマの対応関係を機械識別可能にするため、知識ベースの記述には、クラス間の関係を機械識別可能に記述できる OWL DL を用いる。OWL DL による記述法則の 1 部を表 1 に示す[3]。

##### 3.1.2 記述ルールの確立

OWL DL による記述ルールを要求条件と対応して示す。

- ルートクラスは“知識ベース”としサブクラス“意味情報”と“メッセージフォーマット”を持つ。意味情報スキーマの所属クラスは“意味情報”の下位クラスとし、フォーマットスキーマも同様とする。

- クラススキーマは、owl:subClassOf を用い表現する。
- 対象システムを主語クラス“知識ベース”のプロパティ“belongingSystem”の述語クラス“システム”のインスタンスより識別し、対象メッセージもプロパティ“belongingMessage”より識別する。
- メッセージ要素の識別 ID をクラスとし意味情報スキーマの対応オブジェクトもクラスとして記述する。同様にメッセージ要素をインスタンスとして記述し、意味情報スキーマの対応オブジェクトもインスタンスとして記述する。
- フォーマットスキーマと意味情報スキーマの関係は、クラス間の同値関係を示す owl:equivalentClassOf とインスタンス間の同値関係を示す owl:sameAs で記述する。
- 各クラスのインスタンス(メッセージ要素)がシステム間で交換される際の特性をクラスの構成要素記述により表現する。
- データタイプはフォーマットスキーマの所属クラスが持つプロパティ“hasType”の述語クラス“Type”のインスタンスより識別する。

本ルールより、OWL-S 記述されたサービスのメッセージパラメータを従来型技術のメッセージ要素と対応でき、WSDL のみに対応する OWL-S を従来型技術へ適用できる。

#### 3.2 NMS を対象とした知識ベースの試作

記述ルールの正当性確認のため、実運用されている 3 種の NMS と 1 種の情報取得システムの知識ベースを試作した。

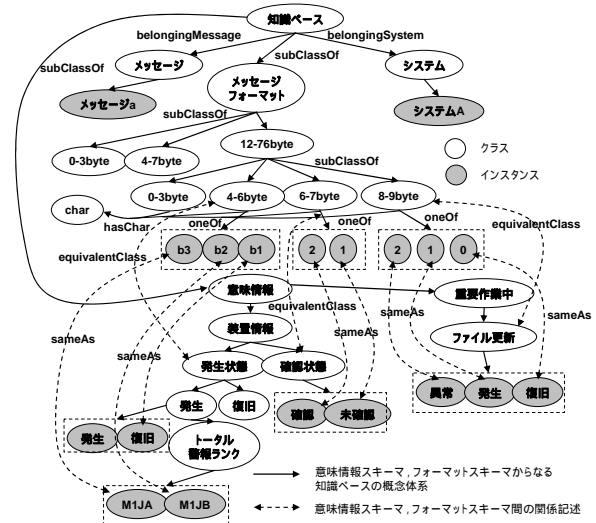


図 2 試作した NMS の知識ベースの一部 (イメージ)

Fig.2 Knowledge base constructed over NMS.

NW オペレーション領域における意味情報は NW やノードなど管理対象の状態情報でありインタフェース仕様書はこうした情報を記載しているため、仕様書より意味情報を抽出し知識ベースを試作した。今回用いた NMS は従来型技術で構築されテーブル型のメッセージフォーマットを備えるが、仕様書には意味情報とメッセージフォーマットが混在して記載され(図 1)、意味情報スキーマとフォーマットスキーマの関係を機械識別出来ない。また現在の仕様書は UML により記述される事が多くなっているため、インタフェース仕

様書を提案する記述ルールに従い UML 記述してから OWL 記述による知識ベースへ変換できればメッセージマッピング手法は実用的になる。そこで知識ベースエディタ protégé[9]が UML 記述されたクラス図を OWL 記述に自動変換出来るため protégé により知識ベースを試作した(図 2)。

試作の結果、従来型技術の知識ベースを記述ルールに従い機械識別可能に記述でき、提案ルールの正当性を確認できた。

#### 4. メッセージマッピング手法の提案

本研究では、複数システムの意味情報スキーマ間をマッピングし、その結果からメッセージフォーマット間を自動マッピングする。マッピング方式として以下を検討した[2]。

1. 名前のみによるマッピング
- 2-1 クラス諸属性利用方式
- 2-2 方式 2-1 に加え蓄積したマッピング結果を再利用
- 2-3 方式 2-1 に加えパターンマッピング

記述ルールに基づきメッセージフォーマットの特徴をクラスの持つインスタンス集合やクラス構成要素を用い記述する。こうしたクラスの諸属性を活用しマッピング精度を高める事が出来ると考え、名前属性のみを用いる方式 1 でなく、方式 2 を選択する。以下、方式 2-1、方式 2-2 を説明する。

##### 4.1 方式 2-1：クラス諸属性利用方式

異なるシステムの意味情報スキーマに所属するクラス間の近似度をクラス名、インスタンス集合、クラス構成要素といったクラスの諸属性より求める[2]。

以下、ある意味情報スキーマ I のクラス  $C_i$  から見た異なる意味情報スキーマ J のクラス  $C_j$  の近似度  $S(C_{ij})$  の計測を基に説明する。 $C_i$  から見た  $C_j$  のクラス名の近似度を  $S(N_{ij})$  とする。クラス名の近似度計測には、クラス名を形態疎解析により形態素に分割し、形態素の一致度に基づき計算する[2]。

次に、クラスの持つインスタンス集合間の近似度を計測する。そのため、 $C_i$  に所属するインスタンス  $I_i$  から見た、 $C_j$  に所属するインスタンス  $I_j$  の名前の近似度  $S(I_{ij})$  を計測する。クラス名の近似度計測と同じく、形態素の一致度より名前属性の近似度を計算する。次に、ヒューリスティックな閾値  $\theta$  を用い、 $S(I_{ij}) > \theta$  (式(1))を満たすならば、インスタンス  $I_i$  と  $I_j$  は、対応すると考える。そして、 $C_i$  と、 $C_j$  の持つインスタンス集合を  $U_i$ 、 $U_j$  とすると、インスタンス集合間の近似度  $S(U)_{ij}$  は  $S(U)_{ij} = |U_i \cap U_j| / |U_i \cup U_j|$  と表される。

続いて、クラスの構成要素間の近似度も計測する。ここで、 $C_i$  から見た  $C_j$  の構成要素間の近似度を  $S(O)_{ij}$  とする。本研究では、クラスに所属するインスタンスのうち 1 つのみが一回のメッセージ交換によりシステム間で交換されるという特徴を owl:oneOf 要素で記述する。そこで  $S(O)_{ij}$  は、 $C_i$  と  $C_j$  両方に owl:oneOf 要素がある場合または無い場合は 1、 $C_i$  と  $C_j$  のどちらかにのみ owl:oneOf 要素がある場合は 0 とする。

次にクラス名、インスタンス集合、クラス構成要素に対し、クラス間の近似度へ与えるヒューリスティックな重み係数  $\kappa$ 、 $\lambda$ 、 $\mu$  を決定する。そして、クラス間の近似度  $S(C_{ij})$  を  $S(C_{ij}) = \kappa \cdot T(S(N)_{ij}) + \lambda \cdot T(S(U)_{ij}) + \mu \cdot T(S(O)_{ij})$  (式(2))により与える。ここで、重みを考慮する関数  $T$  として、知識ベース J に所属するターゲットクラスを母集合とし、平均  $\bar{x}$ 、分

散  $\sigma_x^2$  をもつ式  $T(x_i) = (x_i - \bar{x}) / \sigma_x$  を与える。そして、ターゲットクラスを  $S(C_{ij})$  によりランキングし、クラス間の関係をクラスの持つプロパティ、インスタンスを含めてユーザへ提示する。ランキング結果のみをユーザ確認するため、クラス間・インスタンス間の対応関係の確認回数を抑制できる。

##### 4.2 方式 2-2：マッピング結果再利用方式

本方式では、過去のマッピング知識をマッピングに再利用する事でマッピング精度や自動化を向上する事を試みる。

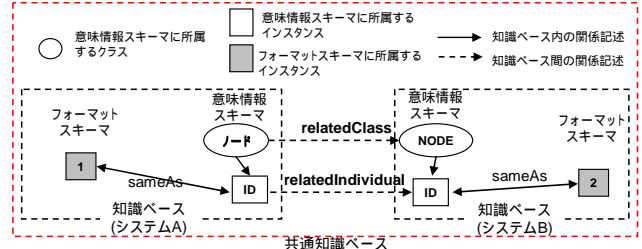


図3 共通知識ベースの具体例

Fig.3 Example of a Public knowledge base.

共通知識ベースの具体例を図 3 に示す。本研究は動的なシステム連携を指向するため、共通知識ベースでは意味情報スキーマ間のマッピング結果を、あるシステム設計者から見て他システムの意味情報が自システムの意味情報と対応関係を示すが、その他のシステム設計者からはその対応関係を保障しない形で蓄積する。つまり、あるシステム設計者から見て他システムのクラスやインスタンスが狭域的に一致すれば、関連性を持つとして関係記述しシステム設計知識として蓄積する。そのため知識ベース間の関係は、整合性を保証しない言語 RDF により記述する。

具体的には、クラス間の対応関係記述は、所属インスタンス全てが一致していなくても一部のインスタンスが一致していれば関係記述(relatedClass 記述)を行い、インスタンス間の対応関係の記述は、共通知識ベースに所属するシステムのうち、一部のシステム間で狭域的に同値であるだけでも関係記述(relatedInstance 記述)を行い再利用性を高める。

例えば、図 3 ではシステム A の設計者から見てシステム A の(ノード- ID)と変換可能な候補は、システム B の(NODE - ID)である事が関係記述より自動的に分かる。一方、システム B の設計者から見ると、システム B の(NODE - ID)とシステム A の(ノード- ID)が変更可能性を持つ事が分かるが、システム連携に再利用するには設計者の確認が必要となる。

##### 4.3 マッピング自動化の評価方法

マッピング自動化の評価のためマッピング結果判定に要する人手回数とマッピング結果の精度を比較する。精度の尺度は、マッピング結果中の正解が全正解に占める割合(再現率)とマッピング結果中の正解の割合(適合率)を用いる。

##### 4.4 NMS を対象とした検証実験

実運用 NMS のインタフェース仕様を題材とした検証を行った。現段階では 407 個のソフトウェアコンポーネント(SC)より構成される 2 種類の NMS の SC のうち 3.2 節で試作した 15 個の SC に提案手法を適用し方式 2-1 と方式 2-2 に対し評価した。なおマッピング判定には実験で用いた仕様書を用いた。

最終的なマッピング結果に誤りがあるとシステム連携として実用的でない。そこで実験では、クラス間の自動マッピング結果を近似度でランキングしユーザ確認させて誤りを除去するユーザ対話型のマッピングツールを実装した。以降の結果は、クラス名、インスタンス名の近似度計測の前準備

としていくつかの無駄語を削除している。また4.1節の式(2)において、 $\kappa = 0.35$ 、 $\lambda = 0.35$ 、 $\mu = 0.3$ とした。これは、クラス名のみではマッピング結果として選択されない正解をインスタンス集合の近似性より選択できるケースが多い事と、クラス構成要素に基づくマッピングよりメッセージ要素の交換形式の異なる結果を除去できる事を定性的に確認した為である。なお形態素解析には Sen[10]を利用した。

表2 従来手法と方式2-1の人手マッピング回数の比較  
Table 2 Comparison of mapping frequencies by the user.

	人手による確認回数
従来手法	25641
方式2-1(ランキング2位まで確認)	468
方式2-1(ランキング1位まで確認)	374

表3 方式2-1によるマッピングの再現率・適合率  
Table 3 Recall and Precision by Method 2-1.

	クラスマッピング	
	ランキング2位	ランキング1位
再現率	229/290 0.790	177/290 0.610
適合率	229/468 0.489	177/374 0.473

表4 マッピング再利用による自動化の検証(方式2-2)  
Table 4 Evaluation of automatic system cooperation.

再利用回数	マッピング結果中の正解数(レベル2)	再利用回数 / マッピング結果中の正解数(レベル2)	再利用結果中のマッピング誤り数
138	229	138/229 60.3%	0

表5 インスタンス間のマッピング結果中の正解数の比較  
Table 5 Comparison of numbers about instance mapping.

属性利用(方式2-1)	共通知識ベース再利用(方式2-2)	増加数
328	359	31

メッセージフォーマットの整合における人手確認回数を、仕様書を参照しマッピング可能性のある全てのクラスの組合せを見比べる従来手法と、知識ベースを用いたクラス間の自動マッピング結果のみを人手整合する方式2-1とで比較した結果を表2に示す。表ではユーザに対しランキングを2位まで確認させた場合と1位のみ確認させた場合を示す。また表3に、方式2-1におけるマッピングの再現率・適合率を示す。

従来手法と比較し方式2-1は人手による対応関係の確認回数を大幅に削減できる(表2)。またランキングを2位まで確認すると確認回数は増えるが再現率が向上し、適合率はほとんど変化がない(表3)。これらより再現率を高くする事でメッセージフォーマットの差異吸収コストを小さくする必要があり、更に最終的なマッピング結果から誤りを完全に除去する必要がある場合は、ランキングの上位結果をユーザ確認させる事が有効である。なお、従来手法の確認回数に関しては、NMS に関し専門的なユーザが確認した場合、実験結果より減少する可能性があるが、実際のNMSを構成するソフトウェアコンポーネントが更に多いことや膨大な量の仕様書を参照する必要を考慮すると、本結果から提案手法がメッセージフォーマットの整合コスト削減に有効であると考えられる。なおランキング2位と3位間の再現率向上は小さかった。

また方式2-2に対し同環境で実験を行った。表4に共通知識ベースによる自動化の検証結果を示す。マッピングを行う中で60.3%のマッピング正解を過去のマッピング知識の再利用に基づき導出できた。これは今回の実験は2システムの知

識ベース間でマッピングを行ったため再利用される結果が多い事もあるが、自動化に対し方式2-2が有効である事が分かる。また表5はインスタンス間の自動マッピング結果中に含まれる正解数を方式間で比較している。方式2-2はインスタンス間のマッピング結果も蓄積・再利用し、方式2-1よりも約9.5%多く正解をユーザ提示できた。これらより方式2-2がマッピング自動化と精度向上に有効である事が分かった。

## 5. 結び

本稿では自動的なシステム連携を実現するため、フォーマットスキーマと意味情報スキーマの対応関係を知識ベースとしてモデル化し、意味情報スキーマ間のマッピングによりメッセージフォーマット間の差異を自動吸収するメッセージマッピング手法を提案し、実運用NMSを対象とする検証により2種類のNMS間でのインタフェース整合において提案手法が有効性を持つ事を確認した。今後は試作した4種類のシステムの知識ベースに対し方式比較実験を進め再現率向上やマッピング結果を用いたインタフェース自動生成の検証を進める。また、連携自動化のためには、ソフトウェアコンポーネント間のメッセージの交換手順であるプロセスの整合まで考慮する必要があると考え、現在研究を進めている。

## 【文献】

- [1] 城田真琴: “ユビキタスネットワーク時代のインテグレーション動向”, NRI 技術創発 (2004).
- [2] 中辻真, 三好優, 木村辰幸: “意味情報に基づくインタフェースマッピングによるシステム連携手法の提案と評価”, DEWS2005 (2005.3).
- [3] 神崎正英: “セマンティック・ウェブのためのRDF/OWL入門”, 森北出版株式会社 (2005).
- [4] Slama, D., Garbis, J. and Russell, P.: “Enterprise CORBA”, Prentice Hall Ptr (1999).
- [5] Booth, D., Haas, H. et al.(eds.): “Web Services Architecture”, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, World Wide Web Consortium, (2004.2).
- [6] Martin, D. et al.: “OWL-s: Semantic Markup for Web Services”, <http://www.daml.org/services/owl-s/> (2004).
- [7] Berners-Lee, T.: “An attempt to give a high-level plan of the architecture of the Semantic Web” (1998).
- [8] Feier, C.(eds.): “D3.1v0.1 WSMO Primer”, <http://www.wsmo.org/TR/d3/d3.1/v0.1/>, DERI (2005.4).
- [9] protégé home page, <http://protege.stanford.edu/>
- [10] Sen Project home page, <http://ultimania.org/sen/>

## 中辻 真 Makoto NAKATSUJI

NTT ネットワークサービスシステム研究所。2003 京都大学大学院情報学研究科システム科学専攻修士課程了。電子情報通信学会、日本データベース学会など会員。

## 三好 優 Yu MIYOSHI

NTT ネットワークサービスシステム研究所。2000 早稲田大学大学院理工学研究科電子・情報通信学修士課程了。2003 年度電子情報通信学会学術奨励賞、電子情報通信学会など会員。

## 木村 辰幸 Tatsuyuki KIMURA

NTT ネットワークサービスシステム研究所。1990 慶応義塾大学理工学部機械工学科卒、入社以降ネットワークオペレーションの研究開発に従事。電子情報通信学会正員。