

## WS-PROVE を用いた Web サービスメトリクスの実験的評価

串戸 洋平 石井 健一 井垣 宏 中村 匡秀 松本 健一

奈良先端科学技術大学院大学情報科学研究科 〒630-0192 奈良県生駒市高山町 8916-5

E-mail: {youhei-k, keni-i, hiro-iga, masa-n, matumoto}@is.aist-nara.ac.jp

あらまし 本論文では、我々の研究グループが提案している 3 種類の Web サービスメトリクス(RFWS, NOWS, NHTWS)について、Web サービスアプリケーションの効率性・信頼性とメトリクスとの関係を調べる評価実験を行った。具体的には、効率性の評価実験として当研究室で開発した WS-PROVE (Web Service Prototyping and Validation Environment)を用いて、Web サービスアプリケーションのプロトタイプを構築・性能計測し、Web サービスメトリクスとの関係について考察した。また、信頼性の評価実験として、Sum of Disjoint Products (SDP)アルゴリズムを用いて Web サービスアプリケーションの信頼性を導出し、Web サービスメトリクスとの関係について考察した。その結果、Web サービスのオーバーヘッドやネットワークを利用する特徴などから、Web サービスメトリクスと効率性・信頼性について関連が認められた。

キーワード Web サービス, ソフトウェアメトリクス, プロトタイピング

## An empirical Study of Web Service Metrics with WS-PROVE

Youhei KUSHIDO Ken-ichi ISHII Hiroshi IGAKI

Masahide NAKAMURA and Ken-ichi MATSUMOTO

Graduate School of Information Science, Nara Institute of Science and Technology

8916-5 Takayama, Ikoma, Nara, 630-0192 Japan

E-mail: {youhei-k, keni-i, hiro-iga, masa-n, matumoto}@is.aist-nara.ac.jp

**Abstract** In this paper, we evaluated in experiments the relationship between the quality of Web Service applications and the Web Service Metrics ( RFWS, NOWS, and NHTWS) that have been proposed by our research group. First, the performance of Web Service application was measured using WS-PROVE developed so that we could build the prototype of Web Service application topology freely, and considered the relationship with Web Service Metrics. Next, the reliability of Web Service application was calculated using the reliability assessment algorithm “Sum of Disjoint Products ( SDP )”, and considered the relationship with Web Service Metrics. As a result, we revealed experimentally a relationship between Web Service Metrics and the quality of Web Service applications.

**Keyword** Web Service, software metrics, Prototyping

### 1. はじめに

ネットワークの進歩とソフトウェアの多様化により、多種多様なサービスが溢れている。このような中、より高度なサービスを効率的に実現するためにサービスの連携が必要になってきており、その一手段として Web サービス[1]が注目されている。Web サービスとは標準化された手順(XML, SOAP/HTTP, UDDI)により異なるシステム間の連携を実現し、既存サービスの再利用性を向上するものである。

近年 Web サービスを用いたシステムがいくつか開発されつつある。しかし、Web サービスを用いたシステムは、実地運用が開始されてからまだ日が浅く、体系だった開発方法論について十分に議論をされてはい

ない。そこで、我々の研究グループでは Web サービスを用いたシステム開発において Web サービスの連携方式と品質との関係について研究を行ってきた[3][4][5][6]。その中で我々は Web サービスメトリクス[6]という 4 つのメトリクス(RFWS, NOWS, EMWS, NHTWS)を提案した。Web サービスメトリクスとは Web サービスを用いたシステムの品質を定量的に評価することを目的としている。しかし、それらのメトリクスについて品質との関係の評価が十分に行えていなかった。

そこで本論文では、Web サービスメトリクスと品質との関係性を評価することを目的に、二つの評価実験を行った。具体的には、我々の研究グループで開発した

Web サービスの連携を自由に構築できる WS-PROVE (Web Service Prototyping and Validation Environment)[3] を用いて効率性の実験の評価を行った。また, Sum of Disjoint Products (SDP)[7]という信頼性評価アルゴリズムを用いて信頼性の評価も行った。実験の結果, NOWS, RFWS, NHTWS の三つのメトリクスで Web サービスアプリケーションの非機能的な部分での効率性・信頼性を評価できることがわかった。

## 2. Web サービスメトリクス

文献[6]において, 我々は以下の4つの Web サービスメトリクスを提案した。

- RFWS(Response For a Web Service)
- NOWS(Number Of Web Services)
- EMWS(Effective Methods per Web Service)
- NHTWS(Number of Hop to Terminus Web Service)

本章では, これら4つのメトリクスについての定義と適用例を説明する。以降の定義において, CA をメトリクスの計測対象とするクライアントアプリケーション,  $W_i$  ( $0 \leq i \leq n$ ,  $n$ : 利用する Web サービス数)を Web サービス(WS)とする。Web サービス  $W_a$  が別の Web サービス  $W_b$  を呼び出す場合には,  $W_a$  を CA とみなしてメトリクスを算出する。

### 2.1. RFWS(Response For a Web Service)

定義:

$W_1, W_2, \dots, W_n$  を CA が直接呼び出す全ての WS とする。このとき, CA の RFWS を以下に定義する。

**RFWS = CA と  $W_i$  間でのメッセージの総和**

例: 図 1(a)では, CA が直接呼び出す Web サービスは WS1 と WS2 の二つであり, それらの Web サービスとの間のメッセージの数は2であり, RFWS は4となる。仮に図 1(b)のように CA が WS3 を新たに直接利用すると, RFWS は WS3 とのメッセージのやり取りの数だけ増加し6となる。

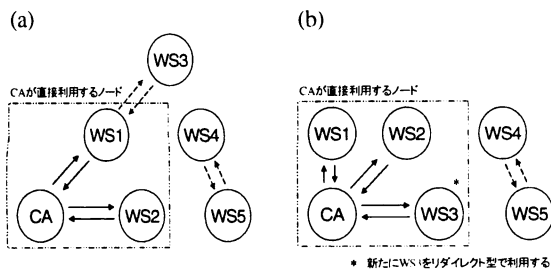


図 1 RFWS の適用例

### 2.2. NOWS(Number Of Web Services)

定義:

$W_1, W_2, \dots, W_n$  を CA が関係する全ての WS とする。このとき, CA の NOWS を以下に定義する。

**NOWS = n**

例: 図 2(a)では, CA が直接利用する Web サービスは WS1 と WS2 の二つであり, 間接的に利用する Web サービスは WS3 の一つである。従って NOWS は3となる。図 2(b)のように CA が新たに WS4 を間接的に利用する(\*1)か, 直接的に WS5 を利用する(\*2)とそれらの分だけ増加し NOWS は5となる。

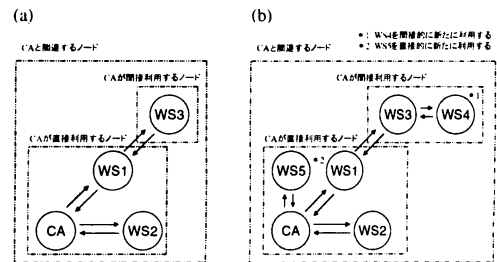


図 2 NOWS の適用例

### 2.3. EMWS(Effective Methods per Web Service)

定義:

$W_1, W_2, \dots, W_n$  を CA が直接呼び出す全ての WS とする。このとき, CA の EMWS を以下に定義する。

**EMWS = EM / PM**

**EM:**  $W_1, W_2, \dots, W_n$  の利用メソッド数の総和

**PM:**  $W_1, W_2, \dots, W_n$  の公開メソッド数の総和

例: 図 3(a)では, CA が利用している Web サービスは WS1 と WS2 であり, WS1 ではサービスを10個公開しており, WS2 ではサービスを2個公開している。また, 図 3(a)中において, CA は WS1 のサービスを二つと WS2 のサービスを一つ利用している。よって, EMWS は  $(2+1) \div (10+2) = 0.25$  となる。仮に, 図 3(b)のように, WS1 の代わりに, WS1 と同じサービスを提供するが公開しているサービス数が4個である WS3 を利用すると, EMWS は  $(2+1) \div (4+2) = 0.5$  となる。

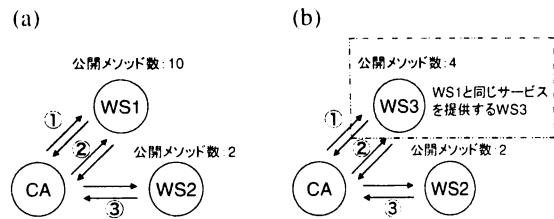


図 3 EMWS の適用例

### 2.4. NHTWS(Number of Hop to Terminus Web Service)

定義:

$W_1, W_2, \dots, W_k$  を WS とする。このとき,  $W_i$  が  $W_{i+1}$  ( $0 \leq i \leq k$ ) を要求するような以下の系列において,  $\rho$  の長さ(=k)をホップ数と定義し,  $\text{hop}(\rho)$  と書く。

**$\rho = W_0 (=CA), W_1, W_2, \dots, W_k$**

このとき, CA が系列  $\rho_1, \rho_2, \dots, \rho_n$  を持つとき, CA の NHTWS を以下のように定義する。

**NHTWS = Max{hop( $\rho_i$ )}**

例：図 4(a)では、CA から WS1 までのホップ数は 1 であり、WS2 までのホップ数は 1 であり、WS3 までのホップ数は 2 である。よって、WS3 までのホップ数が 2 であり最大であるので、NHTWS は 2 となる。仮に、図 4(b)のように、WS3 と同じサービスを提供するが、WS5 を利用する WS4 を WS3 の代わりに利用すると、最大のホップ数が CA から WS5 に至るホップ数の 3 となり、NHTWS は 3 となる。

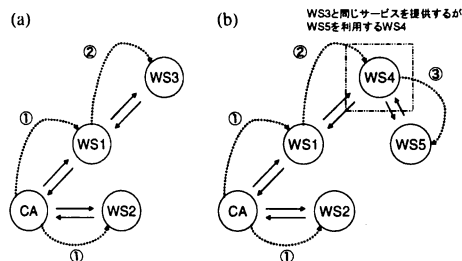


図 4 NHTWS の適用例

### 2.5. Web サービスメトリクスと品質との関係

Web サービスメトリクスと品質において予想される関係について表 1 に示す。RFWS, NOWS, NHTWS についてはメトリクスが大きければそれだけネットワークを通して他の Web サービスを利用している(論理的にもしくは物理的にネットワークを利用する)ということであり、Web サービス利用の際の SOAP メッセージの変換といったオーバーヘッドや、ネットワークを経由することによる予期せぬネットワーク障害の混入等の理由により、信頼性と効率性が評価できると考える。また、NOWS については Web サービスの再利用性といった観点から保守性においても評価できると考える。EMWS については、CA が必要としているサービスにどれだけ Web サービスが特化しているかを示していると言う事ができ、機能性を評価できると考える。

表 1 Web サービスメトリクスと品質の関係

	RFWS	NOWS	EMWS	NHTWS
信頼性	-	-	○	-
効率性	○	○	-	○
保守性	-	○	-	-

### 3. Web サービスメトリクス評価実験

本章では、我々が文献[6]において提案した 4 つのメトリクスと品質特性[2]との関連を評価する実験を行う。一つ目は効率性と Web サービスメトリクスとの関連であり、二つ目は信頼性と Web サービスメトリクスとの関連である。これら品質特性を評価するために WS-PROVE(Web Service Prototyping and Validation Environment)[3]と Sum of Disjoint Products (SDP)[7]という信頼性評価アルゴリズムを用いた。以降、これらについて説明する。

#### 3.1. WS-PROVE による効率性評価

我々は、文献[3]において自由に Web サービスアプリ

ケーションの連携方式を構築できる WS-PROVE を開発した(図 5)。WS-PROVE では、Web サービスアプリケーションの開発において、初期段階で必要とされるプロトタイプを短時間で構築できる。プロトタイプは、実際の WS に類似した仮想のシステム用 WS から構成され、プロトタイプを実際に動作させることにより非機能的な要求(SOAP 変換のオーバーヘッド等)を評価できる。具体的には、WS の連携方式(トポロジ)・個々の WS の処理時間・WS 間での遅延を設定することにより、様々な条件のプロトタイプを構築できる。また、プロトタイプを実際に動作させ、プロトタイプ全体と個々の WS での動作時間(処理の開始から終了までの時間)と実際にかかった WS 間での遅延を計測できる。

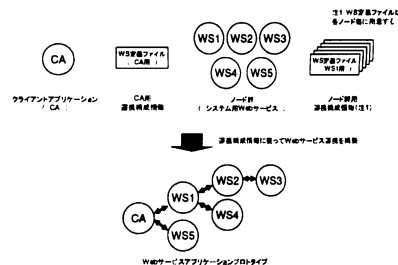


図 5 WS-PROVE

#### 3.1.1. 効率性に関する評価実験方法

効率性に関する評価実験として、本論文では二つの実験を行った。以降、それらについて説明する。ただし、以降の説明において各用語の意味は以下の通りとし、CA はクライアントアプリケーション、WS<sub>i</sub> は Web サービス *i* を示し、ノードは CA もしくは WS<sub>i</sub> を示す。

連携方式: WS の利用の形(トポロジ)

処理時間: ノード単体の処理にかかる固有時間

連携 WS: ノードが利用する次の WS<sub>i</sub>

遅延: ノード間でのネットワーク遅延時間

動作時間: ノードの処理にかかった実際の時間

待ち時間: ノードが連携 WS の処理結果を待つ時間

評価実験 1: 連携方式の違いによる効率性評価

設定項目: 連携方式, 処理時間, 遅延

計測項目: 動作時間, 待ち時間, 実際の遅延

目的: WS 利用数を等しくし遅延を考慮した条件で、連携方式の違いによるプロトタイプとプロトタイプを構成するノードの性能の違いを計測

評価実験 1 では、我々が文献[4]において定義したプロキシ型・リダイレクト型のトポロジを考慮し、図 6 に示すトポロジの(3), (5), (6)の連携方式でプロトタイプを構築し、表 2 に示す処理時間と遅延を WS 定義ファイルに記述し、動作時間と待ち時間と実際の遅延の計測した。実験は 100 回行い、その平均を計測値とした。

**評価実験 2:WS 利用数の違いによる効率性評価**

設定項目:連携方式, 処理時間, 遅延  
計測項目:動作時間, 待ち時間

目的:WS 利用数が異なり遅延を考慮しない条件で, 連携方式の違いによるプロトタイプとプロトタイプを構成するノードの性能の違いを計測

評価実験 2 では, 図 6 に示すトポロジ(1), (2), (3), (4), (6)の連携方式でプロトタイプを構築し, 表 3 に示されるように各プロトタイプでの WS<sub>i</sub> の処理時間の総和が等しくなるように処理時間を設定した. また, ノード間での遅延は考慮しないため, 遅延を 0.00msec と WS 定義ファイルに記述し. 動作時間と待ち時間を計測した. 実験は 100 回行い, その平均を計測値とした.

表 2 表 3 において, WS0 (=CA), WS1, ..., WS6 は 図 6 における各 WS に対応し, 処理時間は対応する WS の処理時間である. WS<sub>i</sub>-WS<sub>k</sub> は *i* 番の WS と *k* 番の WS との間のネットワークにおける遅延である.

表 2 評価実験 1 の設定値

(単位:msec)							
	WS0	WS1	WS2	WS3	WS4	WS5	WS6
処理時間	100	100	100	100	100	100	100
WS0-WS <sub>i</sub>	-	50	100	150	200	250	300
WS1-WS <sub>i</sub>	50	-	50	100	150	200	250
WS2-WS <sub>i</sub>	100	50	-	50	100	150	200
WS3-WS <sub>i</sub>	150	100	50	-	50	100	150
WS4-WS <sub>i</sub>	200	150	100	50	-	50	100
WS5-WS <sub>i</sub>	250	200	150	100	50	-	50
WS6-WS <sub>i</sub>	300	250	200	150	100	50	-

表 3 評価実験 2 の設定値

(単位:msec)							
	WS0	WS1	WS2	WS3	WS4	WS5	WS6
トポロジ(1)	100	900	-	-	-	-	-
トポロジ(2) (4)	100	300	300	300	-	-	-
トポロジ(3) (6)	100	150	150	150	150	150	150

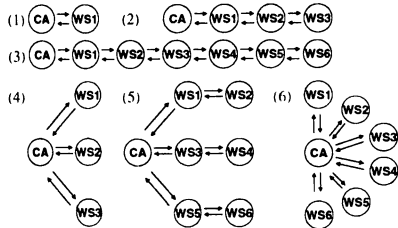


図 6 評価実験用トポロジ

**3.2. SDP アルゴリズムによる信頼性評価**

Sum of Disjoint Products (SDP)[7]とは, ネットワークをグラフで表現することにより, ある特定のノードにおける信頼性を数学的に導出できるアルゴリズムである. 本論文では Web サービスアプリケーションのトポロジを論理的なネットワークとみなしグラフで表現することにより, RFWS, NOWS, NHTWS について SDP による信頼性の評価実験を行う.

**3.2.1. 信頼性に関する評価実験方法**

SDP に対して, トポロジにおいて CA もしくは WS 単体(ノード)が正常に動作する確率(ノード信頼

性:NR)とノード間のネットワーク(リンク)が正常に動作する確率(リンク信頼性:LR)を与えると, トポロジ内の特定のノードが正常に動作する確率(サービス信頼性:SR)を導出できる. SDP を用いて, ノード信頼性やリンク信頼性を様々に変化させてトポロジの信頼性を導出することができる. 本論文では, 図 6 に示される各トポロジを対象に, トポロジを構成するノードのノード信頼性は一律に 1.00 とし, ノード間のリンク信頼性は 0.99 として SDP を適用し, 評価実験を行った.

**4. 評価・考察**

**4.1. WS-PROVE による評価実験結果**

評価実験 1, 2 の実験結果を表 4 表 5 に示す. 表中に各トポロジに対する Web サービスメトリクス (RFWS, NOWS, NHTWS)も計算している. 表 4 表 5 の各行について, System とはトポロジで実現される Web サービスアプリケーション全体のデータを示す. また, Client は CA を表し, CalcWSXX はトポロジにおける各 WS の計算結果を表す.

表 4 評価実験 1 結果

(単位:msec)										
トポロジ	動作時間		連携方式	NOWS	遅延	RFWS	NHTWS	トポロジ		
	System	Client						CalcWS01	CalcWS02	CalcWS03
トポロジ(3)	System	1,188.10	100	3	6	-	-	-	-	-
	Client	100.00	1,181.22	CalcWS01	1,015.12	62.23	2	6	-	-
	CalcWS01	100.00	1,005.65	CalcWS02	835.58	62.16	2	5	-	-
	CalcWS02	100.00	826.05	CalcWS03	655.90	62.21	2	4	-	-
	CalcWS03	100.00	646.48	CalcWS04	476.27	62.23	2	3	-	-
	CalcWS04	100.00	466.50	CalcWS05	296.59	62.21	2	2	-	-
	CalcWS05	100.00	287.07	CalcWS06	117.06	62.26	2	1	-	-
	CalcWS06	100.00	107.56	-	-	0	0	-	-	-
	System	1,440.31	100	5	6	-	-	-	-	-
	Client	100.00	1,433.49	CalcWS01	296.34	62.03	-	-	-	-
トポロジ(5)	CalcWS02	100.00	107.47	CalcWS03	296.48	142.55	6	2	-	-
	CalcWS03	100.00	287.02	CalcWS04	116.75	62.34	2	1	-	-
	CalcWS04	100.00	107.73	-	-	0	0	-	-	-
	CalcWS05	100.00	287.00	CalcWS06	116.83	62.26	2	1	-	-
	CalcWS06	100.00	107.47	-	-	0	0	-	-	-
	System	1,874.83	100	6	6	-	-	-	-	-
	Client	100.00	1,868.20	CalcWS01	117.00	61.70	-	-	-	-
	CalcWS02	100.00	107.55	CalcWS03	116.86	101.75	-	-	-	-
	CalcWS03	100.00	107.62	CalcWS04	116.88	148.73	12	1	-	-
	CalcWS04	100.00	107.51	CalcWS05	117.08	195.56	-	-	-	-
CalcWS05	100.00	107.53	CalcWS06	116.99	304.59	-	-	-	-	
CalcWS06	100.00	107.53	-	-	0	0	-	-	-	

**4.1.1. 評価実験 1 の結果**

表 4 に示すように, System の動作時間と Client, CalcWSXX の動作時間は各トポロジ間で違いが見られた. トポロジ(3)ではトポロジ内の各 WS は順番に次の WS を呼ぶようなプロキシ型の連携をしているが, 表 2 に示すネットワーク遅延の設定値では, プロキシ型で遅延が最小となり, System での動作時間も他のトポロジに比べて最小の 1188 msec となった. しかし, 個々の WS の動作時間に関しては, 他のトポロジに比べて大きくなる結果となった. トポロジ(6)では, トポロジ



では処理時間が 100msec, 動作時間が 107msec になっている。従って, RFWS が大きい WS は処理時間に対して動作時間が大きくなる傾向があると言える。しかし, トポロジ(3)の様なプロキシ型の連携 WS の利用では RFWS での動作時間の評価は適さない結果となった。これは, RFWS では連携 WS の処理を待つ時間の数を評価すると考えられるが, それら連携 WS 単体の処理待ち時間の量の評価には適さないためと考えられる。

NHTWS については, CA もしくは WS が利用する連携 WS の処理を待つ時間(自身の処理時間以外に時間を割かれるか)の量に関して, 関連がみられる結果となった。トポロジ(3)の CalcWSXX では, NHTWS が 1 の CalcWS05 は処理待ち時間が 117msec であり, NHTWS が 2 の CalcWS04 では処理待ち時間が 296msec となっている。以降 NHTWS が大きくなるにつれ処理待ち時間は増加し, NHTWS が 5 の CalcWS01 では処理待ち時間が 835msec となっている。従って, NHTWS が大きい WS は処理時間に対して処理待ち時間が大きくなり動作時間も大きくなる傾向があると言える。しかし, RFWS とは逆にトポロジ(5)のようなリダイレクト型の連携 WS の利用では NHTWS での動作時間の評価は適さない結果となった。これは, NHTWS では WS が利用する連携 WS 単体の処理待ち時間の量を評価すると考えられるが, 連携 WS の処理を待つ時間の数の評価には適さないためと考えられる。

#### 4.3.2. 信頼性に関する考察

NOWS については, 表 6 に示されるように NOWS が 1, 3, 6 となる各トポロジにおいて CA の信頼性が 0.9801, 0.9415, 0.8864 となっており, NOWS と信頼性に関連がみられる結果となった。これは, NOWS では WS の利用数を計測するが, WS の利用数が多くなれば信頼性が下がる要因となるネットワークを通したデータのやり取りが多くなるため, NOWS が大きくなれば信頼性が下がるという結果になったと考えられる。

RFWS と NHTWS については, RFWS かつ NHTWS の値が大きくなれば信頼性が下がる傾向が見られた。しかし, 表 6 の T3 の CA, WS1~WS5 の RFWS は 2 であるが信頼性がそれぞれ異なっていることや, NHTWS が 2 である T2 の WS1 および T3 の WS4 と T5 の CA において信頼性が異なっていることから, RFWS 単体や NHTWS 単体では, 信頼性を評価できない場合がある結果となった。これは, RFWS 単体で信頼性を評価しようとしても, 連携 WS の信頼性がトポロジによって変化するためだと考えられる。逆に, NHTWS 単体では, トポロジによって変化する連携 WS の信頼性を評価できるが, 利用する連携 WS の数を評価できないため, 信頼性を評価できないと考えられる。

#### 4.3.3. まとめ

本論文の評価実験で明らかになった, NOWS, RFWS, NHTWS と効率性・信頼性との間の関連について表 7 にまとめる。3 つのメトリクスは効率性・信頼性について表 7 に示す項目において効率性・信頼性に関連がある結果となった。効率性においては, NOWS は Web サービスアプリケーション全体でのオーバーヘッドを, RFWS は連携 WS の処理を待つ数を, NHTWS では他の WS を待つ時間の量を評価できると思われる。信頼性においては, NOWS は Web サービスアプリケーション全体でのネットワークの利用数を, RFWS と NHTWS は両方のメトリクスを同時に用いることにより対象のサービス信頼性を評価できると思われる。

表 7 評価実験まとめ

	評価できる項目	効率性と信頼性との関係
NOWS	全体のオーバーヘッド	値が大きければ効率性と信頼性が悪くなる
RFWS	連携WSを待つ数	値が大きければ効率性と信頼性が悪くなる
NHTWS	連携WSを待つ時間の量	値が大きければ効率性と信頼性が悪くなる

#### 5. 終わりに

本論文では 3 つの Web サービスメトリクス(RFWS, NOWS, NHTWS)について, WS-PROVE と SDP を用いて, Web サービスメトリクスと効率性及び信頼性との関係を実験的に評価した。実験の結果, NOWS は Web サービスアプリケーション全体のオーバーヘッドと信頼性に関連があることがわかった。RFWS と NHTWS は連携 WS の処理待ち時間と, 連携 WS の信頼性と関連があることがわかった。今後の課題としては, EMWS に関する機能性の評価実験と, プロトタイプによる評価ではなく実際の Web サービスアプリケーションによる効率性の評価をする必要があると思われる。

#### 文 献

- [1] 青山幹雄, “Web サービス技術と Web サービスネットワーク”, 信学技報, IN2002-163, pp.47-52, Jan. 2003.
- [2] 東基衛, ソフトウェア品質評価ガイドブック, 東基衛他編, 日本規格協会, 1994.
- [3] 石井健一, “Web サービスアプリケーションのプロトタイプングおよび性能評価のためのシステム開発,” 信学技報, IN, March 2005. (to appear)
- [4] 石井健一, “異なる設計・実装法を用いた Web サービスアプリケーションの開発および比較評価,” 信学技報, NS2003-315, pp.107-112, March 2004.
- [5] 石井健一, “複数 Web サービス連携手法の実験的評価,” 信学技報, NS2004-99, pp.75-80, September 2004.
- [6] 串戸洋平, “Web サービスアプリケーションのソフトウェアメトリクスに関する考察,” 信学技報, NS2003-316, pp.113-118, March 2004.
- [7] Tatsuhiro Tsuchiya, Tomoya Kajikawa, and Tohru Kikuno, “Parallelizing SDP (Sum of Disjoint Products) Algorithms for Fast Reliability Analysis,” IEICE Transactions on Information and Systems, Vol.E83-D, No.5, pp.1183-1186, May 2000.