

平成12年度 地域新生コンソーシアム研究開発事業
「即効型地域新生コンソーシアム研究開発」
「ITサービス事業共有基盤向けミドルウェアの開発」

成 果 報 告 書

平成14年3月

新エネルギー・産業技術総合開発機構
管理法人 財団法人札幌エレクトロニクスセンター

目次

第 1 章 概要

1.	目的と目標	2
2.	研究開発成果概要	6
2.1.	SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定	6
2.2.	各種プラットフォーム向け SOAP 実装	6
2.3.	総合セキュリティ機能の実装	6
2.4.	他の分散オブジェクト技術とのブリッジングインターフェースの実装	7
2.5.	メッセージ管理機能の実装	8
2.6.	エージェント型ソフトウェアのモビリティに関する仕様策定と実装	8
2.7.	組み込みシステム向け SOAP インターフェースの開発	9
2.8.	ビジネスプロセスソリューションへの適用による評価	9
2.9.	小規模 Web サービスの RAD による評価	10
2.10.	大型計算機サービスへの適用による評価	10
3.	事業化に向けた調査等の概要	12
3.1.	調査方法	12
3.2.	調査結果の概要	12
4.	今後の課題	14
4.1.	SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定	14
4.2.	各種プラットフォーム向け SOAP	14
4.3.	総合セキュリティ機能の実装	15
4.4.	他の分散オブジェクト技術とのブリッジングインターフェースの実装	15
4.5.	メッセージ管理機能の実装	15
4.6.	エージェント型ソフトウェアのモビリティに関する仕様策定と実装	16
4.7.	組み込みシステム向け SOAP インターフェースの開発	17
4.8.	ビジネスプロセスソリューションへの適用による評価	17
4.9.	小規模 Web サービスの RAD による評価	18
4.10.	大型計算機サービスへの適用による評価	18

5.	研究開発スケジュールと主な設備の内容	20
5.1.	研究開発スケジュール表	20
5.2.	主な購入設備（500万円以上）名称と使用目的、設備場所一覧	21
6.	研究開発体制図	22
7.	研究開発者一覧	24
8.	工業所有権等の取得状況	27
9.	外部発表等の状況	28
10.	英文要約	30

第2章 本論

1.	SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定	38
1.1.	概要	38
1.2.	SOAP 仕様および関連仕様に関する調査	39
1.3.	他の SOAP 実装	40
1.4.	開発目標	41
1.4.1.	W3C SOAP 仕様に準拠	42
1.4.2.	複数OS、複数言語をサポート	42
1.4.3.	セキュリティの確保	42
1.4.4.	非同期処理(メッセージング)機能	42
1.4.5.	トランザクション機能	43
1.4.6.	メッセージルーティング	43
1.4.7.	多国語対応	43
1.4.8.	拡張性・プラグインモジュール化	44
1.4.9.	既存IT資産の有効活用	44
1.4.10.	デバッグ・トレース機能のサポート	45
1.4.11.	高速化	45
1.4.12.	開発ツール(API)、ドキュメント、サンプルコードを含む、パッケージ・ソースコードの無償配布	45
1.5.	OpenSOAP アーキテクチャ	45
1.5.1.	メッセージ交換フロー	47
1.5.2.	OpenSOAP ヘッダ	49
1.6.	開発体制と開発計画	50
1.6.1.	開発体制	50
1.6.2.	開発計画	50
1.7.	使用許諾ライセンス	53
1.8.	OpenSOAP における暫定的制約事項	54
1.9.	今後の課題	55

2.	各種プラットフォーム向け SOAP 実装	56
2.1.	はじめに	56
2.2.	C 言語用 API の実装	58
2.2.1.	C 言語用 API 概要	58
2.2.2.	OpenSOAP Message 処理用 API の実装	80
2.2.3.	OpenSOAP Client 構築用 API の実装	80
2.2.4.	OpenSOAP Service 構築用 API の実装	81
2.3.	マルチプラットフォーム対応	81
2.4.	国際化対応	82
2.4.1.	SOAP 仕様における国際化	82
2.4.2.	OpenSOAP API における国際化対応	82
2.5.	他のプログラミング言語用実装	83
2.5.1.	Java 言語用実装	83
2.6.	セキュリティモジュールとの結合	83
2.7.	サンプルコードの作成	83
2.7.1.	OpenSOAP API の処理	83
2.7.2.	共通の処理	83
2.7.3.	クライアントプログラムの作成	84
2.7.4.	サービスプログラムの作成	89
3.	総合セキュリティ機能の実装	94
3.1.	概要	94
3.2.	インターネット上の電子商取引での取引リスク	94
3.3.	基本方式の選定	95
3.4.	取引リスクへの排除	96
3.5.	セキュリティ技術の実装方式	98
3.5.1.	鍵ペアの作成	99
3.5.2.	暗号化処理	99
3.5.3.	復号化処理	100
3.5.4.	電子署名付加処理	100

3.5.5.	電子署名検証処理	101
3.5.6.	電子署名付加／検証に関する留意事項	102
3.6.	認証局の提供	102
3.6.1.	認証局の必要性と概要	102
3.6.2.	認証局の機能	103
3.6.3.	認証局関連API（認証局DBアクセス用）	104
3.6.4.	認証局関連API（電子証明書アクセス用）	105
3.6.5.	認証局サービス機能構築支援	106
3.6.6.	認証局利用側で検討すべき事項	106
3.6.7.	認証局の課題	107
4.	他の分散オブジェクト技術とのブリッジングインターフェースの実装	108
4.1.	目的と目標	108
4.2.	分散オブジェクト技術	108
4.2.1.	分散オブジェクト技術の発展	108
4.2.2.	DCOM(Distributed Component Object Model)	109
4.2.3.	CORBA	111
4.3.	OpenSOAPブリッジングインターフェース	112
4.4.	開発成果	115
4.4.1.	DCOMブリッジング開発成果	115
4.4.2.	CORBAブリッジング開発成果	122
4.5.	開発スケジュール	127
5.	メッセージ管理技術の実装	129
5.1.	はじめに	129
5.2.	OpenSOAPサーバの主要機能	130
5.2.1.	サービスプログラム仲介・連携機能	130
5.2.2.	非同期処理とメッセージ管理機能	131
5.2.3.	サーバ間転送機能	134
5.2.4.	エラー処理とサーバ署名	136
5.3.	サーバオプション機能 —Webトランザクション管理—	138

5.3.1.	トランザクション管理機能概要	138
5.3.2.	OpenSOAP トランザクションクライアント (リクエストメッセージ) 仕様	138
5.3.3.	OpenSOAP トランザクション対応サービス仕様	140
5.3.4.	クライアントへの返信	143
5.4.	OpenSOAP サーバのアーキテクチャとメッセージ処理	145
5.4.1.	OpenSOAP サーバを構成するプロセス群	145
5.4.2.	サーバ内処理手順 (プロセス間関係図)	149
5.5.	OpenSOAP サーバ運用マニュアル	151
5.5.1.	OpenSOAP サーバ制御コマンド	151
5.5.2.	OpenSOAP サーバ設定ファイル	152
5.5.3.	OpenSOAP サービス設定ファイル SSML	156
5.6.	まとめ	158
6.	エージェント型ソフトウェアのモビリティに関する仕様策定と実装	160
6.1.	はじめに	160
6.2.	ソフトウェアアプリケーションのモビリティに関する現状	161
6.2.1.	モバイル向け SOAP 実装の現状	161
6.2.2.	モバイルプラットフォームの現状調査	162
6.2.3.	最新動向	168
6.3.	OpenSOAP MP クライアントの仕様策定と実装	169
6.4.	OpenSOAP MP エージェント型アプリケーション	171
6.5.	まとめ	172
6.6.	用語集	173
7.	組み込みシステム向け SOAP インターフェースの開発	175
7.1.	序論	175
7.2.	オープンなロボットシステムの構築	176
7.2.1.	単機能モジュールロボットシステム	176
7.2.2.	作業実行過程	177
7.2.3.	SOAP によるロボットシステムのオープン化	178
7.3.	ロボットシステムへの実装	181

7.3.1.	通信機能の実装	181
7.3.2.	物体搬送実験	181
7.4.	結論	189
8.	ビジネスプロセスソリューションへの適用による評価	190
8.1.	はじめに	190
8.2.	OpenSOAP 適用と評価	191
8.2.1.	Calc サービスプロトタイプ	191
8.2.2.	CalcAsync サービスプロトタイプ	192
8.2.3.	Shopping サービスプロトタイプ	192
8.2.4.	ShoppingSec サービスプロトタイプ	193
8.2.5.	Transaction サービスプロトタイプ	194
8.2.6.	検証・評価	194
8.3.	Web サービス運用面からの評価	197
8.3.1.	Web サービスの利用上の問題点	197
8.3.2.	Web サービス開発プロセス	199
8.4.	おわりに	201
9.	小規模 Web サービスの RAD による評価	202
9.1.	序言	202
9.2.	プロジェクト内での位置付け	202
9.2.1.	RAD 検証内容	203
9.2.2.	環境構築コスト	203
9.3.	システム概要	203
9.4.	ゲーム内容	204
9.4.1.	概要	205
9.4.2.	カード	205
9.4.3.	モンスターカード	206
9.4.4.	リソースカード	207
9.4.5.	エネルギーカード	207
9.4.6.	畏カード・バトルカード・スキルカード	207

9.4.7.	対戦計算	208
9.5.	構成環境	208
9.6.	接続メッセージの流れ	209
9.7.	検証結果	211
9.8.	ビジネスへの取り組み	211
9.9.	結言	212
10.	大型コンピューターシステムへの適用による評価	213
10.1.	はじめに	213
10.2.	OpenSOAP に基づく大型コンピューティングサービス	214
10.2.1.	ライブラリサービス	214
10.2.2.	プログラムサービス	218
10.3.	議論	230
10.4.	おわりに	231

図表目次

第 1 章 概要

図 1.1	SOAP によるメッセージ交換	3
図 1.2	OpenSOAP プロジェクトの開発体制	5

第 2 章 本論

図 1.4.1	OpenSOAP における SOAP の実装	41
図 1.4.2	OpenSOAP メッセージ転送概念図	44
図 1.5.1	OpenSOAP アーキテクチャ	46
図 1.5.2	OpenSOAP 階層モデル	49
図 1.5.3	OpenSOAP メッセージヘッダ(例)	50
図 1.6.1	開発スケジュール	51
図 1.6.2	OpenSOAP コンポーネントとサブテーマの対応	51
図 1.6.3	開発バージョンと実装機能	52
図 1.7.1	OpenSOAP の使用許諾ライセンス	53
図 1.7.2	各ソースコードのヘッダ部分 (例)	54
図 2.1.1	OpenSOAP API 構成	57
図 2.1.2	OpenSOAP API 言語毎の対応	57
図 2.2.1	OpenSOAP API の処理	59
図 2.2.2	SOAP Message の構成	61
図 2.2.3	OpenSOAP Message Sample (Add Request Message)	61
図 2.7.1	OpenSOAP API 処理の共通処理	84
図 2.7.2	OpenSOAP Request Message (Hello Sample)	85
図 2.7.3	Client Program のリクエスト作成	85
図 2.7.4	Client Program の SOAP Service 呼出	87
図 2.7.5	OpenSOAP Response Message(Hello Sample)	88
図 2.7.6	Client Program のレスポンス解析	88
図 2.7.7	Service Program のサービス登録及び実行	89
図 2.7.8	Service Program サービス関数内のリクエスト解析	91

図 2.7.9	Service Program サービス関数内のレスポンス作成	92
図 3.3.1	PKI による暗号化／復号化の基本方式	95
図 3.4.1	取引リスク排除の基本方式	96
図 3.5.1	秘密鍵の例	99
図 3.5.2	公開鍵の例	99
図 3.5.3	暗号化指示する部位の定義	100
図 3.5.4	暗号化結果	100
図 3.5.5	SOAP メッセージに付加される電子署名	101
図 3.6.1	認証局の概要	103
図 4.3.1	OpenSOAP 構成図	113
図 4.3.2	DCOM と CORBA の構成	113
図 4.3.3	OpenSOAP ブリッジングインターフェース構成	114
図 4.4.1	DCOM サーバの登録確認	116
図 4.4.2	OpenSOAP Request Message OF DCOM In Getds	117
図 4.4.3	OpenSOAP Response Message OF DCOM In Getds	117
図 4.4.4	DCOM Getds 実行画面	118
図 4.4.5	DCOM サーバの動作確認画面	119
図 4.4.6	OpenSOAP Request Message OF DCOM In Add	120
図 4.4.7	OpenSOAP Response Message OF DCOM In Add	121
図 4.4.8	DCOM Add の実行画面	121
図 4.4.9	DCOM Add サーバの動作確認画面	122
図 4.4.10	OpenSOAP Request Message OF CORBA In Add	125
図 4.4.11	OpenSOAP Response Message OF CORBA In Add	125
図 4.4.12	CORBA ネーミングサービス起動画面	126
図 4.4.13	CORBA 簡易演算サーバの起動画面	126
図 4.4.14	CORBA Add ブリッジングの実行画面	127
図 4.5.1	開発スケジュール及び進捗	128
図 5.2.1	OpenSOAP サーバ概念図	130
図 5.2.2	OpenSOAP サーバのメッセージ管理技術	132

図 5.2.3	非同期処理要求のための SOAP ヘッダ拡張	133
図 5.2.4	非同期処理のためのレスポンスメッセージ (Message ID)	133
図 5.2.5	OpenSOAP サーバ間転送による Firewall 越しルーティング	134
図 5.2.6	非同期転送処理の詳細図	135
図 5.2.7	サーバ間転送処理 SOAP ヘッダ拡張	136
図 5.2.8	サーバの生成するエラーメッセージ(1)	137
図 5.2.9	サーバの生成するエラーメッセージ(2)	137
図 5.3.1	OpenSOAP Transaction Request Envelope	139
図 5.3.2	A 銀行へ送信される SOAP Envelope	141
図 5.3.3	B 銀行へ送信される SOAP Envelope	141
図 5.3.4	A 銀行からのレスポンスの例	142
図 5.3.5	トランザクションサービスから A 銀行への COMMIT 要求	142
図 5.3.6	A 銀行から COMMIT 要求に対する返信	143
図 5.3.7	クライアントに返信されるトランザクション処理結果	144
図 5.4.1	OpenSOAP サーバ・内部プロセス関係図	148
図 5.4.2	条件別処理フロー (同期ローカル処理)	149
図 5.4.3	条件別処理フロー (非同期ローカル処理)	150
図 5.4.4	条件別処理フロー (同期転送処理)	150
図 5.4.5	条件別処理フロー (非同期転送処理)	151
図 5.5.1	設定ファイル forwarder.conf の例	152
図 5.5.2	OpenSOAP サーバを利用した転送処理のための SOAP ヘッダ拡張と 処理の流れ	153
図 5.5.3	設定ファイル backwarder.conf の例	154
図 5.5.4	非同期転送処理に伴う拡張ヘッダ追加 (リクエスト時)	154
図 5.5.5	非同期転送処理に伴う拡張ヘッダ参照 (レスポンス時)	155
図 5.5.6	拡張 SOAP ヘッダの詳細例	156
図 5.5.7	Calc.ssml	157
図 6.2.1	I モード対応 Java 搭載端末アーキテクチャ	163
図 6.2.2	Java 搭載端末アーキテクチャ	164

図 6.2.3	J-Phone Java 搭載端末アーキテクチャ	166
図 6.2.4	第二世代 Java アプリケーションへの移行状況	168
図 6.3.1	OpenSOAP MP 仕様概要	169
図 6.3.2	OpenSOAP MP サンプルアプリケーション	170
図 6.4.1	OpenSOAP MP エージェント型ネットワーク	171
図 6.4.2	OpenSOAP MP エージェント型アプリケーションのためのメッセージ ヘッダ拡張仕様	172
図 6.5.1	本サブテーマの研究スケジュール	173
図 7.1.1	単機能モジュール型群ロボットアーキテクチャ	175
図 7.2.1	カメラモジュール	176
図 7.2.2	車輪モジュール	177
図 7.2.3	作業実行プロセス	178
図 7.2.4	適切な運動パタンの例（実装した実験システムにおいて、カメラ モジュールが車輪モジュールの直上約 2.7m の高さに存在する場合）	178
図 7.2.5	モジュールアーキテクチャ	179
図 7.2.6	UDDI の適用	180
図 7.3.1	OpenSOAP を用いた運動データのダウンロード	181
図 7.3.2	搬送実験環境	183
図 7.3.3	構築したインターフェース系	183
図 7.3.4	カメラ画像	184
図 7.3.5	進入禁止領域の表示	184
図 7.3.6	指令方式	185
図 7.3.7	評価結果	185
図 7.3.8	操作過程(1/6)	186
図 7.3.9	操作過程(2/6)	186
図 7.3.10	操作過程(3/6)	187
図 7.3.11	操作過程(4/6)	187
図 7.3.12	操作過程(5/6)	188
図 7.3.13	操作過程(6/6)	188

図 8.1.1	B2B 社内統合 Web サービスモデル	190
図 8.3.1	UDDI	198
図 8.3.2	Web サービス開発プロセス	199
図 8.3.3	業務統合 Web サービスにおける開発プロセス	200
図 9.4.1	ゲーム画面一例	205
図 9.4.2	ゲームの流れ	206
図 9.5.1	ネットワーク構成図	209
図 9.6.1	対戦データ送出フェーズ	210
図 9.6.2	対戦計算フェーズ	210
図 9.6.3	対戦結果受信フェーズ	211
図 10.2.1	OpenSOAP に基づく大型コンピューティングサービスの構成	214
図 10.2.2	乱数サービスにおけるメッセージ例	215
図 10.2.3	乱数サービス実行例 (Linux)	216
図 10.2.4	乱数サービス実行例 (Windows)	216
図 10.2.5	非同期通信に基づく乱数サービスにおけるメッセージ例	217
図 10.2.6	非同期版乱数サービス実行例 (リクエストメッセージ送信)	218
図 10.2.7	非同期乱数サービス実行例 (メッセージ ID の送信)	219
図 10.2.8	ノード使用状況確認サービスにおけるメッセージ例	221
図 10.2.9	ノード使用状況確認サービス実行例	221
図 10.2.10	課金情報確認サービスにおけるメッセージ例	222
図 10.2.11	課金情報確認サービス実行例	222
図 10.2.12	プロセス情報確認サービスにおけるメッセージ例	223
図 10.2.13	プロセス情報確認サービス実行例	223
図 10.2.14	プログラムサービスにおけるメッセージ例 (ログイン失敗)	224
図 10.2.15	プログラムサービス実行例 (ログイン失敗)	224
図 10.2.16	プログラムサービス実行例	225
図 10.2.17	プログラムサービスにおけるメッセージ例	225
図 10.2.18	指定プログラムの実行結果及び Email によるプログラム実行終了通知	226
図 10.2.19	ノード使用状況確認サービス実行例 (Windows)	226

図 10.2.20	課金情報確認サービス実行例 (Windows)	227
図 10.2.21	プロセス情報確認サービス実行例 (Windows)	227
図 10.2.22	プログラムサービス実行例 (Windows)	228
図 10.2.23	ノード使用状況確認サービス実行例 (非同期)	228
図 10.2.24	プロセス情報確認サービス実行例 (非同期)	229
図 10.2.25	課金情報確認サービス実行例 (非同期)	229
図 10.2.26	プログラムサービス実行例 (非同期)	230
図 10.3.1	統合的大型コンピューティングサービス	231

第 1 章 概 要

1. 目的と目標

プラットフォームに依存した作り込み型のアプリケーション開発から、必要に応じてネットワーク上のアプリケーションリソースを利用するサービス指向型への転換が提唱されて久しいが、様々な理由から普及には至っていないのが現状である。その原因の一つとして、汎用的なアプリケーションの呼び出し手段が確立していなかったことが挙げられる。

そのような中で、最近提案されたデータ交換のための規格である SOAP は、急速に普及を果したインターネットに関連する技術を前提としていることや、そのシンプルさなどから、サービス指向型アプリケーションの普及の鍵として大きな期待が寄せられている技術である。

OpenSOAP プロジェクトは、この SOAP 規格をベースとし、ビジネス Web サービスの開発をサポートする様々な機能を付加したミドルウェアを開発・提供することを目的とするものである。

以降では、Web サービスの成り立ちと SOAP の位置付け、そして OpenSOAP プロジェクトの概要について述べる。

Web ブラウザを GUI として用い、HTML でデータ交換を行う Web アプリケーションは、インターネット経由の利用、マルチプラットフォームなどの利点を有するが、HTML 自体が視覚化を目的とする言語であるため、アプリケーションの結果を、人手を介さずに他のアプリケーションから利用するような目的には適さない。

それに対する解のひとつが、任意のデータ構造の記述が可能な XML の利用であり、HTTP を用いて XML 形式でデータ交換を行うような Web アプリケーションは、特に Web サービスと呼ばれるようになって来ている。

このような Web サービスにおけるデータ交換を意図して提案されている規格が、SOAP である (図 1.1)。

SOAP は応用性の高い規格であるが、RPC 向けの基本的な部分が規定されているのみであるため、ビジネス向けの Web サービスを構築する際には、セキュリティやメッセージングなど、SOAP 自体には規定されていない部分を補完する必要がある。

以上のような現状を踏まえ、OpenSOAP プロジェクトでは、サービス指向型のビジネスアプリケーション構築のためのプラットフォームを提供する目的で、次のような目標を掲げてミドルウェア「OpenSOAP」の開発を行って来た。

- ・ ビジネス Web サービスの構築に必要な機能の拡張

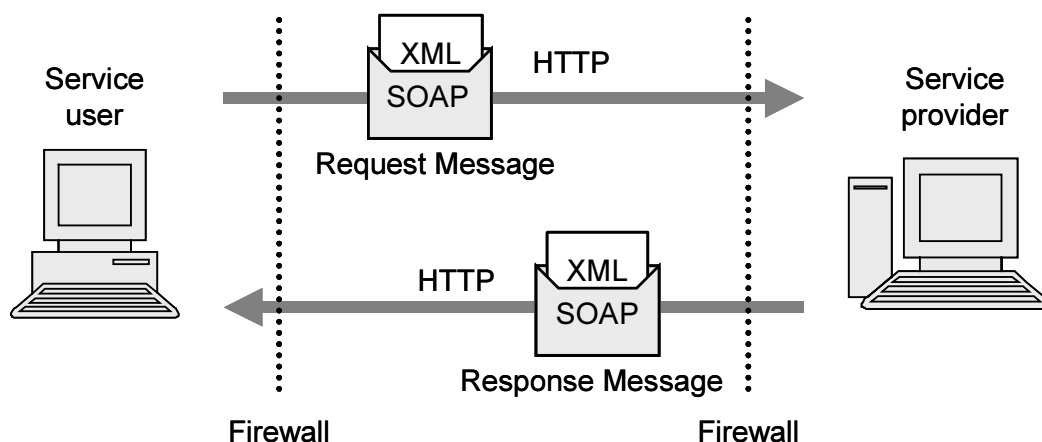


図 1.1 SOAP によるメッセージ交換

- ・ 既存のソフトウェアリソースの再利用性
- ・ マルチプラットフォーム化
- ・ オープンソース・フリーウェアとしての提供

OpenSOAP ミドルウェアのコア部分は、OpenSOAP API、OpenSOAP サーバ、OpenSOAP セキュリティの要素に分割され、同時並行的に開発が進められている (図 1.2)。

OpenSOAP API は、サービスやクライアントを開発する際に用いる関数ライブラリとして実装され、メッセージの送受信や XML 処理などを簡便に行うことを可能とするものである。OpenSOAP サーバは、OpenSOAP における機能拡張の核となる部分であり、各種メッセージ管理機能を実現する。OpenSOAP セキュリティは、OpenSOAP API や OpenSOAP サーバに組み込まれ、標準機能として認証や暗号化を容易に利用することを可能とする。

OpenSOAP プロジェクトでは、開発するミドルウェアのユーザビリティと応用性を評価・検証し、開発にフィードバックする目的で、次のようなパイロットアプリケーションの検討・開発を行っている (図 1.2)。

- (1) モバイルアプリケーション…携帯電話、PDA 向け API、エージェント型アプリケーション
- (2) ビジネスアプリケーション…ビジネスプロセスソリューションへの適用
- (3) 小規模アプリケーション…アミューズメント系インターネットアプリケーションへの適用
- (4) ロボットシステム…分散配置型モジュールロボット系におけるデータ共有
- (5) 大型計算機利用アプリケーション…ライブラリ利用サービス、プログラム処理サービス。

これらのアプリケーション開発は、ミドルウェアの開発と並行して仕様検討・設計を開始しており、その過程で発生した要求は、随時ミドルウェアの開発にフィードバックされている。

以下に、上で述べた OpenSOAP ミドルウェアの構成要素およびアプリケーションに関して、OpenSOAP プロジェクトで設定したサブテーマ毎に具体的な目標を示す。

(1) SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定

本プロジェクトで目標とする、プラットフォーム非依存な弱連結性を有するミドルウェアに求められる各種要件を満たすよう仕様を策定し、ミドルウェア全体のアーキテクチャを設計する。

(2) 各種プラットフォーム向け SOAP 実装

開発ミドルウェアでサポートの対象とする、Windows 系および Unix 系の各プラットフォームで動作する SOAP 基本コンポーネントを実装する。ここで、SOAP 基本コンポーネントとは、SOAP サーバ、SOAP クライアントおよび標準の SOAP メッセージ処理などを行うツール群から構成される。この中には、ネイティブトランスポートプロトコルである HTTP に対応するリクエスト・レスポンス処理の実装も含まれる。クライアント側は、Java、C、C++の各言語に対応した実装を行う。

(3) 総合セキュリティ機能の実装

SOAP メッセージをインターネット上で送受信する際に、セキュリティを確保するための機能を実装する。具体的には、SSL のようなトランスポートプロトコルに応じたセキュリティ技術と、SOAP セキュリティ拡張による認証技術とを組み合わせることにより、EC アプリケーションに求められる各種セキュリティ要件を満たすものとする。

(4) 他の分散オブジェクト技術とのブリッジングインターフェースの実装

SOAP と他の分散オブジェクト技術を用いて開発された既存の分散アプリケーションリソースを連結するためのブリッジングインターフェースを実装する。具体的には、SOAP と CORBA および DCOM との間でメッセージを効率的にトランスレートする機能を実装する。

(5) メッセージ管理機能の実装

EC アプリケーションの構築に不可欠な RDB などのデータベースシステムに、SOAP 経由でアクセスする際に必要となるメッセージ管理機能を実装する。具体的には、トランザクション処理、ログ管理などを実現するための機能を、SOAP ヘッダによるメッセージ拡張を用いて実装する。

(6) エージェント型ソフトウェアのモビリティに関する仕様策定と実装

モバイルエージェントを分散アプリケーションに適用する上で満たすべき要件およびミドルウェアに求められる仕様の策定を行い、それに基づいて、モバイルエージェントをサポートするための SOAP 仕様の拡張および機能の実装を行う。

(7) 組み込みシステム向け SOAP インターフェースの開発

SOAP と家電製品などのハードウェアとのインターフェースに関する研究開発を行う。具体的には、組み込みシステムを SOAP に対応させるための SOAP メッセージの拡張や XML スキーマ定義を行い、それをサポートするための機能を実装する。

(8) ビジネスプロセスソリューションへの適用による評価

本ミドルウェアが実際の業務アプリケーションに適用可能であることを検証するために、大規模かつ複雑なワークフローを有するビジネスプロセスの効率化を目的としたアプリケーション開発に本ミドルウェアを適用し、その開発プロセスやパフォーマンス、スケーラビリティなどについて評価を行う。

(9) 小規模 Web サービスの RAD による評価

本ミドルウェアが、小規模 IT ベンチャー企業の IT サービス事業への参入を容易にするようなものであるかを検証するために、小規模な Web アプリケーションの RAD(Rapid Application Development)を行い、ユーザビリティなどについて評価を行う。

(10) 大型計算機サービスへの適用による評価

本ミドルウェアが、大型計算機センターが提供する大型計算サービスに適用可能であることを検証するために、スーパーコンピュータをベースとした Web サービスを開発し、その利便性や性能等について評価する。

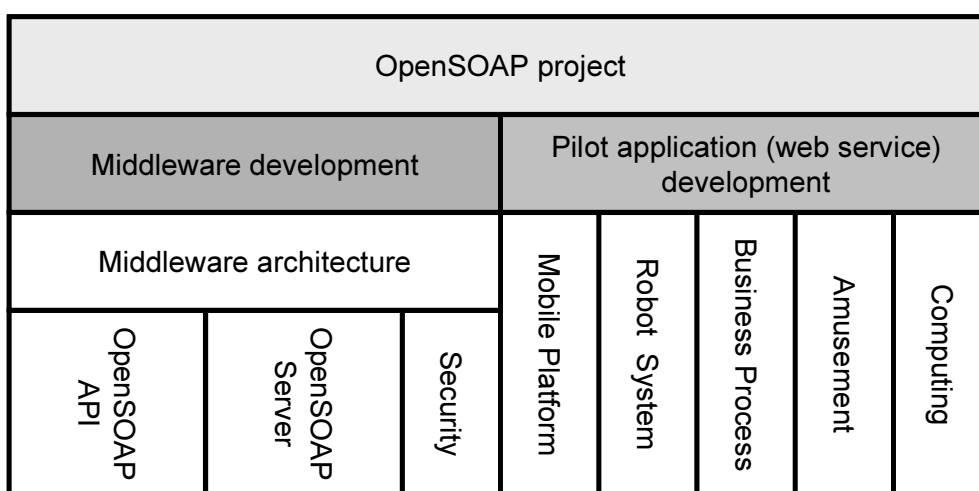


図 1.2 OpenSOAP プロジェクトの開発体制

2. 研究開発成果概要

2.1. SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定

本研究では、ビジネス Web サービスを容易に実現するために、本プロジェクトにおいて開発するミドルウェア(OpenSOAP パッケージ)のアーキテクチャ設計を目的とする。

より具体的には、W3C SOAP 仕様、XML 仕様及びその他の関連仕様の調査、既存の関連ミドルウェア製品等に関する調査、および市場の動向調査等を通じ、本プロジェクトで開発するミドルウェアの全体構造(アーキテクチャ)、ミドルウェアで実現されるべき機能の設計などの概要設計、OpenSOAP コア部分の実装を担当する各サブテーマグループにおいて開発されるライブラリやプログラムなどの外部設計を行った。また、各サブテーマグループ開発担当部分の詳細仕様のすり合わせや、プロジェクト全体の開発スケジュール調整、開発ミドルウェアパッケージの一般への公開に必要な作業も担当した。

2.2. 各種プラットフォーム向け SOAP 実装

本研究では、Windows 系および Linux をはじめとした Unix 系の各プラットフォームで動作する SOAP 実装を行うためのミドルウェアをライブラリ関数群として設計および実装を行った。このライブラリ関数群を OpenSOAP API と呼び、これを C 言語により実装を行った。OpenSOAP API は機能別に分類すると、(1) SOAP Message 処理用 (2) SOAP Client 通信用 (3) SOAP Service 通信用 の三つに分類される。特に、(1) に関しては国際化/多言語化を考慮した実装を行った。実装された OpenSOAP API を利用したサンプルコードの作成を行い、Windows 系では、Windows 2000 において、また、Unix 系プラットフォームでは、Linux をはじめ、NetBSD、Solaris 8 での動作を確認した。C 言語での実装を行うことにより、C++ を始めとした他のプログラミング言語での利用が可能となった。また、Java 言語においては、OpenSOAP クライアントおよびサービスを、ネイティブコードにより、実装可能とした。また、Java 言語に関しては、OpenSOAP 拡張の一つである非同期通信をクライアントにおいて実装した。さらに、OpenSOAP での拡張機能である、セキュリティモジュールとの統合を行った。

2.3. 総合セキュリティ機能の実装

昨今のIT関連インフラの普及に伴い、企業間および企業と消費者間の電子商取引（B2B、B2C）の市場が加速度的に拡大している。こうした実情を踏まえ、OpenSOAPにおいても電子商取引分野へ適用できるための対応が必要となっている。一方、電子商取引はインターネットを介して行われるため、様々な不正が入り込む余地がある。したがって、こうした不正を排除できる環境を提供することが、電子商取引分野への安全な適用のための必須条件となる。

こうした問題を排除するためのセキュリティ技術の実装が電子商取引には必須の条件であるということができるが、W3Cで標準化が進められているSOAP仕様では、セキュリティに関する規定が存在しない。

そこで、今回のOpenSOAP開発において新たにセキュリティ技術に関する実装を行い、OpenSOAPをB2B、B2Cに安全に適用可能とする様な環境を整備した。

今回実装した項目は、暗号化／復号化、電子署名／署名認証および認証局関連機能である。

2.4. 他の分散オブジェクト技術とのブリッジングインターフェースの実装

OpenSOAPは、XMLでフォーマットされたメッセージを主にHTTPを利用して伝える事であり、XMLベースとすることですべてのサービスは容易に利用できることを目的にしている。その結果、SOAP仕様により、ソフトウェアを連携させるための手段が提供され、レガシーコンポーネントを利用しつつ、複数言語、複数企業にまたがる開発が可能になると考えられる。

レガシー分散コンポーネントアーキテクチャには、DCOMとCORBAがある。これらの資産を継承し、OpenSOAPに移行するためには、DCOMとCORBAと接続するブリッジングインターフェースが必要である。DCOMとCORBAとのブリッジングインターフェースの開発を行うことで、これまでの資産の継承、レガシシステムとの相互利用を可能にさせる。

本開発では、OpenSOAPとDCOM、CORBAの分散アプリケーションリソースを連結するために、OpenSOAP ServiceとDCOM、CORBA Serviceの連携により、OpenSOAPとDCOM、CORBAとのブリッジングを実現する構成を提案した。OpenSOAP ServiceとDCOM、CORBA Serviceの連携では、OpenSOAP ServiceがDCOM、CORBA Serviceを利用するクライアント側になる。OpenSOAP Client側がDCOM、CORBA Serviceを利用したい場合、OpenSOAP Client側から送られた

OpenSOAP の Request Message が、OpenSOAP Server を通し、OpenSOAP Service である OpenSOAP ブリッジングのインターフェースに届く。ブリッジングインターフェースは CORBA、DCOM との連携を行い、CORBA、DCOM のサービスを利用可能とする。

2.5. メッセージ管理機能の実装

本研究では、複数のクライアントと複数のサービス間でメッセージ通信を可能とするためメッセージの管理を行う高機能 OpenSOAP サーバの開発を行った。

メッセージの処理としては、他の SOAP 実装でもサポートされているような、主に RPC 向けの同期処理機能を標準で備え、さらに処理に時間を要するサービス用として、クライアントが再度サーバに接続した時点でレスポンスを返す非同期処理も実装した。またファイアウォールの内外との SOAP メッセージをやり取りするための同期転送処理を実装し、転送経路指定と最大転送台数指定によりメッセージ転送の方法を指定できるような機能を備えている。また非同期の場合にもこの機能を使用可能であり、対象とするシーンを大幅に拡張した。非同期処理・同期転送処理・非同期転送処理は、Web サービスや電子ビジネスにおいて必須の機能とされながらも、これまでの SOAP 実装ではサポートされておらず、OpenSOAP プロジェクトが構築したサーバシステムの大きな特徴であるとともに、当該領域における優位性を誇る。

さらに、電子商取引の際に必要なメッセージのコミットとロールバックをサポートするためのトランザクション管理機能も実装している。また、ユーザサイド管理体制のサポートのために、サーバ内で処理されるメッセージの流れを記録するログ管理機能や、サーバの不正終了に対する未処理メッセージの保存機能を備えることで、保守性・安全性を確保している。さらに当プロジェクトにおけるセキュリティ管理機能の成果をいち早く有効活用し、サーバでのメッセージ処理を証明する署名機能も備えている。

OpenSOAP サーバは Linux、Solaris 8、NetBSD、Windows 2000 での動作が可能であり、サーバの開始・終了・再起動を簡便に行うためのツール群も同梱して提供される。

2.6. エージェント型ソフトウェアのモビリティに関する仕様策定と実装

PDA や JVM 搭載携帯端末などモビリティを有するデバイスにおいて、OpenSOAP のプラットフォームを利用して Web サービス等のネットワークアプリ

ケーションに対応させることで、OpenSOAP コミュニティとしてのウェアラブルな利用形態を提案することを目的とし、携帯端末や開発環境の最新動向を調査し、これに基づいて、現状に即したモバイル型 SOAP クライアントおよびサンプルアプリケーションの仕様策定を行った。

本調査結果に基づき、現状において最大の市場規模と最高の開発環境を提供する *i-appli* プラットフォームを開発フィールドとして選択し、これに適用するための詳細な仕様を策定するとともに、実装を行った。

また、エージェント型ソフトウェアによるネットワークサービスの知的自律化を目的として、SOAP メッセージの自動応答や自律的判断機能を搭載したアプリケーションの開発と評価を行った。OpenSOAP プロジェクトのフレームワークをモバイルプラットフォームに適用することにより、より即時的で高度なサービスの形態を提案・実証した。

2.7. 組み込みシステム向け SOAP インターフェースの開発

組み込みシステムの一つとしてロボットを取り上げ、OpenSOAP を適用したシステム構築を行う。具体的には、センシング系（カメラモジュール）、アクチュエーション系（車輪モジュール）等の要素機能を有する単機能モジュールロボットを考え、それらが神経系としてのネットワークで相互結合することで、遠隔で実体ハンドリングが可能となるシステムの構築を目指す。ここでは、搬送作業の中で「通信にもとづく論理的接続」を重視し、それを OpenSOAP アーキテクチャで実現することを考える。作業実現は、A)環境中へのモジュールの配置、B)モジュール間の相対位置・姿勢のパラメータ同定、C)インターフェースを用いた作業指示、の3ステップから構成される。ここで、B)においてパラメータ同定を迅速に行うために車輪モジュールの運動に関するデータベースを用意し、実際の配置に伴い、それを呼び出す必要がある。また C)において人間からの指示を各モジュールに適切に伝達する必要がある。これらの相互通信を、OpenSOAP を用いて行う。上記設計結果を実機に搭載し、椅子をある部屋から別の部屋へ移動する作業を、複数のカメラモジュールと1台の車輪モジュールを利用することで実現できた。これより、提案手法の有効性を示した。

2.8. ビジネスプロセスソリューションへの適用による評価

B2B向けWebサービスの枠組みに合わせて、OpenSOAP ミドルウェアを適用し、その有用性および問題点を評価することに主眼を置いた。評価のためのモデルとして、インターネット上および、イントラネット上に分散する鉄鋼ECサ

サービスの統合利用を想定し、その簡易モデルとしてのプロトタイプを作成して、ビジネスレベルでの実用に向けての各種の検討項目を検証評価した。

また、実際のWebサービスにおけるビジネスプロセスでは、単にサービスの提供がなされればすむものではなく、それらの動的結合にいたる過程が重要となるため、それをサポートする仕組みとしてのUDDIの利用を含めたシステム開発プロセスに関しても言及した。

2.9. 小規模 Web サービスの RAD による評価

インターネットインフラの整備にともない、ネットワークアミューズメント市場は拡大の基調を示している。その市場規模は無視できない規模になりつつあるが、一方でサーバへの投資など規模の増大に比例する投資を強いられている。OpenSOAP はオープンソースの環境であるために、小規模のパイロットケースとしてネットワークシステムを構築するにはリスクの少ないシステムと考えられる。

OpenSOAP がこのようなアミューズメント市場に対する可能性を示すことはこのシステムの普及において大きな意味を持つと思われる。

このような状況を鑑みて今回は OpenSOAP を用いたネットワークゲームの実装を試みた。SOAP を用いる際に大きな問題となるのは実行スピードであるが、今回は転送メッセージと思考ルーチンを一括転送することによってストレスのないシステムを構築した。

2.10. 大型計算機サービスへの適用による評価

北海道大学大型計算機センターにおけるスーパーコンピュータを利用した大型コンピューティングサービスに対して OpenSOAP の適用を行い、OpenSOAP に基づき実用的な Web サービスが実現可能であることを検証する。具体的には、日立製スーパーコンピュータ SR8000 内のライブラリを OpenSOAP に基づき利用可能とするライブラリサービス、更に、スーパーコンピュータにおいて自作したプログラムを実行可能とするプログラムサービスの2種類の Web サービスの構築を行う。

ライブラリサービスにおいては、SR8000 内のライブラリの一例として物理乱数発生ライブラリを取り上げ、これを Web サービス化するものとした。また、プログラムサービスにおいては、C 言語及び FORTRAN によって作成されたプログラムをスーパーコンピュータにおいて実行可能とする Web サービスの構築を行った。これらの Web サービスは OpenSOAP サーバによって管理される構成とし、更に、大型コンピューティングサービスを利用する際のネットワークのコストを軽減す

るため、非同期通信処理に基づく実行を可能とした。実験結果から、OpenSOAPを利用することによって実用的かつ効果的な大型コンピューティングサービスを実現可能であることが示された。

3. 事業化に向けた調査等の概要

3.1. 調査方法

本研究開発のテーマである SOAP に基づく Web サービス構築ツールに関し、一般ユーザがインターネット等から入手して利用できるものについて、その試用を含めた網羅的な調査を実施した。その結果を踏まえ、本研究開発の成果物となる OpenSOAP ミドルウェアの市場における優位性を示す。

また、OpenSOAP ミドルウェアの重要な特長のひとつであるオープンソース・フリーウェアをベースとした事業化の実現可能性について、Web サービスの将来性を含めた調査を実施した。

3.2. 調査結果の概要

SOAP を用いて Web サービスを構築するツールのうち、現在、一般ユーザがインターネット等から入手して自由に利用できるものは、Java による SOAP クラスライブラリである Apache SOAP (現 AXIS) や Microsoft Windows における開発環境をベースとした Microsoft SOAP Toolkit など、フリーウェア、製品ともにいくつか存在している。

しかしながら、本研究開発の成果物となる OpenSOAP ミドルウェアの有する、マルチプラットフォーム対応、および、既存のソフトウェア資産の再利用性重視、という大きな二つの特長の両方を実現するようなツールは存在していない。

たとえば、Apache SOAP は、Java 言語の API によりマルチプラットフォーム対応を実現しているが、既存のソフトウェア資産の有効活用という点では、Java 言語の API では不十分である。また、Microsoft SOAP Toolkit は、さまざまなプログラミング言語で開発が可能であるが、OS が Microsoft Windows に限定されており、さらに、Microsoft が提供する統合開発環境を全体としているため、既存のソフトウェア資産の有効活用という点においても、マルチプラットフォーム対応という点においても十分とは言えない。また、これら以外にも公開されているツールはいくつか存在するが、いずれも開発途上または単純なメッセージ交換の実装に留まっている。

本プロジェクトの成果物である OpenSOAP パッケージは、誰でもがダウンロードし利用できる、いわゆる「フリーウェア」として Web サイトで公開している。これは、プロジェクト当初からの計画のとおりであり、要約すると以下に示すよ

うな事業化戦略を持っているためである。

すなわちそれは、フリーウェアとして配布することによって、OpenSOAP パッケージを Web サービス構築のための基盤ミドルウェアとして世界に定着させ、「Web サービスアプリケーション」や「Web サービスプロバイダ」など、従前存在していない新たな市場を作り、その市場の中で本プロジェクトの参加企業が OpenSOAP パッケージを開発した経験とノウハウを活かしつつ、市場でのイニシアティブや優位性を持ちながら事業を展開するという中長期的な戦略である。このため、現段階においてその市場規模や目標シェアを計り知ることは不可能であるが、

- ・ 我国の情報処理サービス産業が約 10 兆円の市場規模であること。
- ・ アウトソーシングや ASP(Application Service Provider)、ネットワークコンピューティングなどの言葉に代表されるように、自社で情報システムを導入し運用するのではなく、社外の専門業者からサービスを購入することが珍しくなくなっており、今後も一層この市場の拡大が期待されること。
- ・ 現在の ASP 事業者は、特定のアプリケーションや計算機資源を提供しているに過ぎず、企業はより安価で、自社向けの自由度が高い情報サービスを望んでおり、その点からも Web サービスへの期待は大きいこと。

など、これらの点を勘案しても、情報処理サービス産業が Web サービスの方向に向かっているのは明らかであり、本プロジェクトの成果を利用した事業化は十分実現可能である。

4. 今後の課題

4.1. SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定

本プロジェクトで OpenSOAP を開発している途中においても、随時、SOAP や XML 仕様など関連仕様などの変更が発生した。その度に本プロジェクトでも仕様細部の見直しを随時行い対応をしてきたが、今後もそのような対応が必要とされることが発生することが予想される。バグなどの不具合への対応はもちろんのこと、本格的なビジネス Web サービスに適用した場合の機能評価、パフォーマンス評価、安定性評価などは開発期間中に十分に行われていないため、今後は実環境への適用評価を行いつつ、さらなる機能の追加、安定稼働させるための改良を行っていかなければならない。本プロジェクトで開発されたミドルウェアは、オープンソースとして一般に公開をしたため、今後は利用者からの苦情・要望等の反応が期待できる。このため、OpenSOAP コミュニティの拡大を図りながら、Web サービスの容易な構築実現のために、開発を継続して行っていく予定である。

4.2. 各種プラットフォーム向け SOAP

今回行った実装においては、プラットフォームとして、Windows 系 OS と Linux をはじめとした UNIX 系 OS を対象として行ったが、このような系統以外の OS、例えば、Mac OS 等への展開も行う必要があると考えられる。また、多言語対応に関しては、現状では、各 OS に実装されている国際化/地域化の機能を利用して、この機能が不完全あるいは不足しているプラットフォームでは、その機能を補うための実装を行う必要があると考えられる。また、これに関連して、現在の実装では、文字の格納に使用している Code Set がプラットフォームにより locale に依存する可能性があり、これによって、異なる locale で運用されている計算機間で不具合が起こる可能性があるため、一般的な意味での Universal Code Set への変更を考慮する必要がある。

現在の実装では、SOAP メッセージ処理の内、XML ドキュメント処理を別のライブラリを利用するようになっている。しかしながら、このライブラリが利用できない環境が存在する可能性を考え、XML ドキュメント処理の実装も追加する必要性がある。

4.3. 総合セキュリティ機能の実装

今回の一連の実装は、W3Cで標準化が進められているSOAP仕様に規定がないため、基本的には独自仕様にて進めたものである。一方、今後W3CによってSOAP仕様中にセキュリティに関する規定が定められることが予想される。

そこで今後は、こうしたW3Cの動向を注視し、セキュリティに関しての規定が定められた場合の対応が必要となる。ただし、こうした対応作業によって今回の仕様との互換性を失わない様に（すなわち、今回の仕様で作成したアプリケーションがそのまま動作する様に）考慮して設計する必要がある。

また、認証局関連で利用する電子証明書は、今回のOpenSOAP独自仕様であるが、より広い市場で利用可能とすることを目的として、X.509で定められている一般的に多く利用されている電子証明書に対応することも検討課題である。このX.509対応によって、他環境とOpenSOAP環境との親和性を高めて、より一層の市場拡大を期待することができるからである。

4.4. 他の分散オブジェクト技術とのブリッジングインターフェースの実装

OpenSOAP と他の分散オブジェクトのブリッジング開発は予定通り完了したが、今後の開発について説明する。

今回の開発は、OpenSOAP クライアント側がDCOM、CORBAのサービスを利用できるブリッジングのインターフェースを開発したが、その逆に、DCOM とCORBAのクライアント側がOpenSOAPのサービスを利用できるブリッジングインターフェースの開発は今後の課題として考えられる。

また、分散オブジェクト技術の発展に従って、ブリッジングインターフェースの改良が考えられる。分散オブジェクト技術であるDCOMとCORBAは長い道のりを歩んできたが、これからも発展していくのである。新しい機能と拡張はさらに加えられていく。いろいろな言語のサポートと値によるオブジェクト渡しのサポートのような水平な機能から、医療、通信、製造業のような垂直なファシリティの仕様まで、現在いくつかの拡張が提案されている。DCOMとCORBAのこれらの発展に応じて、OpenSOAPのブリッジングインターフェースも改良していく必要がある。

4.5. メッセージ管理機能の実装

本研究で開発を行った、複数のクライアントと複数のサービス間で様々な方法

でのメッセージ通信を可能とする高機能 OpenSOAP サーバは、プロジェクト終了時点において、Linux、Solaris 8、NetBSD、Windows 2000 上で要求仕様を満たし、動作が可能となっている。しかしながら、その機能の豊富さと仕様の獨創性に反して、本プロジェクトにおける工期が約半年と非常に短かったために、今後もブラッシュアップが必要とされる点も多い。主に以下の点が挙げられる。

安定性・対負荷性の向上：Web サービスのようなアプリケーションの特性上、一度起動したサーバが恒久的に安定して稼動する必要がある。また、同時に莫大な数のアクセスを処理する要求がなされる場合も想定し、高負荷下での耐久安定性試験なども行わなければならない。

簡便性の向上：OpenSOAP サーバのユーザは必ずしも計算機技術に熟達しているとは限らず、初心者は簡易的にサーバの機能を楽しむよう最大限に操作・設定を簡便化する一方、多用途に対応できるよう設定項目を十分に用意しなければならない。また、そのためのドキュメントの整備も不可欠である。

透明性の向上：オープンソースというプロジェクトの理念から、OpenSOAP サーバは、誰もがその内部設計と実装内容に接触できる機会をもつ。そのため、ソースコードは可能な限り簡潔で安全なものに改善すべきであるし、OpenSOAP サーバ稼動状態や非常状態の状況をユーザや開発者が容易に把握できるよう、ロギングの方法などを体系的に設計しなければならない。

以上の点について、今後も改善の余地があると認めると同時に、これを作業する開発者の確保や資金調達などの体制について検討が必要である。

4.6. エージェント型ソフトウェアのモビリティに関する仕様策定と実装

モバイルネットワークアプリケーションの業界は現在なお著しい速度で発展途上にあり、今後どのような Web サービス向けのプラットフォームが出現するか未知数であるため、引き続き動向調査を要する。関連して、現状で携帯端末は機種間の個体差が大きく、これに応じて実装を適合させなければいけないため、新機種の出現に対応する必要があるが、もしくは今後、業界におけるモバイルアプリケーションプラットフォームの標準化が進み、個体差を意識する必要がなくなる可能性もある。

また、モバイルビジネスシーンにおけるキラーアプリケーションが未出現であることも事実である。新規のモバイル Web サービスの創出だけでなく、レガシシステムとの接続も容易とする OpenSOAP プロジェクトのフレームワークにおいては特に、今後も十分にキラーアプリの出現を期待でき、探索を続ける必要がある

と考える。このためのキーテクノロジーは本サブテーマで提案したような、知的自律エージェント型ソフトウェアにある可能性も大きく、常に外界との情報交換を行いながらユーザをサポートするウェアラブルセクレタリの開発を目指すものである。

4.7. 組み込みシステム向け SOAP インターフェースの開発

今後の展望としては以下の事柄が挙げられる。

実体の機能表現：ここでは、カメラモジュールと車輪モジュールという二種類の実体モジュールを OpenSOAP プロトコルで記述する方法論について論じたが、より一般的に実体を機能表現することで、それらの協調による実体ハンドリング作業が可能になると考えられる。またそれらの結合による機能表現も別になされるべきである。あるコンフィグレーションの時にはある種の機能を有するが、別のコンフィグレーションをとった時にはまた別の機能を有するようになるという議論が必要であろう。

実体間の相互関係の自律的獲得方法論の構築：実世界に存在する実体間の関係は情報化されておらず、エージェントの自律的な行動により有意義な情報抽出が必要である。ここでは物体間の相互関係を車輪モジュールの運動により迅速に獲得する方法論を提案したが、より一般的に実体あるいは実体間の関係に関する情報として何が必要か？ それは実体のどのような操作により抽出可能か、という議論が今後必要になると考えられる。

通信機能と情報処理機能を有するコンパクトなハードウェアモジュールの開発：単機能モジュールを実際に利用可能な形態にするためにより小型化を指向する必要がある。半導体技術を用いることによりかなりの改善が見込まれるが、通信機能と相互位置姿勢計測機能を統合できることが本来は望ましい。

4.8. ビジネスプロセスソリューションへの適用による評価

本研究により開発された Web サービス基盤としての Open SOAP ミドルウェアを実際のビジネスプロセスに適用する上で、ツールとしての完成度や信頼性の向上が求められるのは言うまでもないが、それ以外にも安全性や機能性の面においてもよりブラッシュアップが必要であると考えられる。しかしながら、それらの対応のための人的リソースやコストの問題もあるため、より重要な要素から、より効率的に対策を進める必要がある。そこで、実際のビジネス環境において早急に OpenSOAP 適用事例を多く生み出し、現実世界において改めて見つけられる問題点の洗い出しや、システム稼働後のメンテナンスフェーズにおけるノウハウの蓄

積などを行い、それらを OpenSOAP コンソーシアムへ対して情報のフィードバックを行うことにより、より高度なレベルへより迅速に、OpenSOAP システムを導く手助けとすることを今後の課題とする。

4.9. 小規模 Web サービスの RAD による評価

インターネット環境の著しい整備により、アミューズメントに対する市場は急速に整いつつある。この市場において、どのようにアカウントिंगの問題を解決していくかは、もはや時間的にも余裕のないものとなっている。OpenSOAP をベースとしたアミューズメントは、単に従来の HTML ベースのコンテンツの代替品ではなく、新しいネットワーク構成によるサーバーコングロマリットからの情報を提供するものである。そこではサーバ間の協調により、受信するコンテンツというよりは、自分のスペースを作成していくようなコンテンツが作成可能になっていくだろう。

本システムにおいて、アイデアさえあれば、比較的安価にサーバアミューズメントシステムが構築できることが判明したことをベースに、今後はこのようなシステムの普及に努めていく必要があると思われる。

4.10. 大型計算機サービスへの適用による評価

OpenSOAP に基づく大型コンピューティングサービスにおける課題・将来展望としては以下の項目が挙げられる。

UDDI との連携に基づく大型コンピューティングサービス：大型計算機センターには、多種多様なライブラリ・アプリケーション・ソフトウェア資産が存在する。OpenSOAP を利用することによって、これらを Web サービスとして公開することは可能であるが、利用者が実際にこれらのサービスを利用する場合には、ネットワークを通して各サービスの仕様を獲得可能でなければならない。これを可能とするしくみが UDDI であり、UDDI との連携が可能な大型コンピューティングサービス環境を実現す必要がある。

OpenSOAP に基づく大型コンピューティングサービスの実運用：OpenSOAP に基づく大型コンピューティングサービスを実運用するためには、技術的な問題の他に、制度面での問題が考えられる。すなわち、利用者申請制度の問題や、如何にして課金を行うかという問題を検討する必要がある。このような制度の見直しも含めた OpenSOAP に基づく大型コンピューティングサービスを実現することが必要とされる。

グリッドコンピューティング的情報処理：他の大型計算機センターや情報基盤

センターが OpenSOAP を採用することによって、センター間での連携が可能となり、グリッドコンピューティング的な情報処理が可能になると考える。このような情報処理の形態は、現在、世界的に必要とされているため、今後の検討を行っていく必要がある。

5. 研究開発スケジュールと主な設備の内容

5.1. 研究開発スケジュール表

	平成 13 年												平成 14 年		
	月	3	4	5	6	7	8	9	10	11	12	1	2	3	
① SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定		→													
②各種プラットフォーム向け SOAP 実装				→									→		
③総合セキュリティ機能の実装				→						→					
④他の分散オブジェクト技術とのブリッジングインターフェースの実装									→				→		
⑤メッセージ管理機能の実装							→								
⑥エージェント型ソフトウェアのモビリティに関する仕様策定と実装					→									→	
⑦組み込みシステム向け SOAP インターフェースの開発					→								→		
⑧ビジネスプロセスソリューションへの適用による評価					→									→	
⑨小規模 Web サービスの RAD による評価					→									→	

⑩大型計算機サービスへの適用による評価													
⑪総合調査研究													

5.2. 主な購入設備 (500 万円以上) 名称と使用目的、設備場所一覧

購入設備なし

6. 研究開発体制図

取りまとめ部門	研究開発項目	研究開発機関
(財)札幌エレクトロニクスセンター	① SOAP をベースとするミドルウェアアーキテクチャに関する仕様策定	北海道大学大型計算機センター 北海道情報大学 (株)情報科学センター
	② 各種プラットフォーム向け SOAP 実装	(株)ジャパンテクニカルソフトウェア (財)札幌エレクトロニクスセンター
	③ 総合セキュリティ機能の実装	北海道大学大型計算機センター 北海道情報大学 (株)アドイン研究所
	④ 他の分散オブジェクト技術とのブリッジングインターフェースの実装	(株)ジャパンテクニカルソフトウェア 北海道エヌエスソリューションズ(株)
	⑤ メッセージ管理機能の実装	(株)情報科学センター 北海道大学 室蘭工業大学
	⑥ エージェント型ソフトウェアのモビリティに関する仕様策定と実装	北海道大学 室蘭工業大学 (株)ジャパンテクニカルソフトウェア
	⑦ 組み込みシステム向け SOAP インターフェースの開発	東京大学 北海道大学 室蘭工業大学
	⑧ ビジネスプロセスソリューションへの適用による評価	北海道エヌエスソリューションズ(株) (株)情報科学センター
	⑨ 小規模 Web サービスの RAD による評価	(有)バードランド (株)ジャパンテクニカルソフトウェア
	⑩ 大型計算機サービスへの適用による評価	北海道大学大型計算機センター
	⑪ 総合調査研究	(財)札幌エレクトロニクスセンター

研究実施場所一覧

財団法人札幌エレクトロニクスセンター	札幌市厚別区下野幌 テクノパーク 1 丁目 1-10
北海道エヌエスソリューションズ株式会社	室蘭市みゆき町 2 丁目 13-1 北海道高度情報技術センター (HITEX)内
株式会社ジャパンテクニカルソフトウェア	札幌市北区北 21 条西 3 丁目 1-14
株式会社アドイン研究所	札幌市中央区南 1 条西 10 丁目 4 タイムビル 5F
株式会社情報科学センター	札幌市北区北 9 条西 3 丁目 1-1 タカノビル 4F
有限会社バードランド	札幌市白石区栄通 3 丁目 南栄ビル 303
北海道大学大学院工学研究科 複雑系工学講座 自律系工学分野	札幌市北区北 13 条西 8 丁目
北海道大学大型計算機センター	札幌市北区北 11 条西 5 丁目
東京大学大学院工学研究科 精密機械工学専攻 新井・湯浅・太田研究室	東京都文京区本郷 7-3-1
北海道情報大学	北海道江別市西野幌 59-2
室蘭工業大学	北海道室蘭市水元町 27-1

7. 研究開発者一覧

[株式会社ジャパンテクニカルソフトウェア]

(6名)

氏名	役職	研究開発項目
許士 博章	第5システム部 部長	②④⑥⑨
内海 顕	第5システム部 課長	
中村 富美夫	第5システム部 主任技師	
藤沢 範浩	第5システム部 技師	
蜂谷 聖一	第1システム部 技師	
高橋 一彦	第5システム部 技師	

[北海道エヌエスソリューションズ株式会社]

(6名)

氏名	役職	研究開発項目
仲尾 哲弥	札幌支店 シニアエンジニア	④⑧
清水目 和幸	システムソリューション第二部 シニアエンジニア	
竹谷 弘	システムソリューション第二部 シニアエンジニア	
伊藤 慎一郎	システムソリューション第二部 シニアエンジニア	
松前 貴志	システムソリューション第二部	
茶谷 哲光	システムソリューション第二部 シニアエンジニア	

[株式会社アドイン研究所]

(3名)

氏名	役職	研究開発項目
藤巻 俊秀	第一事業部札幌ラボラトリー 主任技師	③
山脇 裕二	第一事業部札幌ラボラトリー 副主任技師	
唐 巍立	第一事業部札幌ラボラトリー	

〔株式会社情報科学センター〕 (3名)

氏名	役職	研究開発項目
奥野 拓	研究開発部 主任技師	①⑤⑧
金 丹	研究開発部	
坂東 友則	研究開発部 技師	

〔有限会社バードランド〕 (4名)

氏名	役職	研究開発項目
金沢 良次	代表取締役	⑨
渡辺 作太郎	開発部	
碓井 洋介	開発部	
二村 美智子	デザイン部	

〔北海道大学〕 (9名)

氏名	役職	研究開発項目
嘉数 侑昇	大学院工学研究科教授、 大型計算機センター長	①③⑥⑦
吉井 伸一郎	大学院工学研究科 ポストドクター	
成瀬 継太郎	大学院工学研究科 助手	
石田 崇	大学院工学研究科 ポストドクター	
永山 隆繁	大型計算機センター HINES 運用室長	
杉浦 孝博	大型計算機センター ネットワーク掛	
伊藤 和彦	大型計算機センター システム管理掛	
林 卓也	大型計算機センター システム管理掛	
更科 高広	大型計算機センター システム管理掛	

〔東京大学〕 (4名)

氏名	役職	研究開発項目
新井 民夫	大学院工学系研究科 教授	⑦
太田 順	大学院工学系研究科 助教授	
湯浅 秀男	大学院工学系研究科 助教授	
前田 雄介	大学院工学系研究科 助手	

[北海道情報大学]

(1名)

氏名	役職	研究開発項目
中島 潤	情報メディア学部情報メディア学科 講師	①③

[室蘭工業大学]

(3名)

氏名	役職	研究開発項目
久保 洋	情報工学科 教授	⑤⑥⑦
魚住 超	情報工学科 助教授	
佐藤 裕一	サテライトベンチャービジネスラボラトリー ポストドクター	

8. 工業所有権等の取得状況

出願案件はなし

9. 外部発表等の状況

発表論文

○奥野 拓, 嘉数侑昇: IT サービス事業共有基盤向けミドルウェア OpenSOAP, 計測自動制御学会第2回システムインテグレーション部門学術講演会, 2001.12 (名古屋)

○中島 潤, 嘉数侑昇: OpenSOAP ミドルウェア・アーキテクチャの概要とメッセージ管理技術, 計測自動制御学会第2回システムインテグレーション部門学術講演会, 2001.12 (名古屋)

○坂東友則, 金 丹, 嘉数侑昇: マルチプラットフォーム対応 OpenSOAP API の開発, 計測自動制御学会第2回システムインテグレーション部門学術講演会, 2001.12 (名古屋)

○佐藤裕一, 茶谷哲光, 石田 崇, 嘉数侑昇: OpenSOAP サーバのアーキテクチャと実装, 計測自動制御学会第2回システムインテグレーション部門学術講演会, 2001.12 (名古屋)

○山脇裕二, 嘉数侑昇: OpenSOAP へのセキュリティ技術の実装, 計測自動制御学会第2回システムインテグレーション部門学術講演会, 2001.12 (名古屋)

○長尾光悦, 成田一郎, 林卓也, 更級高広, 中島潤, 嘉数侑昇: OpenSOAP を用いたコンピューティングサービスの検証と評価, 計測自動制御学会第2回システムインテグレーション部門学術講演会, 2001.12 (名古屋)

○松本高斉, 陳 慧瑩, 嶋田賢一, 太田 順, 新井民夫, 嘉数侑昇: OpenSOAP を指向したモジュール型群ロボットシステム, 計測自動制御学会第2回システムインテグレーション部門学術講演会, 2001.12 (名古屋)

○松本高斉, 陳 慧瑩, 嶋田賢一, 太田 順, 新井民夫, 嘉数侑昇: 分散配置型モジュール群によるオープンロボットシステム -SOAP によるオープンなモジュール構築の提案-, 日本機械学会ロボティクス・メカトロニクス講演会, 2002.6 (松江)

○奥野 拓, 中島 潤, 嘉数侑昇: Web サービス開発用ミドルウェア OpenSOAP 概要, 情報処理北海道シンポジウム, 2002.4 (札幌)

○坂東友則, 金 丹, 山脇裕二, 嘉数侑昇: OpenSOAP API における国際化とセキ

ユリティ～Web サービス必須機能の実装～, 情報処理北海道シンポジウム, 2002.4 (札幌)

○茶谷哲光, 佐藤裕一, 石田 崇, 嘉数侑昇: Web サービス運用基盤システム・OpenSOAP サーバの開発, 情報処理北海道シンポジウム, 2002.4 (札幌)

○金 丹, 坂東友則, 佐藤裕一, 嘉数侑昇: OpenSOAP と他の分散コンポーネントアーキテクチャとのブリッジング技術, 情報処理北海道シンポジウム, 2002.4 (札幌)

○中島 潤, 長尾光悦, 嘉数侑昇: OpenSOAP による Web ベース大型計算サービスの構築, 情報処理北海道シンポジウム, 2002.4 (札幌)

○長尾光悦, 吉井伸一郎, 嘉数侑昇: OpenSOAP 向けレジストリサービス TDDI の開発, 情報処理北海道シンポジウム, 2002.4 (札幌)

解説記事関連

「ソフトバンク・コマースと北大が SOAP/UDDI 技術で共同研究」, ZDNet エンタープライズ, 2001.12.17 (<http://www.zdnet.co.jp/enterprise/>)

「異なるシステム間の情報共有 研究 - SBC・北大, 共同研究で -」, 日経産業新聞, 2001.12.17

「ソフトバンク・コマース, Web サービスプロジェクトを開始」, Japan.internet.com, Web テクノロジー, 2001.12.19 (<http://japan.internet.com/webtech/>)

「Web サービスの最新動向と実用に向けた展望」, 日経インターネットテクノロジー, no.056, 2002.3

「北大教授ら VB 設立 - 独自ソフトでシステム構築 -」, 日経産業新聞, 2002.4.4

「ウェブサービスのオープンソースプロジェクト OpenSOAP」, iINTERNET magazine, インプレスコミュニケーションズ, 2002.5

「SOAP ソフトを無償で公開 トランザクション機能を付加 - 北海道の大学やベンチャからなる「OpenSOAP Project」が開発 -」, 日経インターネットテクノロジー, no.058, 日経 B P 出版センター, 2002.5

10. 英文要約

Development of Middleware as Platform for IT Application Services

Summery

In this project, the OpenSOAP middleware, which is based on the SOAP standard, was developed for supporting the web service development. The various expanded functions as followings were added into the OpenSOAP middleware.

(1) The design of specification decision architecture on the middleware architecture based on SOAP

In this study, architecture design of the middleware (the OpenSOAP package) developed in this project is made with the purpose in order to easily realize the business Web service. Concretely, through the investigations of W3C SOAP specification, XML specification and the other relation specification, and trough the investigations on existing relation middleware product, and also through the research on the trend of the market, the followings were carried out: (1)outlines design of the whole structure (architecture) of middleware developed by this project, (2)outlines design such as the design of the function to be realized by the middleware, (3)outside design of libraries and programs developed in sub theme group which is in charge of the OpenSOAP core region. And, it was also in charge of the following works: (1) fitting of the detail specification of development charge part for the each sub theme group, (2) development schedule adjustment of the whole project, (3) the work as a necessity for the open of development middleware package.

(2) The OpenSOAP implementation for the various platforms

One middleware was designed and implemented as a library function group, which carry out SOAP packaging that moves in both of the Unix system and Windows system. This library function group is named as OpenSOAP API, and is carried out according to C language. OpenSOAP API is classified according to the function as followings: (1) SOAP message treatment, (2) SOAP client communication, (3)SOAP service communication. Especially for (1), the implementation for internationalization and multi-lingual was carried out. The sample code was made, and the action was confirmed on Windows 2000 for Windows system, and on Linux, NetBSD, Solaris8 for Unix system. By implementing in C language, the utilization in other programming language such as C++ became

possible. And, for Java language, OpenSOAP client and service were made implemental by the native code. Also, for Java language, the asynchronous transmission, which was one of the expansions, was implemented in the client. In addition, security module, which was the expanded function in OpenSOAP, was integrated.

(3) The implementation of the general security function

The market of electronic commerce (B2B, B2C), between enterprise and enterprise, or between consumers and enterprise, expands at high speed. Based on such actual circumstances, the correspondence is required to OpenSOAP for application in the electronic commerce field. In the meantime, since the electronic commerce is done through Internet, it is possible that various injustices enter in. Therefore, it becomes necessary to offer the environment which can remove such injustice for the safe application to the electronic commerce field. Though it is made that the implementation of the security technology for removing such problem is an indispensable condition for the electronic commerce, in SOAP specification of which the standardization is advanced by W3C, the regulation on the security does not exist. Therefore, the implementation on the security technology was newly carried out in this OpenSOAP development, and the environment, which made OpenSOAP to be safety in B2B, B2C and application, was improved. Items implemented in this development are encryption/decoding, electronic signoff/signature certification and certification station relation function.

(4) The implementation of bridging interface with other distributed object technology

OpenSOAP transmit the message formatted in the XML by using the HTTP, which makes it to be easily for using the various services. As the result, according to the SOAP specification, the method is offered to cooperate with various software, and the development becomes possible which covers multiple languages and multiple enterprises by utilizing the legacy component.

In the legacy dispersion component architecture, DCOM and CORBA are famous. In order to inherit and shift these legacy dispersion component architecture to OpenSOAP, the bridging interface which connect with DCOM, CORBA is necessary.

By developing the bridging interface between OpenSOAP and DCOM, CORBA, it becomes possible for inheriting the exiting assets and using the legacy system.

In this development, in order to connect with DCOM, CORBA, the composition was proposed, which realized the bridging between OpenSOAP and DCOM, CORBA by the cooperation of OpenSOAP service and DCOM, CORBA service. In the cooperation, OpenSOAP service becomes the client side of DCOM, CORBA. When a request message send from the client side of OpenSOAP reaches the OpenSOAP service (the bridging interface) through the OpenSOAP server, the bridging interface carries out the cooperation with DCOM, CORBA, and it makes the service of DCOM, CORBA to be usable.

(5) The implementation of the SOAP message management skill

In order to enable the message communication between multiple clients and multiple services, high-functional OpenSOAP server who controlled the message in this implementation was developed.

As a treatment of the message, OpenSOAP server had the synchronous processing function for the RPC which is mainly standard in other SOAP implementation. Also as a service needing time in the processing, the asynchronous processing was implemented which returned the response in point of time when the client connected for the service. And, the synchronous transfer treatment for exchanging the SOAP message with the inside and outside of fire wall is implemented and has the function which can designate the method of the message transfer by transfer routing and largest transfer unit number appointment. And this function was also expanded to be usable for the asynchronous processing. It is proud that the OpenSOAP server architecture has the functions as asynchronous processing and synchronous transfer treatment and asynchronous transfer treat, because in the exiting SOAP implementation these functions are not supported, although it is made to be an indispensable function in Web service and electron business.

In addition, in order to support the user side management system, the transaction management function for supporting commit and rollback of the message, which requires in the electronic commerce has also been implemented. And maintainability and safety have been ensured by having log control function, which records the flow of the message dealt with in the server and saving function of the untreated message for the unjust end of the server. Also, the result of security control function in this project is quickly used efficiently and also has signature function, which proves the message treatment in the server.

The OpenSOAP server can be executed on Linux, Solaris 8, NetBSD, Windows 2000, and the tool group for conveniently carrying out start and close and restart of the server is also offered.

(6) Specification decision and implementation on the mobility of the agent type software

In the device with mobility such as JVM installation portable terminal and PDA, the latest trend of the portable terminal and the development environment was investigated, for the purpose of proposing utility form as an OpenSOAP community by making to correspond to network application such as the Web service using the platform of OpenSOAP. Based on this investigation, specification decision of mobile type SOAP client and sample application which matched present state was done.

Based on the research result, the i-appli platform, which offers the largest market size and the best development environment in present state, was choose as a development field. And for applying the OpenSOAP to the i-appli platform, the detailed specification was decided, and the implementation was carried out.

Also, the following were carried out for the purpose of the intelligent self-regulation of the network service by the agent type software: the automatic answering of the SOAP message and the development of the application, which implemented the autonomous judgment function and evaluation. By applying the framework of the OpenSOAP project in mobile platform, the form of the advanced and immediate service was proposed and was verified.

(7) The development of the SOAP interface for the integration system

The system construction which applied OpenSOAP was done, by taking up the robot as one of the integration system. Concretely, considering an single function module robot with the element function, such as sensing system (camera module), actuation system (wheel module), it is aimed to construct the system, which can make it possible to handle the substance in the remoteness, by connecting these single function module robot through the nervous system. Here, we emphasized the logical connection based on the communication in the carrier work, and considered to realize it with OpenSOAP architecture. The work is realized by the following 3 steps: A) arrangement of the module to the environment, B) parameter identification of relative position and attitude between modules, C) workmanship instruction using the interface. Here, for B), in order to quickly

carry out the parameter identification, the database prepare for the wheel module, and the database calling according to the actual arrangement are necessary. And for C), it is also necessary to appropriately transmit the indication from human to each module. These intercommunications will be carried out by using OpenSOAP. Through implementing the superscription design on the real machine, the work moving a chair from one room to another room was realized by utilizing multiple camera modules and one wheel module. According to the result, the effectiveness of the proposal technique was shown.

(8) The evaluation by the application to the business process solution

The principal objective was put in for evaluating the usefulness and problem, when applying the OpenSOAP middleware by adjusting frame of Web service for evaluation B2B. As a model for the evaluation, by making the prototype as the simple model on the assumption of integration utilization of the iron and steel EC service which disperses in on Internet on intranet, various examination item for practical use by the business level were verified and were evaluated. And the system development process, including the utilization of UDDI as a mechanism, was mentioned.

(9) The evaluation by RAD of the small-scale Web service

With the improvement of the evaluation for the Internet infrastructure, the network amusement market shows the key of the extension. Though it becomes a scale, which cannot disregard the market size, it forces the investment in proportion to the increase in scales to the server such as the investment in the other. Since OpenSOAP is an environment of the open source, it seems to be a system with small risk as a small-scale pilot case. That OpenSOAP shows the possibility for such amusement market seems to have large meaning in the popularization of this system. By such considering situation, the packaging of the network game using OpenSOAP was tried in this development. Though the execution speed is the large problem when SOAP is used, the system, in which the transfer message and thought routine are transferred by lumping together, was constructed without stress.

(10) The evaluation by the application to the large computer system

OpenSOAP is applied for large computing service using the supercomputer in the Hokkaido Univ. mainframe computer center, and it is verified that the practical web service based on OpenSOAP is realizable. Concretely, 2 kinds of Web services are constructed. One is the library service, which makes it possible to use the library in super

computer SR8000 made in library service base on OpenSOAP. The other is the program service, which makes it possible to carry out the program made by oneself on the super computer. For the library service, web service, which made it possible to use the physical random number generation library in SR8000, was constructed. And for program service, web service, which made the program made by C language and FORTRAN feasible in the super computer was constructed. As a composition controlled by the OpenSOAP server, these web services enabled the execution based on the asynchronous transmission treatment in order to reduce the cost of the network in utilizing the large computing service. According to the experimental results, it was shown that the practical and effective large computing service is realizable by using OpenSOAP.