社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－EMB－1 (4)
2006／6／27

再構成可能な組込みシステム向け分散オブジェクトシステムの開発

谷田貝純†，早川栄一‡
† 拓殖大学大学院工学研究科 〒193-0985 八王子市館町 815-1
‡ 拓殖大学工学部 〒193-0985 八王子市館町 815-1
E-Mail：† epsilon@os.cs.takushoku-u.ac.jp，‡ hayakawa@cs.takushoku-u.ac.jp

本稿では，多様な環境を持つ組込みシステムに対して，その環境に適した機能と規模に再構成可能な分散オブジェクトシステムの開発について述べる．組込みシステム向け分散オブジェクトシステムでは，本来持つ多くの機能を削減し軽量化を行うことが必要であり，開発時にその支援を行うことが有用である．そこで，開発者が分散オブジェクトシステムを利用する環境に合わせてシステムの規模を把握しその機能を構成することが可能なシステム，およびこのシステムに用いる分散オブジェクトシステムの開発を行った．

Development of a Reconfigurable Distributed Object System for Embedded Systems

Jun YATAGAI†，Eiichi HAYAKAWA‡
† Graduate School of Engineering，Takushoku University 815-1 Tatemachi，Hachioji City，Tokyo，193-0985 Japan
‡ Faculty of Engineering，Takushoku university 815-1 Tatemachi，Hachioji City，Tokyo，193-0985 Japan
E-Mail：† epsilon@os.cs.takushoku-u.ac.jp，‡ hayakawa@cs.takushoku-u.ac.jp

This paper describes the development of a reconfigurable distributed object system that has function and scale suitable for the environment for various embedded systems.

In distributed object system for embedded systems many functions should be reduced for lightweighting and supporting environment is useful when it is developed. Therefore the system which developer grasps the scale to environment, and could constitute the function, and the distributed object system for this system was developed.

1. はじめに

現在，携帯電話や情報家電など組込みシステムが用いられる機会が増加している．さらにそれらがネットワークに接続され，互いに情報を交換することにより動作する環境が求められている．このようなシステムを構築する方法の一つとして分散オブジェクトシステムがある．分散オブジェクトシステムはアプリケーション開発者からネットワークを隠蔽するだけでなく，オブジェクト指向開発による資源の再利用が可能という利点がある．

しかし，従来の分散オブジェクトシステム[1]は規模が大きく，資源が限定された組込みシステムでの利用が難しい．また組込みシステム向けに開発された分散オブジェクトシステム[2]は付加的な機能を削減し，機能によるサポートを受けることができない．

そこで，本研究では分散オブジェクトシステムの機能と規模の割当てを開発者が設計し，利用する環境に応じた分散オブジェクト環境を構築可能なシステムを開発した．

2. 問題分析

組込みシステムにおける分散オブジェクトシステムの利用に伴う問題は次のとおりである．

(1)組込みシステムの制限

組込みシステムの資源はPCなどのコンピュータと比較して非常に限定されている．特にROMとRAMの容量の制限が与える影響は大きい．これは大きなシステムの導入における障害となり，ソフトウェア開発者への負担が増加する．

また，組込みシステムはプログラムサイズ10kB以内と

いう制限を持つ携帯電話[3]や，CPU 外部の大容量のファイルシステムやメモリを利用できる場合[4]など，資源は個々に差がある．組込みシステムで導入可能なシステム規模の境界は環境により異なる．

(2)分散オブジェクトシステムの規模

PC で利用される分散オブジェクトシステムでは分散オブジェクト環境を構築するための機能に加え，それを容易に構築可能とするための付加機能が利用できる．これら機能により分散アプリケーション開発において開発効率の向上が期待できる．

しかし，これらの機能は規模が大きく，結果として分散オブジェクトシステムを大規模なものとしている．このようなシステムを組込みシステムへ導入することは上述の資源の制限による問題から不可能である．

(3)組込み向け分散オブジェクトの機能不足

組込みシステムをターゲットとした分散オブジェクトシステムは小規模に設計され，さらにリアルタイム性など組込みシステム向けの特性を持つもの[5]が存在する．これらにより組込みシステムへの分散オブジェクトシステムの導入が可能である．しかし，開発者はこれらに必要に応じて機能の追加や削除を行うことができない．

(4)システム規模の把握

組込みシステムにおいて，アプリケーションのプログラムサイズとメモリ消費量は重要視されている．これが大きい場合はシステムの導入が不可能となる．特に，アプリケーション開発者が導入するシステムのフットプリントを把握することは難しい．システムへの機能追加などの機能の組合せ要素が加わることにより，把握はさらに困難になる．

3. 設計方針

上述の問題を解決するために，本システムの設計方針を次のものとする．

(1)小規模な ORB の提供

分散オブジェクト環境を実現するシステムの中核となる ORB(Object Request Broker)について，その機能を，基本機能と拡張機能の二つに分離し，要求された機能に適したサイズのシステムを提供可能にする．資源が制限された環境での利用を可能とするため ORB の基本機能は軽量なものとし，システムを小規模なものとする．

(2)ORB 機能の再構成支援環境の提供

分散オブジェクトシステムを利用する開発者が，ORB の機能構成を設計し，その ORB を出力するシステムを提供する．これにより分散オブジェクトシステムを用いる環境に合わせた ORB を利用することができる．

また，機能の構成時に，開発者の機能設計に対してその構成の予想メモリ消費量とプログラムサイズを示す．これにより開発者は最終的なシステムの規模を把握し，利用環境に適したシステムを利用できる．

4. 全体構成

システムの全体構成を図 1 に示す．また，システムの利用方法を次に示す．

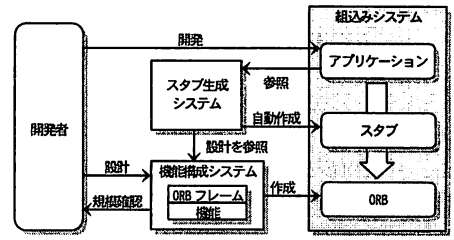

図 1　全体構成

開発者は機能構成システムにより，利用する ORB を作成する．環境で利用可能な規模の範囲であれば，分散システムに多くの機能を持たせることができる．

本システムではアプリケーションから直接に ORB の機能を利用することはなく，スタブによって仲介を行う．開発者は ORB 機能を利用するアプリケーションを開発し，スタブ生成システムによりアプリケーションに対応したスタブを自動生成する．

アプリケーションはスタブを介して ORB の機能を利用することで通信を行う．

5. 設計

5.1. 機能構成システム

機能構成システムは開発者の設計した ORB を作成する．開発者は設計部に対し，用意されたコマンドを用いて機能の追加，削除，現在の機能構成の確認をする．設計部は機能管理部を参照し，システムが要求する機能が提供されているかどうかを検証する．これによって，誤った設計を防止する．また，設計された情報は設計情報として保存する．機能構成システムの全体構成を図 2 に示す．

出力部は ORB の作成時に設計情報を参照する．その設計に基づき，機能管理部より各機能の情報を取得して，ORB フレームに各機能を適用することで ORB を作成する．ORB フレームは機能の衝突を防ぐ枠組みである．これについては次章で述べる．

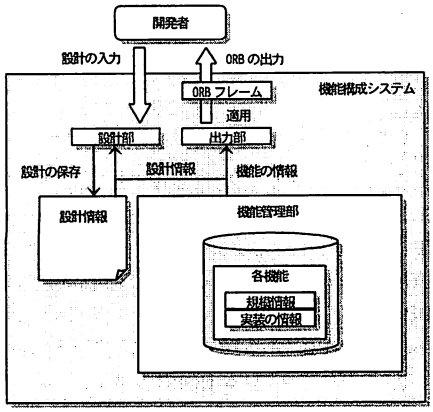

図2　機能構成システム

(1)機能のデータベース管理

各機能の情報は内部のデータベースによって管理する．これは複数の機能の情報を持ち，各機能の情報は次に示すものとする．

・機能のプログラムサイズ
・機能のメモリ消費量
・機能のソースコード，またはライブラリの参照

これらの機能の情報をデータベースで管理することで，今後追加される機能群も含め管理を統一する．

(2)機能の分類

ORB を構成する機能は利用方法などが異なり，大きく次の3種類に分類される．機能構成システムにはこの機能の分類により，異なる管理と構成を行う必要がある．

・内部機能

内部機能は ORB の内部に依存する機能である．これは ORB 間のリクエストに対応する機能であり，直接にオブジェクトの操作を行う．リクエストは通信する ORB の機能呼出し要求である．

内部機能の管理はリクエストに対応するメソッド単位で扱う．内部機能はメソッドレベルで分離され，各機能は自由に組合せることが可能である．各機能のメソッドはデータベースに保存する．

・通信方式

ORB の通信を行う機能である．これは ORB 機能と通信処理の設計を分離し，通信方式の置換えにより他の機能に非依存で通信方式を切り替えることができる．

通信機能の管理は，各通信方式をライブラリとして保存し，データベースの機能情報はその参照を持つ．

・サービス

サービスは ORB 外部の付加的な機能である．これは ORB 本体とは分離されている．アプリケーション開発者は ORB の補助機能としてこれを利用することができる．

サービスの管理は，各機能をライブラリとして保存し，データベースはその参照を保存する．

(3)設計部

設計情報を操作するコマンドを提供し，開発者の入力に対して設計を書きかえる．そのとき，設計部は機能管理部を参照し利用可能な機能を提示する．

また，現在の設計での予想メモリ消費量とプログラムサイズを表示する．各機能は個別に測定したプログラムサイズとメモリ消費量の情報を持つ．設計部は機能管理部を参照し，各機能に対するプログラムサイズと予想メモリ消費量の合計値をそれぞれ求めて表示する．

5.2. ORB フレーム

無秩序に作成された機能による機能の組合せは，各機能の衝突を招く．このような機能の衝突が起こるとシステムが正しく動作しない．そこで機能の構成には，各機能の衝突を防ぐために ORB の基盤となる枠組みを必要とする．その枠組みとして ORB フレームを定義する．ORB フレームの構成を図3に示す．

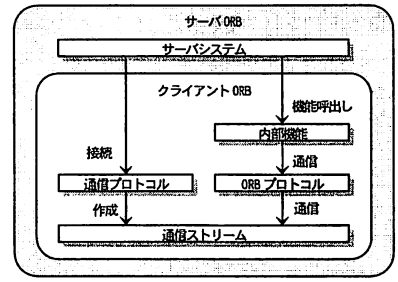

図3　ORB フレーム

ORB フレームは，ORB を構成する機能を各要素に分割し，この機能構成の定義と，機能の構成に関らず利用される実装を合わせたものである．ここで，機能の構成に関わらず利用される要素はサーバシステムと内部機能となる．ORB フレームは作成する ORB の枠組みとなる．

ORB フレームは機能の置換えを想定し各要素を分離する．サーバシステムと内部機能以外の要素は同じ分類の機能と置き換えることができる．

このフレームを基礎に作成される ORB は軽量であること想定しているので，各機能は軽量なものとする．また，リモートオブジェクトの機能利用を目的とする場合はサーバシステムが不要であるので，そのような場合を想定し ORB はクライアントサーバの構成とする．

(1)サーバシステム

サーバシステムはサーバ ORB で利用し，クライアントからのサーバへの接続と，リクエストを受信するサーバの二つの役割を持つ．サーバはリクエストに対し内部機能を呼び出す．

(2)内部機能

内部機能は，機能構成システムにおける内部機能であるメソッドの集合となる．各機能は一つのリクエストに対応し，内部機能はサーバ ORB ではサーバからリクエストに応じて呼び出すメソッド，クライアント ORB ではサーバへのリクエスト送信するメソッドを持つ．機能構成システムによる内部機能の追加は，ORB フレームの要素である内部機能に新たなメソッドを追加する．

(3)ORB プロトコル

ORB の各機能の通信手順を持つ．通信プロトコルと通信ストリームに非依存とし，ORB プロトコルの置換えにより他の ORB の通信プロトコルに対応する．

(4)通信ストリーム

ストリームとそれによる通信機能を持つ．

(5)通信プロトコル

ORB で利用する通信プロトコルを管理する．これにより TCP/IP と HTTP による通信の置換えが可能である．

5.3. スタブ生成システム

スタブ生成システムの構成を図 4 に示す．

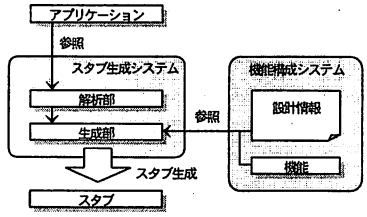

図 4　スタブ生成システム

スタブ生成システムはアプリケーションに対し，スタブを自動で生成する．これにより，サーバとクライアントでメソッドの対応を事前に行うことで，転送するデータ量を削減し通信時のオーバーヘッドが減少する．また，ORB 機能のインタフェースをローカルなオブジェクトの操作に近い形で提供することで，分散環境の利用が容易となる．

アプリケーションはスタブを介して ORB 機能を利用するので，機能の追加に対応する必要がある．そのため，スタブ生成システムは機能構成システムの作成した設計情報を参照し，ORB に対応したスタブを自動生成する．

5.4. 機能

機能構成システムにより扱う機能の一部を分類して次に示す．

(1)通信方式

・iAppli 通信

NTTDoCoMo の携帯電話上で動作する iAppli[6]による HTTP 通信で ORB の機能が利用可能となる．これにより HTTP 通信の隠蔽によるアプリケーション開発の容易化と，携帯電話を端末としたサーバ機能の利用が可能である．

・HORB 通信

Java による ORB である HORB[7]と通信を行うことができる．すでに構築されている HORB システムによるネットワークに導入が可能となる．

(2)内部機能

・オブジェクトの作成機能

オブジェクト作成機能はクライアントからリモートのオブジェクトを作成する機能である．サーバで用意された単一のオブジェクトに接続する方式が機能の呼出しを重視しているのに対して，クライアントからオブジェクトを作成することで，オブジェクトのデータ構造を利用することができる．

・オブジェクト自動解放機能

リモートなオブジェクトの解放を自動化する．これによりクライアントサーバ間で GC によるオブジェクト解放が反映される．

(3)サービス

・ネームサービス

名前によるオブジェクトの利用が可能になる．

・イベントサービス

オブジェクトの非同期イベントを行う機能を提供する．これにより計算機資源を有効に活用することができる．

6. 実現

すべてのシステムは Java 言語を用いて開発した．Java 言語は，オブジェクト指向開発による機能の再利用性など開発効率を向上させる特性を持ち，分散オブジェクトシステムに適した言語である．よって ORB の開発には Java 言語を用いた．表 1 にシステムの規模を示す．

表 1　システムの規模

ORB 機能	行数
機能構成システム	1000
スタブ生成システム	800
ORB フレーム	300

ORB機能を利用するためのインタフェースはHORBとほぼ同様である．これはHORBが，分散オブジェクトシステムの利用を容易とするという方針で開発されたシステムであり，本システムも同様にアプリケーションの開発効率の向上を目的としているためである．

(1)機能構成システム

データベースシステムを除いた総行数が約1000行である．機能構成システムはコマンドの入力によりORBの設計を行なうことができる．また設計時に規模の確認を行い，ORBを作成できる．これにより環境に適したORBを利用することが可能である．

(2)ORBフレーム

ORBフレームはサーバシステムと内部機能が約300行程度である．システム要素の分割とその定義をしたORBフレームを用意することにより，各機能の組合せや機能の部分的な置換えが可能である．

(3)スタブ生成システム

スタブの自動生成により，アプリケーションからのORB機能の利用が容易となった．また，事前にメソッドの対応付けを行うことでデータの転送量増加を抑え，システムパフォーマンスの低下を防止した．

7. システムの利用

機能構成システムによるORBの作成と，スタブの作成手順を次に示す．

(1)ORBの作成

アプリケーション開発者は分散オブジェクトシステムを導入する環境に適したORBを作成する．機能構成システムによるORB作成の手順を図5に示す．

```
>configure

>func
 iappli - C - iAppli通信方式
 horb - C - HORB通信方式
 createObj - F - オブジェクトの作成
 naming - S - ネーミングサービス

>add iappli

>plans
 Com
  iappli, dStream, orb
 Func
  createObj, relObj
 Service
  noFunc.

>scale
 Size server:8.0kB client:4.2kB
 FP server:1.62MB client:1.597MB

>make
```

図5　機能構成システムによるORBの作成

図5では次の手順でORBを作成している．

i. configure
機能構成システムを起動する．

ii. func
利用可能機能名，機能の分類，機能説明を確認する．

iii. add
設計にiAppli通信機能を適用する．

iv. plans
設計したORBを構成する機能を確認する．

v. scale
プログラムサイズとメモリ消費量を確認する．

vi. make
設計を参照しORBを出力する．

(2)スタブの生成

アプリケーションに対するスタブの作成例を図6に示す．

```
>stubGen Server.java
 Server.java compile...  complete.
 Server_Proxy.java creating...  complete.
 Server_Skeleton.java creating...  complete.
```

図6　スタブの生成

このような手順で実際にORBの機能構成と，アプリケーションに対応したスタブを生成することができる．

8. 評価

現在，NTTDoCoMoの現在最新となるFOMA携帯電話のiAppli実行環境は，DoJa-4.x[3]プロファイルある．これはプログラムサイズが100KBまでのJavaプログラムを実行することができる．構成システムによってDoJa-4.xプロファイルの実行環境機能で動作するiApple用ORBを作成した．他のORBとのプログラムサイズの比較と機能の比較を表2，3に示す．

表2　各ORBのサイズ

ORB	プログラムサイズ(kB)
HORB	258815.0
iHORB[2]	6.8
作成したORB	9.0

表3　各ORBの機能

ORB	機能
HORB	オブジェクトの作成 オブジェクトの接続 オブジェクトの自動解放 (他多数)
iHORB	オブジェクトの接続
作成したORB	オブジェクトの作成 オブジェクトの接続 オブジェクトの自動解放

HORBは携帯電話のような資源が限られた環境を想定していないので，この場合プログラムサイズの問題から利用が不可能である．

iHORB[2]はiAppliが利用可能となった初期の携帯電話の環境を想定している．これはプログラムサイズが10kB以内という制限を持つため，iHORBのプログラムサイズは非常に小さい．しかし，そのために機能を削減しているので利用可能な機能が限られている．

機能構成システムによって作成したORBはプログラムサイズ100kB以内という制限の中で複数の機能を利用することが可能である．この場合，アプリケーションはオブジェクト作成機能でクライアントのデータをサーバのオブジェクトとして保存し，データが不要ならばオブジェクトの自動解放機能により自動的にサーバのオブジェクトを解放する，という処理が可能である．

9. 関連研究

現在，組込みシステムにおける利用を想定した分散オブジェクトシステムが存在している．

iHORBは携帯電話での利用に対応した分散オブジェクトシステムである．これは，小さなプログラムサイズと，携帯機器による不安定な接続環境への対応を特徴としている．移動体通信を想定した機能は有効に利用できるが，プログラムサイズを削減するために他の多くの機能を省略しているので，付加的な機能を利用することができない．

情報家電ネットワークによるオブジェクトの増加を考慮した分散オブジェクトシステムにNeko-bus[8]ある．これはオブジェクトのインタフェースを統一することで，大量のオブジェクトインタフェースの管理を不要としている．しかし，静的なインタフェースの定義に対し，この方法では利用する機能名の送信や，機能名からの機能検索などの負担が増加する．多くの機能を利用するアプリケーションにおいて，一度の機能呼出しに対する負担の増加はパフォーマンスの低下の要因となる．

10. おわりに

本稿では組込みシステムで十分な機能を利用するためのORB再構成システムの開発について述べた．本研究ではORBの枠組みとなるフレームと機能構成システム，その機能とスタブの生成システムの開発を行った．

これにより資源が制限された環境で，その環境に適した分散オブジェクトシステム用いて，分散オブジェクト環境を利用したアプリケーションを実行することができた．

今後の課題として，構成する機能が異なるORBによる通信の差異の解決と，システムの実用性の評価を行う必要がある．

参考文献

[1] CORBA CORNER
http://www.omg.org/#CC

[2] iHORB
http://mascot.mis.ous.ac.jp/horb-ous/ihorb/

[3] DoJa
http://www.nttdocomo.co.jp/service/imode/make/index.html

[4] シリコンリナックス CAT709
http://www.si-linux.co.jp/product/cat709/

[5] ORBexpress
http://www.adniss.jp/products/middle/middle_02.html

[6] iAppli
http://www.nttdocomo.co.jp/service/imode/make/content/iappli/index.html

[7] HORB
http://horb.a02.aist.go.jp/horb-j/

[8] 邑中雅樹，"情報家電に向けた，ソフトウェア部品の開発"
未踏ソフトウェア創造事業，2000