M-035

# 多言語プラットフォーム上で動作する P2P 型複製オブジェクト環境の構築
# P2P-based Replicated Object Environment Running on Multiple Language Platforms

市川 泰宏 † 山本 佑樹 ‡ 高田 秀志 †
Yasuhiro Ichikawa Yuki Yamamoto Hideyuki Takada

## 1. はじめに

コンピュータを用いたリアルタイムな協調作業支援システムでは，端末間でリアルタイムにビューや操作の同期を取ることが必要となる．これらを実現する環境として，複製計算モデル [1] に基づく，“ P2P 型複製オブジェクト環境 ”が考えられる．P2P 型複製オブジェクト環境では，P2P ネットワーク上の各端末がオブジェクトの複製を保持し，オブジェクトに対するメソッド呼び出しを他端末に伝播することで，端末間のオブジェクトの状態が同一に保たれる．我々はこの P2P 型複製オブジェクト環境を提供する協調作業支援システム開発基盤 “ CUBE(Collaborative Universal Basic Environment) ” [2] を Java で構築している．

また，近年，プラットフォームが異なるさまざまな端末が普及してきている．これにより，コンピュータを用いたリアルタイムな協調作業支援システムも，多様な端末を用いて利用されるようになると考えられる．このような端末間で動作可能な協調作業支援システムを開発するときに考えられる問題点として，端末によって使用できる開発言語が異なるという点や，CPU やメモリなどの処理性能に差があるという点が挙げられる．このうち，本研究では，端末によって使用できる開発言語が異なるという点に着目し，実装言語の異なるオブジェクト間で，オブジェクトの状態を同一に保つ P2P 型複製オブジェクト環境を構築するためのシステム構成について検討する．

## 2. 協調作業支援システム開発基盤“ CUBE ”

我々はコンピュータを用いたリアルタイムな協調作業支援システムに必要な機能を提供する Java フレームワーク CUBE の構築を行っている．CUBE では，P2P 型複製オブジェクト環境において，各端末に配置された複製オブジェクトの状態を同一に保つために，“ オブジェクトミラーリング ”という同期機能を提供している．また，複数の端末をグループ化し，そのグループを階層的に管理するグループ管理機能や，複製オブジェクトの状態を記録し，端末が作業から途中離脱した場合も，復帰を可能とする回復機能などを提供する．

オブジェクトミラーリングは，各端末にオブジェクトの複製を配置し，その複製オブジェクトに対するメソッド呼び出しを伝播することで，各端末のオブジェクトの状態を同一に保つ手法である．図 1 にオブジェクトミラーリングの仕組みについて示す．オブジェクトミラーリングでは，オブジェクトに対するメソッド呼び出しを伝播するために，Java のリフレクション機能によって提供されている代理オブジェクトを利用する．代理オブジェクトは複製オブジェクトと同じインタフェースをもっており，アプリケーションでは，複製オブジェクトのメソッドを呼び出すのと同様に，代理オブジェクトのメソッドを呼び出すことができる．

図 1 に示すように，代理オブジェクトを通してメソッドが呼び出されたとき，代理オブジェクトは複製オブジェクトのメソッドを呼び出し，さらに，ネットワークを通して他端末にもメソッド呼び出しを伝播する．各端末にはオブジェクトマネージャが存在し，Java のリフレクション機能を用いて，受け取ったメソッド呼び出しを実行する．これにより，端末間で複製オブジェクトの状態を同一に保つ．

次節では，実装言語の異なるオブジェクト間で，複製オブジェクトの状態を同一に保つためにオブジェクトミラーリングを開発言語に合わせて適用することを考える．

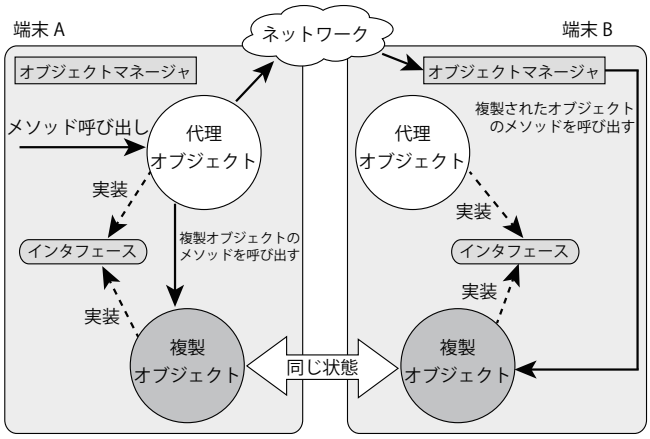


図 1: オブジェクトミラーリングの仕組み

## 3. 多言語プラットフォーム上で動作する P2P 型複製オブジェクト環境

実装言語の異なるオブジェクト間で複製オブジェクトの状態を同一に保つ場合，言語による仕様の違いやシステム構成を考慮する必要がある．本節では，他言語間でオブジェクトを複製するためのシステム構成，および，端末間のメッセージ伝播手法について述べる．

### 3.1 システム構成

図 2 に，実装言語の異なるオブジェクト間で，オブジェクトミラーリングを行うためのシステム構成を示す．図 2 に示す通り，Java で開発されたアプリケーションが動

†立命館大学 情報理工学部
‡立命館大学大学院 理工学研究科

作している端末は，それらの端末間でメソッド呼び出しを伝播することで，端末間の複製オブジェクトの状態を同一に保っている．一方，Java 以外の言語で開発されたアプリケーションが動作している端末は，Java アプリケーションの動作している端末から，任意に親端末を選出する．選出した親端末とメッセージのやり取りを行うことで，親端末のもつ複製オブジェクトと子端末のもつ複製オブジェクトの状態を同一に保つ．

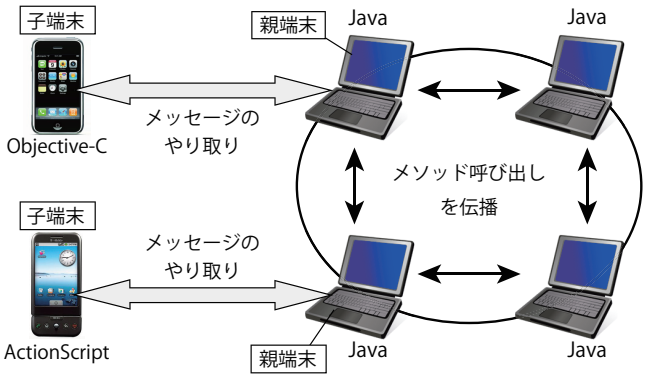


図 2: システム構成

図 3 に示すように，親端末は自身の複製オブジェクトの他に，子端末上の複製オブジェクトと対になる複製オブジェクトを保持する．対になる 2 つの複製オブジェクト間で，メソッド呼び出しに必要な情報をやり取りすることで，実装の異なるオブジェクトの状態を同一に保つ．

また，子端末上の複製オブジェクトは，親端末上の複製オブジェクトを通して，オブジェクトミラーリング以外の機能であるグループ管理や，障害回復のためのチェックポインティング機能などを利用することができる．

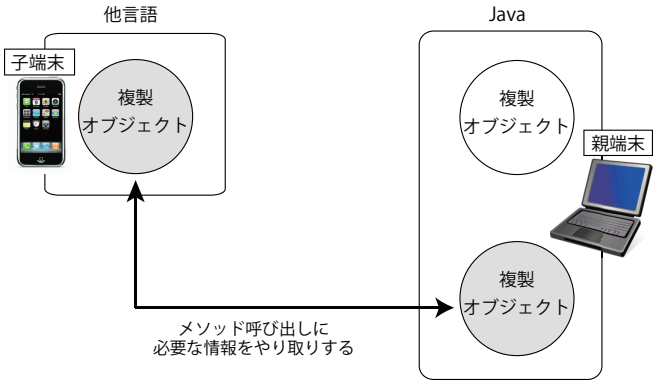


図 3: 実装言語の異なるシステム間におけるオブジェクトの配置

### 3.2 他言語間におけるメッセージ伝播

Java で実装された複製オブジェクト間でメッセージを伝播する場合，マルチキャストによってメッセージのやり取りをしている．一方，実装言語の異なる複製オブジェクト間でメッセージを伝播する場合，対となる複製オブジェクト同士が，ユニキャストでメッセージのやり取りを行う方式を採る．このときやり取りされるメッセージには以下の情報が含まれる．

- メソッドの送信端末を一意に識別するための ID
- 呼び出されたメソッドをもつオブジェクトを一意に識別するための ID
- 呼び出されたメソッドのシグネチャ
- 呼び出されたメソッドの引数の値

このメッセージは，言語が異なっても相互にやり取りできるように，XML を用いて構造的に記述される．

## 4. 適用例

本手法の適用例として，Java と Objective-C で，それぞれ CUBE を用いたパズルアプリケーションを構築した．このアプリケーションでは，パズルのピースを複製オブジェクトとし，各端末でピースの移動を同期する．

Java で開発されたパズルアプリケーションで，ユーザがパズルのピースを移動したとき，パズルのピースに対して Piece.setPoint(x，y) といったメソッドが呼び出される．それと同時に，Java で開発されたパズルアプリケーションが動作している他端末に対して，同様のメソッドが伝播される．そのメソッド呼び出しを受け取った端末の中で，親端末は，自身のパズルピースに対してメソッドを呼び出し，また，子端末上のパズルピースと対になるパズルピースに対して，メソッドが呼び出される．さらに，子端末に対してメソッド呼び出しが伝播され，メソッド呼び出しを受け取った子端末では，パズルピースに対して，Objective-C で書かれた [Piece setPoint:x :y] といったメソッドが呼び出される．これにより，親端末と同様にパズルのピースが移動される．

## 5. おわりに

本稿では，実装言語が異なるオブジェクト間で，オブジェクトの状態を同一に保つための手法を提案し，多言語プラットフォーム上で動作する P2P 型複製オブジェクト環境を構築した．

今後は，子端末とメッセージをやり取りする親端末に障害が発生したとき，親端末を動的に切り替えることで，システム全体の耐故障性を向上させることを検討する．

## 参考文献

[1] David A.Smith, Alan kay, Andreas Raab, David P.Reed, ”Croquet - a collaboration system architecture”, First Conference on Creating Connecting and Collaborating Through Computing 2003(C5 2003), pp. 2-9, 2003.

[2] Shogo Noguchi, Hideyuki Takada: ”CUBE: A Synchronous Collaborative Spplications Platform Based on Replicated Computation”, CollaboTech 2009, 2009.