M-056

オブジェクト同期によるリアルタイム協調作業支援システム開発基盤の構築
The Development of a Software Platform for Real-time Collaborative Work Supporting Systems with Object Synchronization

野口 尚吾 † 野村 俊太 ‡ 植田 亘 ‡ 笠井 康寛 ‡ 高田 秀志 ‡
Shogo Noguchi Shunta Nomura Wataru Ueda Yasuhiro Kasai Hideyuki Takada

1. はじめに

コンピュータを利用した協調作業の支援は，CSCW(Computer Supported Cooperative Work) という分野として盛んに研究が行われている．その中でも，複数人の間でのリアルタイムな協調作業は，作業の効率化や新しいアイディアの創造につながるため有用である．こうしたシステムは，遠隔会議システムなどをはじめとしたコミュニケーションシステムとして多くの職場で用いられている．また，学生同士の協調学習の支援を目的とした教育現場への適用も有用である．

リアルタイム協調作業を支援するシステムを構築するためには，システム間の処理の同期や，ネットワークやノードの参加・離脱の管理など，様々なことを考慮する必要があり，それらは開発者の大きな負担となる．しかし，こうしたシステムの開発を支援するための開発基盤はまだ少ない．

我々は，リアルタイムな協調作業支援システムを開発するための Java 開発環境を提供する “CUBE (Collaborative Universal Basic Environment)” という開発基盤を構築している．CUBE では，TeaTime[1] で提案されている複製計算モデルにもとづき，各システムの処理を同期する機能を提供する．これに加えて，システムで提供されるサービスへのノードの途中参加や途中離脱を可能とするノード管理機能も提供する．さらに，PC や携帯端末などが混在した異種端末同士での協調も可能とする仕組みを提供する．

2. リアルタイム協調作業支援システム開発基盤 “CUBE”

リアルタイム協調作業支援システム開発基盤 CUBE は，ノード同士でリアルタイムに処理を同期することで，協調作業環境の構築をサポートする Java 開発環境を提供する．

TeaTime における複製計算モデルでは，オブジェクトが各ノード上に複製されて配置される．あるノード上であるオブジェクトに対してメソッド呼び出しが発生すると，これが他のノードに伝播され，対応するオブジェクトのメソッドが実行される．これにより，すべてのノード上でオブジェクトの状態が同一に保たれる．

CUBE では，これを Java の Proxy クラスを用いて実現し，リアルタイムにノード間の処理を同期するアプリケーションを構築するためのライブラリを提供する．

†立命館大学大学院 理工学研究科
‡立命館大学 情報理工学部

3. メソッド同期を用いたオブジェクトの状態の同期

Java アプリケーション上での処理は，オブジェクトの状態を変更することで行われ，オブジェクトの状態は，メソッド呼び出しにより変更される．CUBE では，メソッド呼び出しを同期することでオブジェクトの状態を同一に保ち，ノード間の処理を同期する機能を提供する．また，同期するオブジェクトが生成されたさいには，オブジェクトの生成を同期先に通知し，対応するオブジェクトを生成する機能を提供する．これらの機能を用いることで，各アプリケーション内で同じオブジェクトを，同じ状態で保持することが可能となる．

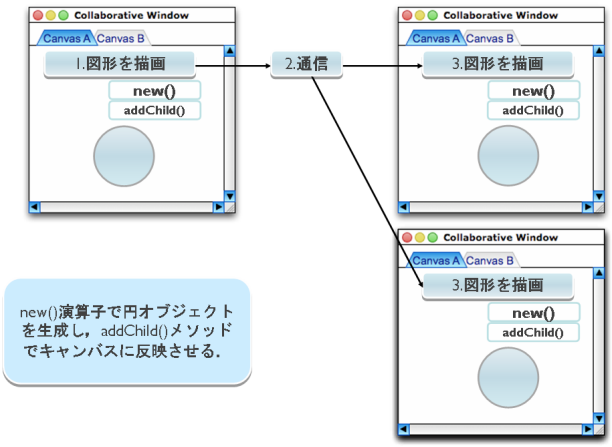

図 1: メソッド呼び出しの同期

CUBE を用いて実装した描画ツールを例として説明する．図 1 のように，複数のユーザが描画ツールを起動している場面を想定する．描画領域に図形を配置するさい，プログラム側では配置する図形のオブジェクトが生成され，そのオブジェクトを描画領域に表示するメソッドが呼び出される．このような処理が起きたさい，そのツールは図形オブジェクトの生成と図形を表示するためのメソッドが呼び出されたことを他のノード上のツールに通知する．通知を受け取ったツールは，同様に図形のオブジェクトを生成し，図形を表示するメソッドを実行する．このような処理を行うことによってオブジェクトの状態を同期し，描画領域上の表示を共有することができる．

3.1 Proxy クラスを用いた代理オブジェクトの生成

CUBE では，オブジェクトの状態を同期するために Java のリフレクション機能により提供されている Proxy

クラスを用いたオブジェクト生成メソッドを提供する．同期させるオブジェクトを引数にして，このオブジェクト生成メソッドを呼び出すことで代理オブジェクトを生成する．代理オブジェクトは同期オブジェクトと共通のインタフェースを実装しているため，同期オブジェクトと同様にアクセスすることができる．代理オブジェクトのメソッドが呼び出されたさい，Proxy クラスの機能によりメソッドの呼び出し情報が InvocationHandler に渡される．そして，呼び出されたメソッドの情報を同期先に通知する．

図 2: CUBE を利用したプログラム例

図 2 に示すように，オブジェクトの状態を同期させるには，CUBE が提供するメソッド (SynchroObjectProxy.newInstance()) を用いて代理オブジェクトを生成する．アプリケーションプログラムからは，同期させるオブジェクトではなく，代理オブジェクトへの参照を保持する．

3.2 メソッド同期の仕組み

CUBE のメソッド同期の仕組みを図 3 に示す．

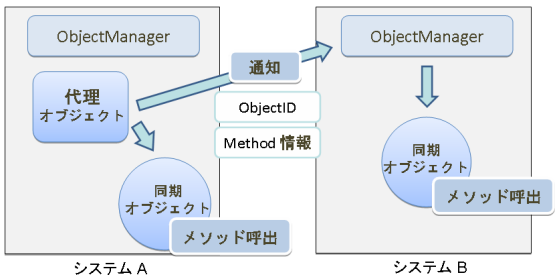

図 3: メソッド同期の仕組み

CUBE では，同期オブジェクトを ObjectManager クラスを用いて管理する．オブジェクト生成メソッドを用いて生成されたオブジェクトは，そのオブジェクトを識別するためのオブジェクト ID と対応づけて ObjectManager で管理される．代理オブジェクトのメソッドが呼び出されたさい，同期オブジェクトのオブジェクト ID と呼び出されたメソッド情報 (Method インスタンス，引数) を同期先に通知する．同期先の ObjectManager は通知を受け取ると，オブジェクト ID をもとに適当なオブジェクトを識別し，そのオブジェクトのメソッドを実行する．

3.3 異種端末間でのオブジェクトの状態の同期

PC 用のシステムを構築する場合と携帯端末用のシステムを構築する場合とでは，性能上の問題などから異なった実装が必要になることがある．CUBE は，こうした実装が異なるシステム同士でも，オブジェクトの状態を同期することができる仕組みを提供する．

CUBE のメソッド呼び出しの同期手法は，オブジェクトの ID とメソッド情報を用いて行われるため，メソッドの実装には依存しない．そのため，共通のインタフェースを実装することで，実装が異なるオブジェクトの状態でも同期させることができる．

4. ノードの参加と離脱

CUBE では，システム間の P2P ネットワークの構築に JXTA フレームワーク [2] を用いている．JXTA には，ピアグループというピアを管理する機能がある．このピアグループの機能を拡張し，ユーザのサービスへの参加および離脱を管理するノード管理機能を提供する．本機能ではノード管理に加えて，サービスに途中参加したノードに対して，その時点でのオブジェクトの状態を伝える．これにより，ユーザがサービスに途中参加してきたさいでも，そのユーザはすでにサービスに参加している他のユーザと同じ状態からサービスに参加することができる．また，サービスに影響なくサービスからノードが途中離脱する機能も提供する．

5. おわりに

本稿では，リアルタイム協調作業支援システムの構築基盤 CUBE の機能と構成について述べた．CUBE では，メソッド同期を用いてオブジェクトの状態を同一に保つことで，システム間の処理の同期を実現する機能を提供する．また，システムで提供されるサービスへのノードの途中参加や途中離脱を実現するためのノード管理機能を提供する．

今後は，複数の処理が同時に行われたさいの排他制御を実装する必要がある．また，CUBE を用いてシステムを実装し，そのシステム間で処理を同期するさいのパフォーマンスなどを確認する必要がある．

謝辞

本研究を進めるにあたり，有益なご助言を頂きました立命館大学情報理工学部島川教授 および研究室の方々に感謝いたします．

参考文献

[1] David P. Reed, “Designing Croquet’s TeaTime - A Real-time, Temporal Environment for Active Object Cooperation,” Invited Talk, OOPSLA, 2005.

[2] Scott Oaks, Bernard Traversat, Li Gong, “JXTA IN A NUTSHELL,” O’REILY, 2002.