M-046

小型情報通信端末による共同作業支援のためのP2P型オブジェクト複製環境

P2P-based Object Replication Environment for Collaborative Work Support Systems Utilizing Mobile Devices

鈴本 悟 † 櫻打 彬夫 ‡ 高田 秀志 †
Satoru Suzumoto Yoshio Sakurauchi Hideyuki Takada

1. はじめに

携帯電話に代表される小型情報通信端末の普及により, 誰もが情報端末を随時携帯し場所を問わずに利用可能になった. さらに近年では, Android を搭載した端末やiPhone に代表されるスマートフォンのように端末の高性能化が進んでいる. これによって, 従来の小型情報通信端末では不可能であった複雑な処理も可能になった. 本稿では, このような小型情報通信端末を用いて日常生活における共同作業支援をおこなうアプリケーションに着目する.

小型情報通信端末を用いた共同作業の支援において, 共同作業の支援者は, 支援対象のグループに対し中途参加をし, 支援作業をした後, グループから離脱をおこなう. 共同作業を行う上で, グループは操作対象となるデータおよびそのデータに対する操作を加えたオブジェクトを所有する. 共同作業を支援するアプリケーションは, 共同作業支援による作業結果をグループ内の端末へリアルタイムに反映させなければならないため, グループ内に存在しているオブジェクトの共有およびグループに属している端末への操作の同期が必要とされる. 特に, 利用者がグループへ中途参加をする上で, アプリケーションはグループ参加時において, グループ内に存在しているオブジェクトの状態を端末間の差異を考慮して複製する必要がある. さらに, 小型通信端末上に構築される共同作業を支援するアプリケーションが複製するオブジェクトは, 対象作業に必要とされるオブジェクトのみとすることが望ましい.

本稿では, 小型情報通信端末による共同作業支援を実現するために, 利用者が作業グループへ中途参加をし, 支援のために必要とされるオブジェクトに操作を加え, 作業グループから離脱する一連の動作をおこなうことが可能になる P2P 型オブジェクト複製環境を提案する.

2. P2P 型オブジェクト複製環境

共同作業をおこなうアプリケーションには, 共同作業に参加している各端末上のオブジェクトの共有および操作の同期が必要とされる. その実現方法として, 共同作業に参加する各端末上に同一オブジェクトを複製し, ある端末でそのオブジェクトに対する操作があった場合, 他の端末の当該オブジェクトへその操作を通知することが考えられる. 本稿では, この仕組みを P2P 型オブジェクト複製環境と呼ぶ.

CUBE [1] は, Java によるリアルタイムな共同作業支援システムの構築支援を目的とした P2P 型オブジェクト複製環境を実現するフレームワークである. その主な機能として, オブジェクトのふるまいを同期させるオブジェクトミラーリングがある. オブジェクトミラーリングとは, 共同作業を行う各端末上へ同じオブジェクトを複製し, ある端末上でオブジェクトのメソッド呼び出しが発生すると, 他の全ての端末でも同じメソッド呼び出しを発生させ, オブジェクトのふるまいを同期するものである. ここで, オブジェクトミラーリングの対象となるオブジェクトをミラーオブジェクトと呼ぶ.

CUBE を利用したアプリケーションでは, リアルタイムな共同作業が可能であるが, 中途参加をおこなうときにミラーオブジェクトの状態を表す属性に対して同期をおこなう必要がある. ミラーオブジェクトの属性に対して同期をおこなわないと, 中途参加をした端末とグループに参加していた端末では, 中途参加をした時点でミラーオブジェクトの状態が異なってしまう.

そこで, CUBE を用いた共同作業の支援をおこなうアプリケーションにおいて, 新しく参加する端末に対してミラーオブジェクトの属性を同期することにより, 共同作業をおこなっているグループに対する中途参加を可能にした P2P 型オブジェクト複製環境が必要とされる.

3. 中途参加におけるオブジェクト同期手法

中途参加をおこなう端末は, グループに存在する全てのミラーオブジェクトを複製 / 同期するのではなく, 共同作業の支援に必要とされるミラーオブジェクトのみを複製 / 同期すればよいと考えられる. 複製 / 同期を行う上で, 小型情報通信端末と PC ではオブジェクトの実装が異なることも考えられるため, 対象のミラーオブジェクトを直接複製するのではなく, ミラーオブジェクトに対して共通に実装させるインターフェースを用意し, そのインタフェースで定義されたメソッドを介して状態の同期をおこなう.

3.1 オブジェクト同期の流れ

中途参加時におけるミラーオブジェクトの同期の流れは図 1 のように, 大きく分けて 5 つのステップがある. 以下に, 端末 B に存在するミラーオブジェクトに対して, 中途参加をする端末 A が同期をおこなう手順を示す.

1. オブジェクトの生成
   まず, 共同作業の支援をおこなう端末 A は, 状態を同期させるミラーオブジェクトを, 参加するグループに存在しているミラーオブジェクトのオブジェクト ID について整合性をとった上で生成する.
2. オブジェクト同期要求
   次に, 端末 A は, 同期させるミラーオブジェクトのオブジェクト ID とともに, オブジェクトの同期要求をグループに対しておこなう.

†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

3. オブジェクトの同期に使用する情報の収集
　グループに所属している端末の中で要求を受け取った端末 B は, 指定されたミラーオブジェクトの状態を同期させるのに必要なメソッド, および, そのミラーオブジェクトの状態をクラス定義に基づいて取得する.
4. メソッド実行要求
　その後, 端末 B は, 端末 A へ, 同期を要求されたオブジェクト ID とともに値を設定するメソッドの実行通知, および, そのメソッドの引数となる現在の状態を送信する.
5. 受け取った情報からメソッドを実行して状態を同期
　最後に, 要求を受け取った端末 A は, 受け取った情報からミラーオブジェクトの状態を同期する.

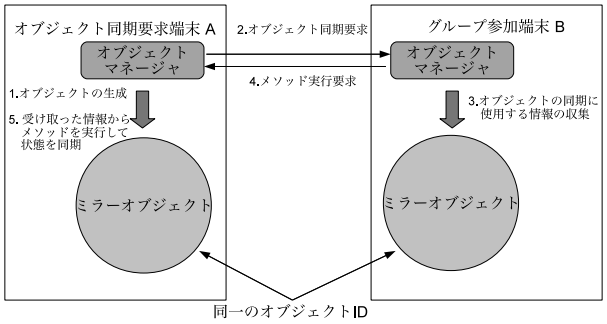

図 1: オブジェクトの状態同期の流れ

3.2 オブジェクト同期のためのクラス定義手法

3.1 節で述べた通り, ミラーオブジェクトは, ミラーオブジェクト自身に定義されているメソッドを使用して状態の同期をおこなう. オブジェクトの状態同期を要求した端末側で状態を設定する時に使用するメソッドを状態同期メソッド, 状態同期要求を受信した端末でミラーオブジェクトの状態を取得するために使用されるメソッドを状態取得メソッドと呼ぶ. 状態同期メソッドは, 端末間による差異を吸収するために, インタフェースで定義されている.

リスト 1 に, Java による実装例を示す. インタフェースで定義された状態同期メソッドをインプリメントしたメソッドに対し, アノテーション @stateMethod をつけ, その引数である getterMethods に状態同期メソッドの引数に対応した返り値を持つメソッドのメソッド名を指定する. @stateMethod は, インタフェース IMirrorObject に定義されており, 状態同期をおこなうミラーオブジェクトを生成するクラスは, IMirrorObject をかならず実装しなければならない. この例では, IPerson インタフェースの実装クラスである Student クラスから生成されるミラーオブジェクトにおいて, ミラーオブジェクトの状態を示す属性 name を同期させるために, 状態同期メソッドには “void setName(String name)” を使用し, 状態取得メソッドでは “String getName()” を使用している.

状態同期メソッドの引数が複数ある場合, @stateMethod の引数である getterMethods へ, 状態同期メソッドの引数順にそれぞれの引数に対応したメソッドを “,” で区切って設定する.

リスト 1: アノテーションによるメソッドの指定

```
public interface IPerson{
   :
  // 状態同期メソッドはインタフェースで定義
  public void setName(String name);
   :
}

public class Student implements IPerson,IMirrorObject{
   :
  String name; // 同期対象の属性
   :
  // 状態同期メソッド
  // getterMethodsで状態取得メソッドを指定
  // この場合はgetNameが状態取得メソッド
  @stateMethod(getterMethods={"getName"})
  public void setName(String name){
    this.name = name;
  }

  // 状態取得メソッド
  public String getName(){
    return name;
  }
   :
 }
```

4. 適用例

本手法の適用例として, 教員による児童への学習支援が考えられる. 我々は, CUBE を基盤とした協調学習環境 SnowBoy[2] の構築をおこなっている. SnowBoy では, 3D グラフィック作成し, 作成した 3D グラフィックスに対しプログラムを付与することで想像力や論理的思考能力を高めることを目的としている. SnowBoy を用いてオブジェクトやプログラムを編集している児童らが, 行き詰まりを覚えたときに, 教員が小型情報通信端末を用いて児童らが作成したオブジェクトやプログラムを取得して共同作業に加わり手元で操作をすることにより, 的確な助言や作業の手本を示すことが可能になると考えられる.

このように, 本手法を適用することで, 小型情報通信端末を用いて支援者がグループへ中途参加をし, 共同作業を支援するアプリケーションの構築が可能になる.

5. おわりに

本稿では, 小型情報通信端末による共同作業支援をおこなうため, 中途参加時におけるミラーオブジェクトの同期を可能にする P2P 型オブジェクト複製環境の提案をおこなった. 今後は, この手法を適用した共同作業支援アプリケーションを製作し, その有用性を確かめるために実験および検証をおこなっていく予定である.

参考文献

[1] Shogo Noguchi, Hideyuki Takada: “CUBE: A Synchronous Collaborative Applications Platform Based on Replicated Computation”, Proceedings of The Fifth International Conference on Collaboration Technologies (CollabTech 2009), 2009.

[2] 柿内達真, 取越翔太郎, 桜打彬夫, 大東和忠幸, 野口尚吾, 高田秀志: “SnowBoy:教室内でのプログラミング作品共有による共同創作が可能な初等教育向け協調学習支援システム”, 情報処理学会研究報, 2009-GN-72, Vol.2009, No.2, 2009.