

## 共同作業ツール作成支援のための P2P フレームワークの開発

大内 雄矢 安井 浩之 松山 実

武蔵工業大学

## 1 まえがき

オンライン共同作業ツールは、通常はクライアントサーバ(C/S)モデルで構築、運用されるが、サーバの導入、管理には多大なコストがかかり、少人数では手を出しにくい。そこで、サーバを設置しなくても済む P2P を用いることで、遠隔地同士の少人数の集まり等でも気軽にブレインストーミングや、デバッグ等の共同作業ツールが利用できるようになると期待される。本報告では、共同作業ツールの作成を支援する P2P のフレームワークの提案と開発について述べる。

## 2 フレームワーク概要

Skype 社が無料で提供している Skype API A2A を使用することで、Skype の P2P ネットワークを利用したアプリケーションの作成が可能となる。そのため、ここではオンライン共同作業ツールの作成を容易にするためのフレームワークを、A2A を使用して開発することを目的とする。これにより、ネットワークを利用した情報収集や Skype の会議通話を利用した共同作業ができるようになり、作業効率が向上する。本フレームワークは Skype とアプリケーション間の通信や、後述する競合の回避、作業中断の防止機能を含む。

本フレームワークは A2A の仕様により、以下のような制約が発生する。

- ①Skype のインストールが必要[1]
- ②最大同時接続人数：25 人(Skype の会議通話の最大接続人数)[2]
- ③切断までの無通信時間：約 8 分[1]
- ④アプリケーション名：32 バイト[1]
- ⑤通信相手がコンタクトリストに登録されている必要がある[1]

これらの中、共同作業の障害になりえるのは②最大同時接続人数と⑤コンタクトリストである。しかし、②は多くても十数人と想定しているので本フレームワークの仕様には十分な人数であり、⑤のコンタクトリストに登録する作業も少人数の利用であれば時間もかからず、大きなデメリットにはならない。

## 3 競合とネットワークの再構築

オンラインで共同作業を行う際、複数のユーザから更新要求がくることがある。そこで、ユーザの一人をホストとし、ホストがその他のユーザであるゲストを、以下のように制御することで対応する。

データの更新をユーザ全員に対して行う際、全ての更新はホストからゲストに送信し、反映させる。ゲストからのデータを更新する場合は、一度ホストに更新を反映させてからゲストに送信する。(Fig.1)

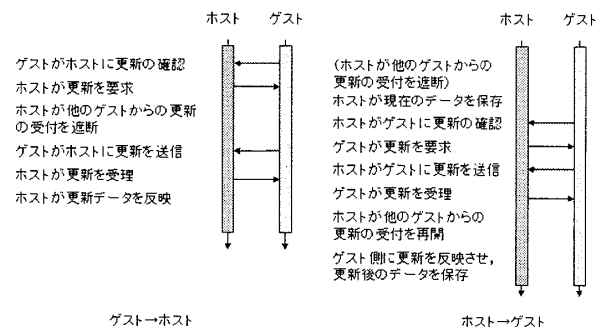


Fig.1 データの更新の流れ

更新の競合が起こらないよう、更新を送信する際には他のゲストからの更新を全て遮断する。なお、図中の「データの保存」とは、作業を前の時点に戻すことができるように更新履歴を保存しておくものである。

本フレームワークでは、ホストが不慮の事故で切断された際に作業が中断されるのを防ぐために、次のホストとなるゲストを予め決めておき、ホストが切断されても作業の続行を可能とする。

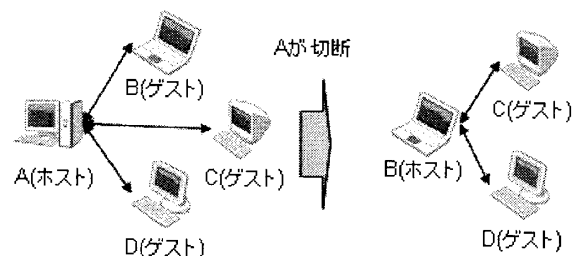


Fig.2 A(ホスト)が切断した場合

ホストが Skype のネットワークから切断されると、予め決められた次のホストとなるゲストが参加していた他のユーザを誘いなおすことで、Fig.2 のようにネットワークを再構築し、ホストが切断された際の作業中断を防ぐ。ネットワーク形態はスター型である。

#### 4 C/S 型との比較

導入コスト、データの管理、対障害性において C/S 型と P2P 型の本方式を比較する。

Table1 C/S モデルと本方式の比較

	C/S 型	本方式
導入コスト	×	○
データ管理	○	△
対障害性	△	○
導入の手間	○	△

(○：良い，△：どちらともいえない，×：悪い)

C/S モデルでは、サーバの導入コストが不可欠であるが、本方式では P2P を使用しているので不要である。作業履歴のデータの管理は、サーバで全てのデータを一括管理する C/S モデルに対し、サーバを持たない本方式では更新の競合を防ぐためにホストを通して更新を反映させているが、その更新履歴は全てのユーザのローカルに保存しなければならないため、共同作業ツールの開発者がその管理を考慮しなくてはならない。耐障害性は、サーバがダウンするとシステムが使えなくなる C/S に対し、本方式ではホストがダウンしても別のゲストがホストになることで作業を続行することが可能である。最後に導入の手間は、本方式は Skype のインストールが必須となっており、その分の手間がかかる。

以上から、本フレームワークは、導入コストが会議通話で利用するマイクのみであり、対障害性も優れているといえる。しかし、データ管理は C/S モデルに劣り、手間がかかる。大量のデータ管理や、大人数での作業には向かず、少人数でのコストをかけないで安定した作業を行うのに向いている。

#### 5 評価

本フレームワークを利用して作成したチャットプログラムを用いて以下の環境 (Table2) でテストを行った。A, B, C それぞれがホストとなった時、ゲストからホストを通して他のゲストに「あいうえお」というデータが反映されるまでの時間を測定した。6 パターンの送受信をそれぞれ 20 回ずつ行い、平均値を求めた結果を Table3 に示す。

Table2 マシンスペック

	CPU	RAM	通信速度
PCA	1.6GHz	1.23GB	54Mbps
PCB	2.01Ghz × 2	2.00GB	100Mbps
PCC	2.40GHz	512MB	11Mbps

Table3 測定結果

経路	A→B→C	A→C→B	B→A→C
時間(s)	0.27	0.23	0.22
経路	B→C→A	C→A→B	C→B→A
時間(s)	0.30	0.26	0.28

どのルートでも平均 0.3 秒以内に送受信を行えることが確認できた。この速度は Skype での会議通話の遅延も考慮すると共同作業に影響を与えない範囲の値である。しかし、本フレームワークは Skype の P2P ネットワークを利用しているので一定の通信速度を保つことは不可能と考えられ、通信速度が変動する可能性がある。

また、本フレームワークを利用してサンプルプログラムを作成したところ、約 220 行に収めることができたのに対し、利用していないものは約 400 行になった。

#### 6 まとめ

本フレームワークを使用することで、少人数での使用を目的とした共同作業ツールの開発を容易にすることが可能になる。P2P ネットワークを使用しているため、C/S 型と違い高いコストも必要なく、Skype の会議通話を同時に使用することにより、キーボードで文字を打つよりも情報の交換が速く的確に行えることを実感することができた。

ただし、本フレームワークは作業履歴のデータの管理を行っておらず、共同作業ツールの開発者にゆだねられるので、バックアップの支援などの改良が必要である。

#### 参考文献

- [1] Japanese Developers Forum  
API Book Project 監修  
石田真一, rゆ/xai, 池島俊, 大谷弘喜,  
須藤雅道, 寺田亮, 谷萩毅之, 山田達也  
Skype API Book Vol.1, pp.223~227,  
Talpa-Tech, 2006
- [2] Skype 公式サイト  
<http://www.skype.com/intl/ja/welcomeback/>