

# 操作変換を用いた双方向変換網の競合解決における 操作変換制御の一検討

仲野 祐希<sup>†</sup>

法政大学情報科学研究科<sup>†</sup>

日高 宗一郎<sup>‡</sup>

法政大学情報科学部<sup>‡</sup>

## 1 導入

双方向変換とは、2つもしくはそれ以上の情報源の間で変換を介して一貫性を維持する仕組みのことである。多数の情報源の間で同期を取る場合、図1のように自然な双方向変換の結合方法として、(a)逆方向変換である *put* が直列に並ぶ場合、(b) *put* が向き合う場合の2通りの場合が考えられるが、(b)の場合は、競合が不可避である。土生ら [1] は、操作変換を応用したアルゴリズムを適用することで2つのビューの更新内容をどちらも考慮したデータ更新ができる双方向変換を実現しているが、3つ以上の更新に現実的な操作変換で対処するには操作変換の適用方法の制御が求められる。本研究では3つ以上のビューの更新内容を考慮した双方向変換を実現することを目的とする。

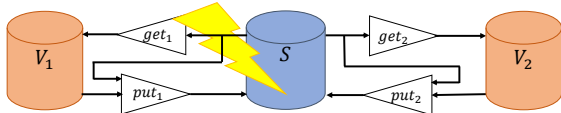


図1: 多数の情報源の間での *put* が向き合う双方向変換の例

近年、多数の双方向変換を組み合わせた多方向変換について研究されている。本研究では、図2のような、双方向変換が辺となるネットワークを想定している。ソースやビューはネットワークを介した多数のサーバーであると考えられる。

### 1.1 操作変換

文字列を対象とした操作変換 [2] では、挿入、削除といった操作を引数とする変換関数  $T$  により、互いの文字列が一致するような編集操作を算出することで、共同編集文書の一貫性を保つことができる。操作変換の主な特性として、データを更新する主体の数に応じて  $C_1$  と、 $C_2$  がある。 $C_1$  では2つの場合に、 $C_2$  では3つ以上の

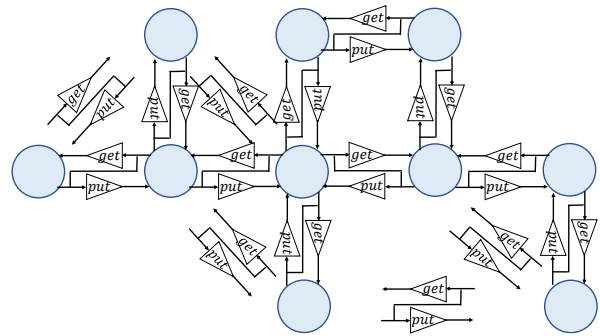


図2: 本研究で想定する双方向変換網

場合も、適用順によらず編集結果が等しくなる。図3は、共同編集ソフトウェアで起こる操作変換の過程を例示したものであり、本研究で利用する特性  $C_1$  に対応する。

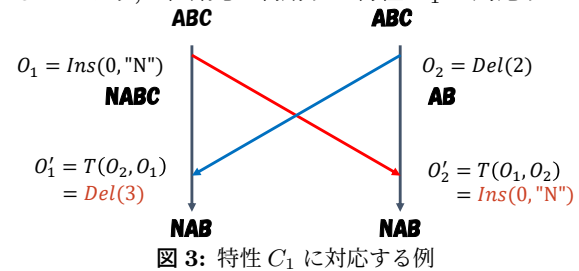


図3: 特性  $C_1$  に対応する例

## 2 関連研究

一般的な双方向変換では、ソースとビューは1対1の関係であるが、土生ら [1] はソースとビューが1対2の場合に対応している。操作変換を利用することにより、2つのビューの更新内容が異なる場合においても、どちらの更新も考慮したデータ伝播を可能にした。実装の実行結果から、X言語の特性 GETPUTGET, PUTGETPUT, 操作変換の特性  $C_1$  を満たすことが確認されている。

仲野ら [3] は土生らの実装を三者の場合に拡張し、競合解決を行っている。Randolphら [4] が挿入・削除といったシンプルな操作であっても、特性  $C_1, C_2$  をともに満たすことは不可能であると指摘しているため、 $C_2$  に頼らず、 $C_1$  と、操作変換適用の制御という立場のもと、三者での競合解決を実現している。仲野らはXML形式のファイルに対する三者による更新適用を行う評価実験を行った。操作変換により競合解決が行われ、双方向変換の特性である PUTGETPUT, GETPUTGET を満たしていること、期待した操作変換結果が得られ

A study on Controlling Operation Applications in Conflict Resolution of Bidirectional Transformation Network based on Operational Transformation

<sup>†</sup> Yuki Nakano Graduate School of Computer and Information Sciences, Hosei University

<sup>‡</sup> Soichiro Hidaka Faculty of Computer and Information Sciences, Hosei University

ることを実験により確認しているが、操作変換の制御法は具体化されていなかった。本研究では、Hyperledger Fabric を用いた実装の実現方法の具体化に取り組む。

### 3 本研究の特徴

本研究では Hyperledger Fabric[5] を実装に利用している。Hyperledger Fabric を利用することで、Peer to Peer 型のネットワークを想定し、特定の参加者間で異なるデータベースを持ち、共同編集文書に改ざん耐性が求められるような状況へ対応することができる。

### 4 操作の適用順

ここでは2節で述べた、特性  $C_1$  のみが満たされている場合に、三者以上がデータを更新すると、編集結果は操作の適用順に依存してしまうという問題について詳しく見ていく。図4では、頂点が状態、辺は操作を表しており、辺をたどる経路は操作の適用順を示している。図中では操作変換の特性は以下のように表現される。特性  $C_1$  のみが満たされている場合、2つの操作が合流し、面となる。さらに特性  $C_2$  が満たされている場合、3つの操作が合流し、直方体となる。

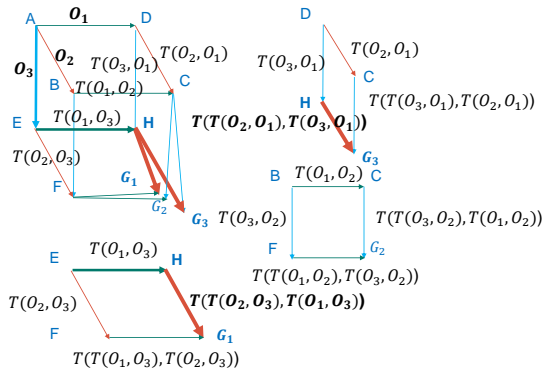


図4: TP1 のみを満たした場合の三者での操作変換

双方向変換として扱うことを考えると、頂点 A の状態をソースの初期状態、D,B,E を3つの異なる更新反映後のソース、それぞれ A との差分を  $O_1, O_2, O_3$  として、 $G_1, G_2, G_3$  のいずれかの頂点に向かわせることとなる。

例えば、Orderer が  $O_3 \rightarrow O_1 \rightarrow O_2$  のように操作変換の適用順を決定したとする。その場合、 $G_1$  への経路： $O_3; T(O_1, O_3); T(T(O_2, O_3), T(O_1, O_3))$ 、 $G_3$  への経路： $O_3; T(O_1, O_3); T(T(O_2, O_1), T(O_3, O_1))$  の2通りの異なる結果をもたらす2つの経路が考えられ、結果が1つに収束しない。この問題に対応するためには、Orderer が単純な操作適用順序決定だけでなく、操作自体も交換する、もしくは受け取った順序をもとに Peer ごとに解釈を変え、最終的に一意に決定する必要がある。

### 5 操作変換制御

実装環境として、以下のような3つの Organization からなる、Hyperledger Fabric Network を構築する。Peer から受け取ったトランザクションの順序を決定してブロックを作成し、接続している Peer にブロックを配布する、Ordering Service を利用して、操作変換の制御を行うことを想定している。現段階では、競合する3つのトランザクションをほぼ同時に発行し、Ordering Service に送信することが可能となった。

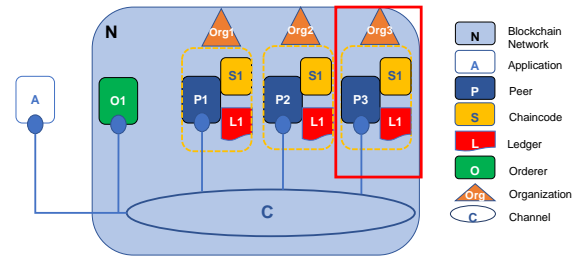


図5: 3つの Org を持つ Hyperledger Fabric Network の構築例

Hyperledger Fabric 公式ドキュメント addOrg3 [6] をもとに構成

### 6 評価実験方針とまとめ

三者以上により、同時刻に同じ XML 要素に対し更新を行った場合、検討した操作変換が期待する結果と本研究の提案手法により得られた結果が一致しているか確認する。本研究では、Hyperledger Fabric の持つ Ordering Service を用いた制御を試み、多数の更新が競合する共同編集ソフトウェアを想定した評価実験結果を行う予定である。

### 参考文献

- [1] M.Habu and S.Hidaka. Conflict resolution for data updates by multiple bidirectional transformations. In G.Fletcher, K.Nakano, and Y.Sasaki, editors, *Software Foundations for Data Interoperability*, pp. 62–75, Cham, 2022. Springer International Publishing.
- [2] S.Sergey, et al. Verified operational transformation for trees. In *International Conference on Interactive Theorem Proving*, pp. 358–373. Springer, 2016.
- [3] Y.Nakano and S.Hidaka. 双方向変換網における操作変換による競合解決手法. IPSJ 全国大会 22, pp. 273–274, 2022.
- [4] A.Randolph, et al. On consistency of operational transformation approach. *Electronic Proceedings in Theoretical Computer Science*, Vol. 107, p. 45–59, Feb 2013.
- [5] E.Androulaki, et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proc of EuroSys'18*, p. 15, 2018.
- [6] Hyperledger fabric 公式ドキュメント. [https://hyperledger-fabric.readthedocs.io/ja/latest/channel\\_update\\_tutorial.html](https://hyperledger-fabric.readthedocs.io/ja/latest/channel_update_tutorial.html). 取得: 2023/1/12.