卒業論文

ブロックチェーンに基づく
非中央集権型辞書システム

指導教官　　村上 陽平 准教授

立命館大学情報理工学部
先端社会デザインコース 4 回生
2600170514-7

張 禹王

2020 年度 秋学期 卒業研究 3(CH)
令和 3 年 1 月 31 日

ブロックチェーンに基づく非中央集権型辞書システム

張 禹王

内容梗概

政府の同化政策，共通語などのマジョリティ言語の普及力強化などに伴い，少数民族言語などのマイノリティ言語の使用率の低下が著しい．現在，国際連合教育科学文化機関によると世界で 2464 の言語が消滅の危機にあるとされている．このようなマイノリティ言語を保護し，言語の多様性を保つために，多くの人々が危機言語の辞書データの生成に貢献することが望まれる．これまでの多言語の言語資源の成功例としては Wikipedia がある．

Wikipedia は著作権の問題をクリエイティブコモンズライセンスによって解決し，多くの人々の協働を可能とした．しかしながら，同じ枠組みの辞書作成を目的とした Wiktionary は成功しているとは言いがたい．その原因は，Wikipedia に働く「記事作成の楽しさ」というモチベーションが，Wiktionary には働かないためと考えられる．例えば，作成に多くの労力を要する辞書は，内容が単なる意味や事実のため著作権で保護されるような創作性がなく，クリエイティブコモンズライセンスのため経済的代償も確約されない．

そこで，本研究では，辞書データの利用量に応じて作成者が利用者から直接報酬を分配することができるシステムを提案する．具体的には，データの利用量に応じて作成者と評価者に報酬を分配する．ブロックチェーンを用いることで，利用者と作業者の間に立つ辞書サービス提供者を排除し，利用者と作業者の直接のやりとりを可能とする．本提案手法を実現にあたり，取り込むべき課題が以下の２点である．

作業履歴の高信頼データ管理

利用量に応じて報酬を直接に作業者に分配するために，辞書の訳文データから作成者，評価者，利用者を正引き，逆引きできるようなデータモデルは必要がある．さらに，報酬分配の根拠となる辞書の訳文データの作成情報，利用履歴などのデータが改ざん耐性を持つシステムも必要である．

報酬分配の透明性

ブロックチェーンに記録された作成，評価履歴に基づいて自動的に報酬額を計算し分配するメカニズムが必要である．このメカニズムは作業者の作業モチベーションを妨げないように，報酬をもらえる作業者ともらえるべき報酬額を

誰でも確認できるよう透明性を確保した手段で実現する必要がある．

前者の課題に対しては，作成履歴，評価履歴，利用履歴の三つの中間テーブルを導入し，辞書の訳文データの作成者，評価者，利用者をユーザとして，辞書の訳文データとの間に多対多のリレーションシップを設定する．ユーザが辞書の訳文データを作成，評価，利用する際に対応している中間テーブルに履歴を記録する．そして，報酬分配の根拠となる履歴情報が改ざんされないようにブロックチェーン上に記録する．

後者の課題に対しては，辞書の訳文データの作成者と辞書の訳文データの正誤を評価した評価者をその辞書の訳文データの貢献者として設定する．各貢献者の貢献度に応じて報酬額を計算する．訳文データが利用される際に利用者から利用料を引いて，貢献者に報酬を分配する．報酬をもらえる作業者ともらえるべき報酬額を誰でも確認できるよう透明性を確保するには，報酬額を計算するロジックと報酬を分配するロジックをスマートコントラクトとしてブロックチェーンに記述する．このようなことに通じて，ブロックチェーンネットワークに参加する全ピアノードが報酬の計算，分配のロジックを確認し，許可することが可能となる．本研究の貢献は以下の通りである．

作業履歴の高信頼データ管理

利用量に応じて報酬分配するためのデータモデルの設計を行い，改ざん耐性を持つシステムを構築した．辞書の訳文データが作成，評価，利用される時に履歴も残され，報酬分配の根拠になる．データ数に応じたトランザクションの性能評価を行った．

報酬分配の透明性

報酬を計算，分配するアルゴリズムを決め，訳文データが利用されるたびに自動的に報酬を分配できるメカニズムの実装を行なった．なお，このメカニズムの実装によって，訳文データが利用される際にリアルタイムで報酬分配することも可能になった．

Blockchain-based Decentralized Dictionary System

ZHANG Yuwang

Abstract

The utilization rate of crisis languages such as minority languages is decreasing remarkably with the government's assimilation policy and the popularization of majority languages such as common languages. At present, according to the United Nations Educational, Scientific and Cultural Organization, 2464 languages are in danger of disappearing. In order to protect such crisis languages and to keep the diversity of languages, it is desired that many people contribute to the generation of dictionary data of crisis languages. One successful example of a multilingual language resource to date is Wikipedia.

Wikipedia solves the problem of the copyright by the Creative Commons License, and it enables the cooperation of many people. However, Wiktionary, which aims to create a dictionary with the same framework, has not been successful. The reason seems to be that the motivation of "Fun of writing articles" working in Wikipedia does not work in Wiktionary. For example, a dictionary which requires a lot of labor for its production has no creativeness such that its contents are protected by copyright because of mere meaning and fact, and no economical compensation is promised because of a creative commons license.

In this paper, I propose a system that enables the creator to distribute the reward directly from the user according to the amount of dictionary data used. Specifically, rewards are distributed to the creator and evaluator according to the amount of data used. By using the blockchain, the dictionary service provider which stands among the user and worker will be eliminated, and the direct communication among the user and worker will be enabled. In the implementation of the proposed method, the following two problems should be taken in.

Highly reliable data management of work history

In order to distribute the reward directly to the creator and evaluator according to the amount of data used, it is necessary to develop a data model that can detect the creation, evaluation, or use of the translated data. In addition, a system is required in data are not tampered.

Transparency of reward allocation

A mechanism is needed to automatically calculate and distribute the reward amount based on the creation and evaluation history on the blockchain. This mechanism needs to be realized by a transparent means so that anyone can confirm the amount of compensation that should be received by the worker who receives the reward so as not to hinder the work motivation of the worker.

For the former problem, three intermediate tables of creation history, evaluation history, and usage history are introduced, and a many-to-many relationship is set between translation data of the dictionary, creator, evaluator and user. Data added to the intermediate table when a user creates, evaluates or uses translation data. Then, it is recorded on the block chain so that the history information is not falsified.

For the latter problem, the creator and the evaluator are set as contributors of the translation data. The reward amount is calculated according to the degree of contribution of each contributor. When the translation data is used, the fee is deducted from the user and the reward is distributed to the contributors. To ensure transparency so that anyone can see who gets reward and how much should get, the logic to calculate the reward amount and the logic to distribute the reward are described in the blockchain as smart contracts. Through this, all the peer nodes participating in the blockchain network can confirm and permit the logic of calculation and distribution of the reward. The contribution of this research is as follows.

Highly reliable data management of work history

Designed a data model to distribute the reward according to the amount of data used, and constructed a system with falsification resistance. The history will be record, when translation data of the dictionary are created, evaluated, and used, and it becomes a basis of the reward distribution.

Transparency of reward allocation

An algorithm to calculate and distribute the reward was decided, and a mechanism to automatically distribute the reward every time when translated data was used. In addition, a real time reward distributing system was implemented in this mechanism.

ブロックチェーンに基づく非中央集権型辞書システム

目次

第1章　はじめに

現在，国際連合教育科学文化機関によると世界で 2464 種の言語が消滅の危機にあるとされている．これらの消滅の危機に迫っているマイノリティ言語の保護支援を行い，言語の多様性を保つために多くの人々が危機言語の辞書データの生成に貢献することが望まれる．これまでの多言語の言語資源の成功例としては Wikipedia がある．

Wikipedia は著作権の問題をクリエイティブコモンズライセンスによって解決し，多くの人々の協働を可能とした．本稿を執筆する時点に Wikipedia に 298 種類の言語の言語資源が貢献されている．しかしながら，同じ枠組みの辞書作成を目的とした Wiktionary は本稿を執筆する時点に 180 種類の言語の辞書しか作成されなく，成功しているとは言いがたい．その原因は，Wikipedia に働く「記事作成の楽しさ」というモチベーションが，Wiktionary には働かないためと考えられる．例えば，作成に多くの労力を要する辞書は，内容が単なる意味や事実のため著作権で保護されるような創作性がなく，クリエイティブコモンズライセンスのため経済的代償も確約されない．

本研究では，より多くの人が多言語の辞書データの生成に貢献するために，辞書データの利用量に応じて作成者が利用者から直接報酬を分配することができるシステムを提案する．ブロックチェーンを用いることで，利用者と作業者の間に立つ辞書サービス提供者を排除し，利用者と作業者の直接のやりとりが可能とする．この提案手法を実現するにあたって，以下の課題に取り込む必要がある．

作業履歴の高信頼データ管理

利用量に応じて報酬を直接に作業者に分配するために，辞書の訳文データから作成者，評価者，利用者を正引き，逆引きできるようなデータモデルは必要がある．さらに，報酬分配の根拠となる辞書の訳文データの作成情報，利用履歴などのデータが改ざん耐性を持つシステムも必要である．

報酬分配の透明性

ブロックチェーンに記録された作成，評価履歴に基づいて自動的に報酬額を計算し分配するメカニズムが必要である．このメカニズムは作業者の作業モチベーションを妨げないように，報酬をもらえる作業者ともらえるべき報酬額を誰でも確認できるよう透明性を確保した手段で実現する必要がある．

本稿の残りは以下のような構成となっている．第二章で非中央集権型辞書システムにおけるブロックチェーンの意義とブロックチェーンの種類を紹介する．第三章で非中央集権型辞書システムの概要，構成，データモデルの設計およびシステム内の各処理の振る舞いについて具体的に説明する．その後に，第四章でこの辞書システムの性能評価を行い，第五章で本稿をまとめる．

第2章　ブロックチェーン

この章では，ブロックチェーンの概要と本システムにおけるブロックチェーンの意義について説明する．その後，本システムの機能要件を満たす視点から，ブロックチェーンの各種類について説明する．

2.1　ブロックチェーンの意義

ブロックチェーンはブロックチェーンネットワークの参加者（ピアノード）にデータベースを分散して共有する分散型台帳システムである．従来の分散型台帳システムと異なって，合意形成アルゴリズムでデータの正当性を相互に保証することで中央集権的な管理主体を必要としないことができる．

図 1 のようにブロックチェーンでは，ネットワークに送信されたデータと一個前のブロックのハッシュ値などで次のブロックを生成し，ブロックチェーンを形成される仕組みになっている．つまり，ブロックを生成する毎に，一つ前のブロックの補強するのである． [1]

そのため，任意のピアノード上のデータが改ざんされると，ネットワークに全ピアノード間のデータの一貫性が保てなくなることから，データの改ざんは極めて困難である．

また，従来の分散型台帳システムは中央集権的な管理者がデータの同期などの管理を行う．それに対して，ブロックチェーンでは合意形成アルゴリズムでネットワークの参加するピアノードの間に相互の証明と承認をもって，データの一貫性と真正性を担保する．

従って，報酬分配の根拠となる辞書の訳文データの作成情報，利用履歴などが改ざんされることを防げる上に，特定の管理者機関に負荷を集中するリスクによってシステムの信頼性を下げることも防げる．

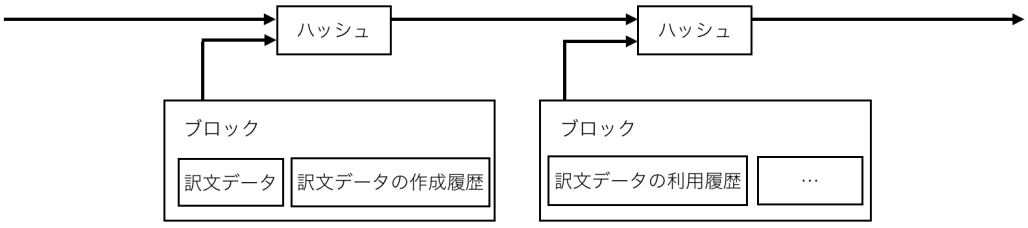

図 1: 一個前のブロックのハッシュ値で次のブロックを生成

そして，ブロックチェーンにスマートコントラクトが備えている．スマートコントラクトはブロックチェーンネットワークの外部にあるクライアントアプリケーションによって呼び出され，トランザクションを通してブロックチェーン上に特定なロジックを自動的に実施するコードである．[1)] 一部のブロックチェーンのディストリビューション[2)]のスマートコントラクトは「チューリング完全」と呼ばれて，ループを含む任意のプログラムを実行することも可能である．

辞書の訳文データの登録，評価，利用のロジックをスマートコントラクトとしてブロックチェーンに記述することでデータの作成履歴，評価履歴，利用履歴が正確かつ確実に作成されることを保証できる．

2.2 ブロックチェーンの種類

ブロックチェーンには大きく分けると，誰でも参加することができるパブリック型と特定の複数または一つの組織から構成されるプライベート型/コンソーシアム型が存在する．

2.2.1 パブリック型

この型の最も代表的なブロックチェーンシステムがビットコインシステム [1] である．ビットコインシステムからはじめ，多くのパブリック型ブロックチェーンシステムは PoW（Proof-of-Work）[1] または PoW から派生したアルゴリズムで合意形成する．

PoW の役割としてはブロックのハッシュを取得する際に，最初の n ビットが全て 0 で始まるハッシュ値を見つかるまで，ブロック内の Nonce を変化させ，試し続ける．要求されたハッシュ値を見つかるまでの計算量（work）は 0 のビット数に応じて指数関数的に増加する．その一方，ハッシュ値を一回計算すればその計算量の証明（proof）を検証することが可能である．PoW アルゴリズムによる合意形成では，最初に発見された Nonce を内蔵するブロックが新たにブロックチェーンに接続される．Nonce の最初の発見者に経済的なインセンティブを与えることでブロックが次々と生成される．インセンティブをもらえるための計算行為が金などの貴金属を採掘することにたとえて「マイニング」と呼ばれる．マイニングを行うピアノードが「マイナー」と呼ばれる． [2]

[1)] 「Glossary (用語集) — hyperledger-fabricdocs master ドキュメント」https://hyperledger-fabric.readthedocs.io/ja/latest/glossary.html#smart-contract (最終検索日: 2021/01/31)

[2)] 例えば，Ethereum と Hyperledger Fabric

インセンティブをもらえるために各マイナが競争的にマイニングを行うのため，合意形成されるまで複数の Nonce が発見されることもありうる．その結果，ブロックチェーンが図 2 に示すように複数のチェーンに分岐され，互いに競合する．

そこで，PoW の計算量が最も使用されたチェーンを最長チェーンと決め，ネットワークに参加する全ピアノードの意思決定の代表になる．各ピアノードが常に最長チェーンを正と判断し，マイニングし続ける．チェーンが競合する時に，短い側のチェーンに取り込んでいたノードは長いチェーンに切り替えることになる． [1]

そのため，トランザクションが決定されたかどうかを確認するにはそのトランザクションを含むブロックが属しているチェーンが最も長いチェーンになるまで待てなければならない. 例えば，ビットコインの場合には約一時間後となっている [1)]

辞書システムを通じて辞書引きする際に，利用者を長時間を待たさせるのはあまりにも非現実的だと考えられる．辞書システムがパブリック型ブロックチェーンに基づく場合にはトランザクションの処理時間が辞書システムのネックになる．

また，悪意を持つ攻撃者が制御する CPU パワーが全部の善意のピアノードを持つ計算力より上回る時に，ブロックチェーンに格納されたデータの正確さを保証できなくなり，データが改ざんされるリスクもある．

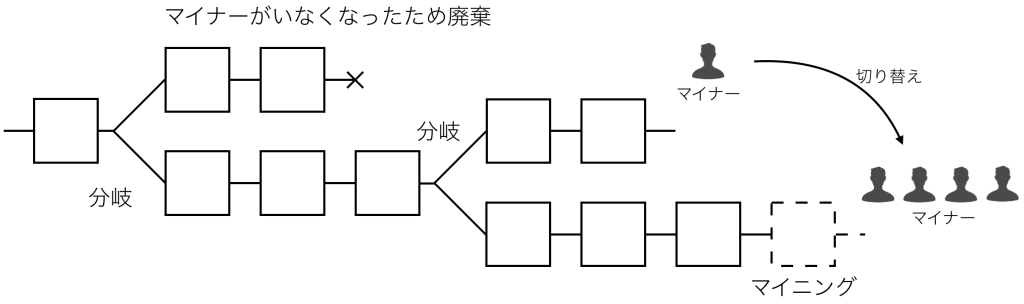

図 2: ブロックチェーンの分岐

[1)] 白石 善明 「ブロックチェーンの合意形成アルゴリズム」 [2]（電子情報通信学会 通信ソサイエティマガジン Vol.14，No.1，2020 年，ページ 23）より

2.2.2 プライベート型/コンソーシアム型

プライベート型/コンソーシアム型ブロックチェーンの特徴はパーミッションドである．許可を与えられた組織あるいは個人しか参加させない．

このタイプのブロックチェーンの代表の一つと言える Hyperledger のドキュメントによると「許可型のブロックチェーンは、 既に知っていて素性のわかっている (多くの場合、深く吟味された) 参加者によって、ある程度の信頼をおく管理モデルで運用されます。 許可型のブロックチェーンは、共通する目標をもちつつ、完全には互いを信用していないグループの中でのセキュアなやりとりを行う場を提供します。 参加者の素性が明らかであることで、許可型ブロックチェーンでは、コストのかかるマイニングを必要としない、 クラッシュ故障耐性 (CFT) やビザンチン故障耐性 (BFT) をもつ、従来から存在する合意形成プロトコルを使用することができます。

くわえて、許可型のブロックチェーンでは、参加者が、意図的にスマートコントラクトに悪意のあるコードをもたらすリスクも低くなります。 まず第一に、参加者は互いを知っています。 そして、アプリケーショントランザクションの発行、ネットワークの設定変更、スマートコントラクトのデプロイといった全ての行動は、 ネットワークとトランザクションの種類ごとに設定されたエンドースメントポリシーに従って、ブロックチェーンに記録されます。 このため、完全な匿名の場合と異なり、問題ある参加者は簡単に特定され、管理モデルの規定に従ってそのインシデントが処理されることになります。」 [3]

従って，ブロックチェーンネットワークの参加者（ピアノード）数が既知であり，参加者にある程度の信頼を持たれる．そのため，PoW ではなくより効率的な合意アルゴリズムで意思決定することが可能である．このタイプのブロックチェーンに基づいて辞書システムを構築する場合にはパーブリック型より高速な辞書作成，引きすることが可能となる．

その上，参加者が意図的にスマートコントラクトに悪意のあるコードをもたらすリスクを低くなることによって，スマートコントラクトに記述されている辞書の訳文データの登録，評価，利用のロジックの改ざん防止と信頼性向上に期待できる．

2.3　ブロックチェーンの応用

現在，多くの企業はブロックチェーンの「非中央集権」と「改ざん耐性」の特長を活かし，実証実験や応用を行なった．

1. **ウォルマート社と IBM 社による食品サプライチェーンの追跡**

   「食品の安全性確保」と「流通経路の透明性」を保障するために，ウォルマート社と IBM 社がブロックチェーン技術に基づく食品サプライチェーンの追跡の実証実験を行なった．

   ウォルマート社は 2016 年 10 月に中国市場で実証実験を始めた．豚肉の生産地から小売店舗に並ぶまでの仕入れルート情報をブロックチェーンに記録することで，豚肉に付けられた追跡コードを読み取ると数秒でトレースできる．[1)]

2. **アート市場におけるブロックチェーンの応用**

   贋作の混入は美術作品の信頼性担保や価値証明を損なう大きな問題である．そして，二次流通市場における作品の著作権管理などもアート市場の重要な課題である．

   スタートバーン社はこのような問題を解決するため，ブロックチェーン技術を活用して，アート作品の証明書発行や来歴管理、売買履歴や規約を管理するインフラ基盤を構築し，実社会に応用した．[2)]

非中央集権と改ざん耐性の特長があるブロックチェーンを適切に設計し，応用することで，複数の業界を横断し，コストを削減し，透明性とトレーサビリティを高める．上記の二つの例は追跡コードと証明書で実世界の物事とブロックチェーンに紐つくことで，豚肉と美術作品を管理，追跡可能になる．しかしながら，追跡コードや証明書の発行は第三者の個人または組織に依存しなければならない．システムの信頼性のボトルネックになる．

一方，辞書と例に挙げられた豚肉と美術作品と大きな違いがある．豚肉や美術作品はデジタル化すると価値が失うため，直接でブロックチェーンに格納できない．それに対して，辞書の場合には紙に印刷しても，デジタル化にしても伝わる情報が変わらないため，ブロックチェーンに格納できる．

---

1) 「ブロックチェーンで|生産から消費まで「食のサプライチェーン」を可視化する|IBM THINK」 https://www.ibm.com/think/jp-ja/business/food-trust/ (最終検索日: 2021/01/31)

2) 施井 泰平「ブロックチェーン技術のアート産業への応用可能性」，研究 技術 計画，Vol.34, No.4，pp. 367-376，2019.

辞書システムの信頼性を向上し，辞書を管理，追跡するために，辞書データもブロックチェーンに格納する必要がある．

そして，辞書システムには辞書データの利用量に応じて作成者が利用者から直接報酬を分配する．利用量を高速に確定するために，誰が，いつ，どの辞書を利用したかの利用履歴をブロックチェーンに残す必要がある．同様に，報酬をもらえる作成者を特定するためにも作成者，作成物などの情報を持つ作業履歴もブロックチェーンに残す必要がある．

第 3 章　非中央集権型辞書システム

第二章の検討を踏まえて，ブロックチェーンに基づく非中央集権型辞書システムをコンソーシアム型なブロックチェーンのネットワーク上に実行させる．

3.1　辞書システムの概要

辞書システムを利用できる個人ないし組織をユーザと定義する．そして，このシステムは作成者，評価者，利用者の三つのユーザロールがあると想定している．また，一つのユーザに複数のロールを割り当てえる．

辞書システムと各ユーザロールとの関係と対訳の作成・評価・利用の流れを

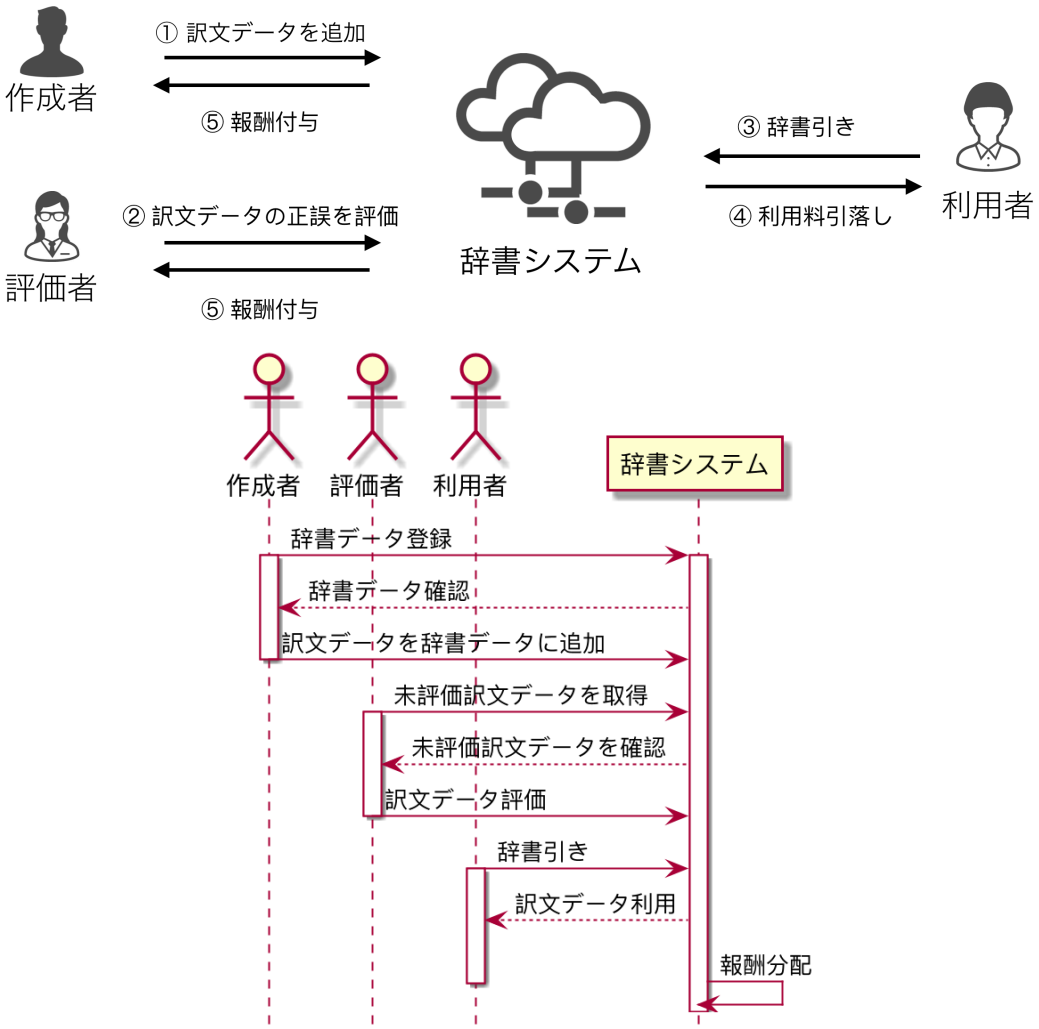

図 3: 辞書システムの概要

図 3 に示す.

3.1.1　作成者

このシステムでは辞書データと辞書の訳文データを作成し，システムに登録する作成者としてマルチリンガルを想定している．主たるユースケースとしては，システムに存在されていない辞書情報を作成し，システムに登録するケースとシステムに既存の辞書の下に訳文データを作成し，追加するケースと想定する．作成者が辞書に追加した訳文データが利用される際に，システムから報酬を付与されるケースもある．

3.1.2　評価者

不特定多数の作成者が作成された辞書では，作成者の能力にばらつきがあり，品質が保証されることが困難である．さらに，危機言語を含む複数の言語を話すことができる作成者はさらに限られるため，能力の高い作成者はあまり見込めない．辞書の品質を向上させるために，辞書の訳文データの正誤を評価する評価者のユーザロールを導入する．

ネイティブ以上のレベルで複数言語を話すマルチリンガルに評価者のユーザロールを付与することを想定している．評価者の主なユースケースとしては，自分が話せる言語の辞書に作成者が追加した訳文データを確認し，正誤を評価するケースと想定する．また，評価者が評価された訳文データが利用される際にも，システムから報酬を付与されるケースがある．

3.1.3　利用者

全てのユーザが利用者のユーザロールに割り当たられる．利用者が自分のニーズに応じて訳文，評価，作成者などあらゆるの条件で訳文データを検索し（辞書引き），利用するケースと利用者が利用された各訳文データに利用料を払うケースが想定されている．

3.2　システムの構成

実装の容易性を配慮して Hyperledger Fabric [1] という Linux Foundation が管理するオープンソースのブロックチェーン基盤を利用する．

Hyperledger Fabric を基盤とした辞書システムは図 4 のように構成されている．

[1] Hyperledger Fabric https://github.com/hyperledger/fabric

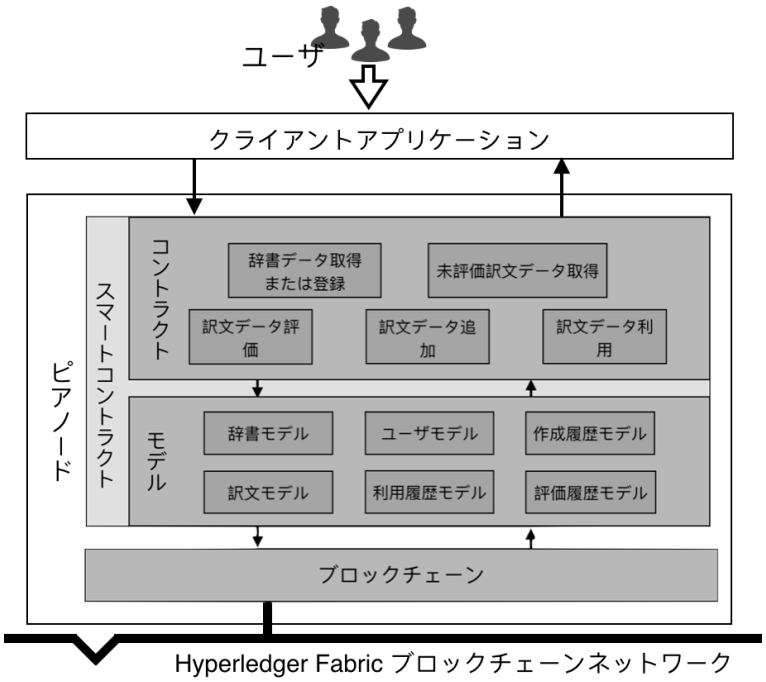

図 4: システムの構成

3.2.1 クライアントアプリケーション

ユーザはクライアントアプリケーションを通じて辞書の訳文データの追加，評価，利用などの操作を行う．辞書システムのユーザインタフェースである．

クライアントアプリケーションはブロックチェーンネットワークに参加するピアノードにトランザクション提案を送信することによってブロックチェーンネットワークの外部からスマートコントラクトを呼び出せる．ピアノードはスマートコントラクトを実行し，提案応答をクライアントアプリケーションに返す．

3.2.2 ピアノード

ブロックチェーンネットワークに接続されている各コンピュータ端末をピアノードと呼ぶ．ピアノードはブロックチェーンを保有する．ブロックチェーンにデプロイされているスマートコントラクトが呼び出される際に，スマートコントラクトを実行，検証する役割も持っている．

3.2.3 スマートコントラクト

このシステムが訳文データの利用量に応じて報酬分配するために必要なデータを定義，操作するロジックとユーザがシステムとやり取りするときにシステムの処理手順を定義するビジネスロジックがスマートコントラクトに記述する．

前文に言及したように，スマートコントラクトがブロックチェーンにデプロイするために，一定数のピアノードの承認が必要である[1)]．そのため，スマートコントラクトに記述されたロジックの透明性が保証される．

このシステムはスマートコントラクに記述されたロジックをモデルとコントラクトの二つの部分に分け，実現する．

コントラクト

コントラクトの部分は辞書，訳文，履歴情報などのデータに対する処理手順を示すビジネスロジックが記述される．辞書，訳文などのデータがユーザにアクセスされる時，確実に作成者，評価者，利用者を記録し，作成者と評価者が事前に合意した報酬分配メカニズムで報酬分配されることを保証する．詳細の振る舞いをこの章のトランザクションの節に論述する．

モデル

このシステムは関係モデルに基づいて辞書，訳文，履歴情報などのデータを定義，管理する．スマートコントラクトの中にブロックチェーンにデータを刻み込み，データのステートを更新するアプローチを抽出し，Active Record のデザインパターン [4] にしたがってモデルの部分に実装する．モデルの詳細をこの章のデータモデルの節に論述する．

このような設計で，ビジネスロジックが明白になり，ピアノードがスマートコントラクトを承認する段階にロジックの不正と不備が排除されやすくなる．システム全体の信頼性と透明性を向上する．

3.2.4 ブロックチェーン

スマートコントラクトと辞書，訳文，履歴情報などのデータを格納する場所である．データとスマートコントラクに決めたロジックの信憑性を保証する．

3.3 データモデル

このシステムは関係モデルに基づいて訳文データの利用量に応じて報酬分配するために必要なデータを定義し，組織化する．

まず，前文に言及したようにこのシステムのユーザは辞書を作成したり，辞書に訳文を追加したり，評価したり，利用したりすることをできる．必然的にユーザ，辞書，訳文の三つのエンティティが考えられる．

[1)] デフォルトでは過半数のピアノードの承認が必要となる．

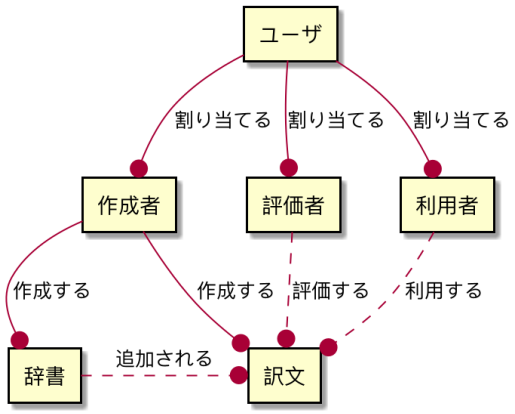

図 5: 各エンティティとエンティティ間の依存関係

そして，各ユーザロールに応じてユーザのユースケースも変わる．権限管理のために作成者，評価者，利用者の三つのエンティティも必要である．

ユーザは各ユーザロールを割り当てられる．作成者は辞書と訳文を作成する．作成された訳文が辞書に追加される．評価者と利用者は訳文を評価，利用できる．

各エンティティとエンティティ間の依存関係は図 5 のように表れている．

そして，訳文データの利用量に応じて報酬分配するため，利用者が訳文を利用する際に利用履歴を残すのが必要となる．訳文を作成した人と評価した人たちに報酬を分配するために訳文の作成履歴，評価履歴も残す必要がある．従って，利用履歴，作成履歴，評価履歴の三つの仲介エンティティも不可欠である．

以上のステープに踏まえて，図 6 のような概念モデルを定義できる．

次のステップはユースケースに満たすために，概念モデルに対してアトリ

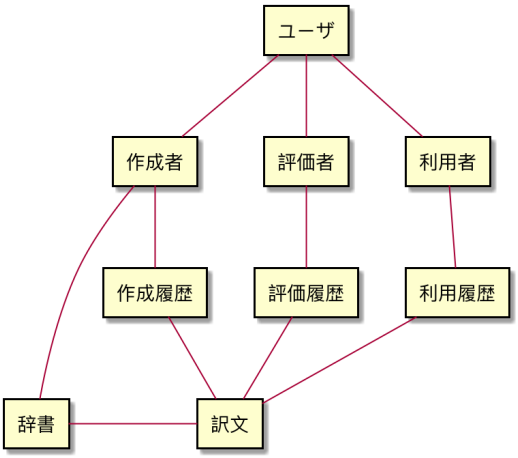

図 6: 概念モデル

ビュート，アイデンティファイア（主キー），外部キーの定義や、リレーションシップにカーディナリティといった要素を追加し，論理モデルを定義する．

まず，一人の作成者が複数の辞書を作成し，一個の辞書に複数の訳文を追加できる．一個の辞書または訳文に対して作成者が一人しかいない．

同様に一人の評価者は複数の訳文の正誤を評価できる．一つの訳文が複数の評価者に正誤を評価られる．利用者は辞書引きする際にも複数の訳文を利用する．一個の訳文が複数の利用者に使えられる．

そして，一つのユーザは複数のユーザロールに該当できる．作成者，評価者，利用者のユーザロールも複数のユーザに付与することが可能となる．

必然的に各リレーションシップに付けるカーディナリティを図 7 のように定義する．

次に，各エンティティのレコードを一意に識別するための UUID[1)] をアイデンティファイア（主キー）と定義する．親エンティティの主キーを子エンティティの外部キーと定義する．

ユーザが該当しているユーザロールに応じて利用量を払ったり，報酬をもらったりするケースがあるため，ユーザのエンティティ内にカレンシーがアトリビュートとなる．

また，評価者が訳文データを確認し，正誤を評価するケースがあるため，評価履歴のエンティティに評価者が評価した評価値のアトリビュートが必要となり [2)]，訳文データのエンティティに訳文データが正しいかどうかを表す評価値がアトリビュートとなる必要もある．

その他，辞書エンティティに辞書が対応する言語を説明するディスクリプションがアトリビュートとなり，訳文エンティティに評価値以外には元言語と訳語のアトリビュートも必要となる．

最終に得た論理モデルは図 7 である．

[1)] スマートコントラクトに非決定論的なロジックが記述されるとコンセンサス取れなくなるのため，UUID version 3 または UUID version 5 を利用する必要がある．

[2)] 履歴に作成時間をアトリビュートとして定義されるパータンはよく見られる．しかしスマートコントラクトの中に現在時刻を取得すると非決定論的なロジックになり，コンセンサスが取れなくなる．外部の「現在時刻」を参照すると時刻の信憑性が疑われる．その故，履歴に作成時間をアトリビュートとして定義しない．

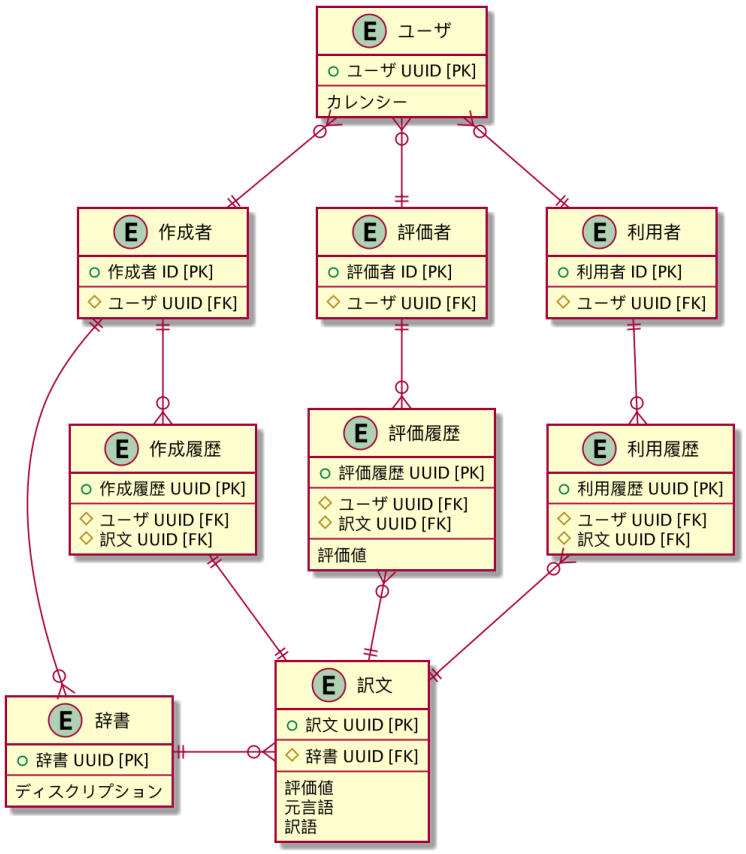

図 7: 論理モデル

3.4　トランザクション処理

トランザクションは，スマートコントラクトがクライアンドアプリケーションから呼び出され，データに対して登録，追加，利用などを行うときに作成され，ブロックチェーンに追加される [1)]．この小節はユーザがクライアンドアプリケーションを通じてスマートコントラクトを呼び出すから，トランザクションを作成され，トランザクション結果をユーザに返すまでのシステムの一連の振る舞いを説明する．

3.4.1　辞書データ取得または登録

作成者はシステムに存在しない辞書情報をシステムに登録するケースまたはシステムに既存の辞書に訳文データを追加するケースと想定される．そのため，システムは既存の辞書を引き出せるまたは新たな辞書を登録できる処理が必要

[1)] 「Glossary (用語集) — hyperledger-fabricdocs master ドキュメント」https://hyperledger-fabric.readthedocs.io/ja/latest/glossary.html#transaction (最終検索日: 2021/01/31)

となる．

作成者が辞書を取得または登録したい場合には，クライアンドアプリケーションを通じて，対応するスマートコントラクトを呼び出す．

スマートコントラクトの中には，まず辞書モデルを通じてブロックチェーンの中に目標辞書が存在しているかどうかを確認するトランザクションを発行する．目標辞書が存在した場合には目標辞書をインスタンス化して，クライアンドアプリケーションに返す．

目標辞書が存在されなかった場合には辞書モデルを通じて目標辞書のインスタンスを作成し，ブロックチェーンに格納する．辞書モデルは目標辞書のインスタンスをブロックチェーンに追加するトランザクションを発行する前に，目標辞書のインスタンスにバリデーションをかけ，辞書データの妥当性を確認する．その後に作成した目標辞書のインスタンスをクライアンドアプリケーションに返す．

最後にクライアンドアプリケーションは受信した辞書データを整形し，作成者に確認させる．

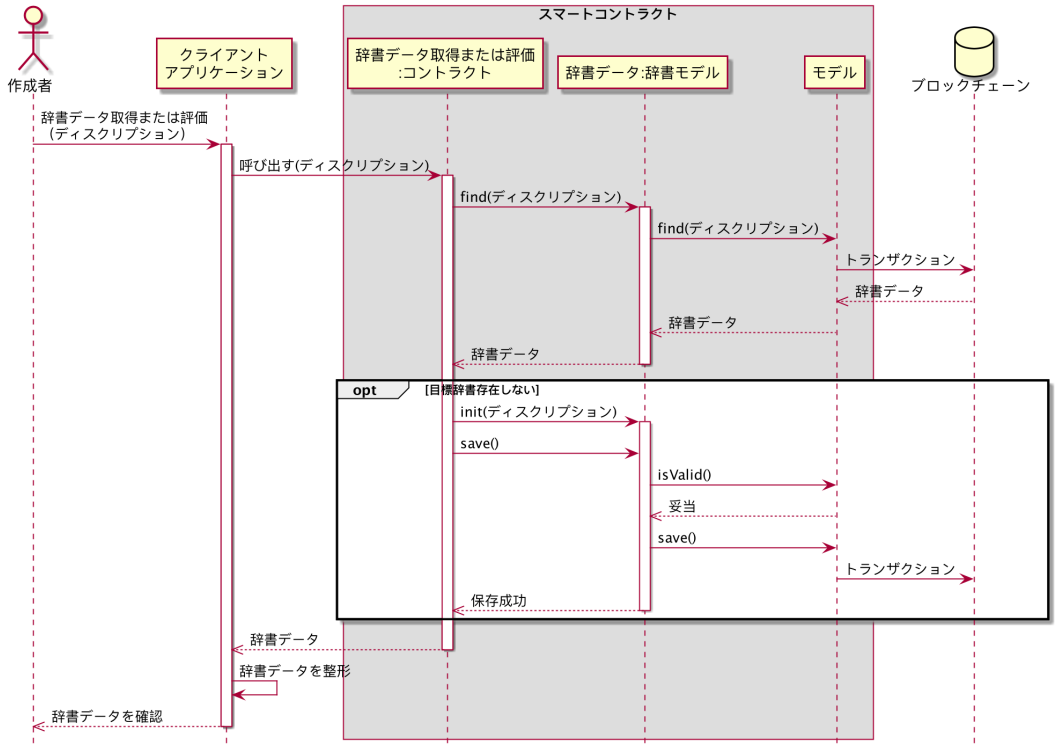

図 8: 辞書データ取得または登録のシーケンス図

図 8 は辞書データ取得または登録の時系列を表現するシーケンス図である．

3.4.2　訳文データ追加

作成者は訳文データを作成し，システムに既存の辞書に訳文データを追加するケースと想定される．また，その訳文データが利用される度に作成者に報酬分配する．そのため，システムは既存の辞書に訳文データを追加できる処理と訳文データが追加される同時に作成履歴を残す処理が必要となる．

作成者が既存の辞書に訳文データを追加するには追加先の辞書 UUID，元言語，訳語をクライアンドアプリケーションに渡し，クライアンドアプリケーションが渡されたデータを JSON 形式に整形し，整形されたデータを引数として訳文データ追加のスマートコントラクトを呼び出す．

スマートコントラクトの中には，まずユーザモデルを利用し，現在のユーザのインスタンスを取得する．同時に引数として渡されたデータを訳文モデルに渡し，訳文データのインスタンスを作成する．

現在のユーザのインスタンスと訳文データのインスタンスを取得した後に，現在ユーザインスタンスと訳データインスタンスを作成履歴モデルに渡す．作成履歴モデルは現在ユーザの UUID と訳文データの UUID を取得し，作成履歴インスタンスを作成する．

そして，訳文モデルと作成履歴モデルを通じて訳文データインスタンスと作成履歴インスタンスをブロックチェーンに格納するトランザクションを発行する．同様にトランザクションが発行される前に訳文データインスタンスと作成履歴インスタンスにバリデーションをかけ，データの妥当性を確認する．その後に訳文データインスタンスをクライアンドアプリケーションに返す．

最後にクライアンドアプリケーションは受信した訳語データを整形し，作成者に表示する．

図 9 はスマートコントラクトが呼び出されてから，処理結果を返すまでの時系列を表現するシーケンス図である．

3.4.3　未評価訳文データ取得

評価者は訳文データの正誤を評価するケースが想定される．正誤を評価するには未評価の訳文データを取得し，確認しなければならない．そのため，システムは正誤が確定されていない訳文データを検索する処理が必要となる．

評価者が未評価の訳文データを取得したい場合には，クライアントアプリケーションを通じて未評価訳文データ取得のスマートコントラクトを呼び出す．絞

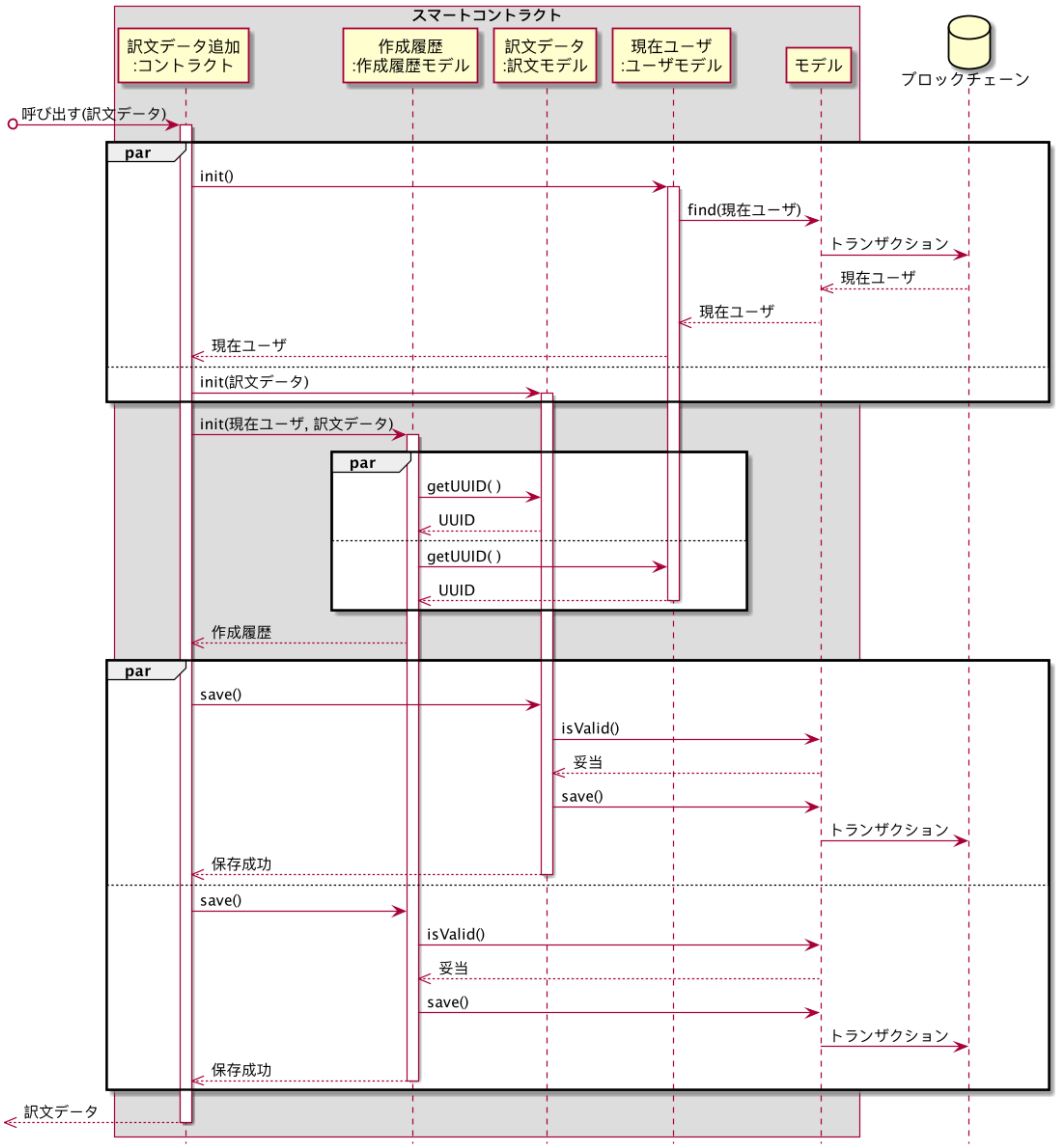

図 9: 訳文データ追加のシーケンス図

り込む訳文データに条件を付加したい場合には絞り込み条件をクライアントアプリケーションに渡す．クライアントアプリケーションは絞り込み条件を適切なクエリに整形し，引数として未評価訳文データ取得スマートコントラクトを呼び出す．

スマートコントラクトの中には，まず渡されたクエリにバリデーションをかけ，インジェクションを防ぐ．そして，訳文モデルを通じてクエリの条件と一致する訳文データセットを取得し，クライアントアプリケーションに返す．

クライアントアプリケーションは訳文データセットを受信すると，訳文デー

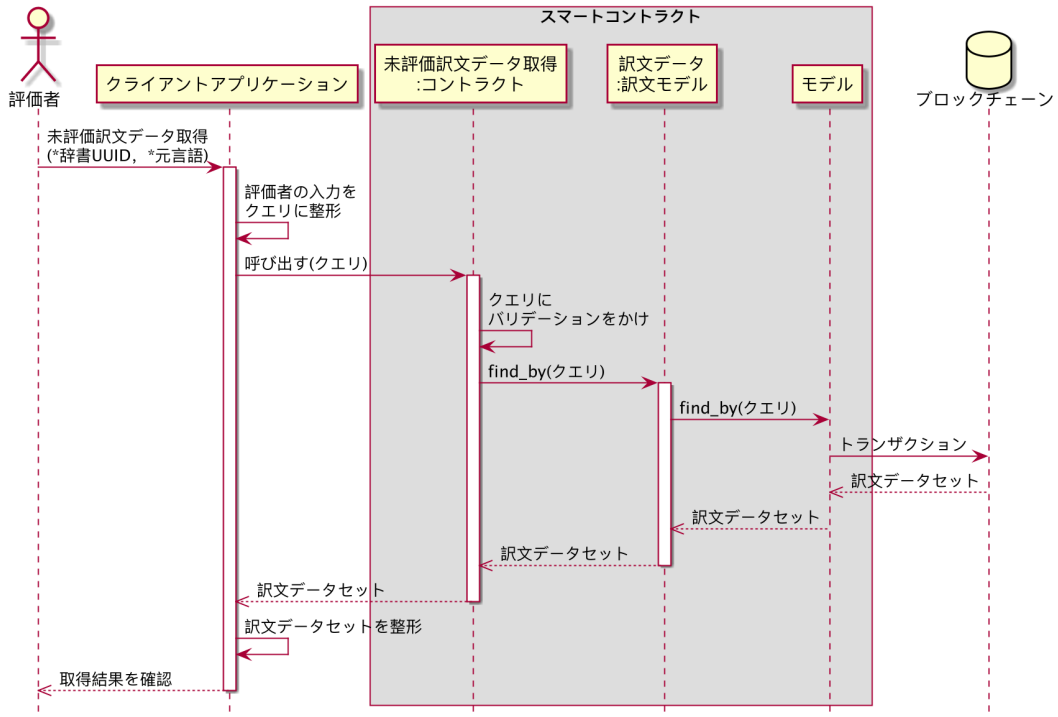

図 10: 未評価訳文データ取得のシーケンス図

タセットを評価者に見やすい形に整形し，評価者に確認させる．

図 10 は未評価訳文データ取得の時系列を表現するシーケンス図である．

3.4.4 訳文データ評価

評価者が訳文データを確認し，訳文データに正または誤の評価値を付けた行為を記録するために，システムは評価履歴を作成し，ブロックチェーンにブロックチェーンに格納する処理が必要となる．また，評価者の誤判断を防ぐために，一つの訳文データは 5 人以上の評価者から評価を受けた場合のみに評価者の評価を多数決で訳語に正誤をつけ，訳文データの評価値を更新する．

評価者が訳文データを評価したい場合には，クライアントアプリケーションでその訳文データの UUID を指定し，評価値を正または誤に決める．クライアントアプリケーションは評価者から入力された情報を引数として訳文データ評価のスマートコントラクトを呼び出す．

スマートコントラクトが呼び出されたら，まずユーザモデルを通じて，現在のユーザを取得する．同時に，渡された訳文データの UUID を訳文モデルに渡し，指定された訳文データが存在していることを確認する．確認できると評価値，現在ユーザインスタンスと訳データインスタンスを評価履歴モデルに渡す．

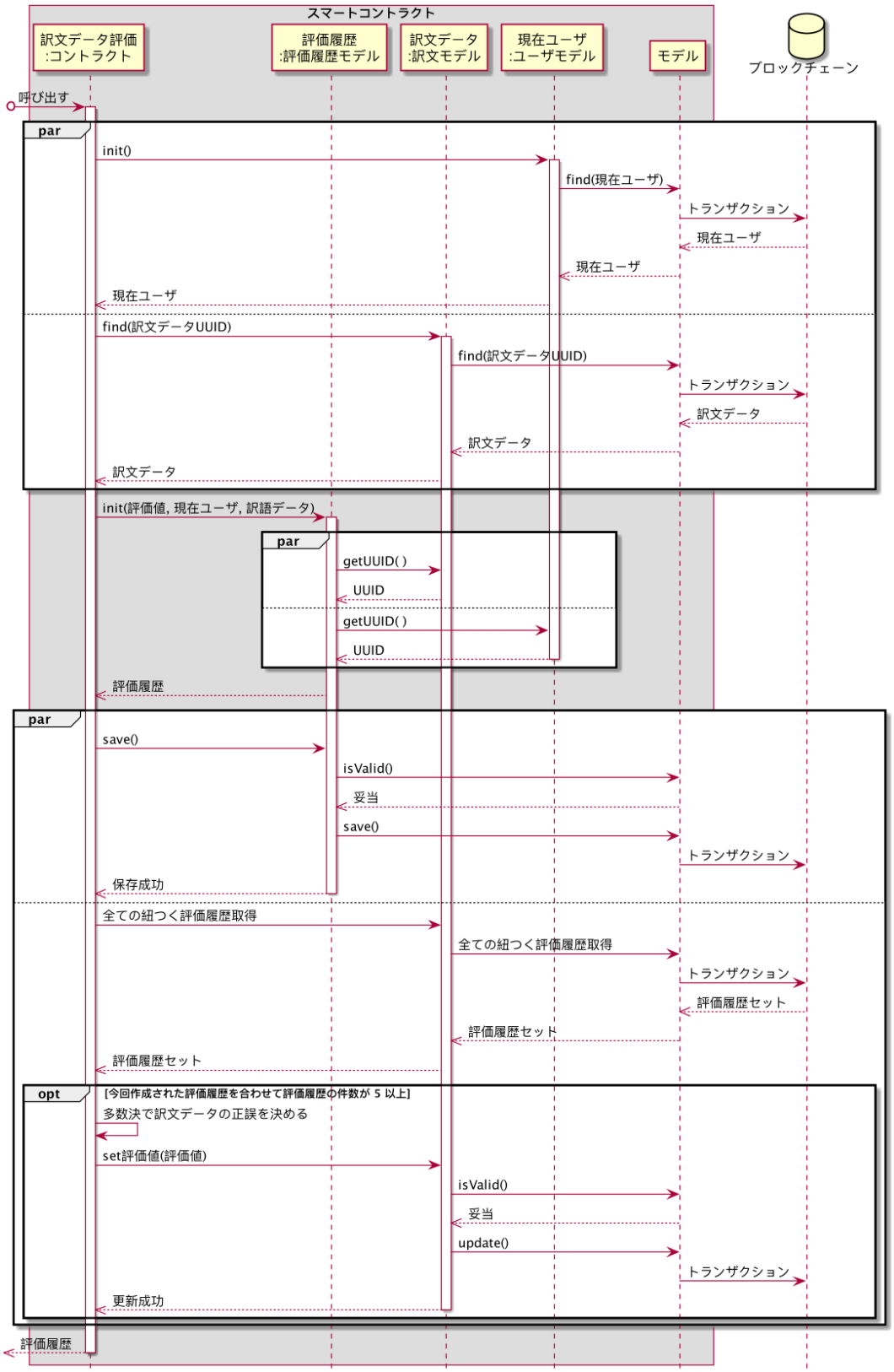

図 11: 訳文データ評価のシーケンス図

評価履歴モデルは現在ユーザの UUID と訳文データの UUID を取得し，評価履歴インスタンスを作成し，バリデーションを行い，ブロックチェーンに刻み込むトランザクションを発行する．

そして，今回作成された評価履歴を合わせて，訳文データと紐つく評価履歴の件数が 5 件未満の場合は今回作成された評価履歴をクライアンドアプリケーションに返す．5 件以上の場合にはクライアンドアプリケーションに返す前に，多数決で訳文データの評価値を算出し，訳文モデルを通じて評価値を更新するトランザクションを発行する．トランザクションが発行される前にも評価値にバリデーションをかけ，データの妥当性を確認する．

最後にクライアントアプリケーションは受信したデータを整形し，評価者に通知する．

図 11 は訳文データ評価のスマートコントラクトが呼び出されてから，処理結果を返すまでの時系列を表現するシーケンス図である．

3.4.5　訳文データ利用

利用者は辞書を引き，訳文データを利用する時には利用履歴が記録される．同時に利用者から利用料を取り，利用された各訳文データの作成者と評価者に報酬を分配する．このようなユースケースを実現するにはシステムは以下のような振る舞いを実現した．

利用者がクライアントアプリケーションに絞り込み条件を指定する．クライアントアプリケーションはもらった条件を適切なクエリに整形し，引数として訳文データ評価のスマートコントラクトを呼び出す．

スマートコントラクトが呼び出されると渡されたクエリにバリデーションを行い，インジェクションを防止する．そして，ユーザモデルを通じて現在のユーザを取得する．同時に，訳文モデルを通じてブロックチェーンからクエリの条件と一致する訳文データセットを取得する．訳文データセットを取得した後に各訳文データに以下の処理を繰り返す．

現在ユーザインスタンスと訳文データインスタンスを利用履歴モデルに渡す．利用履歴モデルは現在ユーザの UUID と訳文データの UUID を取得してから，利用履歴インスタンスを作成し，バリデーションを行い，ブロックチェーンに格納するようにトランザクションを発行する．

同時に，訳文データの作成者と評価者たちの情報を取得し，インスタンス化する．現在のユーザから利用料を引落し，報酬として作成者と評価者たちに分

配する．ここでは開発の便宜上の理由で作成者と各評価者が訳文データに対する貢献度を同等に扱う．引落した利用料を作成者と評価者たちの人数で均等に分け，報酬として作成者と各評価者に分配する．

繰り返し処理を完了すると，現在のユーザと利用された各訳文データの作成者と評価者の情報をユーザモデルを通じて更新し，ブロックチェーンに格納する．取得した訳文データセットをクライアントアプリケーションに返信する．

最後にクライアントアプリケーションは受信した訳語データセットを整形し，利用者に提示する．

図 12 は報酬分配までのスマートコントラクト内の振る舞いを表現するシーケンス図である．

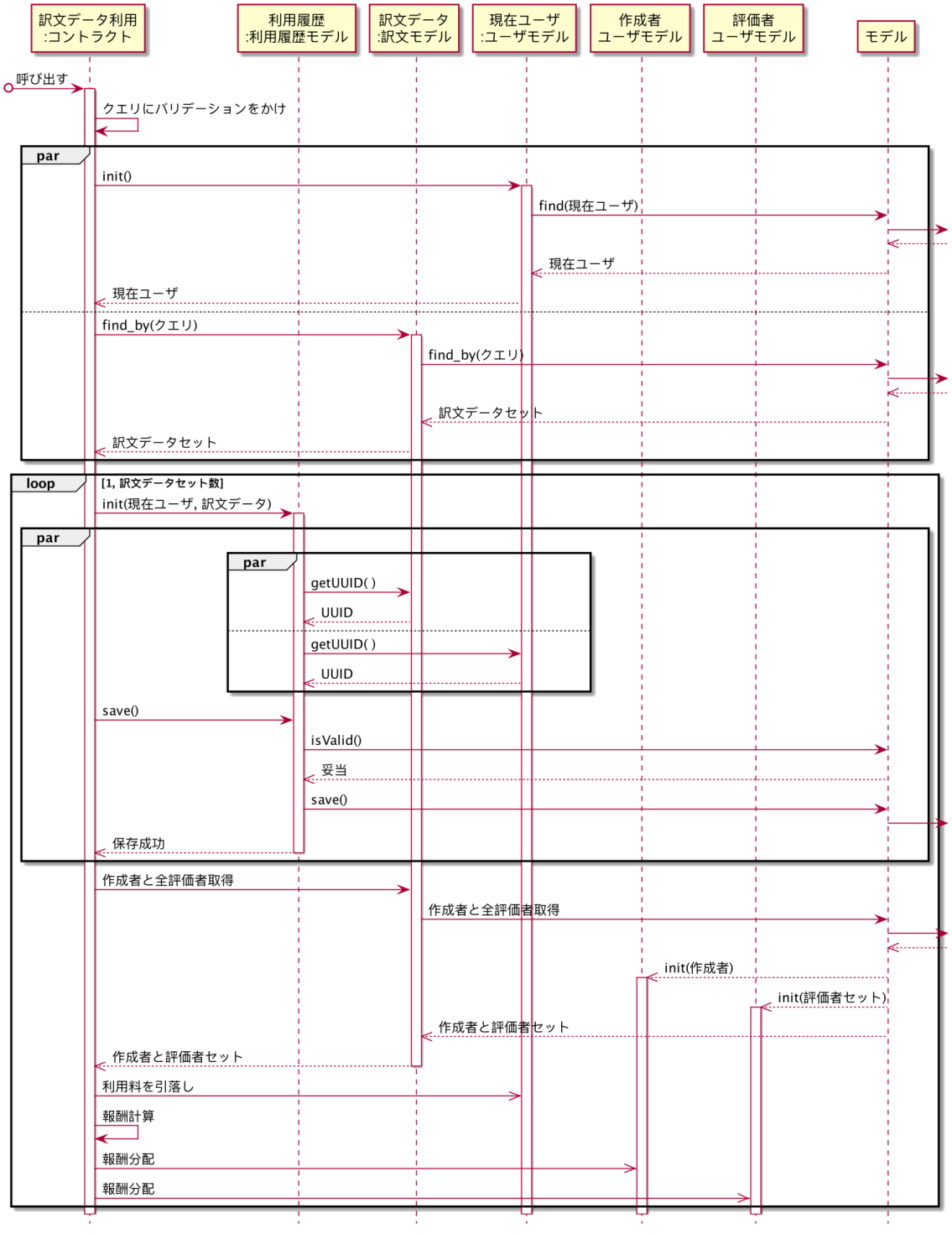

図 12: 訳文データ利用のシーケンス図

第4章　評価

本研究では，許可型ブロックチェーン OSS（Hyperledger Fabric）を利用し，辞書データの利用量に応じて作成者が利用者から直接報酬分配することができるシステムを提案し，構築した．この章では辞書システムにおける許可型ブロックチェーン OSS（Hyperledger Fabric）の有効性の確認を行う．

4.1　実験環境

実際に評価を行った検証システムについて説明する．三つのコンソーシアム（作成者コンソーシアム，評価者コンソーシアム，利用者コンソーシアム）を仮定し，ブロックチェーンを構築する．図 13 に構成図を示す．検証システムは 6 台の仮想マシン（VM1〜6）から構成される．VM0〜VM2 にはブロックチェーンへの参許を可認する認証局（CA）を配置する．VM3 には Orderer と呼ばれるブロック生成機能を配置し，VM4〜5 には Peer と呼ばれる検証／記録機能を配置する．Raft プロトコル [5] で合意形成する．ブロックチェーンには大部分の参加者が検証，承認されたデータのみ登録するようにエンドースメントポリシーを設定する．

クライアントアプリケーション[1)](図の Application) をホストマシンに配置し，検証システムに負荷をかけることで，性能評価を実施する．

表 1: 検証システム環境構成

項目	内容
CPU	Intel Core i7-7700K
メモリ	8 GB
ブロックチェーン OSS	Hyperledger Fabric v2.2.2
開発言語	Node.js v12.16.1 & TypeScript v3.9.7
Hyperledger Fabric SDK	fabric-network v2.2.4

[1)] Hyperledger Fabric SDK for Node.js を利用

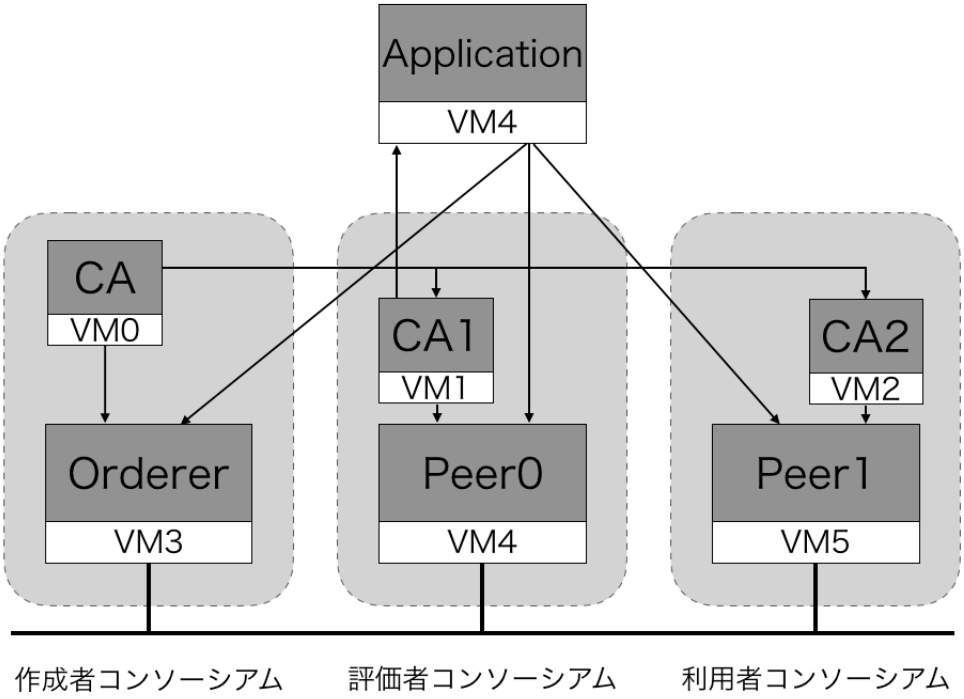

図 13: 検証システム構成

表 2: 検証システム環境での対訳作成と評価の処理性能

処理名	スループット	平均応答時間
対訳作成処理	95 [件/秒]	3.2176 [秒]
評価処理	5 [件/秒]	10.8727 [秒]

4.2 性能評価

上記の検証システムに対して性能評価を実施する．具体的には対訳作成と評価時と辞書引き時のスループットと応答時間を計測する．対訳作成と評価は訳文データ追加の処理と訳文データ評価の処理，辞書引きは訳文データ利用の処理とする．応答時間はクライアントアプリケーションがスマートコントラクトを呼び出すからトランザクションが承認され実行結果を受信するまでの時間とする．

4.2.1 対訳作成と評価

対訳作成と評価時のスループットとその時の平均応答時間を表 2 に示す．

対訳作成では一定の単位時間内に 30 リクエストから 1739 リクエストまでを

表 3: 検証システム環境での辞書引きの処理性能

処理名	スループット	平均応答時間
辞書引き処理	4 [件/秒]	20.4145 [秒]

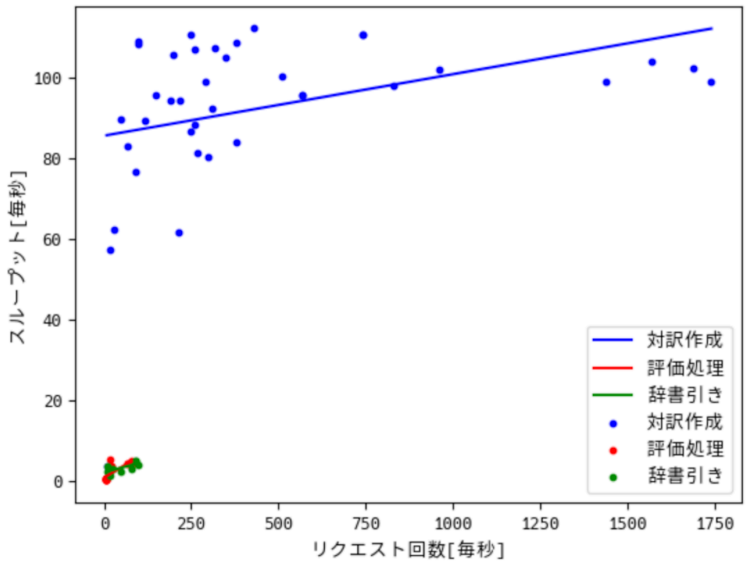

図 14: 各処理のスループットの散布図と線形回帰直線

請求する．評価処理では単位時間内[1)]のリクセストが 200 回以上に上回る途端に検証システムがコンフリクトのエラーを吐き出し，または無反応になり，計測不能になる．そのため，リクセストが 200 回以内で性能を計測する．

4.2.2 辞書引き

辞書引き時のスループットとその時の平均応答時間を表 3 に示す. 辞書引き処理は評価処理と同様に単位時間内に 200 回以上のリクセストを請求すると，検証システムがコンフリクトのエラーを返し，または無反応になり，計測不能になる．そのため，リクセストが 200 回以内で性能を計測する．

また，各処理の計測結果で図 14 と図 15 のスループットと平均応答時間の散布図と線形回帰直線を作成した．

4.2.3 考察

対訳作成では 1 秒間に 95 回の辞書引き処理が可能であると判明した．このスループ値は，1 利用者あたり毎分 2 回対訳作成処理を行うとした際に 1900 人の作成者の同時作業が可能である．

1) 一秒以内

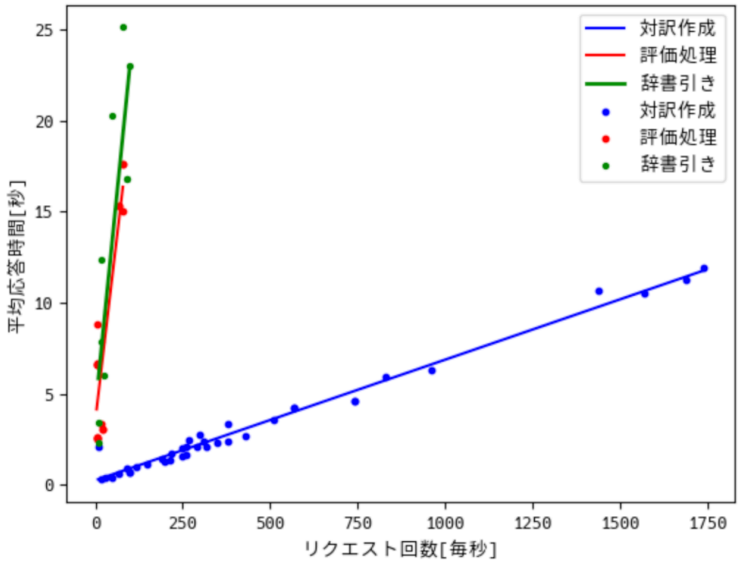

図 15: 各処理の平均応答時間の散布図と線形回帰直線

しかしながら，評価処理と辞書引き処理の際に検証システムはスループットが毎秒 5 件前後に落ち，平均応答時間が対訳作成処理の三倍から七倍になった．非常に悪いパフォーマンスが出された．

パフォーマンスが悪くなる原因は検索処理と推測する．評価処理と辞書引き処理に比べて，対訳作成処理はデータを検索する処理がない．各ピアノードはトランザクションを検証するためにスマートコントラクトを実行するだけで良い．一方，各ピアノードは評価処理または辞書引き処理のトランザクションを検証するために，ブロックチェーンから対応するデータを探し出す必要がある．そのため，処理が重く，パフォーマンスが悪くなったと考えられる．

Hyperledger Fabric にはワールドステートという概念がある．ワールドステートはブロックチェーンと同様にデータを保持している．ブロックチェーンがデータが現在値になるまでの変化を記録するのに対して，ワールドステートはデータの現在値を記録する．Hyperledger Fabric はワールドステートをピアノード上のデータベース (Couchdb) で管理し，ブロックチェーン中の記録でデータベース内のワールドステートの正確性を保証する．検索処理を行う際に，データベースにクエリを発行することで検索処理の時間を短縮する．

一方，Couchdb は楽観的ロックである．そのため，短時間内に同一ユーザに報酬分配にするとデータがコンフリクトが生じ，エラーが発生する．ピアノードの間に合意が形成できない．評価処理の時も同様に，短時間なに同一訳語の

正誤を判断すると訳語の評価値にコンフリクトが生じやすくなる．

また，Couchdb にはデータの検索を高速に行うために，インディクスを事前に作成する必要がある．今回の検証システムには適切なインディクスが作成されなかった原因で，検索処理に莫大の計算力を消耗し，無反応になると考えられる．

今後の課題として適切なインディクスを作成すれば検索処理を軽くなり，システムが良いパフォーマンスを出せれるかを検証する．

そして，図 15 から，単位時間内のリクエスト数の増加と共に各処理の平均応答時間が長くなる傾向が分かれる．しかしながら，図 14 から単位時間内のリクエスト数を増やすとスループットが大きくなり，負荷が軽い時より良いパフォーマンスを出したことを判明できる．

リクエスト数を増やすとスループットが大きくなることには以下の原因を考えられる．一ブロックに複数のトランザクションが記入されてから，ブロックチェーンに繋げる．負荷が軽い時にはトランザクション数が少ない，ブロックの最大容量に達していないのため，一定の時間[1)]が経たないとブロックが生成されない．リクエスト数を増やすとトランザクション数も増やされ，ブロックの最大容量を上回るため，新しいブロックが次々と生成され，ブロックの生成に待つ時間が大幅に短縮される．

そのため，クエスト数を増やすとスループットが大きくなった．

その他，検証システムでは参加者の大部分の承認がないとブロックチェーンにトランザクションを格納されないようなエンドースメントポリシーを設定した．エンドースメントポリシーが厳しく設定すれば，トランザクションの正確性を保障できる一方，十分な承認をもらうのも困難になり，システムのパフォーマンスが影響されると予測できる．

それに対して，エンドースメントポリシーは十分の承認数をもらえやすくように設定する場合には，システムのパフォーマンスが上がる一方，システムのデータ完全性が保障しにくくなる．

辞書システムの場合には，エンドースメントポリシーをどう設定すれば，パフォーマンスとデータ完全性のバランスを取れるのも今後の課題になる．

[1)] 検証システムは二秒と設定した．

第5章　おわりに

本研究では，辞書データの利用量に応じて作成者が利用者から直接報酬を分配することができるシステムを提案した．そして，ブロックチェーン技術の改ざん耐性と非中央集権の特性を用いて辞書システムの健全性を向上することを示した．

本研究の貢献は以下の2点である．

作業履歴の高信頼データ管理

利用量に応じて報酬分配するためのデータモデルの設計を行い，改ざん耐性を持つシステムを構築した．辞書の訳文データが作成，評価，利用される時に履歴も残され，報酬分配の根拠になる．データ数に応じたトランザクションの性能評価を行った．

報酬分配の透明性

報酬を計算，分配するアルゴリズムを決め，訳文データが利用されるたびに自動的に報酬を分配できるメカニズムの実装を行なった．なお，このメカニズムの実装によって，訳文データが利用される際にリアルタイムで報酬分配することも可能になった．

また，本稿では未解決の問題も存在する．今回は単純化のために，評価者が未評価訳文データを自ら取得し，訳文データの正誤を評価者の多数決で決めた．今後の課題として，評価者の過去の評価の正答率に基づいて評価者の動的な信頼値評価手法を導入することで，信頼値に応じて評価者に未評価訳文データを自動的に割当たる．さらに，信頼値に基づいた評価に重み付けを用いることで，多数決の精度を向上させる． [6]

そして，報酬分配のメカニズムをスマートコントラクトでブロックチェーンに記述することで，報酬分配のメカニズムの透明性を確保した．しかしながら，今回は開発の便宜上の理由で作成者と評価者が訳文データに対する貢献度を同等に扱った．正しい訳文データに誤った評価値を付けた評価者にも訳文データが利用される際に報酬を分配される．このような報酬分配手法は公平であるとは言いがたい．今後の課題として，シャープレイ値に基づいた分配手法を導入することで，報酬分配メカニズムの公平性を向上させる．

謝辞

本研究を行うにあたり, 熱心なご指導, ご助言を賜りました指導教官の村上陽平准教授に深謝申し上げます. また，普段からご協力，ご意見を下さいました大久保弘基先輩にも感謝の意を表します.

参考文献

[1] Nakamoto Satoshi: Bitcoin: A peer-to-peer electronic cash system, *http://www.bitcoin.org/bitcoin.pdf* (2009).

[2] 白石善明，掛井将平: ブロックチェーンの合意形成アルゴリズム，電子情報通信学会 通信ソサイエティマガジン， Vol. 14, No. 1, pp. 19–25 (2020).

[3] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains, pp. 1–15 (2018).

[4] Fowler, Martin : Patterns of Enterprise Application Architecture: Pattern Enterpr Applica Arch. Addison-Wesley (2002).

[5] Diego Ongaro and John Ousterhout: In Search of an Understandable Consensus Algorithm, Philadelphia, PA, USENIX Association, pp. 305-319(2014)

[6] 地田大樹，村上陽平: 対訳辞書作成のための信頼に基づくクラウドソーシングの評価，電子情報通信学会 信学技報, Vol. 119, No. 482, pp.91-96 (2020).