歴史マイクロナレッジの提唱と HIMIKO（Historical Micro Knowledge and Ontology）システムの実装

小川　潤・北本　朝展（ROIS-DS 人文学オープンデータ共同利用センター）

大向　一輝（東京大学）

概要：本稿は，歴史資料に記述された出来事や状況，人やモノの関係性といった歴史事象の具体的な記録そのものを「歴史マイクロナレッジ」として知識グラフで構造化するモデルである HIMIKO（Historical Micro Knowledge and Ontology）について論じる．HIMIKO は，上述のような歴史事象に言及する資料の特定箇所（Factoid）をデータの基本単位として構造化する Factoid プロソポグラフィモデルを拡張し，プロソポグラフィ以外のデータ構造化や一次資料記述のより精緻な表現を実現する．本論では，HIMIKO におけるデータ構造化モデルの詳細を論じるとともに，モデルに基づいたデータ構築を支援するために開発した HIMIKO エディタを紹介する．このエディタを利用することで，デジタル技術に精通していない歴史研究者であっても，従来のテクスト読解の延長線上にある直感的な操作で資料記述に基づく RDF データを作成することが可能になり，歴史研究者の知見に基づく精緻な歴史データの蓄積が可能になる．

キーワード：デジタル・ヒストリー，リンクトデータ，知識グラフ，テクスト構造化

The Concept of Historical Micro Knowledge and the Implementation of HIMIKO (Historical Micro Knowledge and Ontology) System

Jun Ogawa / Asanobu Kitamoto (ROIS-DS Center for Open Data in the Humanities)
Ikki Ohmukai (University of Tokyo)

Abstract: This paper discusses HIMIKO (Historical Micro Knowledge and Ontology), a model designed to structure the specific records of historical phenomena, such as events, situations, and relationships found in historical documents as 'Historical Micro-knowledge' represented as an RDF graph. While HIMIKO extends the Factoid Prosopography model, which structures specific points of reference (factoids) mentioning historical events in documents as the basic unit of data, it enables a structurization of data beyond prosopography and a more elaborate representation of primary source descriptions. Besides explaining the HIMIKO data model in detail, we also introduce the HIMIKO editor developed to support data construction based on the model. Utilizing this editor allows historians, even those not proficient in digital technology, to create RDF data based on document descriptions through intuitive operations, which is to some extent close to the traditional text reading, facilitating the accumulation of detailed historical data based on their insights.

Keywords: Digital History, Linked Data, Knowledge Graph, Text Encoding

１．はじめに

デジタル・ヒストリー研究における重要なテーマの一つは，複雑かつ多様な歴史情報をいかにデータとして構造化し，専門的な歴史研究から一般に向けた情報発信にまで活用可能なデータを生成するかという問題である．とくにビッグデータ時代の歴史情報の利用を考えるに[1]，個々の資料単位でのメタデータ付与やアノテーションに加え，それらの資料中に含まれるより粒度の細かい情報，すなわち，そこで言及される個々のエンティティや，それらが関与する出来事や関係性についての記述を構造化し，高い一次資料参照性を確保しつつ利用可能にすることが重要となる．

２．関連研究と課題

2-1　テクストマークアップとエンティティ・リンキング

近年，資料そのものから得られる歴史情報をリンクトデータとして構造化し，Web 上の広範なデータ資源と接続可能な形で利用しようとする動きが活発になっている．その具体的な手法はさまざまに提案されており，国内でも，「小城藩日記データベース」や「みんなで注釈」といったプロジェクトが[2][3]，資料の記述内容のリンクトデータ化に取り組んでいる．ここでいうリンクトデータ化とは具体的には，資料の中に現れる人物や場所，時間といったエンティティ

情報をマークアップし，それを外部の典拠データと接続すること，すなわちエンティティ・リンキングである．また，こうしたエンティティ・リンキングを実現するためには，共通して参照可能な典拠データの整備が不可欠であることから，地名典拠を提供する GeoLOD や歴史地名データ[4][5]，時間情報を表現する HuTime など[6]，基礎データそのものの整備も着々と進みつつある．

他方，国外でも同様に，基礎データの整備やエンティティ・リンキングに基づく資料のリンクトデータ化が急速に進んでいる．こうした，リンクトデータを基盤とした歴史研究，ひいては人文学研究基盤構築において，とくに欧米で重要な役割を担っているのが Pelagios Network という研究コミュニティである．Pelagios Network は各地の研究機関やプロジェクトとパートナー提携を行うことで，そうした機関・プロジェクトが生成するデータやツールの幅広い共有を実現することを目指しており[7]，大規模歴史地名典拠である World Historical Gazetteer や Pleiades[8][9]，時間情報の LOD 化に取り組む PeriodO などもパートナーに名を連ねている[10]．また，こうした典拠に依拠した資料のアノテーションを支援するツールも開発しており，Recogito がその代表的な例である[11]．もちろん，歴史情報の整備やリンクトデータ化を進めているのは Pelagios Network とそのパートナーのみではなく，各所で同様の動きがみられる．

2-2　テクストのセマンティック表現に向けて

ここまでにあげた研究やプロジェクトは，典拠データとしてのリンクトデータ整備や，エンティティ・リンキングという意味での資料のリンクトデータ化の事例が主であった．他方，近年ではエンティティ・リンキングによる同定というレベルを超えて，資料で記述されるエンティティ間のセマンティックな関係性や，テクスト記述の語彙や表現そのもの，いわば「ニュアンス」そのものをデータとして表現しようとする研究もみられるようになってきている．

たとえばチェコの Dissident Networks Project は，‘Source Criticism 2. 0’というコンセプトのもと，テクスト中で用いられる個別の語彙や表現までを含む意味内容を構造化し，知識グラフとして表現するモデルとして CASTEMO を提唱しており[12]，そうしたデータ構造化を比較的容易に行うための InkVisitor というエディタの開発も進めている．ただし CASTEMO はグラフデータベースである Neo4J を用いて知識グラフを記述しており，厳密なオントロジーを設計しているわけでもなければ，RDF を用いたデータ記述を行なっているわけでもないため，厳密にはリンクトデータを用いたプロジェクトではない．こうした資料記述そのものに基づく構造化という点では，TEI コミュニティにおいても，<relation>エレメント等を用いてテクストに含まれるさまざまな関係性や出来事に関する「言及」を XML で構造化し，それを RDF などのリンクトデータ形式に変換し利用する，といった試みが提起されるようになってきている[13]．TEI による構造化であれば基本的にはテクスト自体が保持されるため，高い一次資料参照性を確保することができる．

そのほか，より歴史学に特化した研究としては，会計資料構造化を対象に，資料の中の商取引に関する言及を，物品の価格や数量といったきわめて詳細な情報を含めて構造化するデータモデルである DEPCHA や[14]，ある特定の人物が関与した関係性や出来事，経歴に関するプロソポグラフィ情報構造化のための Factoid Prosopography Ontology（FPO）などがある[15]．これらのプロジェクトにおいても近年，TEI との連携，すなわち TEI マークアップに基づくリンクトデータ構築の試みが進んでおり，テクスト情報との接続が強く意識されていることが窺える[16][17]．

2-3　課題

このように，リンクトデータおよび知識グラフを用いた歴史情報構造化の試みは，典拠データ整備から資料の意味内容構造化までさまざまなレベルで行われている．そうした既存の研究を踏まえて，デジタル・ヒストリー研究の観点からリンクトデータによる歴史情報の構造化を鑑みるに，以下の諸点がなお課題となっていることがわかる．

1. 特定の資料形態や情報の種類に限らず，多様な資料の記述内容を柔軟に表現し，かつ広範に接続しうる汎用的なデータモデルの設計
2. 資料記述そのもの，そこで用いられる語彙や表現といった詳細な一次資料テクスト情報を，標準的な RDF リンクトデータとして表現するための手法とワークフローの確立
3. 複雑なデータ構築作業を簡易化し，デジタル技術に詳しくない研究者によるデータ構築を可能にするシステムの構築

以上の課題を踏まえ次章では，資料における個々の出来事や関係性，状況への言及を最小の情報単位とみなし，資料記述の詳細までを含めて構造化する「歴史マイクロナレッジ」という概念を提示するとともに，その概念をデータとし

て構造化するための HIMIKO（Historical Micro Knowledge and Ontology）モデルを定義する.

３. HIMIKO オントロジーの導入

3-1　歴史マイクロナレッジ

HIMIKO モデルが依拠する「歴史マイクロナレッジ」とは，資料全体や巻・章等のひとまとまりのテクスト単位ではなく，その中で言及される個別の出来事や状況，関係性といったあらゆる歴史事象に関する「断片的な」情報を知識グラフとして構造化し，独立したデータ資源として扱うための概念である．このように，小さな単位での構造化を行うことで，資料の複雑な情報構造を比較的シンプルかつ共通のモデルで扱うとともに，個々のリソースについてのきわめて詳細な情報記述が可能になる．こうしたマイクロナレッジの概念はむろん，鈴木親彦が提唱する「人文学資料マイクロコンテンツ」と密接に関連するものであるが[18]，マイクロコンテンツが，切り抜かれた画像など，あくまで個々のエンティティを対象にするのに対し，マイクロナレッジはむしろ，そうしたエンティティ間の関係性を知識グラフとして繋げて記述しようとするものである．また，マイクロコンテンツが現在のところ図像資料に限定されているのに対して，マイクロナレッジはテクストを対象として知識の構造化を行う．

3-2　HIMIKO オントロジーのコンセプト

このような歴史マイクロナレッジの概念を具象化したオントロジーが HIMIKO であり，RDF に基づく知識グラフとして歴史知識を構造化する[1]．HIMIKO オントロジーは，資料における特定の歴史事象への「言及」を基本単位とする点で上述の Factoid モデルを基盤とするが，既存のモデルには，「歴史マイクロナレッジ」の記述を実現するうえでいくつかの課題がある．第一に，FPO はプロソポグラフィに特化したモデルであるため，データ記述の対象は基本的にプロソポグラフィで扱われる情報に限られ，それ以外の情報は扱われない．たとえば，地理情報や気象情報，モノを含む空間情報，また人物に関する情報であっても一回的な行為等のあまりに詳細な情報は記述の対象とはされない．HIMIKO オントロジーでは，こうした情報にまでデータ構造化の範囲を拡大し，より網羅的かつ粒度の細かい歴史情報記述を実現する．第二に，Factoid は資料の特定箇所における歴史事象の「言及」に基づいて歴史情報を構造化するモデルであるものの，詳細な文言や表現までを含めた一次資料の参照については，現状では課題が残る．そのため HIMIKO では，TEI によるテクストのマークアップと RDF による知識記述を接続し，各文字レベルで構造化されたテクストデータに基づく高い一次資料参照性を確保することで，資料における具体的な表記や語彙，さらには文字情報といったきわめて詳細な情報へのアクセスを可能にする．以上を踏まえた HIMIKO の基本的な構造を以下図１に示す.

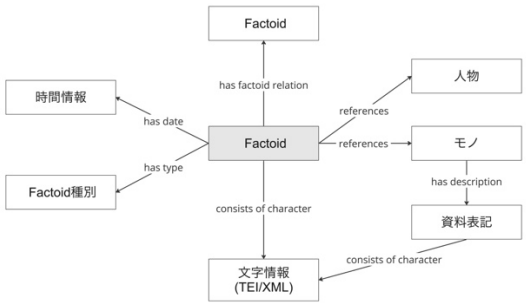
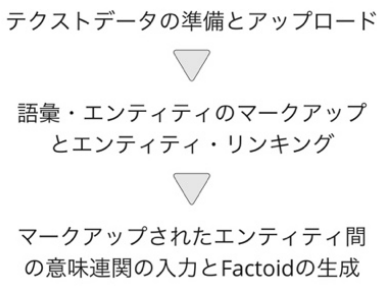


図１：HIMIKO オントロジーの概念略図

HIMIKO は資料記述から抽象的な知識までを一連のモデルで表現することのできる汎用的で表現性の高いモデルとして，歴史資料記述のデータ構造化に資するものである．しかし，それゆえに複雑な知識構造を有し，データ作成作業におけるコストはどうしても高くなる．そこで本研究では，テクストインターフェイス上で，実際にテクストを読み進めるような感覚でデータを作成・接続し，RDF 知識グラフを構築することのできるシステムとして，HIMIKO エディタを開発した[2]．以下ではこのエディタについて詳しく述べることとするが，これは HIMIKO オントロジーに基づくデータ作成を補助するシステムであるため，オントロジー自体の詳細な構造についても併せて論じることになる.

４．HIMIKO エディタとオントロジーの詳細

HIMIKO エディタを用いたデータ構築のフローを端的に示すならば，以下のようになる.



[1] https://w3id.org/HIMIKO/himiko.owl.
[2] 試作版は https://himiko-editor.vercel.app/.

4-1 基礎テクストデータの準備

HIMIKO は可能な限り細かい粒度での一次資料参照性を確保するため，文字レベルでのデータ構造化を行う．これは，歴史研究においてはときに文字自体の情報も重要になるためである．写本や碑文の研究においては，個々の文字の字形や大きさが分析の対象になるし，場合によっては文字の同定そのものが議論の対象になる．そうした事例を考えれば，文字データそのものに情報を集積することができるシステムを整備する必要がある．

HIMIKO は，テクストに含まれるすべての文字を RDF リソースとして扱うことでこれを実現する．そのためにはすべての文字に一意の ID を付与しなければならない．また，テクストとしての可読性を保つためには，文字間の順序情報を保持することも必要である．こうした順序情報の記述に関しては RDF よりも XML が優れていること，また，すでに多くの歴史資料が TEI/XML で構造化されていることを考慮すれば，基礎的なテクストデータについてはこれを利用することが最適である．それゆえ，既存の TEI/XML データを基盤に，以下の図２のように xml:id 属性を持つ<c>エレメントタグを付与することで，HIMIKO エディタにアップロードした際，自動的に文字レベルでの URI が生成されるようにする．

```
xml:id="shibusawa_char_3">栄</c><c xml:id="shibusawa_char_4">一</c>) <c
xml:id="saga_char_1_1_1"> (</c><c xml:id="saga_char_1_1_2">五</c><c
xml:id="saga_char_1_1_3">月</c><c xml:id="saga_char_1_1_4">) </c><c
xml:id="saga_char_1_1_5">三</c><c xml:id="saga_char_1_1_6">十</c><c
xml:id="saga_char_1_1_7">一</c><c xml:id="saga_char_1_1_8">日</c><c
xml:id="saga_char_1_2_1"> </c><c xml:id="saga_char_1_2_2"> </c><c
xml:id="saga_char_1_2_3"> </c><c xml:id="saga_char_1_2_4"> </c><c
xml:id="saga_char_1_2_5"> (</c><c xml:id="saga_char_1_2_6">五</c><c
xml:id="saga_char_1_2_7">月</c><c xml:id="saga_char_1_2_8">) </c><c
xml:id="saga_char_1_2_9">三</c><c xml:id="saga_char_1_2_10">十</c><c
xml:id="saga_char_1_2_11">一</c><c xml:id="saga_char_1_2_12">日</c><c
xml:id="saga_char_1_2_13"> </c><c xml:id="saga_char_1_2_14">曇</c><c
```

図２：基盤となる TEI/XML データの例

なお，現状の HIMIKO エディタにおいては，既存の TEI/XML ファイルから自動的に<c>タグ付きファイルを生成する機能は実装されておらず，その作成はエディタ外部で行う必要がある[1]．

<c>タグが付けられた TEI/XML ファイルは，HIMIKO エディタの「アイテム」画面からアップロードする．HIMIKO エディタの構成は「プロジェクト」＞「ドキュメント」＞「アイテム」となっており，個々の TEI/XML ファイルに相当するのは「アイテム」である．複数のアイテムを束ね，ひとまとまりの文書群を表すのが「ドキュメント」，さらに複数のドキュメントを束ねるのが「プロジェクト」である．アップロードされたテクストデータは Cloud Firestore のストレージに格納される．またファイルをアップロードする際には，エディタのテクストインターフェイスに表示させる TEI の要素名，属性名，属性の値を入力することができる．これにより，インターフェイス上に表示したいテクスト本文が格納されている要素ブロックがファイルによって異なる場合にも対応することができる．

ファイルをアップロードしたうえで，インターフェイス上の「エディタ」ボタンを選択すると，以下の図３のようなエディタ画面が開く．

図３：エディタ画面

エディタの構成としては，右側がテクスト表示画面，左側が種々の情報を表示するパネルとなっており，基本的にはテクストインターフェイスからデータを作成し，情報パネルでデータの編集や削除を行うことになる．

4-2 語彙やエンティティのマークアップと外部典拠データとの接続

データ作成者がまず行うのは，テクストに現れる語彙やエンティティのマークアップである．HIMIKO では，歴史事象に言及する資料記述において用いられる詳細な語彙や表現をデータとして保持することも重要であるため，エンティティのみならず語彙もリソースとして構造化する．テクストインターフェイス上で特定のテクスト領域を指定し，画面右上のアイコン群から「語彙」（黒い星形のボタン）を選択すると，語彙データ入力画面が開くので，語の辞書形を入力し，外部の語彙データとの接続を行う．同様に，人や場所といったエンティティについても，テクスト領域を指定したうえで，アイコン群からエンティティ種別を選択してデータ入力画面を開き，典拠データとのリンキングやタグ付けを行う．

ここでマークアップされた語彙やエンティティに関するデータは，TEI/XML ファイルにタグを追記する形で保持されるのではなく，HIMIKO オントロジーにおける LemmaReference クラスおよ

[1] ローマ碑文データベースが提供する TEI/XML ファイルをもとに自動作成した<c>タグ付きデータセットとして https://github.com/junjun7613/EDH_Ctagged_Inscriptions.

びSourceDescriptionクラスのインスタンスとしてCloud Firestoreに保存される．これらのクラスは，語彙やエンティティの「資料における表記」を表すものであり，資料記述そのものの厳密な構造化というHIMIKOモデルの趣旨に密接に関わるものである．歴史研究においてはときに，個々の語や，人物・場所といったエンティティが実際にどのように表記されているかを知ることがきわめて重要になる．これらのクラスに属するインスタンスは，XMLファイルを読み込んだ際に，<c>タグに基づいて生成される文字ごとのリソース（Charクラスのインスタンス）を参照することで，テクストにおけるその開始位置と終了位置の情報を保持し，一次資料参照性を確保する．この開始位置と終了位置の記述は，インターフェイス上でテクスト領域を選択する際にエディタに保存される．また，TEI/XMLファイル上での位置情報を記述しているため，XMLファイル側に校訂記号等のマークアップが施されている場合，RDF側からそれを参照することが可能になる．以上の操作によって生成されるRDF知識グラフは図４のようになる．

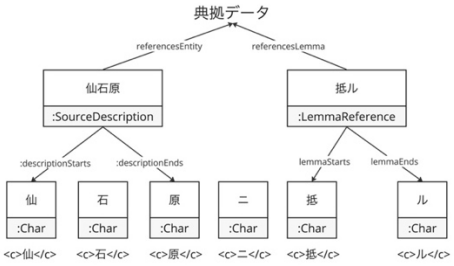


図４：マークアップによって生成されるデータ

4-3　エンティティ間の意味連関の入力とFactoidデータの構築

次にHIMIKOエディタでは，以上のような語彙とエンティティのマークアップ，および資料表記の構造化に基づいて，より抽象的な知識の構造化を行う．画面左上の「ファクトイド」タブを選択すると，マークアップを行うためのエディタからFactoidに基づく知識グラフを構築するためのエディタに移行する．この段階ですでにマークアップがなされている語彙やエンティティはフォント色によって可視化されている（「仙石原」が緑になっている）．テクストインターフェイス上の「ファクトイドを作成」を選択すると，テクスト表示ブロックの右側に新たなブロックが開き，ファクトイド記述のための入力インターフェイス（図５）が表示される．

ここではまず，何らかの出来事や状況，関係に言及する記述箇所をテクストインターフェイス上で指定し，入力画面最上部にある「テクスト範囲を決定する」を押下することで，Factoidを構成するテクスト領域を選択する．次に，「Factoid Type」という項目からFactoidの種別を表すクラスを選択する．このクラスには，「行為」や「発話」，「職業・官職」「親族・社会的関係」「状況」「地理環境」など，資料記述の内容についての比較的大きな区分が含まれる．

図５：Factoid入力インターフェイス（右部分）

クラス選択の他に，当該項目においてはFactoidの「詳細種別 DetailedType」と「説明 Note」を任意で入力することができる．「詳細種別」は，クラスでは表現しきれないFactoidの詳細な内容を記述する場合に用いられる．この段階で，システム内部ではFactoidインスタンスが生成され，入力されたクラス，詳細種別，説明がRDFで記述される．また，生成されたFactoidインスタンスはテクストの領域指定に基づいてCharインスタンスを参照することで，エンティティと同じく開始位置・終了位置を与えられる．ここまでの操作で生成されるデータは図６のようになる．

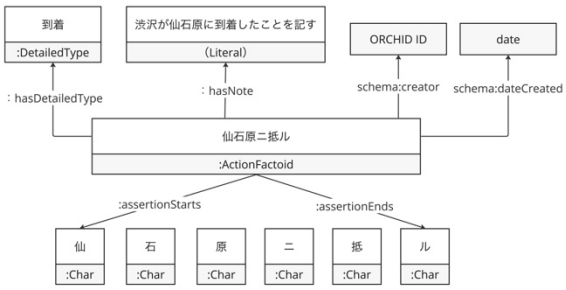


図６：最低限の情報が入力されたFactoidデータ

図６から分かるようにHIMIKOエディタにおいては，ユーザが直接インターフェイス上で入力するわけではない「データ作成者 Creator」や「作成日時Date」に関する情報も自動的にFactoidリソースに紐づけられる．これにより，資料の特定箇所の記述を表すFactoidごとにメタデータが付与され，きわめて細かい粒度での歴史情報へのメタデータ記述と，データ作成に際しての「解釈責任」の保持が可能になる．

このように，テクスト領域の選択とクラス選択（および詳細種別・説明の入力）によって最小限のFactoidデータが生成されるが，実際には，

Factoid において言及される人物や場所といったエンティティとその役割, 時間情報, Factoid 間の関係性といった情報を記述することで, より精緻な歴史情報を構造化することが望ましい. HIMIKO エディタにおいてはこうした詳細なデータの構築も, Factoid 入力画面から行う.

まず, 入力画面における「Factoid Type」ブロックの下に「Has Reference」というブロックがあり, ここから, Factoid の記述内容に関与したエンティティに関する情報を記述する（図７）.

図７：Reference のプロパティ入力画面

「Reference を選択してください」という項目のプルダウンを開くと,「関与者 Had Participant」,「時間 Had Time Span」,「場所 Took Place At」といった選択肢が表示されるため, Factoid に記述すべき情報に沿って選択する. そのうえで,「カードを追加」ボタンをクリックすると, 以下の図８のような入力画面が新たに表示される.

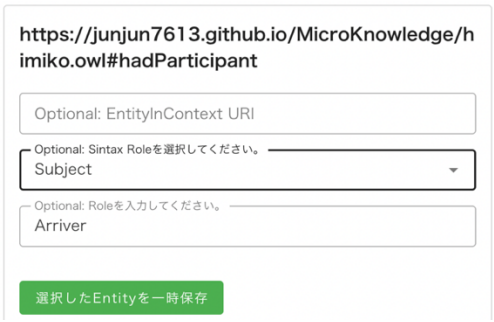


図８：Reference の内容入力画面

まずトップには, 先に選択した「関与者 Had Participant」のプロパティ URI が表示されている. すなわち, この画面で詳細を入力するエンティティは,「関与者 Had Participant」として Factoid に紐づけられるということである. 次に, テクストインターフェイスで「関与者」として Factoid に紐づけられるべきエンティティ（「関与者」の場合には大抵, 人物エンティティ）を選択し,「選択したエンティティを一時保存」ボタンをクリックすると, そのエンティティが「関与者」として Factoid にリンクされたトリプルが生成される. ただし, ここで Factoid リソースに直接的にリンクされるデータは先のマークアップのプロセスにおいて生成された SourceDescription インスタンスではなく, Factoid 入力に際して新たに生成される Reference クラスのインスタンスである. Reference は, 既存の Factoid モデルの概念を継承したもので,「特定の Factoid というコンテクストにおけるエンティティ」を表す[19]. すなわち, そのエンティティが特定の Factoid においてどのような文脈で言及されているかを表現するものであるため, 図８の画面において入力可能な Role, すなわち言及されるエンティティがある資料記述において有する役割に関する情報は, この Reference インスタンスに紐づけられることになる. また, Reference インスタンスと, 先にマークアップされた SourceDescription インスタンスの関係についていえば, 前者が hasDescription プロパティによって後者を参照することで, 資料における具体的な表記情報を保持するという形を取る.

以上のようなエンティティ情報の入力に加えて HIMIKO エディタでは, Factoid 同士の時系列や因果関係といった関係性, そして Factoid において言及される出来事や関係性を表すために資料で実際に用いられている「述語」「補語」等を記述することも可能であり, エンティティ記述と同じく Factoid 情報入力画面からデータを入力する. 基本的な操作方法はエンティティ・データ入力と同様であり, まずは Factoid 間の関係性の種類や述語・補語の別をプルダウンから選択し, 新たなカードを作成する. そのうえで, 関係づける他の Factoid や, 述語・補語に相当する語彙, すなわち, 先に生成された LemmaReference インスタンスをテクストインターフェイスから選択し保存する.

Factoid 情報入力画面におけるこれら一連の操作によって生成される Factoid データは, 図９のような構造を持つことになる.

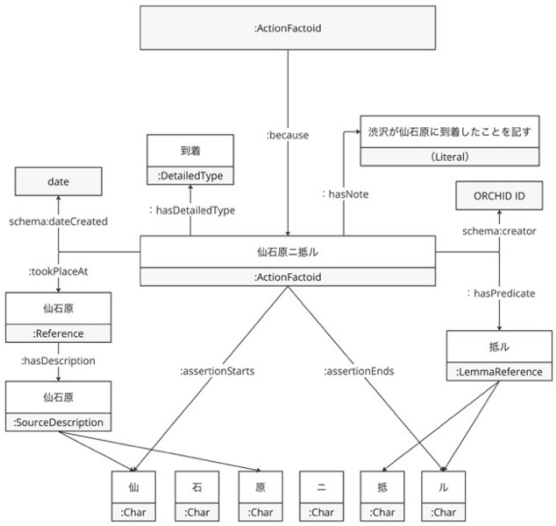


図９：最終的に生成される Factoid データ

4-4　小括

図９からわかるように, HIMIKO モデルに基づいて生成されるデータは, 歴史事象とその関与者の関係性, 時, 場所といった抽象的な知識記述と, それらのエンティティが資料中でどのように表記・表現されているか, というきわめて具体的なテクスト参照性の双方を含むものであり, 個々の文字に関するミクロな情報から, グローバルなデータリソースを含むマクロな知識ネットワークまでを一連の知識グラフとして表現することができる.

エディタ上で入力されたデータはすべて, Cloud Firestore に JSON 形式で保存されるようになっており, これを RDF データとしてダウンロードする際には, アイテムごとあるいはドキュメントごとに, JSON-LD および TURTLE 形式に変換して出力する. ダウンロードされたデータは, SPARQL による検索や可視化に用いることが可能である.

５．デジタル・ヒストリー研究における HIMIKO の貢献

2-3 で示した諸課題に対し、HIMIKO はそれぞれ次のような形で解決策を示した。

1. 個々の資料言及を表す Factoid に基づいて「歴史マイクロナレッジ」を構築することで、多様な情報に柔軟に適用可能なデータモデルを設計
2. TEI によるテクストマークアップと RDF による知識記述を統合することで、一次テクスト情報への文字レベルでの高い参照性を確保
3. HIMIKO エディタを開発することによって、データモデルやデジタル技術に精通していなくともデータ構築を行える環境を整備

HIMIKO が対象とするような資料の内容に関わるデータ構築においては, 高度な資料読解能力を持つ歴史研究者がデータ作成を担うことが不可欠である. この点において HIMIKO は, 歴史研究者にとってのデータ作成作業への参入障壁を引き下げ, 彼らが持つ専門知識に基づく精緻な歴史データ構築を実現するための基盤インフラを提供する.

そして、そうした専門家の知見に基づき、歴史資料の内容にまで踏み込んだ「深い」データが蓄積されていけば、資料外部の広範な知識空間と資料内の意味内容, そして厳密なテクスト情報を縦横に参照しつつデジタル空間で歴史知識の広範かつ詳細な探索を行うことが可能になり, 質的な情報を扱うデジタル・ヒストリー研究の進展にも寄与することになるはずである.

６．課題と議論

HIMIKO は歴史情報のデータ構造化に大いに貢献する研究ではあるが, 関連研究の項でも述べたように, 歴史資料の記述内容そのものをデータとして構造化しようとする研究は端緒についたばかりであり, なお試行錯誤の段階にあるため, いくつもの課題があることも事実である. 本稿では最後に, 現状の HIMIKO が有する課題と議論を提示しておきたい.

6-1 文字レベルでの構造化に関して

HIMIKO は, 各文字に<c>タグが付与された XML を基盤とするデータ構築モデルであるが, これまでに作成された TEI/XML テクストデータの多くは文字レベルでの構造化は行っていないため, これを HIMIKO で利用しようとする際には, 既存の構造は保持しつつ<c>タグを付与するという事前作業が必要になる. この<c>タグ付与という作業を HIMIKO エディタそのものが実装するという選択肢はもちろんあるものの, 既存の TEI データが有する構造は多様であり, そうした構造を保持しつつタグを付与するためには, それぞれに異なる処理が必要になるため, あらゆるテクストデータに対応可能な変換機能の実装はきわめて困難である. この点についてはむしろ, HIMIKO に限られないデータ基盤整備の一環として, 既存の TEI/XML データに基づく<c>タグ付きテクストデータの整備を進める方向性も考えられる. HIMIKO が依拠する文字レベルでの歴史情報の構造化は, CODH の「くずし字データセット」のような文字単位でのデータ蓄積も進む中で, その有用性がますます高まることが想定される以上, デジタル・ヒストリー分野全体で取り組むべき歴史情報基盤整備の一環とみなされるべきであろう.

6-2　歴史典拠データの整備

上で述べた歴史情報基盤整備と関わるもう一つの論点として, 歴史典拠データの整備がある. HIMIKO は, 個々の Factoid というコンテクストを重視して歴史情報を記述するモデルであるが, 個々のコンテクストにおいて言及される語彙やエンティティを最終的に同定するための参照点としての典拠データへの接続がきわめて重要になる. 加えて HIMIKO が, 歴史資料の詳細な記述内容を構造化するモデルである以上, 参照する典拠データの粒度や範囲もまた, より詳細かつ網羅的である必要がある. それゆえ, HIMIKO のようなデータ構築システムを真に有用なものとするためには, 歴史典拠データの整備・拡充も並行して進展する必要がある.

6-3　Factoid 記述の自動化に関して

Factoid は資料における出来事や関係性への言及を構造化するモデルであるが, 自然言語処理

技術を応用して，そうした言及を機械的に抽出する可能性については当然ながら議論がなされてきた．人文情報学においてそうした手法を部分的に導入した例としては，Chinese Text Project がある[20]．ただし，歴史資料の中で言及される出来事や関係性は多様であり，微細な表現の相違やニュアンスも表現する必要があるため，機械的な手法のみで Factoid を構造化することは現状ではきわめて困難であり，いずれにしても人手での修正やデータ入力が必要になる．

HIMIKO エディタでは，こうした技術的な限界に加えて，もう一つの理念的な理由から，現在のところ自動での Factoid 生成システムは導入していない．5 章で述べたように，HIMIKO エディタの目的は歴史研究者によるデータ作成作業を補助することで精緻な歴史情報の構造化を実現することであり，それを効率的に進めるためには，歴史研究者が資料読解を進める過程でメモを取るかのようにエディタを利用し，データが生成されていく環境を構築することが最善である．すなわち，HIMIKO エディタは資料の読解と関連情報のアノテーションをデジタル・プラットフォーム上で行うことを意図して開発されたものである以上，これは全自動化されたデータ作成とは相容れない．あくまでも人が資料を読み，その解釈に基づいて情報を抽出・マークアップするという前提のもとで，機械可読な構造化データを構築することが HIMIKO の基本的な理念である．

6-4　「歴史マイクロナレッジ」の拡張

本稿で扱ったのはあくまでテクスト資料を対象としたデータ構造化の手法である．だが HIMIKO モデルそれ自体はテクストという媒体のみに限られるものではなく，原理的には図像資料や 3D データも，テクストと同様に Factoid として構造化することができる．現在のエディタではこうした図像や 3D を扱う機能は実装されていないが，今後の発展の方向性としてはテクスト以外の資料を対象としたデータ構築システムの開発，ひいては，テクストや図像といった異なるメディア形式を統合したマルチモーダルな歴史情報構造化手法の確立を目指す．

謝辞

HIMIKO エディタの開発にあたっては，東京大学史料編纂所の中村覚・助教に多大なご協力をいただきました．謹んで感謝の意を表します．

参考文献

[1] “歴史ビッグデータ”．http://codh.rois.ac.jp/historical-big-data/, (参照 2023-11-03).
[2] 吉賀夏子，只木進一，伊藤昭弘．小城藩日記データベースの構築．情報処理学会研究報告，2018, Vol.2018-CH-117 No.3 .
[3] “みんなで注釈”. https://ansei2.vercel.app/, (参照 2023-11-03).
[4] GeoLOD. https://geolod.ex.nii.ac.jp/, (参照 2023-11-03).
[5] “歴史地名データ”. https://www.nihu.jp/ja/database/source_map, (参照 2023-11-03).
[6] 関野樹．時間名による時間参照基盤の構築：Linked Data を用いた期間の記述とリソース化．人文科学とコンピュータシンポジウム論文集，2019, pp. 267-272.
[7] R. Kahn et al. Pelagios: Connecting Histories of Place. Part II: From Community to Association, 2021, International Journal of Humanities and Arts Computing, Vol. 15, No. 1-2, pp. 85-100.
[8] World Historical Gazetteer. https://whgazetteer.org/, (参照 2023-11-03).
[9] Pleiades. https://pleiades.stoa.org/, (参照 2023-11-03).
[10] PeriodO: A gazetteer of periods for linking and visualizing data. https://perio.do/technical-overview/, (参照 2023-11-03).
[11] G. del Rio Riande and V. Vitale. Recogito-in-a-Box: From Annotation to Digital Edition. Modern Languages Open, 2020, Vol. 1, No. 1, p. 44.
[12] D. Zbíral et al. Model the source first! Towards source modelling and source criticism 2.0. Zenodo, https://doi.org/10.5281/zenodo.5218926, (参照 2023-11-03).
[13] P. Boot and M. Koolen. Connecting TEI Content Into an Ontology of the Editorial Domain. In E. Spadini et al. (eds.), Graph Data-Models and Semantic Web Technologies in Scholarly Digital Editing. 2021, Norderstedt, pp. 9-29.
[14] DEPCHA: Digital Publishing Cooperative for Historical Accounts. https://gams.uni-graz.at/context:depcha, (参照 2023-11-03).
[15] The Factoid Prosopography Ontology. https://www.kcl.ac.uk/factoid-prosopography/ontology, (参照 2023-11-03).
[16] 小風尚樹．構造化記述された財務記録史料データの分析手法の開発：イギリスの船舶解体業を事例に．人文科学とコンピュータシンポジウム論文集, 2019, pp. 183-190.
[17] D. L. Schwartz et al. Modeling a Born-Digital Factoid Prosopography using the TEI and Linked Data. Journal of the Text Encoding Initiative [Online], 2022, Rolling Issue, https://journals.openedition.org/jtei/3979, (参照 2023-11-03).
[18] 鈴木親彦・北本朝展．人文学資料マイクロコンテンツの実世界との双方向結合とデータポータル「edomi」．人文科学とコンピュータシンポジウム論文集, 2021, pp. 96-103.
[19] FPO References. https://www.kcl.ac.uk/factoid-prosopography/fpo-references, (参照 2023-11-03).
[20] D. Sturgeon. Digitizing Premodern Text with the Chinese Text Project. Journal of Chinese History, 2020, Vol. 4, Special Issue 2, pp. 486-498.