人文学のためのテキストデータ構造化のチュートリアル

第3章

構造化テキストの様々な記述方法

永崎研宣

version 1.0
2026.3.21 作成

本資料は、文部科学省委託事業「人文学・社会科学のＤＸ化に向けた研究開発推進事業」（JPMXP1624）において、学校法人慶應義塾が、大学共同利用機関法人人間文化研究機構から再委託を受けて作成したものです。

1．はじめに

1-1．データモデルと記述形式の区別

構造化テキストを理解するうえで重要なのは、**データモデル**と**記述形式**を区別することである。

データモデルとは、テキストをどのような構造をもつ対象として捉えるかという概念的枠組みであり、たとえば、階層構造（木構造）、グラフ構造、表構造、あるいは文字列上の区間集合などがある。データモデルと構造は必ずしも一対一対応するものではなく、一つのデータモデルが複数の構造を有している場合もある。

一方、記述形式とは、そのデータモデルを実際にファイルやデータとして書き下すための構文的手段であり、XML、JSON、CSV、RDF/Turtle などがこれに該当する。同一のデータモデルが、複数の記述形式によって表現されることも珍しくない[1)]。

1-2．語彙・スキーマの役割

データモデルや記述形式とは別に、語彙やスキーマと呼ばれる共通の記述規約が存在する。これは、どのような構造要素や関係を、どの名称で、どのような制約のもとで記述するかを定めるものであり、特定の研究分野や目的に即した知識の共有を可能にする。

たとえば TEI ガイドラインは、人文学研究におけるテキストをどのような構造的・意味的単位として捉えるかを体系化した語彙・指針の集合であり、XML という記述形式を用いるが、XML そのものと同一視されるものではない。

1-3．構造化と注釈の関係

構造化テキストにおける構造化と注釈は、しばしば連続的な関係にある。章や段落のように文書全体の骨格を与える記述は構造化と呼ばれ、語の品詞や意味、校異情報のように特定範囲に付与される情報は注釈と呼ばれることが多い。しかし両者の境界は必ずしも明確ではなく、どこまでを「構造」と見なすかは、研究目的やデータモデルの選択に依存する。

2．様々な記述形式

2-1．プレーンテキスト（plain text）

①プレーンテキストの定義

プレーンテキスト（plain text）とは、特定のアプリケーションや表示形式に依存せず、文字コードにもとづく文字列の列として記述されたテキストを指す。書体、レイアウト、色、フォントサ

イズといった視覚的情報を含まず、基本的には文字、改行、空白、記号のみから構成される点に特徴がある。このため、プレーンテキストはしばしば「非構造的なテキスト」と説明されることがあるが、この理解は必ずしも正確ではない。

②「非構造的テキスト」という理解の再検討

実際のプレーンテキストには、多くの場合、暗黙的な構造が含まれている。改行による段落や行の区切り、空白による語の分離、番号や記号による区分、見出しが行頭に置かれるといった慣習的配置などは、明示的なマークアップを伴わないにもかかわらず、人間の読解においては構造として理解されてきた。この意味で、プレーンテキストとは構造を欠いたテキストではなく、構造が明示的に記録されていないテキストであると捉える方が適切である。

③プレーンテキストと解釈の役割

プレーンテキストにおける構造は、データの内部に固定されているのではなく、読者や研究者の解釈行為によってその都度再構成される。そのため、解釈の自由度が高く、異なる研究目的や文脈に応じて柔軟に読み替えることができる一方で、解釈の再現性は低く、他者と同一の構造理解を共有することが難しいという側面も持つ。人文学研究においては、このように解釈に委ねられた状態そのものが、必ずしも欠点ではなく、研究の柔軟性や多義性を支えてきた要因でもあった。

④構造化テキストとの連続性

構造化テキストは、こうしたプレーンテキストを否定するものではなく、その延長線上に位置づけられる。多くの場合、構造化はまずプレーンテキストとして文字列を確定し、次にそこに含まれる暗黙的構造を分析し、最終的にその構造を明示的な記述として外在化するという段階を経て行われる。OCR 結果や翻刻テキストなどがプレーンテキストとして扱われることが多いのは、このような段階的理解に基づいている。

⑤自然言語処理におけるプレーンテキストの位置づけ

一方、自然言語処理（NLP）の観点から見ると、プレーンテキストは必ずしも暫定的・不完全な形式とは見なされない。むしろ、多くの言語処理手法、とりわけ統計的手法や機械学習、深層学習に基づくモデルにおいては、人為的な構造付与がなされていないプレーンテキストが好まれる傾向がある。

自然言語処理では、語の分布、共起関係、文脈的連続性といった性質を、文字列そのものから計算的に抽出することが重視される。そのため、あらかじめ特定の解釈や理論にもとづく構造が明示的に付与されていないプレーンテキストは、モデルにとって中立的で再利用性の高い資源として扱われる。この点において、プレーンテキストは、人文学的構造化とは異なる方向性での「有

益さ」を持っている。

⑥利点と限界

構造化の観点から見た場合でも、プレーンテキストには重要な利点がある。特定の理論やデータモデルに拘束されず、長期保存に適しており、編集や差分管理が容易である点は、将来的な再解釈や再構造化の可能性を確保するうえで大きな価値を持つ。一方で、プレーンテキストだけでは構造の機械的処理が困難であり、複雑な注釈や関係を安定的に保持することはできない。そのため、研究の進展やデータ共有を見据える場合には、どの段階で、どの程度まで構造を明示化するかが重要な設計上の判断となる。

⑦まとめ

このように、プレーンテキストは構造化テキストと対立する概念ではなく、構造が暗黙的に存在する状態を表す記述形態である。構造化テキストとは、その暗黙的構造を分析し、再利用可能な形で明示的に記録しようとする試みであり、両者は連続的な関係にあると理解すべきである。

2-2．XML

① XML とは何か ― 構造をもつテキストとしての記述

XML (Extensible Markup Language) の観点から見ると、構造化テキストは単なる文字列ではなく、要素（element）と属性（attribute）によって明示的に組織化された情報として扱われる。とりわけ、人文学を含む多くの分野で用いられてきた XML 系の記述では、章・節・段落・引用・注記といった単位を入れ子構造として表現することにより、文書の論理的構造を外在化し、機械処理可能な形で固定化することが重視される。そのため XML は、構造を共有し、同一のテキストに対する処理や解釈の再現性を高める手段として位置づけられる。

②タグによって構造をはっきり示す

XML による記述では、どの範囲が見出しであり、どこが注であり、どの語句が固有名であるかといった区分が、タグとして明示される。これにより、検索・抽出・変換・表示といった処理が安定し、長期保存や異なる環境への移行も容易になる。たとえば、見出しを見出しとして検索し、注記を注記として抑制して表示し、引用箇所だけを抽出する、といった操作は、プレーンテキストだけから規則的に推定するよりも、はるかに確実に実行できる。また、同一の構造を前提に、HTML や PDF への変換、索引生成、校訂情報の表示など、複数の利用形態へ展開しやすいという点でも、XML は基盤的な記述形式として強みを持つ。

③ XML だけでは決められないこと

ただし、XML が提供するのはあくまで記述のための構文であり、XML そのものが「何をどう記述するべきか」を決めるわけではない。どの要素名を用い、どの単位を区別し、どの属性を付与するかは、分野ごとの語彙やスキーマ、すなわち記述規約に委ねられる。たとえば人文学領域では、見出し・段落・注記といった一般的単位に加え、話者、地名、人名、引用の出典、校異、語形や表記の揺れなど、研究目的に応じた多様な区分が必要となりうるが、そのどれを記録するか、どの粒度で記録するかは、研究上の判断である。したがって XML は、強力な記述形式であると同時に、「どの構造を記録し、どの構造を記録しないか」という編集上の選択を不可避に伴う枠組みでもある。

④共通の記述規約としての TEI

このような記述上の選択を、個々の研究者やプロジェクトごとに完全に独立して行うだけでは、データの共有や再利用は困難になる。そのため、人文学分野では、どのような構造や単位を想定し、それをどのように記述するかについて、一定の共通理解を形成しようとする試みが行われてきた。その代表例が、TEI によって策定されているガイドラインである。TEI は、文学作品、歴史資料、写本、書簡、辞書など、人文学における多様なテキスト資料を対象に、どのような要素や属性を用いて構造や意味を記述するかを体系的に示した記述規約を提供している。

⑤ XML と TEI の役割分担

TEI ガイドラインは、すべてのテキストに同一の記述方法を強制するものではなく、研究目的や資料の性質に応じて、どの要素を採用し、どの粒度で記述するかを選択できる柔軟性を備えている。一方で、要素や属性の意味、使用条件、相互関係については明確な定義が与えられており、異なる研究者やプロジェクトが作成したデータであっても、一定の互換性と可読性を保てるよう設計されている。この点において TEI は、XML という記述形式の上に、人文学的テキストを扱うための共通の記述規約を与える枠組みとして機能している。

したがって、XML による構造化テキストの実践において重要なのは、XML という構文だけでなく、その上でどのような記述規約を採用するか、あるいは必要に応じてどのように拡張・調整するかを明示的に意識することである。TEI は、そのような判断を行う際の参照点を提供する存在であり、XML による構造化を個別的な作業から、共有可能な研究基盤へと位置づけ直す役割を果たしてきた。

⑥入れ子構造の得意な点と苦手な点

XML が得意とするのは基本的に階層構造（木構造）の表現であり、入れ子として整合的に表現できる範囲では、文書構造を明快に記述できる。しかし、実際のテキストには、複数の区分が同じ箇所を異なる観点から横断する場合がある。たとえば、韻律単位と文法単位が一致しない場

合、ページや行の区切りが文の途中に入る場合、あるいは校訂や注釈の対象範囲が段落や文の境界をまたぐ場合などがそれにあたる。このような重なり（overlap）を単一の入れ子構造として記述しようとすると困難が生じるため、XML を用いる実務では、どの構造を本文に埋め込み、どの情報を参照として外部化するか、すなわち記述方針の設計が重要になる。なお、TEI のデータモデルは階層構造には収まりきらないものであり、そのようにして階層構造としては記述しきれない構造に対応するために XML の機能を活用したいくつかの記述方法を提供している。

⑦ XML は「タグ付け作業」ではない

この点を踏まえると、XML による構造化は、単に「タグを付ける」作業ではなく、文書や資料の性質に即して、どの構造を中心に据えるかを決め、必要に応じて属性や参照、あるいは本文と分離した注釈の仕組みを組み合わせながら、利用可能性を確保する営みであると言える。XML の強みは、構造の明示と共有によって処理の再現性を高められる点にあるが、その効果は、語彙・スキーマの設計、記述粒度の選択、そして運用上の目標（検索なのか、表示なのか、分析なのか、保存なのか）をどのように定めるかによって大きく左右される。したがって、XML は構造化テキストのための中立的な容器であると同時に、研究目的と編集方針を具体的なデータの形に結びつけるための、方法論的な枠組みとして理解されるべきである。

⑧ XML は木構造だけの形式ではない

なお、XML は階層構造（木構造）を表現する記述形式として理解されることが多いが、必ずしもそれに限定されるものではない。この点を理解するうえで重要なのが、グラフ構造のデータモデルを持つ RDF との関係である。RDF は、主語・述語・目的語からなる三項関係の集合として対象間の関係を記述する枠組みであり、文書の階層構造ではなく、概念や実体の相互関係を表すことを目的として設計されている。

⑨ RDF を XML で書くという考え方

このように、データモデルとしては階層構造を前提としない RDF であるが、その策定初期においては、記述形式として XML が積極的に採用され、RDF/XML と呼ばれる構文が標準として推奨されていた。RDF を XML で記述する場合、表面的には XML の入れ子構造が用いられるため、木構造の文書のように見えるが、それはあくまで書き方の問題にすぎない。RDF の本質は、同一のリソースが複数の関係に参加し、複数箇所から参照されうるという点にあり、XML で記述した場合であっても、その全体像は木構造ではなくグラフ構造として理解される。

⑩木構造とグラフ構造を意識した XML の使い方

このことは、XML が階層構造の記述にのみ適した形式であるという理解が必ずしも正しくないことを示している。XML は、要素の入れ子によって階層的な関係を明示できる一方で、識別

子や参照を組み合わせることにより、階層を超えた関係、さらには循環を含む関係構造を表現することも可能である。RDF/XML の歴史的経緯は、XML が文書構造の記述だけでなく、グラフ構造のデータモデルを持つ情報の記述にも利用されてきたことを示す具体例であり、XML を用いた構造化テキストの設計においても、木構造とグラフ構造の関係を意識する必要があることを示唆している。

⑪ 身近なソフトウェアにおける XML の利用例

なお、XML は学術研究や専門的なデータ記述のためだけに用いられてきた形式ではない。現在広く利用されている Microsoft Office の文書形式（Word、Excel、PowerPoint）は、内部的には Office Open XML（OOXML）と呼ばれる XML にもとづく形式で保存されており、文書の構造、見出し、表、スタイル、注記などが XML として記述されている。このことは、XML が特殊な研究用途に限られたものではなく、一般的な文書作成や情報共有の基盤としても広く採用されてきたことを示している。

同様に、Web 上で用いられてきた各種設定ファイル、電子出版（EPUB）、業務システムにおけるデータ交換形式などにおいても、XML は長らく重要な役割を果たしてきた。これらの利用例は、XML が人間にとって可読性を保ちつつ、同時に機械処理にも適した形式であるという特性を持つこと、そして文書構造を明示的に保持したまま長期的に利用できる形式として評価されてきたことを示している。

2-3．JSON データ

① JSON とはどのような形式か

JSON（JavaScript Object Notation）は、キーと値の組、および配列を基本単位として構造を表現する記述形式である。もともとは Web アプリケーションにおけるデータ交換を目的として設計されたが、その簡潔さと可読性の高さから、現在では Web API、設定ファイル、データ保存形式など、幅広い場面で利用されている。構造化テキストの文脈においても、JSON は、テキスト断片とそれに付随する属性情報や解析結果をまとめて扱うための、実用的な記述形式として位置づけられる。

② JSON による構造表現の特徴

JSON では、構造は主に「オブジェクト」と「配列」によって表現される。オブジェクトはキーと値の対応関係を示し、配列は複数の要素を順序付きでまとめる。この仕組みにより、章や段落のような階層的関係を表現することも可能であるが、その構文は XML に比べて簡潔で、記述上の自由度が高い。

この簡潔さは、プログラムによる処理や他システムとの連携において大きな利点となる。

JSON は多くのプログラミング言語で標準的に扱うことができ、Web API やデータベースとの相互運用性にも優れている。そのため、構造化テキストを「文書」として保持するというよりも、「処理・交換されるデータ」として扱う場面では、JSON が選択されることが多い。

③ XML との違いと役割の違い

JSON は XML と同じく構造化データを記述できる形式である。その設計思想の違いに着目するなら、XML が、文書全体の論理構造を明示的に記述し、長期保存や処理の再現性を重視してきたのに対し、JSON は、データ項目と属性のまとまりを軽量に表現し、即時的な処理や交換を重視する傾向が強い。

たとえば、文書中の各段落に対して ID、位置情報、注釈、自然言語処理の結果などを付与する場合、JSON はそれらを一つのまとまりとして管理しやすい。一方で、章・節・段落といった文書構造そのものを厳密に記述しつつ、複雑な関係を長期的に保持したい場合には、XML の方が適していることが多い。そのため実務では、本文は XML で保持し、付加情報や解析結果を JSON で管理する、といった併用がみられることもある。

④記述規約と JSON の柔軟性

XML と同様に、JSON そのものは「何をどう記述するか」を規定するものではない。どのキーを用い、どのような構造を想定するかは、プロジェクトや分野ごとの取り決めに委ねられる。JSON にはスキーマを定義する仕組みも存在するが、実際の運用では、厳密な制約よりも柔軟性が優先されることが多い。

この柔軟性は、迅速な開発や試行錯誤には有利である一方、構造の意図が暗黙化しやすく、第三者による再利用や長期的な共有の際には注意が必要となる。そのため、JSON を用いた構造化テキストでは、キーの意味や想定される構造を文書として明示するなど、運用上の工夫が求められる。

⑤ JSON と XML の変換に見られる非対称性

実務上しばしば指摘される点として、**JSON 形式のデータを XML に変換することは容易である一方、XML で記述されたテキストを完全に JSON へ変換することは一般には困難である**、という非対称性がある。この違いは、両者がテキストと構造の関係をどのように扱うかの違いに由来する。

JSON では、テキストは通常、値としてまとまった単位で保持され、その内部構造は JSON 自身の構文としては扱われない。そのため、JSON で記述された階層的データ構造は、そのまま XML の要素の入れ子構造として写し替えることができる。一方、XML では、語や句、文の途中に要素を挿入する**インラインマークアップ**という記述方法を取りうる。このような記述が用いられる場合、連続する文字列の流れと構造要素の境界とが密接に結びつく。

インラインマークアップを含むXML文書をJSONへ変換しようとすると、どこまでを一つの文字列として扱い、どこからを構造情報として分離するかについて、追加の設計判断が必要となる。つまり、XMLからJSONへの変換は、単なる形式変換ではなく、**テキスト内部の構造をどのように再表現するかという解釈を伴う操作**になる。

この点から見ると、XMLは、連続するテキストの内部に構造を埋め込むことを可能にする記述形式であり、その特性がインラインマークアップとして活用されてきた。一方、JSONは、テキストと構造情報を比較的明確に分離して扱う傾向のある形式であると言える。この違いは、どちらが優れているかという問題ではなく、テキストと構造の関係をどのように設計するかという立場の違いを反映したものである。

⑥構造化テキストにおけるJSONの位置づけ

以上を踏まえると、JSONは、文書全体の論理構造を厳密に記述するための形式というよりも、**テキストに付随する情報を柔軟に保持・交換するための形式**として理解するのが適切である。自然言語処理の結果、注釈情報、メタデータ、表示制御用情報などを扱う場面では、JSONは非常に有効である。

このように、JSONはXMLを置き換えるものではなく、異なる目的に適した補完的な記述形式である。構造化テキストを設計する際には、どの情報を長期的な文書構造として保持し、どの情報を可変的な付加情報として扱うかを見極めたうえで、XMLやJSONを使い分け、あるいは組み合わせて用いることが重要となる。

2-4．その他の様々なデータ形式

① LaTeX ― 組版を目的とした論理構造記述形式

LaTeXは、文書の見た目を直接指定するのではなく、章・節・注・引用・数式といった**論理的構造を命令として記述する**ことを特徴とする記述形式であり、学術論文を組版するための記述形式として広く普及している。人文学においても、学術論文、校訂本文、注付きテキストなどを作成するための形式として一部では長く用いられてきた。

LaTeXでは、`\section`や`\footnote`、`\cite`といった命令を用いて、文書中の構造的役割を明示する。これにより、本文・注・参照といった区分を論理的に記述することができ、最終的には高品質な組版結果（主にPDF）として出力される。この点でLaTeXは、**文書の論理構造と視覚的表現とを密接に結びつけた記述形式**であると言える。

一方で、LaTeXは基本的に出力結果を最終目的とする形式であり、記述された構造は、再利用や再解釈を前提とした汎用的なデータ構造としては扱われにくい。たとえば、人名や地名、引用範囲などを細かく意味付けして機械的に処理したり、異なる利用目的に応じて構造を再編成したりすることは、XMLなどと比べると容易ではない。そのためLaTeXは、人文学においても重

要な役割を果たしてきた記述形式である一方で、**研究データ基盤としての構造化テキスト**という観点からは、XML に置き換えられてきた側面もある。

② Markdown ― 可読性を重視した軽量マークアップ形式

Markdown は、プレーンテキストに近い可読性を保ちながら、見出し・段落・リスト・強調などの**基本的な文書構造を簡易に表現するための記述形式**である。記号による簡単な構文を用いることで、人間にとって読みやすい状態のまま構造を付与できる点が特徴である。有名なところでは、ソフトウェアのソースコードや文書をオンラインで保存・共有・共同編集するためのサービスである GitHub において、文書作成の形式として採用されている。

Markdown では、# による見出し、空行による段落区切り、* や - によるリストなど、ごく限られた構文によって文書の骨格を示すことができる。このため、草稿作成、教材、研究ノート、Web 公開用テキストなど、迅速な執筆や共有を目的とした場面で広く利用されている。

しかし、Markdown が表現できる構造は基本的な文書構造に限られており、注釈の細かな区別、語句単位での意味付け、重層的な構造の記述といった、人文学的な精密構造化には対応していない。また、Markdown には統一された厳密仕様が存在せず、処理系によって解釈や拡張が異なる場合がある。そのため、Markdown は**最終的な構造化テキストの記述形式というよりも、構造化に至る前段階や補助的形式**として位置づけられることが多い。人文系の研究データとしては、XML のタグ付けが困難なアラビア語／文字のために採用している例がある。

3．人文系研究データのデータモデル

3-1．記述形式からデータモデルへ ― 人文系研究データ設計の視点

XML、プレーンテキスト、JSON、LaTeX、Markdown といった記述形式の違いを理解した上で、人文系研究データを設計する際により重要になるのは、どの形式で書くかという選択そのものではなく、研究対象をどのような単位と関係の集合として捉えるか、すなわちデータモデルの選択と組み合わせである。データモデルとは、テキスト、資料、人物、出来事、概念、画像といった対象を、どの抽象レベルで区別し、どのような関係として記述するかを定める概念的枠組みであり、それぞれが特定の学術分野や専門コミュニティによって策定・維持されてきたという点に特徴がある。同一のデータモデルが、XML や JSON など複数の記述形式によって表現されうることも、この点を理解する上で重要である。

①テキスト中心のデータモデルとしての TEI

人文学においてテキスト中心のデータモデルとして最も広く用いられてきたのが TEI ガイドラインである。TEI ガイドラインは、国際的な人文学研究者コミュニティによって共同で策定・

維持されているガイドラインであり、文学、歴史学、宗教学、言語学などの研究実践に根ざした豊富な語彙体系を備えている。テキストを線形に読まれる対象として保持しつつ、その内部構造（章・段落・行・語句・注・校異など）と、書誌情報や来歴、注釈を同一の枠組みで記述することを目的とする点に特徴がある。本文（`<text>`）に加え、資料内の要素間の様々な関係情報を示す ID 参照、書誌情報を扱う `<teiHeader>`、そして資料内外の関係情報を記述可能な `<standOff>` を備えることで、TEI は本文を記述する規約にとどまらず、**テキストを核とした研究データ全体を束ねる枠組み**として機能する。

3-2．書誌的抽象レベルを整理する IFLA LRM ／ FRBR

書誌的対象の整理という観点では、FRBR (Functional Requirements for Bibliographic Records: 書誌レコードの機能要件) およびそれを統合・発展させた **IFLA LRM** (Library Reference Model: 図書館参照モデル) が重要である。これらは 国際図書館連盟（International Federation of Library Associations and Institutions, IFLA）によって策定されてきた書誌情報の概念参照モデルであり、図書資料に対して**作品（Work）**、**表現形（Expression）**、**体現形（Manifestation）**、**個別資料（Item）**という異なる抽象レベルを区別することで、翻訳、版、媒体、所蔵といった差異を体系的に整理する。人文学研究においては、文献学的判断と書誌情報を概念的に接続するための基盤として、TEI と補完的に用いることが可能である。なお、FRBR は近代的な出版物の整理には有効であるが、書写や伝承の過程でテキストが連続的に変化する古典籍に対しては、作品や表現形の境界を一意に定めにくいという点で扱いにくい場合がある。

3-3．グラフ型のデータ形式と RDF

人・場所・概念などのつながりを表現するためのデータ形式としてはグラフ形式のデータがよく用いられる。この場合、「ノード（実体）」と「エッジ（関係）」で世界を記述することになる。多対多・階層・ネットワーク状の関係を自然に表すことができ、人物の関係や典拠の統合、出来事の年表、資料間のリンクなど、人文学においても様々な用途がある。Web 上で情報資源やその要素間の関係をオープンな形で表現するデータは LOD (Linked Open Data) と呼ばれ、Web の仕組み（URI とリンク）を、文書だけでなくデータそのものにも適用し、さらに再利用もできる形で公開する仕組みとして徐々に広まりつつある。従来の Web（Web of Documents）が「ページ同士をリンクする」ものだとすると、LOD は Web of Data（データの Web）を目指すものであるとされる。

このような、対象間の意味的関係を表現するグラフ型データの抽象的データモデルとして広く用いられているのが **RDF** (Resource Description Framework) である。RDF は **World Wide Web Consortium**（W3C）によって策定されており、現在広く用いられているバージョン 1.1 では主語・述語・目的語からなる三項関係（トリプル）の集合として情報を表現する枠組みを提供する [2)]。RDF は国際標準規格の一つではあるものの、それ自体では具体的な語彙を定義せず、「関係をど

う表すか」を定めるモデルに徹しているため、RDF に準拠しただけでは具体的なデータ共有にはつながりにくい。実際の運用では、国際的な共通語彙である Dublin Core や知識組織化体系の表現モデルである SKOS、人物・組織に関するオントロジーである FOAF などを用いることで、関係の内容を国際的に相互運用可能な形にする。さらに人文学の資料に特化しようとすると、博物館資料に用いられる CIDOC CRM（後述）や図書館で用いられる IFLA LRM を利用することになる。そして、さらに人文学の資料の内容に踏み込もうとした時には、TEI の語彙にもとづく概念を RDF で関係付けるという使い方が有用である。このように、RDF と TEI は対立するものではなく、**RDF が関係モデルを、TEI が語彙と文脈を提供する**という補完的関係として活用可能である。

グラフ型データの運用や検索、横断的利用においては、関係情報を効率的に処理すべく設計された RDF 等のグラフ型データを管理するデータベースが用いられる[3)]。これらのソフトウェアは関係の検索や推論、横断的利用に優れるが、一方で、データベースや特定の運用環境に依存するため、保存やバックアップの単位としては管理上の負担が大きくなりやすい[4)]。また、外部の要素とのリンクを前提としたグラフ型データの場合には、対象となった外部要素も共に保存しておかなければデータとしての意味を保持することが難しい。個人が作成したデータを保存しておくことでその人のデータ作成者としての評価が行なわれることが基本になるが、そのデータの価値がリンク先（の URI）同士を接続することであった場合には URI の永続性が評価と少なからず関わりを持つことになってしまう。これは第〇章の課題になるので参照されたい。

3-4．長期保存を支える設計原則と保存標準

人文系研究では研究成果が長く参照されることは近年の論文引用調査でも裏付けられているとおりである。したがって、研究向けのデータとして作成された場合には長期にわたって保存され利用されることが強く期待される。すなわち、長期保存においては、単にファイルを保持するだけではなく、将来の利用者が意味情報を再構築できる状態を維持することが不可欠である。この点は、OAIS 参照モデル（ISO 14721）が示す「表現情報」と「環境独立性」、FAIR 原則における「再利用可能性」、および PREMIS (PREservation Metadata: Implementation Strategies) による保存メタデータ設計の方針など、多くの保存標準で共通して強調されている。長期保存を主たる目的とする場合には、自己完結性、可搬性、バックアップの容易さ、将来の再解釈可能性といった要件が特に重要となる。

3-5．出来事中心モデルとしての CIDOC CRM

文化遺産分野では、出来事を中心に据えたモデルとして **CIDOC CRM** が用いられる。CIDOC CRM は、国際博物館会議（ICOM）の国際委員会である CIDOC によって策定されており、制作、使用、所蔵、移動といった出来事（イベント）を媒介として資料や作品を記述する。来歴情報を厳密に表現できる点で優れており、RDF との親和性も高い。一方で、同様の来歴情報は TEI に

おいても <history> 要素を用いて叙述的に記述できるため、TEI がテキスト理解のための要約的来歴を担い、CIDOC CRM が横断的・関係的な詳細記述を担うという役割分担があり得る。

3-6．アーカイブズ記述モデル

アーカイブズ分野では、長らく ISAD(G) が用いられてきた。ISAD(G) は 国際公文書館会議 (International Council on Archives, ICA）によって策定された国際標準であり、fonds、series、file、item といった階層構造を前提に、文書がどの組織や活動の中で生成されたかという文脈を重視する。一方、こうした固定的階層モデルの限界を踏まえ、同じく ICA が策定を進めているのが Records in Contexts（RiC）である。RiC は、資料、主体、機能、出来事、場所などを関係のネットワークとして捉え直す概念モデルであり、RDF や CIDOC CRM と親和的な方向性を持つが、現時点では概念モデルとオントロジーが主に提示されている段階にあり、確立した記述規約としての運用は発展途上にある。

3-7．画像とテキストを結ぶ IIIF

テキストと画像の関係を扱うモデルとして不可欠なのが IIIF (International Image Interoperability Framework) である。IIIF は、図書館・博物館・研究機関の連携によって形成された国際的コンソーシアムである IIIF Consortium によって策定・維持されており、世界中の研究図書館でデジタル画像化した資料を公開する際に採用されている。これは画像そのものの保存形式ではなく、ページ順、領域、表示方法といった構造を定義する。そのような構造は TEI で記述することが可能だが、TEI 記述のままではそれに従って Web 上の画像を操作することは容易ではない。それを IIIF が定義する API に変換することによってはじめて実質的に可能となる。これにより、写本や古典籍の画像と、TEI で記述されたテキストとを対応付ける基盤が提供されているのである。

3-8．複数モデルの役割分担と統合ハブとしての TEI

以上のように、人文系研究データで用いられるデータモデルは、それぞれが異なる分野的要請と専門コミュニティを背景に成立しており、単一のモデルで完結するものではない。重要なのは、これらを対立的に捉えるのではなく、**どの情報をどのモデルに割り当て、どのように接続するか**という設計判断である。

この点で、TEI を**統合ハブ**として扱う可能性は極めて重要である。TEI は、本文、ヘッダ、スタンドオフ記述を通じて、テキストを核にしながら、書誌情報（IFLA LRM 的整理）、来歴情報（CIDOC CRM 的整理）、RDF 的な関係データ、IIIF への参照などを一体として保持できる。とりわけスタンドオフ記述は、RDF のデータモデルにもとづく関係情報を、TEI という人文学的語彙を用いて XML としてシリアライズし、**自己完結的な保存単位に内包するための仕組み**として理解できる。

3-9．Cambridge Digital Collection Platform に見る実践例

このような設計の有効性は、ケンブリッジ大学図書館が開発・運用する **Cambridge Digital Collection Platform** [5] における実践にも見られる。同プラットフォームでは、TEI ファイルをデジタルコレクションの基盤データとして保持し、TEI に含まれる本文構造、ヘッダ情報、スタンドオフ記述をもとに、検索、表示、画像連携といった機能を実現している。これは、TEI を保存の正本（canonical）とし、必要に応じて他のデータモデルへ展開するという設計が、大規模デジタルコレクションにおいて現実的かつ有効であることを示す例である。

人文系研究データモデルの組み合わせと展望

以上の検討をまとめたのが表 1 である。人文系研究データにおいては、複数のデータモデルを役割分担させつつ、TEI を統合ハブとして据える設計が、保存・運用・再利用の要請を同時に満たす現実的なアプローチの一つとなり得る。

情報の種類	TEI ガイドラインにおける 対応する要素	想定される データモデル	説明
テキスト本文	<text>	TEI	章・段落・行・語句など、線形的に読まれるテキスト構造
本文中の言及（人物・地名・概念）	<text>	TEI	テキスト中に現れる言及そのもの（参照先は ID や URI で管理）
書誌情報・版情報	<teiHeader>/<fileDesc>	FRBR / IFLA LRM	作品・表現形・体現形の区別、版・出版・媒体の情報
資料の出典・所蔵	<teiHeader>/<sourceDesc>	LRM / ISAD(G)	資料がどの文脈で生成・管理されてきたか
写本・物理的記述	<teiHeader>/<msDesc>	TEI / LRM	形態・支持体・書写状況などの記述
制作・来歴・移動（概要）	<teiHeader>/<msDesc>/<history>	TEI	テキスト理解のための説明的・要約的来歴記述
制作・来歴・移動（詳細・横断）	<standOff> または RDF ストア・グラフ DB 等	CIDOC CRM	出来事中心モデルにもとづく厳密・再利用可能な記述
人物関係・概念関係（保存用）	<standOff> または RDF ストア・グラフ DB 等	RDF（データモデル） ＋ TEI（シリアライゼーション）	RDF 的な関係データを、保存のために XML として内包
人物関係・概念関係（運用・検索）	RDF ストア・グラフ DB 等	RDF 等のグラフ型データ	複数資料を横断する検索・推論・連携
注釈・校異・解釈層	<standOff>	TEI / RDF 等のグラフ型データ	後付け・多層的・可変的な注釈情報
画像との対応	<teiHeader> ＋参照	IIIF	ページ順・領域とテキストの対応関係

人文学研究データにおける役割分担（表 1）

注

1　本章におけるグラフ型データと RDF に関する記述は、東京大学大学院人文社会系研究科の大向一輝先

生にご助言をいただいた。

2 原稿執筆時点ではワーキングドラフトの段階だが、RDF のバージョン 1.2 では、RDF のトリプルに対して graph name を付与することで区別するクワッド構造が提唱されており、データ構造としては複雑化してしまうものの、人文系のニーズにはよりフィットする場面が出てくるだろう。

3 RDF を格納するいわゆるトリプルストアと呼ばれるソフトウェアには、GraphDB、Virtuoso、Apache Jena Fuseki などがある。一方、グラフの各要素（ノードやエッジ）に任意のプロパティをする、プロパティ・グラフ系と呼ばれるものとしては、Neo4J、ArangoDB などがある。

4 特にテキスト資料をデータ化する場合には、それを TEI 化したテキストデータに関係情報を ID 参照で埋め込んでおき、必要に応じてそこから RDF を生成・展開するという設計は有効である。

5 https://cudl.lib.cam.ac.uk/about-dl-platform