

XML による歴史の知識基盤表現

The Use of XML to Create a Historical Knowledge Base

松井 正志[†]
Masasahi Matsui中平 勝子[†]
Katsuko T. Nakahira三上 喜貴[†]
Yoshiki Mikami

1. はじめに

HTML は文書構造を定義するものであり、文章の論理的構造を記述するものではない。そのため、HTML で記述された文書からデータを取り出して利用することは難しい。そのような問題を解決するため、セマンティック Web¹⁾ が提唱された。セマンティック Web は、文書構造ではなく意味的構造を記述することで、コンピュータが自動的にデータを利用できるような Web、すなわちデータの Web を目指す。

一方、筆者らは歴史知識、とりわけ技術史に関する歴史知識をいかにして共有可能な知識基盤として構築するかという課題に取り組んでいる。この点で、従来の Web 上にある歴史知識は文書・年表形式で記述されていたり、データベースとして蓄積されているが、その仕様や記述項目について標準となるフレームワークが存在しない、あるいは公開されていないため、第三者がコンピュータを用いて自動的に再利用することは難しい。個別に独立した当事者たちによって構築される歴史知識を第三者が自由に再編・統合して利用できる環境を構築することが筆者らの目指すところであるが、その実現への道はまだ遠い。

そこで、本稿ではこの目標に向けた第一歩として、歴史的事象を記述するフレームワークを提案すると共に、これを具体化するための Microformats の語彙を定義した。

2. 先行研究

歴史知識を記述する試みとして、HSML (Historical Space Modeling Language)²⁾ がある。HSML は、電気学会の電気技術史技術委員会が 1996 年に電力系統技術歴史データベース作成³⁾ に用いた歴史記述言語である。HSML によって記述されたデータは、GUI「曼荼羅」によって閲覧できる。また、海外では HEML(Historical Event Markup and Linking) プロジェクト⁴⁾ が歴史記述のための XML スキーマと、それによって記述された歴史情報を SVG を用いて年表形式や地図形式で表示する Web アプリケーションの開発を行っている。

また、記述に関する試みが行われる一方で、歴史を Web 上で視覚化する様々な試みが行われている。年表表示の場合、国内では @nifty Timeline⁵⁾、海外では SIMILE の Timeline⁶⁾ が挙げられる。@nifty Timeline は @nifty の提供するサービスで、Web 上に年表を作成できる。@nifty Timeline の特徴は、ユーザが縦軸を自由に設定できるという点である。横軸は時間の経過を示すが、縦軸が何を示すかは年表を作成するユーザの自由である。またインターフェースは Flash、システムは



図 1 @nifty Timeline



図 2 Knecht クロニクル

Ruby on Rails によって実装されており、非常に軽快な操作感を実現している。SIMILE の Timeline は、SIMILE というマサチューセッツ工科大学コンピュータ科学・人工知能研究所、マサチューセッツ工科大学図書館、W3C の 3 機関による共同プロジェクト内で開発された AJAXy なソフトウェアである。Timeline は JavaScript によって実装されているため、個人が自由に Web サイトやブログなどに組み込めることができる。また、XML・JSON・SPARQL といった多様な形式に対応しているため、他のソフトウェアや Web 上で公開されている様々な API と連携が行いやすい。

これらのような時系列での歴史表現の他にも、Knecht クロニクル⁷⁾ のような地図上に歴史を表示させる試みもある。Knecht クロニクルは Knecht の開発したサービスで、Google や Wikipedia などの API を利用し地図上に事象をマッピングする。スクロールバーによって年を指定することで、その年にどこで何が起こったかを表示する。

このように歴史に関しての記述や視覚化表現などを行う様々

[†] 長岡技術科学大学

な試みが行われている。

3. 歴史記述のフレームワーク

本稿では、歴史的事象の記述フレームワークを格文法における深層格に基づいて構築し、これを実装するための Microformats の語彙を定義した。

3.1 記述項目

フィルモアの深層格は、格文法で用いられる格の考え方を意味的なものにまで拡張し、動詞に対しての役割を示したものであり、「動作主格・経験者格・道具格・源泉格・目標格・場所格・時間格」の8つの格からなる。これを歴史的事象に当てはめると次のようになる。まず、動作主格は事象を起こした人物あるいは組織となる。経験者格は、事象は事実の記述であり心理事象ではないため、該当する項目はない。道具格はある事象の原因を示すために必要である。対象格は事象の主題となることが多い。例えば「エジソンが蓄音機を発明した」の場合は、発明したという動詞よりも蓄音機という対象が重要である。よって、対象は記述項目として必要である。源泉格と目標格は動詞に何をを用いるかによって大きく変わる。そのため、本稿では記述項目として含めない。場所格は事象の地理的分布を示すために必要である。時間格は事象を時間的分布を示すために必要である。

これらの検討結果から歴史的事象の記述項目として「人物・対象・原因・場所・時間」の5項目が必要ということになるが、対象格は先に述べたように事象の主題となることが多い。他方、個々の歴史的事象をする特定のための識別子が必要であるが、事象名は多くの場合「動詞+対象格」で表されるため、対象格を識別子として利用することが適当ではないかと考えた。そこで、対象格の代わりにこれをイベント名として記述フレームワークに加えた。また、他の参考文献へのリンクがあった方が便利である。HEML や予定の記述に用いられる iCalendar など、多くの形式で参考 URL へのリンクが記述できる。そのため、深層格には該当する項目がないが、参照情報を記述する項目が必要である。以上をまとめて、歴史記述のフレームワークは「イベント名、人物、場所、時間、原因、参照情報」の6項目とした。(表1)

表1 歴史記述のフレームワーク

| 項目 | 対応する格 | 格の説明 |
|-------|-------|--|
| イベント名 | 対象格 | 移動する対象物や変化する対象物。あるいは、判断、想像のような心理事象の内容を表す役割。 |
| 人物 | 動作主格 | ある動作を引き起こす者の役割。 |
| 場所 | 場所格 | ある事象が起こる場所および位置を表す役割。 |
| 時間 | 時間格 | ある事象が起こる時間を表す役割。 |
| 原因 | 道具格 | ある事象の直接原因となったり、ある心理現象と関係して反応を起こさせる刺激となる役割。 |
| 参照情報 | - | - |
| - | 経験者格 | ある心理事象を体験する者の役割。 |
| - | 源泉格 | 対象物の移動における起点、および状態変化と形状変化における最初の状態や形状を表す役割。 |
| - | 目標格 | 対象物の移動における終点、および状態変化と形状変化における最終的な状態や結果を表す役割。 |

3.2 Microformats

本稿ではメタデータの記述方法に Microformats⁸⁾ を用いた。Microformats は、(X)HTML の属性値に一定の規則に沿った名前づけを行い、文書からメタデータを抽出できるようにする。Microformats は (X)HTML の文法に用意されている属性値を利用するため、現存するコンテンツに対して適用しても、その構造をほとんど変更せずに済むという利点がある。本稿では既にある文書に対しての適用も考慮しているため、このような利点を持つ Microformats によって歴史的事象の記述を行った。本稿が提案する Microformats の語彙を表2に示す。

4. 因果関係の記述と表現

歴史的事象間の因果関係を表現しうるフレームワークの開発は、本研究の大きな目標の一つである。本研究では歴史的事象をノード、因果関係をエッジとしたグラフ構造として因果関係を表現し、図3のような木構造として表現する描画ソフトウェアの設計及び実装を行った。

4.1 因果関係の記述

因果関係において原因となる歴史的事象を原因ノードと呼び、結果となる事象を結果ノードと呼ぶ。本稿では因果関係を結果ノードから原因ノードへの単方向のエッジで表した。

エッジを双方向のエッジにした場合、2つのエッジに矛盾が発生しないようにする必要がある。そのため、ある事象について原因あるいは結果を記述したときに、その対象となる事象に対してもデータの変更を行わなくてはならない。本稿では Web 上での共有を想定しているため、必ずしも対象となる事象

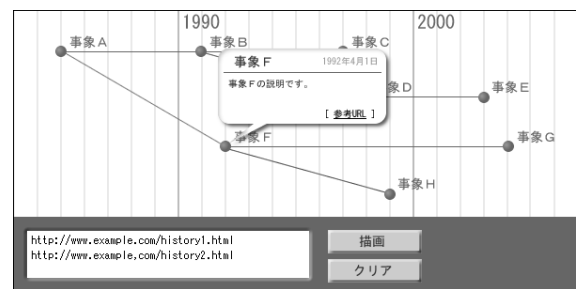


図3 因果関係を表示するソフトウェアのイメージ図

表2 本稿で定義した Microformats の語彙

| 分類 | 要素 | 属性 | 値 | 項目 | 記述箇所 | データ型 | 備考 |
|------|------|-------|-------------|----------|----------|------|-----------------|
| イベント | 指定なし | class | event | イベント名 | title 属性 | 文字列型 | |
| 時間 | 指定なし | class | start_date | 開始期日名 | 要素内 | 文字列型 | |
| | | | | 開始期日 | title 属性 | 日付型 | |
| | 指定なし | class | end_date | 終了期日名 | 要素内 | 文字列型 | 期日に範囲の無い場合は省略。 |
| | | | | 終了期日 | title 属性 | 日付型 | |
| 場所 | 自由 | class | location | 地名 | 要素内 | 文字列型 | |
| | | | | 緯度経度 | title 属性 | 数値型 | 緯度経度をカンマ区切りで記述。 |
| 人物 | a | rel | participant | 人物名 | title 属性 | 文字列型 | |
| | | | | 人物 URL | href 属性 | URL | |
| 原因 | a | rel | cause | 原因名 | title 属性 | 文字列型 | |
| | | | | 原因 URL | href 属性 | URL | |
| 参照情報 | a | rel | evidence | 参照情報名 | title 属性 | 文字列型 | |
| | | | | 参照情報 URL | href 属性 | URL | |

のデータに対して変更を行えるとは限らない。このような理由から、単方向のエッジとした。単方向のエッジの場合、片方の方向からの遡及波及関係しか辿ることができないという問題がある。この問題については、解決の方策について現在考察中である。

また、原因と結果のどちらを用いて関連を表すかについて、本稿では原因を用いた。ある事象について記述する場合、その原因について特定することは比較的容易であるが、波及先の事象の全てを網羅することは非常に困難であると考えられるからである。

4.2 ソフトウェアの概要

本稿で作成するソフトウェアは、サーバ側とクライアント側の二つに分けられる。サーバ側のソフトウェアは PHP 5.1、クライアント側のソフトウェアは ActionScript 2.0(Macromedia Flash 2004 MX) によって実装した。特にクライアント側については Java アプレットや JavaScript などの選択肢が存在するが、プラットフォーム間の差異やプラグインの普及率、描画機能や開発効率を考慮した結果、Flash が最も適当であると判断した。

サーバ側のソフトウェアは、Web 文書にアクセスし、文書中の Microformats によって埋め込まれた情報を取得する。そして、取得した情報をクライアント側のソフトウェアの対応する形式に変換し、クライアントに返す。

クライアント側のソフトウェアは、Microformats によって歴史が記述されている Web 文書の URL をサーバに送信し、レスポンスとして送られた情報を画面に描画する。

4.3 データ変換

本稿では Microformats によって記述されたデータの抽出に XSLT(XSL Transformation) を利用した。XSLT は XML 文書を読み込み、データを加工・出力するための変換言語である。

また本稿では、XSLT で直接ソフトウェア固有の形式に変換するのではなく、RDF に一旦変換してから、それをソフトウェ

ア固有の形式に変換するという方法をとった。これは、RDF をハブとすることで、様々なソフトウェアと組み合わせることを容易にするためである。また、RDF はトリプルと呼ばれる主語・述語・目的語の3つからなるグラフ構造を持つ。因果関係もまたグラフ構造を持つため、RDF のトリプルによって表せる。このような背景から、本稿では RDF をソフトウェア間のハブとして使用する。

5. おわりに

本稿では、歴史的な事象を XML を用いて記述するための6項目からなるフレームワークを提案し、これを具体化するための Microformats の語彙を定義した。このうち因果関係の表現については既存の視覚化表現ツールでは対応不可能なため、これを表現しうる独自のツールの設計を行った。今後の課題として、このツールを完成させると共に、本フレームワークに基づくデータのオーサリングツールの開発を行い、歴史の知識基盤表現を記述/表示の両面から支援するツールとして完成させたい。

参考文献

- [1] W3C Semantic Web Activity, <http://www.w3.org/2001/sw/>.
- [2] Matsumoto, Y. and Yamada, A.: An association based management of reusable software components, *Annals of Software Engineering*, Vol. 5, pp. 317-347 (1998).
- [3] 電気技術史データベース体系化調査専門委員会：電気技術史データベースに関する調査検討, 電気学会技術報告, No. 991, pp. 9-11 (2004).
- [4] Historical Event Markup and Linking Project, <http://www.heml.org/heml-cocoon/>.
- [5] @nifty TimeLine, <http://timeline.nifty.com/>.
- [6] SIMILE Timeline, <http://simile.mit.edu/timeline/>.
- [7] Knecht クロニクル, <http://chronicle.knecht.jp/>.
- [8] microformats, <http://microformats.org/>.