Vol.2017-CH-114 No.7
2017/5/13

Miradorを利用したクラウドソーシングによるコラボレーション・システム ——人文学の共同研究における分析と理論の構築を支援するシステムの提案

*佐藤 正尚[1,a)]　*太田 一行[2,b)]

概要：クラウドソーシングによるコラボレーションは，これまで人文学研究の共同研究の中では，翻刻・注釈の労働集約的作業のため行われる傾向が強かった．それに加えて，本発表では，共同での分析と理論の構築を支援するシステムを提案する．具体的には，IIIF 対応ビュワー Mirador の拡張版を提示し，その理論的背景を述べる．そして，IIIF の広がりとデジタルコレクションの利用法の変化を踏まえて，技術的な課題や今後の展開を述べる．

Collaboration system based on crowdsourcing with Mirador – Proposal of a system to support analysis and theory in collaborative research of humanities

*Masanao Sato[1,a)]　*Ikko Ota[2,b)]

Abstract: So far, collaboration in crowdsourcing has tended to be used for labor-intensive tasks of transcription and annotations in collaborative research on humanities. In addition, we propose a system that supports analysis and construction of theory in collaboration. Specifically, we present an extended version of Mirador, the viewer for IIIF, and describe its theoretical background. Based on the spread of IIIF and the change of usage of digital collection, finally, we show some technical problem about our viewer and next steps.

1. はじめに

本発表では，人文学における共同研究を分析と理論から支援するウェブアプリケーション「VCS-Mirador」の理論的背景とその開発について報告する．

2. 研究の背景

人文情報学の主要な議論の 1 つにオープンデータに関する話題がある．最近の永崎研宣 [6] の議論を参照すれば，オープンデータで公表される資料画像は研究成果の公表に際して，資料画像を添付することができるので成果物を実証的に確認できる点で読者に対する利便性が向上し，研究者コミュニティでの成果物の検証コストを下げることができると言える．また，そのことによってアーカイブが資料画像を公開できない状態になっても，デジタル資料画像が Web から消失するのを防ぐことができる．なぜなら，オープンデータであれば誰かがそれを著作権法の範囲内で複製

[1] 東京大学
The University of Tokyo, 3-8-1 Komaba, Tokyo, 153-8902
[2] 京都大学
Kyoto University, Honmachi, Sakyo, Yoshidahoncho, 606-8501
[a)] masasato@phiz.c.u-tokyo.ac.jp
[b)] *は共同筆頭著者。ota.ikko.48v@st.kyoto-u.ac.jp

して公開しつづけることができるからだ．

最近では，資料画像の公開と共有の方法を共通化する国際的なフォーマットとして IIIF[*1]が提案された．IIIF の Image API と Presentation API に対応して公開することでデジタル資料画像は機械処理しやすいフォーマットをもち，IIIF に対応したビュワーを使うことでローカルでも閲覧が容易になる．これによって，オープンデータをアーカイブスとの関係を切ることなく利用することができるようになった．現在，DPLA や Gallica をはじめとする各機関が IIIF に対応した URI を提供し[*2]，今後のオープンデータ公開に際して IIIF の利用が広まっていくと考えられている．

しかし，オープンサイエンス・オープンデータにも少なからず問題がある．例えば，資料画像を編集した後に付与した座標情報は元画像とのリンクが切れてしまうとデータとして意味をなさなくなってしまう．また，そもそもオープンデータによって各地で資料画像を公開できるようにすることでかえって元のアーカイブスへのアクセス回数が減ってしまい，アーカイブスの維持自体が難しくなってしまうというジレンマを抱えている．こうした問題が解決されることがオープンデータを利活用していくうえで解決されることが望まれる．

さて，こうしたオープンデータの状況は，人文学者にとって利益でもあり問題でもある．利益である点は，資料取集の時間と費用が大幅に削減されることだ．一方で問題となる点は，大量に公開されているオープンデータをどのように利活用していくかの議論をし続けていく必要があることだ．増え続けるオープンデータの利活用は，現在，クラウドソーシングや Web コラボレーションによって資料の翻刻作業を進めるといった形で利用されてる [2], [7]．本発表はこうした翻刻作業の他に，発表者の研究領域である文学研究に注目して，すでに開発されている IIIF 対応ビュワーである Mirador[*3]を拡張することで，共同研究における分析と理論の構築を支援するシステムを考案した．

3. どうして Mirador なのか

発表者が Mirador の拡張を共同研究に際して目指した理由は，2 つある．

最初の理由は，その人文学に適した諸機能である．まず，Mirador の複数画像をマルチウィンドウで表示・拡大縮小できる機能は，複数の資料画像の比較が行える点で研究に資するところが大きいからである．例えば，異本を研究対象としている場合，それぞれの異本を比較検討する必要があるので，マルチウィンドウは研究者にとって優れたものとなる．また，IIIF Presentation API の規格に対応した注釈機能があり，これを利用することでクラウドソーシングによるコラボレージョンを行いやすい．

次の理由は，Mirador を利用した共同研究の事例がすでにあり，その有効性が確かめられているからである．例えば，Jeffrey C.Witt と Rafael Schwemmer は，IIIF Presentation API の注釈のフレームワークに着目した共同研究のシステムを作り，Mirador など IIIF 対応ビュワーに対応させている [5]．

Mirador を中心に共同研究を行なっている事例として他に代表的なものには，大規模な仏教図像データベースである SAT 大正蔵図像 DB を挙げることができる [7]．SAT のデータベースで配信されている資料画像を閲覧する際には Mirador のマルチウィンドウが有効である．また，画像の注釈をつける独自の検索システムを構築することで，図像の一部分からでも検索できることを可能している．そうした注釈は各地の研究者による Web コラボレーションによってなされている．

以上の理由から，Mirador はオープンデータの利用状況に適応していて，人文学における共同研究で用いるツールとして有効であると考えられる．したがって，本研究では，現時点では Mirador をシステムの中心とした開発を目指していきたい．なお，紹介した共同研究の事例はいずれも，Web コラボレーションでの共同での翻刻作業だが，人文学研究では，翻刻の後に行う解釈も重要であることから，本研究では解釈をいかにしてシステム上に効果的に組み込んでいくかについても検討したい．

4. システム開発の理論的背景

人文学における共同研究の可能性は，上述のような実作業における Web コラボレーションとは別の文脈での議論と実践も行われてきている．その一例として，本発表で紹介するシステムの開発手法の理論的背景をなす，Rockwell と Sinclair が提唱する *Agile Humanities*（以下，AH）という概念を取り上げたい．

Rockwell らは，デカルトが『方法序説』で示した思考法が現在の人文学者に深い影響を与えていると考え，「私たちが徹底的に考えるべきであるとする様々な思考にいまなお強い影響力を持っている，単独であること，疑うこと，反省を実践するという基本的なモデルを人文学者に与えた[*4]」と指摘している．それに対して，Rockwell らは人文学とコンピュータサイエンスが協同するための新しい研究法が必要だと主張している．そこで，Rockwell らはコンピュータサイエンスの具体的実践の一例としてソフトウェア開発を取り上げており，特に Extreme Programming（以下，XP）という開発方法に注目している．

XP では，作成するソフトウェアの全体を細部まで決定

[*1] http://iiif.io
[*2] http://iiif.io/community/#participating-institutions
[*3] http://projectmirador.org
[*4] 文献 [3]，1 頁．

することを優先しコードの作成を後回しにすることが，伝統的な手法とみなされている．XP はそうした伝統的な手法に対して，まず必要なコードを作成してから，実験・考察・改善を何度も繰り返すことでソフトウェアを完成させる手法である．具体的には，全体的な細かい指示に沿うのでなく，小さな部分を必要だと判断したものから順次コードしてテストを重ねる．そうすることによって，生産性を上げ、システムの安全性を担保することが可能だとされている[*5]．

こうした XP の主要な特徴の 1 つはペアプログラミングを要求していることにある．ペアプログラミングでは，1 人がコードを書いている時についテストを省略してしまったり，リファクタリングを延期してしまうのを，もう 1 人が会話を通して是正することでソフトウェア開発を円滑にしようとするものである[*6]．Rockewell らはこれを人文学研究の中でも，とりわけ自然言語処理や統計的分析などのコンピュータ処理が必要な研究に応用しようとした．ペアプログラミングがちょうどコーダーとコーチのような役割を分担しているとすれば，AH では「設計者／プログラマー」と「解釈者／学者」の役割を分担しながら作業を進めることが提案されている[*7]．

XP の立場からみると従来のソフトウェア開発が事前に対象の分析を重ねることでコードを書くのを後回しにしてしまうという問題があったように，AH から見ると従来のデカルト的研究は，理論を作り上げる前に研究者による長い準備期間が必要になるという問題を抱えていたと言える．

AH においては，理論とは，テキストの読解の訓練を積んだ人文学者とテキスト解析ソフトの扱いに慣れている技術を持った研究者や，プログラミングを書ける技術者が組むことによって短いサイクルで解釈を次々に検討していくことで作られてるものとされている[*8]．いわば「設計者／プログラマー」と「解釈者／学者」による共同作業と言える．

人文情報学ではすでに取り上げたように，クラウドソーシングによるオンラインの共同作業が注目されてきた．それに対して，AH のペアワークの概念が重要なのは，テキストの解釈や理論の構築といった側面でも，共同作業による研究が大きな意義を持つ点を強調していることである．そして，Rockwell らの共同研究は以下に述べる点でさらに発展させることができる．というのも，ペアワークを前提としている AH には XP の他の利点も取り入れることができるはずだからである．XP を定式化した 1 人である Kent Beck はペアワークを XP の基本としているものの，あるプロジェクトを進めるために大人数のプログラマーをどのように配置すれば最大の効果が出るのかがそもそも課題とされていた．その一方で，Rockwell らは XP のペアワークについてのみ取り上げており，XP がもともと複数人が参加するプロジェクトのマネジメントの方法である点を検討していない．そこで，この複数人が参加するプロジェクトでテストやリファクタリングするコード（人文学者にとっての分析対象となる資料）をどのように扱っているのか重要な点となる．すると，XP を代表とするアジャイルなソフトウェア開発の現場では，ソースコードを管理するための Version Control System（以下，VCS）に注目する必要がある．本発表者は，この VCS こそ，Rockwell らの提案する AH をさらに拡張できるものであると考えている．

5. VCS とテクスト生成論の共通点

VCS は，全体で 100 行以上を超えるようなコードが求められるプロジェクトを 2 人以上で共同で作業する必要がある時に，コードやファイルの履歴をリポジトリと呼ばれるファイルに保存したり，テスト用のコードとプロジェクトで提出するコードを別々に発展させたりできる．また，近年では Distributed Version Control System（分散型バージョン管理システム）のように，リポジトリ自体をクライアントのマシンにコピーすることでサーバに依存しないシステムがソフトウェア開発では一般的となっている．

こうした VCS の歴史を調査した Nayan B. Ruparelia によれば [4]，1972 年にベル研究所で Marc J. Rochking が開発した Source Code Control System（以下，SCCS）が VCS の初期事例だとされている．その後，1980 年代になると，SCCS でのファイル操作では一連のコマンドを入力していく必要があったのを部分的に自動化していくことでより操作性の向上した Revision Control System（以下，RCS）が Purdue 大学の Walter F. TIchy によって開発された．例えば，UNIX の diff コマンドのように変更以前以後のファイルの差分を簡単に見ることができるようになった．ところが，現在の VCS では一般的だと考えられているブランチのインポートやチェックアウトしたブランチでの複数人による同時作業ができるといった「クライアント／サーバ・モデル」は RCS の設計には含まれていなかった．つまり，RCS では，他の人のブランチをマージすることもできなかったし，チェックアウトしたブランチで複数人が作業することもできなかった．

この「クライアント／サーバ・モデル」が RCS に組み込まれるのは，1985 年 11 月 23 日である．その日，Dick Grune が Amsterdam Compiler Kit と呼ばれる C コンパイラーを学生との共同作業で開発するために，Bourne シェルのシェルスクリプトで書かれた Concurrent Versioning System（CVS）を UNIX コミュニティの 1 つであった comp.sources.unix に投稿した．1986 年 6 月 23 日に改訂版が Grune によって投稿されて，RCS のフロントエンドとして CVS が開発されたことが説明された．

*5 文献 [1]，45-51 頁.
*6 文献 [1]，104 頁.
*7 文献 [3]，6 頁.
*8 文献 [3]，8 頁.

現在では，VCS は「クライアント／サーバ・モデル」が前提とされているが，CVS でこのモデルが導入されて以来，Subversion や，さらに分散型の特性を持った git といった多くの VCS を生み出すことになり，大規模人数でのソフトウェア開発で一般的に用いられるようになっていた．

ところで，これと同じように複数人が作業に従事する場合が文学の研究においてもある．発表者の専門領域では，フランスの近代テクスト草稿研究所[*9]で行われているテクスト生成論的研究[*10]が代表的である．生成論は，Gallica が大量の草稿群の資料画像を公開しているために，今後ますます重要な研究方法になっていくものと考えられる．

生成論はフランスでは 20 世紀初頭に生まれた．生成論以前では，文学研究で対象となるのは刊行された書籍（いわゆる決定版）だけであるとされ，作品の成立過程は研究では注目されなかった．日本に本格的に生成論を紹介した松澤和宏によれば，「理念上の決定稿を目指すいわゆる本文校訂と異なって，生成批評版は，様々な執筆段階の資料を生成過程の所産として受け止め，読解可能な対象として構築して維持することを眼目としている[*11]」．つまり，生成論の登場によって文学研究でメモ書きや草稿，赤入れされたゲラ刷りなどをいわば作者のコミットメントの履歴として扱うことになったのだ．ただ，こうした生成論的研究の成果を校訂版でどのように表現するのかは 20 世紀中頃から議論が続いている．人文情報学でも，Text Encoding Initiative に所属するグループの 1 つに，生成論に基づいた校訂をどのように TEI P5 に従って記述するかを話し合う Genetic Encoding が 2008 年より本格的に活動を始めている[*12]．

ところで，生成論において注意したい点は，書籍として刊行された後に，作者自身が書き込んだ作品へのコメントや，第 2 版での変更点なども研究対象となっているので，作品とみなされているものは，作者がその時点でコミットメントをしなくなった，あるいはできなくなったものであると考えられる点である．AH で参照されていた XP によるプロジェクト運営でも似たような点がある．それは，XP のプロジェクトもまた，始まる時点でその終わりが具体的に決まっていない点である．XP では，ユーザの運用しているソフトウェアに新しい機能を追加することがなくなった時がそのソフトウェアの寿命であると Beck は指摘している[*13]．完成品としてのソフトウェアに向かってコーディングしていくのではないという考えは，生成論における草稿がある作品の完成の状態に向かって漸進していくものではないという考えと似ている．XP と生成論は，複数人での作業という点のほかに，終わりの定まっていないものをどう取り扱うかという点でも共通している．

すでに見たように，生成論と VCS の共通点があり，XP と生成論に共通点があった．だとすれば，XP に着想を得ている AH を拡張するために必要なのは，VCS のような考え方で研究者の解釈の蓄積を管理するツールであると考えられる．

6. Mirador への AH の適用

AH で技術者と学者の間でアイディアがその場でテストされるのとは異なり，生成論は大規模な人数が参加するものの，その情報交換やアイディアを出す場面は AH で想定されているように頻繁ではないし，同じ草稿を同時に 2 名以上で扱うこともあまり一般的ではない．理由はいくつか考えられるが，各研究者が世界中に散らばっているために共同研究が難しいということと，草稿に対する注釈作業を総合するのは書籍となる場合だけであり，「クライアント／サーバ・モデル」のように，各研究者がローカルに蓄積した注釈の情報を，最終的にはサーバサイドで管理するシステムが整っているとは言えないからだと考えられる．

以上の問題点と前節の内容を踏まえると，Mirador に AH を適用するためには次の 4 つの条件が求められる．

(1) IIIF に対応した GUI によって，共同研究を円滑にする．
(2) 作者のコミットメントの履歴を残しつつ，研究者の調査のコミットメントの履歴も残すことができる．
(3) 共同研究としての成果をまとめることができる．
(4) データの保存・出力・交換形式としては TEI の Genetic Encoding などに対応することができる．

(1) の条件が必要なのは，前節で述べたように，IIIF とオープンデータ化によって進むはずなので，共同研究が人文学でも今後盛んになってくると考えられるからである．データはサーバに預けておき，閲覧や操作はローカルに行うことが一般的になるを想定した場合に求められる実装の条件である．また，技術者だけでなくより多くの研究者に利用してもらうために，その操作にはコマンド操作を廃して，できるだけ簡単な操作にする必要があるため WYSIWYG を備えた GUI のツールで研究できるようにすべきである．(2) は，これまでの共同研究で主流だったクラウドソーシングや Web コラボレーションによる翻刻作業に加えて，資料の解釈をすることを考えた場合に必要な条件である．資料を単に活字にするだけでなく，ある文章がどこからの引用なのか，また，ある文章はどのような意味を持っているのかなども注釈に加える必要がある．こうした共同での注釈の積み重ねが，新しい解釈の基礎となる．(3) では Rockwell らが提唱した AH のペアワークを例にとると，その作業で行われたテストの履歴を全て保存できるような機能が求められる．これは生成論の

*9 http://www.item.ens.fr
*10 La génétique des textes が原語．他にも作品生成論（génétique textuelle）や生成批評（critique géngétique）などといった表現がなされるが，本発表では以下で生成論と表記する
*11 文献 [8]，88 頁．
*12 http://www.tei-c.org/Activities/Council/Working/tcw19.html#index.xml-front.1_div.1
*13 文献 [1]，138-139 頁．

共同研究でも重要であり，ある解釈を誰がいつどの時点で行ったのかをすぐに調べておくことができるので，その後ある解釈を発展させていく場合に，担当者の割り振りや研究の方向性を定めやすくなるからである．最後の(4)は，IIIF Presentation API のフレームワークに従って必要となる条件である．Presentation API では，資料画像に追加する翻刻のデータは TEI や METS/ALTO など別の形式で提供しつつリンクすることになっている．よって，今回の開発に関わりのある Genetic Encoding だけでなく，TEI P5 の Representation of Primary Sources[*14]や Critical Apparatus[*15]に対応することも検討している．

本発表では，この 5 つの条件を備えたツールを VCS-Mirador（以下，VM）という名称で呼ぶことにする．VM は，AH を背景とした人文学における共同研究を，注釈機能を実装したビュワーの Mirador によって，VCS の仕組みを参考にしながら，文学研究を促進させるツールである．

7. ウェブアプリケーションの仕組み

VM は Mirador を利用した Web アプリケーションである．メールアドレスなどを登録してアカウントを取得することで利用することができる．

アカウントを取得したユーザはグループを作成することができ，グループ内でだけ閲覧できる資料のリンク情報を管理でき，資料に付与された注釈もまた閲覧できる．グループの管理を行うシステムは Ruby on Rails のログイン機構のフレームワークに従って開発した．

Mirador の注釈機能は IIIF Presentation API 2.1 に準拠しており，アノテーションの構造は Open Annotation[*16]に対応している．Mirador でアーカイブスの資料画像を閲覧するのに必要な Manifest URI には，その資料に付された注釈を束ねる Annotation List URI，注釈の座標情報に関係する Segment URI，資料画像の上に乗せるのではない資料画像へのコメントをまとめる Comment Annotation URI，それぞれのアノテーションをリンクさせる Hotspot Linking URI の情報を付与することができる．VM のユーザが直接閲覧できるのはこのうち Annotation List URI に記載されたテキストである．

VM では，このテキストを Open Annotation とは別の形で，「注釈」「アイディア」「仮説」の 3 種類に分類し，固有の ID を割り当てアプリケーションのサーバで管理している．こうした分類は前節で述べた Mirador に AH を適用する条件の 2 と 3 に対応するものとして用意されている。

すなわち，作者のコミットメントの履歴を書くのが「注釈」であり，研究者のそれは「アイディア」であり，共同研究での一定の成果は「仮説」に対応している．VM での研究者の最初の仕事は，アーカイブスの資料を閲覧する際に対象となっている作者が残したメモを翻刻し，それが何らかの事件を参照にしてはいないか，あるいは引用なのではないかといったノートをつけるのが「注釈」である．次に，そういったノートをまとめていくことで問いを立てたりするのが「アイディア」である．最後に，そうした「アイディア」を検討して一定の仮説を作り上げるのが「仮説」である．これらはすべてハッシュ ID が付与され，データベース内でリンク関係にある．また，これら全てに対してグループのユーザはコメントをつけることができる．それと同時に，ユーザが自分の書いた文章の内容を誰かに検討して欲しい場合は，「コメントリクエスト」を出すことで，グループ内のユーザにコメントを促すことができる．

VM は VCS のシステムに着想を得ているため，注釈機能によって書き込まれたデータは全てリポジトリに保存してある．よって，ある時点でグループメンバーが削除したデータも削除されたデータの ID から容易に復元することができる．

注釈機能以外でも，VM では，ユーザの利便性を向上するために，Manifest URI をできるだけ簡単に取得できるようにした．ユーザはあるアーカイブの資料のリンクをフォームに入力すると，サーバサイドでリンク先のサーバーに Manifest URI をリクエストし，取得できる仕組みがそれに当たる．これは IIIF の知識がないユーザにも VM を利用してもらうことを目的としている．

以上の機能を有した VM は利用法を記したドキュメントとともに公開し，今後はオープンソースで開発していく予定である．

8. 今後の課題

VM には技術的な課題が多く残っているので，その解決が望まれる．課題として真っ先に考えられるのは，オープンデータといえども公開先の事情によってアノテーションの対象が変更されてしまった場合, 注釈が参照先をなくして意味を失ってしまう点である．他にも，現在の VM では横書きでしか注釈機能を用いることができないので縦書きが求められる分野では扱いにくい点が考えられる [6]．また，今回は Mirador ではなくて，本発表の時点では、Mirador の採用を見越しつつ、OpenSeadragon[*17]に基づく実験的な簡易実装を試みている。今後は、Mirador を用いた本格的な運用を目指している。最後に、発表者の属している研究分野やそのほかの分野で共同研究が実際に促進するのかどうかの調査を継続的に行い，利用に際して変更が求められる点を改善していく必要がある．現在，発表者が自身の研究で用いる以外にも，フランスの中世の写本や旧フランス

[*14] http://www.tei-c.org/Vault/P5/3.1.0/doc/tei-p5-doc/en/html/PH.html
[*15] http://www.tei-c.org/Vault/P5/3.1.0/doc/tei-p5-doc/en/html/TC.html
[*16] http://www.openannotation.org/spec/core/
[*17] https://openseadragon.github.io

植民地文学を研究している大学院生などの協力を予定している．

謝辞 本原稿の執筆に際して永崎研宣先生に多くの助言を戴いた．ここに謝意を表する．

参考文献

[1] Kent Beck, 長瀬嘉秀監訳. XP エクストリーム・プログラミング入門: ソフトウェア開発の究極の手法. ピアソン・エデュケーション, 2001.
[2] Tim Causer, Justin Tonra and Valerie Wallace. Transcription maximized; expense minimized? Crowdsourcing and editing The Collected Works of Jeremy Bentham. Literary and Linguistic Computing 27(2). pp.119-137, 2012.
[3] Geoffrey Rockwell and Stéfan Sinclair. Hermeneutica: Computer-Assisted Interpretation in the Humanities. MIT Press, 2016.
[4] Nayan B. Ruparelia. The history of version control. ACM SIGSOFT Software Engineering Notes. Association for Computing Machinery 35(1). pp. 5-9, 2010.
[5] Jeffrey C. Witt and
Rafael Schwemmer. IIIF, Webmentions, and Collaboration between Institutions and Research (online). http://lombardpress.org/2016/04/16/iiif-webmentions/. (2016. 04. 16).
[6] 永崎研宣. 人文系オープンデータと IIIF がもたらす意義・可能性・課題. 研究報告人文科学とコンピュータ（CH）113(6). pp.1-6, 2017.
[7] 永崎研宣, 津田 徹英, 下田 正弘. SAT 大正蔵図像 DB をめぐるコラボレーションの可能性. 研究報告人文科学とコンピュータ（CH）113(8). pp.1-4, 2017. SIAM (1998).
[8] 松澤和宏. 生成論の探究——テクスト・草稿・エクリチュール. 名古屋大学. 2004.