

多義的な情報の分類に基づくインタフェースの試作

津田 光弘
イパレット

本報告では、多義性やあいまいさを有する情報を分類し、相互に関連付けるためのデータモデル、さらに、そのデータモデルを使って情報をブラウジングするインタフェースのプロトタイプを紹介する。具体的には、画像コンテンツをリソースとするアノテーションと、グルーピングを役割とするアノテーションから成る二部グラフ表現をデータモデルの基礎として用いる。このモデルはその構造上、多義的な情報を表現し得る。このデータモデルの実現形態としてコレクションとレイアウトについて説明し、最後に、レイアウトの拡張表現であるコンテキスト・フィールド「文脈場」と名付けたインタフェースによって情報を共有する可能性について考察する。本データモデルの実装においては、オープンソースのリレーショナル・データベースとXMLを出力するサーブレットを用い、データを視覚的に表現するためのクライアント・インタフェースを作成した。

An Interface Prototype in order to Classify Ambiguity Information of Contents

Mitsuhiro TSUDA
IPallet

In this paper, it introduces a data model to classify information that is ambiguity or has vagueness, and to connect them mutually. And it presents an interface prototype that browses information by using the data model. More in detail, a base of the data model is expressed by a bipartite graph that consists of annotations having an image resource and having a role of a grouping. This model can handle data with ambiguity by the structure. It explains a collection and a layout that are expression of this data model. And it considers possibility to share information by an interface that named "Context-Field" that expanded a layout. In implementation of this data model, it used an open-source relational database and a servlet program that outputted XML, and made a client interface to visualize data.

1. はじめに

記録を目的とした写真には主題となる被写体以外にも様々な事物が映っている。報告者らはこれまで特にデジタルアーカイブ画像に対して、それらの情報を有効に記録するため、アノテーションによる記述とその活用を支援するツールを開発してきた。^[1] アノテーションによって、サイズの大きな画像を閲覧する場合の指標となるだけでなく、詳細に画像を比較分析してゆくことができる。アノテーションと共に記録するメタデータのテキストを検索することで、ファイル単位ではなく被写体単位で横断的な画像収集が可能となった。

しかしながら、これまで幾つかのデジタルアーカイブ画像に対して上記手法を適用してきたが、アノテーションを介して画像を横断的に得る場合に、メタデータのテキストによる検索結果だけでは不十分である。アノテーションに対応するサムネイル画像には被写体や被写体の一部だけが映っており、それだけでは意味があいまいになる。そのため、メタデータに記述をしっかりと行う必要があるが、アノテーションの被写体に対し検索に耐えるメタデータを付与してゆくことは負担が大きい。分析目的や内容にも拠るが、情報の粒度が小さくなるために検索の目的を達成できるような差異のあるテキストを付けることが難しいことも多い。元画像への関係をたどって構造的なメタデータとする方法もあるが、あくまで補完的である。

アノテーション化した大量のサブ・コンテンツを有効に活用するためには、従来のメタデータによるテキスト検索だけでなく、より画像というメディアの直感性や多義性を活かした方法が必要と考える。例えば、あいまいながら「ここを指す」というようにアノテーションを作成し、構造的な概念を基にして関連付けを行うことが望ましい場合もあるだろう。このようなアノテーションとオントロジーの関連付けの重要性に対する認識は高まっており、データモデルやアプリケーションの公開も行われている。^[2] ただし、オントロジーが不明瞭な段階あるいは使用方法での関連付けでは、かえってその後の情報利用の妨げとなってしまう場合があるとも考える。本報告では、よりプリミティブに「分ける」という行為について考え、多義性を持った情報、特にアノテーションによって抽出されたアノテーション・イメージを活用するためのデータモデルと試作したインタフェースを紹介する。

2. 「分ける」インタフェースの考察

先ず、あたりまえと思うことをよく考えてみたい。「わかる」ことは「分ける」こと、「分ける」ことは人間の知的プロセスの原点だと言われる。^[4] 対象が多数の場合には「分ける」ことは新しく「まとめる」ことであり、言い換えれば「整理」することである。私たちの思考は分類、その概念化、概念の関連付けなどを繰り返しながら進んでゆく。データや情報にかかわるときだけでなく、日常的にも様々なレベルで意識的、無意識的に行っており、思考をまとめてゆく数々の手法にはこのプロセスは欠かせない。^[5]

アノテーション・イメージに対して適用する前に、グラフィカル・ユーザ・インタフェース(GUI)という身近な例によってオブジェクトを分ける行為を考えてみたい。デスクトップ・メタファ、フォルダ、アイコンなどで構成されるウィンドウ・システムは、既に私たちの情報生活に溶け込んだ馴染み深いデザインであるが、ここには分類や整理の考えも凝縮されている。操作上の行為は周知のことではあるが、次の2種にまとめることができるだろう。(図1参照)

- (1) コレクション(Collection)
- (2) レイアウト(Layout)

コレクションはデータ構造で複数のデータを格納することであり、リスト、集合、配列などと言い換えることもできる。GUIではオブジェクト(ファイル)をフォルダに移動・複製して分ける行為であり、何らかのいれものに割り当てることである。もうひとつのレイアウトはオブジェクトの配置、あるいは配置行為であるが、GUIではフォルダ内の空間を使って意図的に分けることに相当する。特定のオブジェクトをその他と区別する場合や、強調する場合、また、あいまいに意味合いを空間上で表現したい場合にも用いる。私たちはGUIを使ってオブジェクトを扱ってゆく場合に、これらをうまく組み合わせている。分けると同時にフォルダ名やその構造、位置によって分類の意味の概念化を行って整理している。GUIではフォルダの構構性や階層性は概念と無関係に作成でき、フォルダ名が任意の記号であっても整理はできるが、分類の意味や構造をあいまいにしておくとも時間が経つにつれて初期の記憶が薄れ最後には分けた目的がわからなくなる。また、サーバーの共有フォルダで規則を決めたつもりでも他者の区分が時に分からず、データの共有化の支障となるのは、検索語彙やオントロジーの課題と類似点があると推察する。以上のことは、カードや日常生活で物品などを分ける場合にも当てはまり普遍性があると言えるだろう。

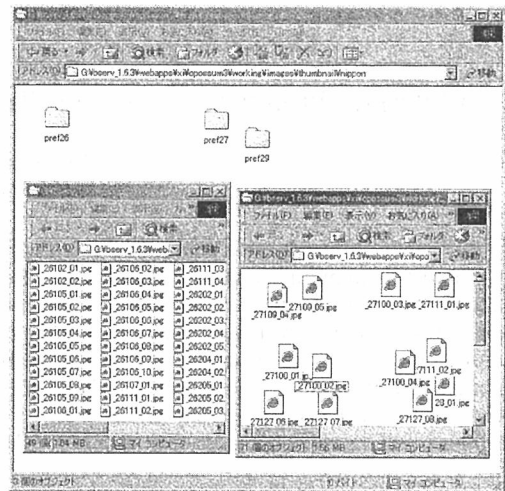


図1. GUIを例とした分類の分析
(コレクションとレイアウト、その混在例。
Microsoft社のWindows2000 Systemから)

3. 画像の多義性を表現するデータモデル

本報告のデータモデルは、既に報告者らが開発を進めてきたアプリケーション iPalletnexus (イパレットネクサス)^[1]の拡張である。先ず、iPalletnexusの旧データモデルについて説明する。アノテーション Pal とそれらの関係 Nexus を基礎としており、次の特徴がある。

- (1) Pal は、リソース(例えば画像)への1つ以下のリンクを含む
- (2) Pal は、0個以上の図形情報を含む
- (3) Pal は、それ自身のメタデータを記録する
- (4) Pal は、0個以上の他の Pal への関係記述(リンク)である Nexus を含む

ここで、リソース0個、図形情報0個を持つ特別な Pal を定義していた。この Pal は、他のアノテーションを取りまとめるコレクションの役割を果たすもので、役割によって INDEX もしくは POD と呼んでいた。メタデータをテキスト検索した結果もこのコレクション用の Pal で記述していた。図2にこのモデルの概念図、また、表1に代表的な Pal を示す。このモデルは、アノテーションを示すノード SELF と、他のアノテーションへのリンク Nexus を基本とするシンプルな構造である。Pal にはその役割 Role を定義しており、表1のように PARENT はリソースへのリンクと共にリソース全体にかかる図形情報を持つアノテーションである。CHILD はリソースについては構造上の親である PARENT に依存

し、自身は図形情報とメタデータを持つ。POD(及び INDEX)はリソースや図形情報を持たず、コレクション機能だけを有し Nexus によって相互につながる。この旧データモデルでも、アノテーションは複数のコレクションに関係付けることができる。メタデータによって概念に相当する情報もコレクションに記述できる。しかしながら、iPalletnexus のインタフェースは、アノテーションというデータを作成するための機能を主体としており、コレクションとアノテーションの関係記述部分は一部を除き、編集のための十分なインタフェースを備えてはいない。アノテーション・イメージをコレクションやレイアウトの表現の下に再編集するためには、GUI のウィンドウ・システムに相当する柔軟なユーザ・インタフェースが必要であり、そのためには若干のデータモデルの拡張が必要である。

そこで、データモデルの見直しを行い、アノテーションとコレクションの関係を図 3 に表すような二部グラフの形態にまとめた。図の各ノードが Pal の SELF パート、リンクが Nexus に相当する。A サイドのアノテーションは PARENT と CHILD である。G サイドはコレクションに相当し INDEX、PARENT と POD が属する。A から見て G は所属先グループ、逆に G から A は属するメンバーと位置付けられる。メンバーは Nexus によって複数のグループに属することができる。この構造によって分類と同時にアノテーションの多義性も表現することができる。

ここで新たに、G サイドの POD を概念、文脈、検索ストリングなどを扱うことが可能な「汎用グループ」として定義した。この汎用グループ POD はアノテーションと同じ構造を基礎とし、リソースを持つことを可能とした。(表 1 の網かけ部分を参照)なお、二部グラフは社会科学分野における所属関係ネットワーク(アフィリエイト・ネットワーク)として知られている^[6]が、その理論までをコンテンツのアノテーションに適用できるかは今後の研究が必要である。

もうひとつの分類表現であるレイアウトに関しては、二部グラフだけでは表現できない。そこで、NEXUS の属性として位置データなどを追加記述することとした。汎用グループに画像リソースを適用した場合、文脈を反映する画像や図形を背景としてアノテーションのレイアウト表現が可能である。以上の拡張によって、このデータモデルは「分ける」ための要件を満たすものとなった。

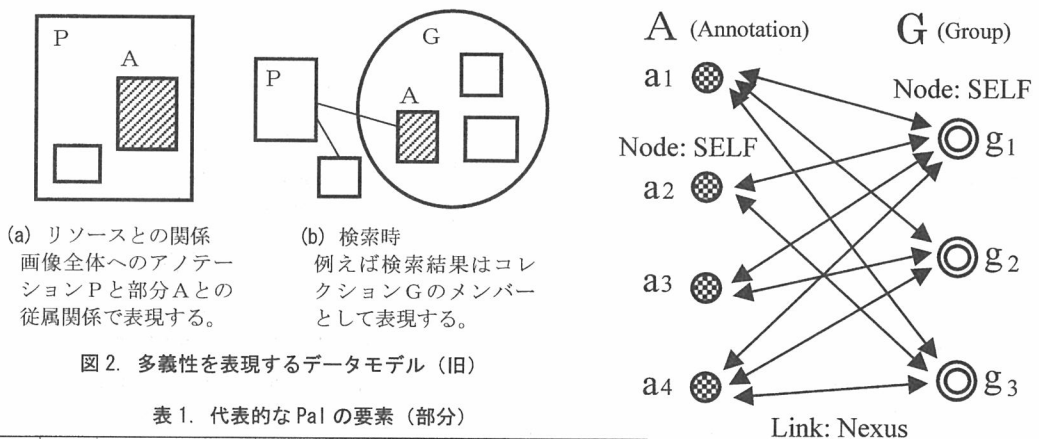


図 2. 多義性を表現するデータモデル (旧)

表 1. 代表的な Pal の要素 (部分)

種別 A	種別 B	項目 (要素, あるいは属性)	補足説明	PARENT	CHILD	POD
PAL (SELF パート)						
パラメータ						
		アノテーション自身の種別				
		アノテーションの種別				
		アノテーションの空間座標値				
		リソースの種類 (フォーマット)				
		リソースへのリンク	URL 相当			
		リソースの大きさ	画素数, 解像度			
		アノテーションの作成日				
		アノテーションの更新日				
		アノテーションの作成者情報				
メタデータ						
		識別記号				
		タイトル				
		主題				
		記述				
		作成者				
		作成年月日				
NEXUS						
リンク						
		親へのリンク	URL 相当			
		子へのリンク	URL 相当			
		関連するアノテーションへのリンク	複数			
		座標	関連するアノテーションへのリンクの属性として			
		強度	現状はシステム内部のデフォルト値			

■ 記述せず
 // 記述 (新) で記述

図 3. 多義性を表現するデータモデル (新)

[補足] このモデルは二部グラフを基礎とし、A と G の双方を順次経由しながらアノテーション・データ空間をブラウズしてゆくことを可能とする。レイアウト表現は Nexus の属性とすることで、G ごとに多重にレイアウトが定義できる。

4. プロトタイプ

本データモデルに基づくプロトタイプを、インターネットで利用可能なクライアント/サーバーの構成で作成した。表 2 にサーバーの仕様を示す。Java とオープンソースを基本に構築しており、Linux でも動作することを確認している。

また、表 3 にデータベースのテーブルの主要部を示す。データモデルを実現するテーブルの構成はごく一般的なものであり、移転可能なように単純にした。テーブル `annotes` には、全てのアノテーションと、同じフィールドを定義した汎用グループを登録する。作成したアノテーションごとにユニークな ID を割り振り、URI に相当する名前を付与する。メタデータは基本的に Dublin Core の 15 要素である。リソースとなる URI と共に、アノテーションの (図形) 種別 `an_draw_type` と座標群 (もしくは SQL 等の記述) `an_data` を記録するフィールドを持つ。汎用グループの場合には、リソースとアノテーションは `null` な場合もある。

テーブル `annotes` の各レコードすなわちノードを関連付ける二部グラフのリンクをテーブル `nexus` で管理する。リンクごとにユニークな ID を割り振り、関係するノードの ID を登録する。`pr_id` はコレクションの入れ物となる汎用グループ `POD` もしくは `PARENT` アノテーションが登録される。`an_id` にはコレクションのメンバーである `CHILD`、`PARENT` アノテーションの ID が登録される。このリンク関係は、`nx_role` と `nx_reverse` で相互に定義しておく。プログラム言語 `Xi` によって、メンバー側から汎用グループを得る場合、汎用グループ側からメンバーを得る場合を SQL と共に記述し、このテーブルにアクセスし、クライアントの指令からコレクションに相当する XML を得る。

一方、クライアントのインタフェースは Macromedia 社の `Flash` を使って作成した。サーバーからクライアントへはコレクションを中心とする単階層の XML、クライアントからはクエリストリングによって情報を送受信している。

図 4 にインタフェースの主要構成を示す。コレクションを表現する 2 種類のツリー・コンポーネントと、レイアウトを表現するフィールド「場」から成る。

このデータモデルの特徴である汎用グループとメンバー・アノテーションを連携させたブラウジング動作を簡単に記載する。

- (1) 汎用グループ・ツリーに初期インデックスを表示。
- (2) ツリーのノードを選択し、メンバーをもうひとつのツリーに読み込む。
- (3) メンバーのアノテーション・イメージ (サムネイル) がフィールドに表示される。
- (4) メンバーを選択し、リソース (画像) やメタデータを閲覧する。
- (5) メンバーの所属グループのコレクションを汎用グループ・ツリーに呼び出す。
- (6) (2)以降を繰り返し、多義情報をブラウズする。

ここで、(5)の部分によって二部グラフによる多義性の表現が可能になる。

表 2. プロトタイプ (サーバー) の仕様

OS	Windows 2000 / XP
Java	Java SDK 1.4.2
	Java Advanced Imaging API 1.1.2
Servlet	横浜ベイキット BayServer 1.6.3
	同 Xi 1.4.3 (サーブレットと同梱)
Database	PostgreSQL 8.0.3

表 3. テーブルの主要フィールド (未使用の実験用フィールドも含む)

```
CREATE TABLE annotes (
  an_id          INT8,

  an_role        VARCHAR(64),
  an_field       VARCHAR(1024),
  an_nest        VARCHAR(1024),
  an_name        VARCHAR(1024),
  an_root        INT8 DEFAULT -1,

  dc_title       TEXT,
  dc_subject     TEXT,
  dc_description TEXT,
  dc_creator     TEXT,
  dc_publisher   TEXT,
  dc_contributor TEXT,
  dc_date        TEXT,
  dc_type        TEXT,
  dc_format      TEXT,
  dc_identifier  TEXT,
  dc_source      TEXT,
  dc_language    TEXT,
  dc_relation    TEXT,
  dc_coverage    TEXT,
  dc_rights      TEXT,

  resource_uri   TEXT,
  resource_hint  VARCHAR(64),

  an_draw_type   VARCHAR(64),
  an_data        TEXT);

CREATE TABLE nexus (
  nx_id          INT8,

  pr_id          INT8,
  an_id          INT8,

  nx_role        VARCHAR(64),
  nx_reverse     VARCHAR(64),

  nx_serial      INT8 DEFAULT -1,
  nx_x           INT4 DEFAULT 0,
  nx_y           INT4 DEFAULT 0,
  nx_z           INT4 DEFAULT 0,
  nx_layer       INT2 DEFAULT -1,
  nx_timeline    VARCHAR(64),
  nx_effect      INT2 DEFAULT 10,
  nx_note        TEXT);
```

このプロトタイプは、また、画像の閲覧機能と共に簡単なアノテーションも作成することができる。アノテーションを汎用グループにまとめる機能、汎用グループに属するアノテーションをコレクション、レイアウトの方法で表示する機能を持つ。汎用グループのリソースに SQL の問い合わせ言語を記述とすることで、従来の検索も汎用グループの枠組みに含めることができる。インタフェースの外観や主要な機能については図 5 (a)~(d)に示す。

5. コンテキスト・フィールド

クライアント・インタフェースの主要機能として、コレクション内のレイアウト機能を盛り込んだ。図 5(b)のように利用者がアノテーション・イメージを移動してレイアウトを任意に編集することができる。このレイアウトの移動データは、nexus テーブルの nx_x、nx_y の属性フィールドの数値を更新することで記録する。座標系はコレクションに割り当てた背景イメージに依存する。図 5(b)ではグリッドイメージを用いている。

これらのレイアウト機能と、コレクションの背景イメージを発展させると、図 5(d)のようなインタフェースが実現できる。このような表現は「地」と「図」によるメディア表現として研究されており、

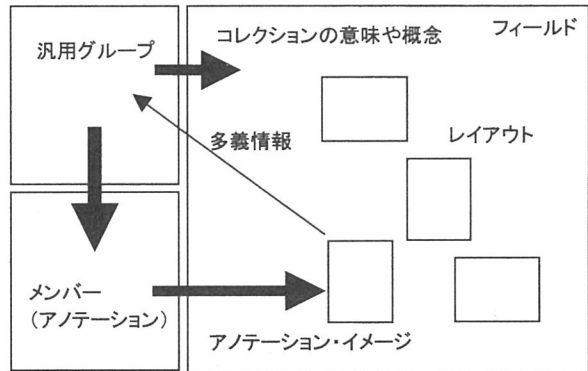
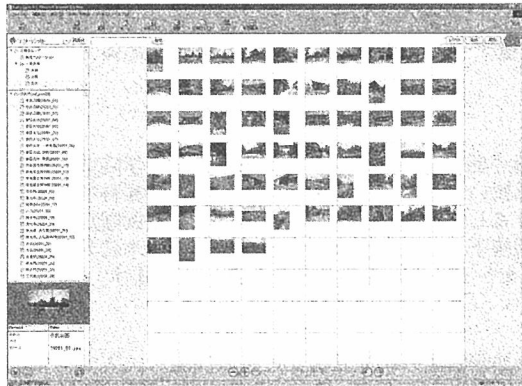
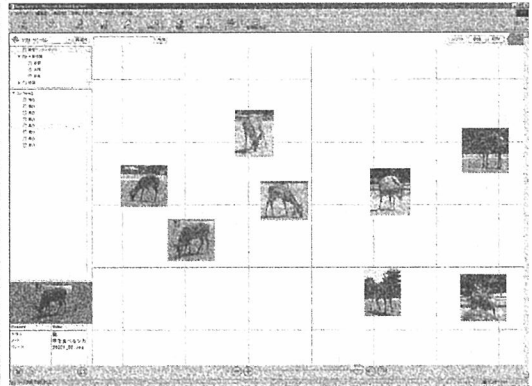


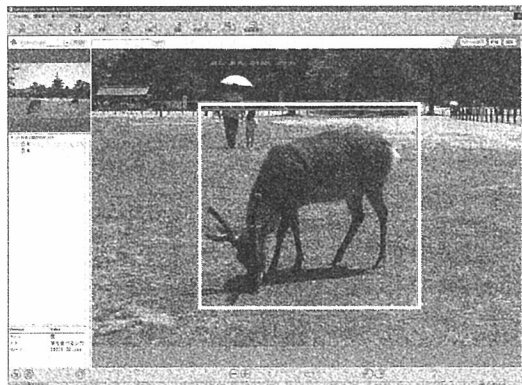
図 4. クライアント・インタフェースの概略



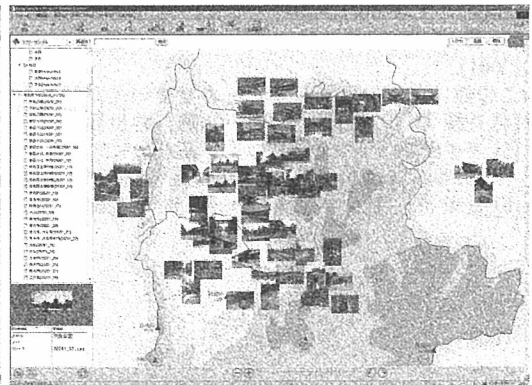
(a) アノテーション・イメージのコレクション



(b) アノテーション・イメージのレイアウト



(c) アノテーションの編集



(d) コンテキスト・フィールド

図 5. プロトタイプの表示例

その有効性も指摘されている。^[3] 本モデルではこのようなフィールドの利用を特に、コンテキスト・フィールド「文脈場」と定義した。コンテキスト・フィールドは、レイアウトの指標を与えると共に、Web 上で汎用グループの意味を共有するためのわかりやすい手段になる可能性がある。これは、動的なクリッカブル・マップであり、コンテキスト・フィールドは汎用グループごとに定義できるため、同一のメンバーを多視点で表現する場合にも有効な役割を果たすと考える。

ただし、汎用グループを利用してコンテキスト・フィールド間を移動する場合、汎用グループのみを切り換えると、対象としたアノテーションを再度探さねばならない。そこで、選択中のアノテーション画像の表示位置を中心から移動せずに、汎用グループを切り換える表示方法を実装した。この機能によって、周囲のアノテーションとの関係性をレイアウトによって確認しつつ多義的な空間をブラウズしてゆくことができる。類似検索に相当するような情報のブラウジングが行えると予想する。

6. おわりに

デジタルアーカイブの横断的な活用を目指し、画像からアノテーションによって抽出したデータを分類し整理するためのひとつの試みを紹介した。二部グラフのデータモデルを基にしてアノテーションを関連付け、多義性やあいまいさを有する情報をブラウジングするためのインタフェース・プロトタイプを説明した。本プロトタイプには、スケーラビリティの問題や、レイアウト及びコレクションの自動生成方法を含め、まだ多くの課題がある。二部グラフをマニュアル操作でナビゲートできる仕組みに目処を得たが、その効果的な表現、更にどう有効に活用できるかについては、分野を異にするアフィリエイト・ネットワークの適合性を実際の利用によって検証しなくてはならない。

今後は、検討したデータモデルとインタフェースをオープンな研究支援ツールとしても実装を行ってゆくと共に、Weblog のように誰もが使えるわかりやすい道具として開発を進め、具体的な事例を増やしてゆきたいと考えている。

なお、本プロトタイプではデータモデルの実現にリレーショナル・データベースを用いたが、同様の構造は XML によっても作成できる。また、実用には、記載した 2 種類のテーブルだけでは充分ではない。コンテンツ管理システム (CMS) に相当する各種データ管理用のテーブルも併せて必要であることを付け加える。

謝辞

iPalletnexus のデータモデルの拡張において各種の助言をいただきました東京大学大学院情報学環の馬場章教授と、同学際情報学府の碓谷紀夫氏に感謝いたします。北海道大学大学院情報科学研究科の田中譲教授の諸研究からは多くの知的刺激を受けました。また、プロトタイプの初期段階からご使用いただき分類インタフェースの多くの問題をご指摘下さった奈良女子大学文学部の鈴木広光助教授、そして、株式会社堀内カラー・アーカイブサポートセンターの川瀬敏雄氏に感謝いたします。

文献

- [1] 津田 光弘, 馬場 章, 川瀬 敏雄, 肥田 康: 「絵地図画像の研究支援ツールの開発」, 人文科学とコンピュータシンポジウム, pp241-244, (2002).
 - [2] 川添 愛, ナイジェル・コリアー: 「オントロジー構築・アノテーション支援ツール Open Ontology Forge」, 人工知能学会研究会資料 SIG-SWO-A501-05. (2005).
 - [3] 田中 譲: 「ミームメディア技術の展開と知財アクセス空間」, 学術月報, Vol.54 No.4 pp349-356, (2001).
- この他、以下の書籍を参考とさせていただいた。
- [4] 斉藤 孝: 「「記録・情報・知識」の世界」, 中央大学出版部, (2004).
 - [5] ネイサン・シェドロフ: 「情報インタラクションデザイン」, ロバート・ヤコブソン編「情報デザイン原論」, pp214-255, 東京電機大学出版局, (2004).
 - [6] 金光 淳: 「社会ネットワーク分析の基礎」, 勁草書房, (2003).