空間ドキュメント管理システムの設計と実装

白石 陽，有川正俊，相良毅，浅見泰司

2007

DEWS 論文集

(電子情報通信学会 第 18 回データ工学ワークショップ)

空間ドキュメント管理システムの設計と実装

白石 陽† 有川 正俊† 相良 毅‡ 浅見 泰司†

†東京大学 空間情報科学研究センター 〒277-8568 千葉県柏市柏の葉 5-1-5
‡東京大学 生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1
E-mail: †{siraisi, arikawa, asami}@csis.u-tokyo.ac.jp, ‡sagara @tkl.iis.u-tokyo.ac.jp

あらまし 本稿では，住所情報を含むデジタルドキュメントを，直感的なインタフェースで扱うことのできるシステムとして，空間ドキュメント管理システム（SDMS: Spatial Document Management System）を提案する．SDMSは，ドラッグ&ドロップという簡単な操作で，即座に，ドキュメント中の住所情報を，点として地図上に表示することができる． SDMSでは，非定型の一般的なデジタルドキュメントから，住所情報を抽出し，アドレスマッチング技術を利用してPOI（Point of Interest）を生成し，地図上に表示する．SDMSを用いることにより，多くの一般的なユーザが，簡単に，デジタルドキュメントを空間的に整理し，地図化することが可能となる．

キーワード 空間ドキュメント，位置情報，GIS，ジオコーディング，ジオパース，Point of Interest (POI)

1. はじめに

近年の位置情報関連技術の進歩は目覚しい．位置測位技術の発達，GPS搭載携帯電話の普及，位置情報サービスの高度化，地図APIの公開，Webコンテンツへの緯度経度情報の埋め込みなど，位置情報を扱う様々な枠組みが整ってきている[1,2,3,4]．緯度経度という絶対的な位置情報は，こうしたサービスを支える基盤となる情報であるが，多くの人にとっては，ただの数字の列にしか見えないため，必ずしも人間にやさしい表現とは言えない．それに対して，住所や地名などの位置情報は，そのままでは地図に表示することはできない間接的な表現であるが，人間の理解しやすいものである．日常的に利用される自然言語による表現は，コミュニケーションにおいても効果的であり，案内文書，報告書，店舗一覧，施設一覧，光化学スモッグ発生情報など多くのデジタルドキュメント上で利用されている．今後，緯度経度のような“直接位置参照情報”が目に見えない形で埋め込まれたドキュメントが増えていくことは容易に想像できるが，依然として，住所や地名などの“間接位置参照情報”を含む様々なドキュメントが，多くの場面で利用されていくことは間違いない．本研究では，こうした住所や地名などの位置情報を含むデジタルドキュメントを「空間ドキュメント」と呼ぶ．インターネット上のWebページ，メールの添付ファイル，自分のコンピュータの中のファイルなど様々な空間ドキュメントが存在する．そのファイル形式も，HTML，Word，Excel，TEXTなど多様である．

本稿では，一般ユーザが日常的に利用している，このような多種多様な空間ドキュメントを，容易に管理，検索，地図化できるツールとして，空間ドキュメント管理システム（Spatial Document Management System，以下SDMSと呼ぶ）を提案する．SDMSは，ドラッグ&ドロップという単純な操作で，住所や地名などを自動的に抽出し，緯度経度を算出して，Point of Interest（POI）として地図上に表示することができる．SDMSを用いることによって，インターネット上やパソコン内に存在する様々なドキュメントを空間的に整理し，把握することができる．また，地域行政，保健医療，公衆衛生，危機管理などの現場において，こうした場所情報を含むドキュメントは，貴重な情報源であるにも関わらず，地図化する簡易な手段がなかったために，死蔵されている場合も多く，空間ドキュメントを扱うことのできるSDMSに対する潜在的なニーズは高いと考えられる．

以下，本稿では，2章で関連研究を挙げ，提案システムの位置付けについて述べた後，3章で提案システムの説明を行う．4章では，動作例を説明した後，考察と今後の課題を述べる．最後に，5章でまとめを行う．

2. 関連研究

2.1. 地理情報システムとの比較

住所，地名，郵便番号などの間接位置参照情報を緯度経度などの直接位置参照情報に変換することをジオコーディング（geo-coding）と言う．住所を緯度経度に変換する処理は，特に，アドレスマッチングと呼ばれる．それに対して，ドキュメント中から住所や地名などの地理情報を抽出することをジオパース（geo-parse）処理と呼ぶ．

空間情報を管理できるシステムとして地理情報システム（GIS: Geographic Information System）[5]がある．GISは，様々な空間情報の管理，解析，統合，可視化を行うことのできる便利なツールである．ジオコーデ

ィングは，GIS 分野でも古くから重要な課題であり，現在では，商用のサービス，あるいは，フリーの API を介してジオコーディングサービスを利用できる．既存の GIS では，入力データとして表として整理された，すなわち構造化された空間データを必要とする．多くの研究者は，GIS を利用する前に，緯度経度を含む表形式のデータを作成する必要がある．元データが住所である場合には，CSV などの表形式で住所情報を整理した後，ジオコーディングサービス（例えば，CSV アドレスマッチングサービス[6]）を利用することにより，緯度経度の付与された構造化された空間データを準備することができる．このように GIS では，GIS を利用する前の手順が，住所情報の収集・抽出も含めて，非常に手間のかかる仕事となっている．

空間ドキュメントのファイル形式は，HTML, Word, Excel, TEXT など多様であり，半構造や未構造の空間データである．既存の GIS では，構造化された空間データを必要とするため，こうした非定型（半構造あるいは未構造）のドキュメントを，そのままの形で直接扱うことはできない．多数のドキュメントを対象とする場合，ドキュメント中に多数の住所が含まれている場合には，大量の住所情報を手動で抽出し，構造化データとして整理するのは現実的ではない．

本稿で提案するシステムでは，空間ドキュメント中から住所を抽出して，緯度経度に変換し，地図上に表示できる必要があるが，空間ドキュメントに対するこうしたジオコーディングとジオパースの処理を，ユーザの手を返さず，自動化できると望ましい．また，GIS は，高度な機能を持つため，専門家にとっては強力なツールとして利用できるが，その反面，一般のユーザが操作を習得するためには訓練が必要であり，日常的に使いやすいものとは言えない．

空間ドキュメントを管理するためのシステムの要件としては，(a)種々の形式のファイルを扱えること，(b)非定型のドキュメントから自動的に住所を抽出して緯度経度を算出できること，(c)算出した緯度経度情報をドキュメント中に埋め込むこと，それに加えて，(d)それらの処理を簡単な操作で実行できることが必要である．

2.2. 住所情報の抽出とドキュメントの構造化

本研究では，住所情報を抽出するために「住所地名テーブル」を作成し，ドキュメント中の文字列とテーブル内の地名語とを比較することでジオパース処理を行っている．WWW 文書から住所情報を抽出し，文書を構造化したり，地理情報を付与（geo tagging）したりすることで，位置指向の検索やコンテンツの整理が可能となる[7,8,9]．Amitay らは，地名辞典（gazetteer）を利用して，Web コンテンツを地名語（place name）と関連付ける手法を提案している[7]．横路らは，WWW 文書のための位置指向検索手法を提案しており，文書中から住所情報を抽出し，文書の構造化を行っている[8]．その際，住所名を辞書に加えて形態素解析を行い，各形態素を住所辞書と比較することで，住所抽出を行っている．住所辞書には，都道府県名，市区町村名，町字などを単独もしくは上位の住所階層からの省略のない形で登録している．そして，住所を正しく抽出するために，「都道府県名から省略なく正確に記述されている」，「都道府県や市区など住所を明確に示す接尾辞がついている」などのルールを適用することで，住所抽出の精度を向上させようとしている．形態素解析のみを利用する住所抽出では，その精度が辞書に登録されている住所情報に依存し，登録された文字列が長いと検索漏れが生じ，文字列が短いと住所認識の精度が下がる可能性がある．長屋らは，国土交通省の提供する「街区レベル位置参照情報」[10]に基づいて，住所抽出のためのキーワードを生成し，キーワード群の状態遷移表を用いて文字列マッチングを行っている[9]．ただし，キーワードとしては，「丁目」以降が削除された住所文字列を選択し，「丁目」以降の文字列は構文解析によって処理している．一般的なドキュメントに含まれる住所情報では，上位の住所階層が省略されていることが多いため，そういった住所情報を漏れなく抽出できることも重要である．地域行政，地方自治体，地域ポータルといった対象を考えると，上位階層の住所表記は省略される可能性が高い．本研究では，文字数の比較的短い住所文字列をテーブルに登録することで住所抽出の再現率を向上させ，アドレスマッチングの変換結果の精度に基づいて最終的な住所判定を行うことで住所抽出の精度を向上させようとしている．

さらに，最近では，GeoRSS[3]や Microformats [4]など，ドキュメント中にタグを用いて属性情報を埋め込む枠組みについての議論が行われている．住所や地名などを属性としてドキュメント中に埋め込み，ドキュメント中のオブジェクトと関連付けることができれば，その位置情報を活用して様々な応用が期待できる．本研究においても，ジオパース処理とジオコーディングの結果である位置情報をタグの属性としてドキュメント中に埋め込むことで，ドキュメントの管理や POI の検索を行う．位置情報タグを用いてドキュメントを半構造化する手法については，先行研究[11]においても提案し，プロトタイプシステムの実装を行っているが，本稿では，よりユーザビリティの高いシステムの実現を目指す．

3. 空間ドキュメント管理システム(SDMS)

3.1. 人間中心型システムとしての設計

本研究は，既存の GIS のような専門家が利用するツールではなく，一般のユーザが日常的に利用できるツールの開発を目指している．基本的な設計方針は，次の通りである．

- ファイルのドラッグ＆ドロップという単純かつ直感的な操作によって，ドキュメントの入力を簡単化する
- 入力された非定型のドキュメントに対して，自動的に，住所を抽出し緯度経度に変換する仕組みを開発する
- 抽出した緯度経度情報を，タグを利用してドキュメント中に埋め込むことで，空間ドキュメントの管理や検索に役立てる

空間ドキュメントに対するジオコーディングを自動的に行うためには，ジオパース機能が不可欠であり，ジオコーディング機能と連携させる仕組みが必要である．本稿では，ジオコーディング機能そのものについては，既存のサービスを利用するものとして，ジオコーディングのインタフェースとなる部分に焦点を当てる．当然のことながら，提案システムのジオコーディングの性質や性能は，利用するジオコーディングサービスに依存するが，人間中心型のシステムを開発するためには，ユーザとインタラクションを行う部分と，ジオコーディングプロセスとの仲介を行う部分の設計も重要である．

3.2. システム構成

図 1 に，SDMS のシステム構成を示す．

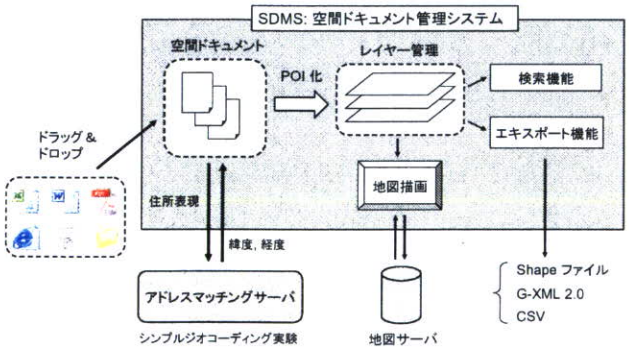

図 1: システム構成

SDMS は，ドラッグ＆ドロップされたドキュメントから住所を抽出し POI に生成する部分（POI 生成部），抽出した POI をレイヤーとして管理する部分（レイヤー管理部），POI レイヤーと背景地図を描画する部分（地図描画部）などから構成される．POI 生成部は，ネットワークを介してリモートのアドレスマッチングサーバと通信することにより緯度経度変換を行う．地図描画部は，リモートの地図データサーバから要求した縮尺の地図画像データを取得し，背景地図として表示する．その他，レイヤー管理された POI に対する検索機能とエキスポート機能を持つ．

3.3. 処理の流れ

次に，SDMS における処理の概略（図 2）を説明する．

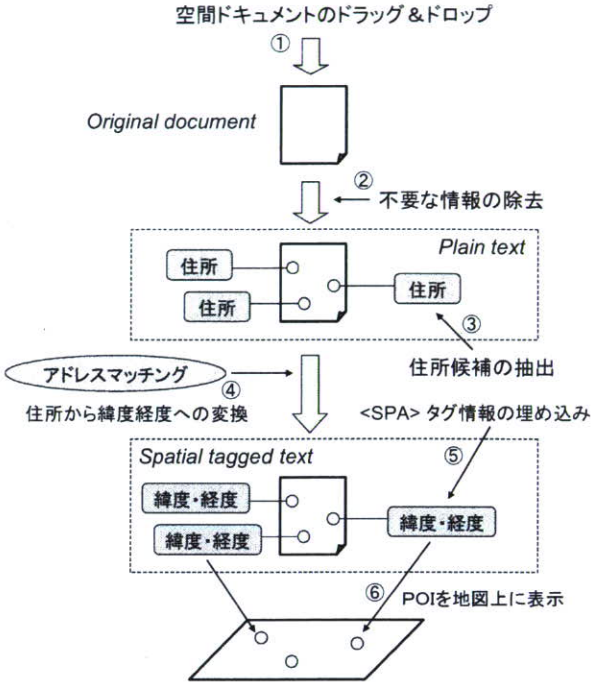

図 2：処理の流れ

① ファイルアイコンをドラッグ＆ドロップすることにより，ドキュメント（original document）を SDMS に入力する．この際，複数のドキュメントを含むフォルダアイコンをドラッグ＆ドロップすることにより，複数のドキュメントをまとめて入力することも可能である．

② SDMS は，入力されたドキュメントから，不要な情報を除去して，プレインテキスト（plain text）を生成する．例えば，HTML ドキュメントを対象としている場合は，そこに含まれる HTML タグを取り除く．

③ プレインテキストを先頭から走査し，住所候補を見つけるごとに，その住所候補文字列をアドレスマッチングサーバに送信する．この部分の処理の詳細については，3.3.4 にて説明する．

④ アドレスマッチングサーバは，送信された文字列

に対してアドレスマッチングを行い，緯度経度情報を生成し，SDMS に返信する．

⑤ アドレスマッチングが成功した場合には，プレインテキスト中の該当する部分に，タグを埋め込み，緯度経度とともにサーバより返信された情報（3.3.1 参照）をタグの属性として埋め込む．サーバに送信した文字列が住所でない場合や変換結果の精度が十分でない場合には，タグの埋め込みを行わない． SDMS は，このタグ付けされたテキスト中の各タグの属性情報を POI として管理する．複数のドキュメントがドラッグ＆ドロップされている場合には，各ドキュメントが一つのレイヤーとして管理され，各ドキュメント中の POI の集合が別々のレイヤーとして管理される．SDMS は，この空間タグ付けテキスト（spatial tagged text）の集合を POI のデータベースとみなして，POI に対する種々の操作を行う．

⑥ SDMS は，空間タグ付けテキストを参照し，生成された POI を点分布として地図上に表示する．背景地図は，ネットワークを介してリモートの地図サーバから取得される．フォルダアイコンがドラッグ＆ドロップされた場合には，各ドキュメント中の POI がすべて地図上に表示されるが，ドキュメントごとに異なる色のアイコンとして表示される．

3.3.1. シンプルジオコーディング実験

SDMS は，アドレスマッチングサーバとして，東京大学空間情報科学研究センターで運用している「CSIS シンプルジオコーディング実験[12]」を利用している．シンプルジオコーディング実験は，日本語で記述された住所・地名・駅名・公共施設名を緯度経度に変換し，結果を XML 形式で返す実験的なサービスとして運用されており，SDMS では，住所候補を緯度経度に変換するために利用する．国土交通省より無償で公開されている「街区レベル位置参照情報[10]」をアドレスマッチングのためのテーブルとして利用しているため，変換精度は街区レベルまでであるが，無料で利用できるサービスとなっている．さらに，表記の揺れや省略に強いといった特徴も持っているため，SDMS も同様の利点を持つ．より詳細な号レベルの精度を持つ有償のサービスもあるが，非常に高価であり，対象地域が広がるほどコスト負担も大きくなる．多くのユーザによる利用を考えると，街区レベル位置参照情報に基づくマッチングは，現時点での望ましい選択肢と言える．

シンプルジオコーディング実験は，REST（Representational State Transfer）サービスとして実装されており，住所候補文字列を含むクエリを送信すると，アドレスマッチングの結果を XML 形式で返信する．例えば，「千葉県柏市柏の葉 5-1-5」という文字列を送信すると，次の XML が返される．

```
<?xml version="1.0" encoding="EUC-JP" ?>
<results>
<query>千葉県柏市柏の葉 5-1-5</query>
<geodetic>wgs1984</geodetic>
<iConf>5</iConf>
<converted>千葉県柏市柏の葉 5-1-</converted>
<candidate>
<address>千葉県/柏市/柏の葉/五丁目/1 番</address>
<longitude>139.935455</longitude>
<latitude>35.901711</latitude>
<iLvl>7</iLvl>
</candidate>
</results>
```

図 3：シンプルジオコーディングによる住所変換結果

<query> は送信文字列を表し，<converted>は送信文字列の中で変換に成功した文字列を表している．<iConf>は変換結果の信頼度を表し，iConf が 1 や 2 の場合はエラーである．<candidate>には，変換候補が格納される．「中央区」や「富士見」などの地名は二本中に数多く存在するため，住所文字列の長さが短い場合には，複数の変換候補が返される．<address>は正規化された住所情報，<longitude>は経度，<latitude>は緯度，<iLvl>は変換された住所の階層レベルを表している．

なお，「柏の葉 5-1-5」という文字列を与えた場合にも，同様の緯度経度が返される．都道府県名や市区町村名が省略された場合でも，変換候補が一意に決められる場合には精度の高い変換が可能である．

3.3.2. POI（Point of Interest）情報の保持

本システムでは，一般的なドキュメントに対して，タグを用いて位置情報を埋め込み，ドキュメントの管理に活用する．具体的には，シンプルジオコーディング実験より返信されたアドレスマッチングの結果（緯度経度および関連情報）に基づいて，次のようなタグ（<SPA>タグ）が埋め込まれ，空間タグ付けテキストが作成される．

「柏キャンパス〒277-8568<SPA id="1" address="千葉県 柏市 柏の葉 五丁目 1 番" level="7" lat="35.901711" lon="139.935455" label = "千葉県柏市" content="ス〒277-8568 千葉県柏市柏の葉 5-" message

=”” url=”” alias=”” lastmodified = "20070106171427" dictionaryid = "-1" converted = "千葉県柏市柏の葉 5-1-">千葉県柏市</SPA>柏の葉 5-1-5 東京大学空間情報科学研究センタ-」

<SPA>タグの属性のうち，address は正規化された住所，level はマッチングレベル，lat は緯度，lon は経度，converted は変換された住所表現を表し，いずれも，シンプルジオコーディングによるマッチング結果に基づいて設定される．content は，変換に成功した住所に文章中の前後の文字列を追加したものが格納され，POI ごとの検索を行う際に利用される．

SDMS は，空間タグ付けテキスト中の <SPA> タグの属性を一つの POI 情報として管理し，表示や検索の対象とする．

3.3.3. 住所地名テーブルの作成

本システムでは，非定型のドキュメントから住所を抽出するために，一定の長さの地名語（住所の部分文字列）を蓄積したテーブルファイル（「住所地名テーブル」）を用意する．例えば，「柏市柏の葉」，「目黒区駒場」といった文字列をファイルに格納し，ドキュメント中から取り出した文字列と比較することで，取り出した文字列が住所候補かどうか絞り込むことができる．

アドレスマッチングサービスによって文字列を変換した結果十分な精度が得られなければ，その文字列は住所ではないと考えられるので，ドキュメントの先頭から，一文字ずつシフトしながら，ある一定の長さの文字列を順番にアドレスマッチングサービスに送信する方法でも，ドキュメント中の住所を特定し緯度経度を算出していくことはできる．しかし，この方法では，大量の住所でない文字列をサーバに送信することになり，無駄が多く，処理時間も長くなると考えられる．

本システムでは，国土交通省の「街区レベル位置参照情報」[10]に基づいて，“hook.txt”と呼ばれるファイルを作成し，最長5文字の約20万語の地名語を登録し，住所候補の絞込みに利用する．この住所候補テーブルを利用することで，アドレスマッチングサーバに送信する文字列の数を大幅に制限することができる．この際，単純に街区レベル位置参照情報（住所と緯度経度の対応表）の各レコードに含まれる住所文字列の連続する5文字を取り出すわけではなく，次に述べる方針にしたがって，地名語を作成し，テーブルに登録する．

日本における住所表記は，都道府県，市区町村，町丁字，丁目というように，住所階層レベルの異なる複数の文字列から構成されているが，hook.txt には，「千葉県柏市」，「柏市柏の葉」など，2階層を結合した文字列を住所候補として登録する．その結合された文字列が5文字より長い場合には5文字までの部分文字列を登録する．逆に，市区町村や町丁字の名前が短い場合も存在するため，「柏の葉 5」や「柏 1」といった5文字より短い長さの文字列も登録されている．5文字より長い住所文字列を登録することで，さらに絞り込むことも可能ではあるが，現実的ではない．また，「丁目」の省略に対応するために，「駒場 4 丁目」は「駒場 4」という形式で登録する．

住所地名テーブルを用いたマッチングの目的は，住所らしい文字列を漏れなく抽出することであり，この段階で住所であることを確定する必要はない．本研究のアプローチでは，アドレスマッチングサーバ側での緯度経度変換の結果から住所であるかどうか判断するため，同様の処理をクライアントである SDMS 側で実施するのは冗長であり，効率的でない．

3.3.4. ジオパース処理のアルゴリズム

ここで，プレインテキスト中の i 番目から j 番目までの文字列を $S(i,j)$ と表記する．3.3 の手順③では，プレインテキストを先頭から走査しながら，次の手順を繰り返す．なお，i の初期値は，1 であり，$S(1,5)$ はプレインテキストの先頭5文字を表す．図 4 に，このジオパース処理のアルゴリズムの概略を示す．

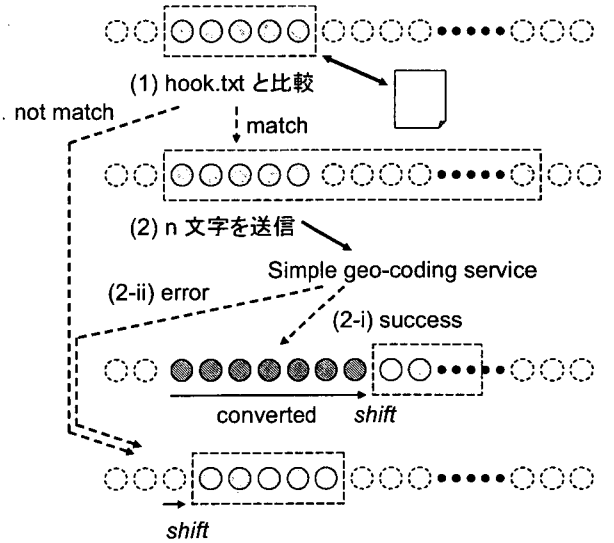

図 4：ジオパース処理のアルゴリズムの概略

(1). プレインテキストから，5 文字の文字列 $S(i,i+4)$ を取り出して，hook.txt 中の文字列と比較する．hook.txt には，5 文字以内の異なる長さ L_m $(m<=5)$ の文字列が登録されているため，各 L_m に対して，文字列 $S(i,i+m-1)$ との比較を行う．$S(i,i+m-1)$ は，$S(i,i+4)$ の最初の文字から m 番目の文字までを含む．

(2). hook.txt 中に，*S(i,i+m-1)* とマッチするものがあれば，住所候補として，*S(i,i+4)*を先頭に含む長さ *n* の文字列 *S(i,i+n-1)*をサーバに送信する．現在の実装では，*n=30* としている．

(i). サーバからマッチング結果が返され，信頼性が高い，すなわち，iConf が 3 以上の場合には，住所変換が成功したものとして，<SPA> タグに属性情報を設定し，変換に成功した文字数分 *k*(3.3.1 の<converted>の文字数に相当) だけシフトして（*i=i+k*），(1)の処理へ進む．

(ii). iConf が 0 もしくは 1 の場合には，1 文字シフトして（*i=i+1*）して，(1)の処理へ進む．

(3). hook.txt にマッチするものがない場合には，1 文字シフトして（*i=i+1*），(1)の処理へ進む．

3.4. 空間ドキュメン管理トシステムの特徴

その他の機能も含めて整理すると，SDMS は，次の特徴を持つ．

(a) 単純な操作による空間ドキュメントの入力
ファイルやフォルダのアイコンのドラッグ＆ドロップだけでなく，URL 指定による Web ページの入力もサポートしている．

(b) ロバストなアドレスマッチングの提供
シンプルジオコーディングサービスを利用することにより，表記の揺れや省略に強いアドレスマッチングを提供する．街区レベルではあるが，日本全国をカバーしている．

(c) 空間ドキュメントのレイヤー管理
空間ドキュメントに含まれる POI の集合は一つのレイヤーとして管理される．複数のドキュメントが入力されている場合には，各ドキュメントに含まれる POI が別々のレイヤーとして管理され，地図上に表示される．

(d) 日本全国の背景地図の提供
背景地図は，ネットワークを介してリモートの地図サーバから取得する．国土地理院発行の数値地図 25000 をベースとした複数の縮尺の地図画像を利用できる．

(e) エキスポート機能のサポート
GIS や統計ソフトなどの他の解析ツールとの連携を考慮して，地図上の POI を別形式としてエキスポートすることが可能である．出力形式としては，Shape ファイル，G-XML，CSV 形式をサポートしている．

(f) テキスト検索や空間検索のサポート
ドキュメント単位，POI 単位でのキーワード検索をサポートしている．

4. 実装及び考察

4.1. 実装

SDMS は，Java 言語を用いて実装しており，Windows OS 上で動作する．アドレスマッチングおよび背景地図管理の機能は，リモートサーバに配置しているため，ネットワークを利用できる環境が必要となるが，それらの機能を外部に持つため，本体プログラムのサイズは比較的小さく，可搬性は高い．また，前述した通り，街区レベル位置参照情報を利用しているため，アドレスマッチング機能を無料で利用できる．アドレスマッチングの精度が重要になる場合もあるが，街区レベルのマッチングでも十分な状況が多いと考えられる．例えば，SDMS を利用して，概略的に，空間的な分布と傾向を把握し，その後で，時間をかけて，高精度のアドレスマッチングと高機能の GIS を使って，より詳細な分析を行うといった役割の分担も考え得る．多くのユーザに SDMS を利用してもらうためにも，無料での配布・利用という点は重要である．

4.2. 動作例

ユーザは，コンピュータ内のファイルやフォルダのアイコンを SDMS にドラッグ＆ドロップするだけで，そこに含まれる住所情報を，空間的な分布として見ることができる．以下，具体例について説明する．

図 5 は，ファイルのドラッグ＆ドロップの結果である．入力されたドキュメントは，あらかじめパソコン中に保存された東京都消防庁の消防署一覧のページ（http://www.metro.tokyo.jp/ANNAI/TOCHO/SOSHIKI/shoubo_c.htm）であり，ドキュメント中に含まれている各消防署の住所が POI に変換され，地図上に点分布として表示されていることがわかる．

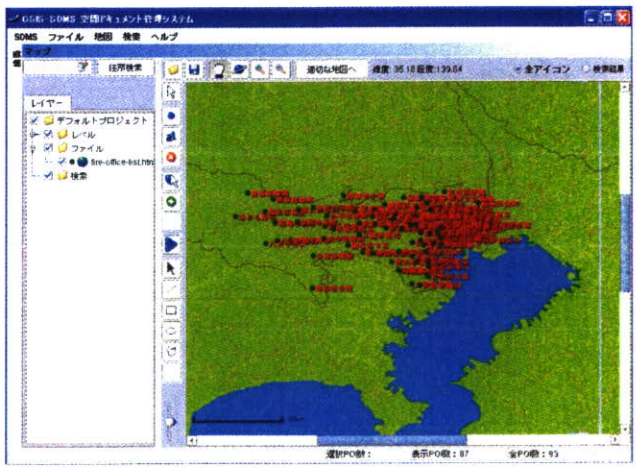

図 5：ファイルアイコン（消防署一覧の HTML ファイル）をドラッグ＆ドロップした時の実行結果

図 6 は，複数のファイルを含むフォルダのドラッグ＆ドロップの結果である．このフォルダには，光化学スモッグの発生に関する情報（東京都福祉保健局，光化学スモッグによる被害と思われる発生状況について，http://www.fukushihoken.metro.tokyo.jp/kanho/smog/higaiindex.html）が日付ごとに保存されており，各ドキュメントには発生場所に関する住所情報が含まれている．

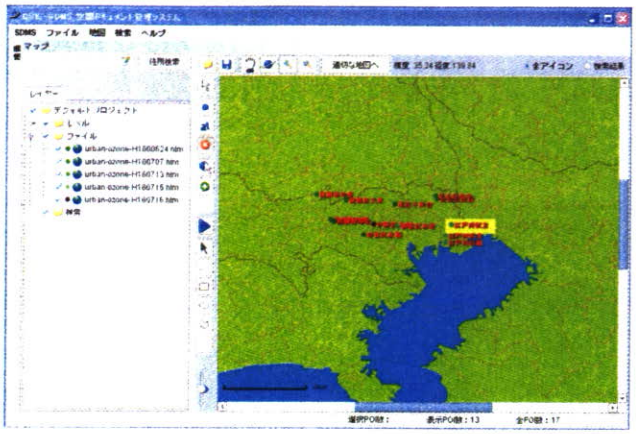

図 6：複数のファイルを含むフォルダのアイコンをドラッグ＆ドロップとした時の実行結果

図 6 の左側のレイヤー管理部において，各ドキュメントが別々のレイヤーとして管理され，それぞれのドキュメント中の POI が色の違うアイコンとして地図植に表示されていることがわかる．地図上の各アイコンは元ドキュメントと関連付けられており，マウス操作で，その POI を含むドキュメントを簡単に開くことができる．

4.3. 考察

3.3.1 で述べた通り，ドキュメント中に含まれる住所文字列の長さが短い場合には，複数の住所候補がサーバから返信される．SDMS では，複数の住所候補から一つの住所に絞り込むために，次のいずれかの基準を用いる．一つ目の基準では，変換されている緯度経度と地図の中心座標とを比較して，地図の中心点に近いものを選択する．もう一つの基準では，「東京都」や「柏市」といったキーワードを指定し，そのキーワードを含む住所を選択する．例えば，「駒場 4 丁目」という住所文字列に対して，「茨城県取手市駒場 4 丁目」と「東京都目黒区駒場 4 丁目」が候補として返されるが，あらかじめキーワードとして「東京都」を指定しておくことで，後者のみを選択し表示することが可能である．地域単位での分析や自治体など地域単位での利用を考えた場合には，都道府県名や市区町村名が省略される場合が多いため，複数の住所候補が返される可能性が高くなるが，このような場合，キーワード指定によるフィルタリングが有効であると考える．

本研究では，住所候補を抽出するために，hook.txt を用いるアプローチを取っているが，ごくまれに，住所でない文字列が住所として認識される場合がある．例えば，「昭和 45 年」「平成 6 年」といった文字列を含む場合，「昭和 4」や「平成 6」といった地名が，hook.txt に登録されているので，住所として認識され，地図上に表示されてしまう．ただし，これらの地名を hook.txt から削除してしまうと，本来住所であるものも住所として抽出されなくなってしまうので，省略された住所表現に対応するためには，hook.txt にこうした文字列を登録しておくことは必要である．この場合でも，前述したキーワード指定をすることで，ある程度の絞り込みは可能である．

現在，SDMS は，国立保健医療科学院の健康危機管理支援情報システム（H-CRISIS）[13]でダウンロード可能となっている．H-CRISIS は，保健所などの地方の公衆衛生機関が危機管理情報を閲覧し，情報を共有するためのポータルサイトである．公衆衛生や健康危機管理といった場面では，場所情報が重要であるにも関わらず，多くのドキュメントが死蔵されてしまっている現状があり，本システムの潜在的なニーズは非常に高い．健康危機に関する空間ドキュメントを SDMS によって地図化することができれば，過去の情報を活性化できるだけでなく，災害時のリアルタイムでの意思決定支援にも役立つと考えられる．専門性を必要とする高価な GIS ソフトウェアではなく，日頃から利用しているドキュメント管理ツールの方が非常時や緊急時にも効果的に活用されると考えられる．

4.4. 今後の課題

今後，ツールの配布に向けて準備を進めていく予定であるが，一般ユーザからのフィードバックを考慮して，インタフェースを改善し，より使いやすいツールとして開発を進めたいと考えている．また，現状では，地図上に表示されるのは点だけであるが，線や面などの幾何オブジェクトへの変換や地図上での重ね合わせ表示ができると空間的な関係性を把握するのに有用であると考えられる．

SDMS は，ドキュメント中から住所情報を抽出して緯度経度に変換することができるが，現時点では，郵便番号，電話番号，地名などの他の間接位置参照情報はサポートしていない．特に，地名に関しては，応用分野に特有の地名用語辞書を作成し，その辞書を選択・共有することで，SDMS を専門的な用途で活用できると考えられる．変換した POI の緯度経度情報は，地図上のドラッグ操作により容易に修正することがで

きるため，SDMSを利用して，修正されたPOIの位置情報を登録しながら，ユーザ辞書を整備していくことも考えられる．

SDMSは住所情報の抽出と表示は可能であるが，その住所が対応している実世界オブジェクトの情報をドキュメント中から探し出すことは現時点ではできない．例えば，消防署一覧の例で考えても，消防署の点分布は表示されるが，各点がどの消防署に対応しているかは自明ではない．全自動で対応するのは難しいとしても，ユーザの負荷を軽減するような対策は考案したいと考えている．

火災や交通事故などの災害事故情報，光化学スモッグ発生情報などのセンサ情報，インフルエンザ発生情報などの健康危機管理情報などを公開しているサイトでは，掲載されている情報が定期的に変化するため，そこに含まれている住所情報も変化する．空間的な分布が，時間的にどのように変化していくかを把握するためには，指定されたURLに定期的にアクセスして，Webページ中のドキュメントから住所を抽出して，生成したPOIをスナップショットとして保存し，アニメーションとして再生できると望ましい．現在，その時系列管理機能を実装中である．

5. まとめ

本稿では，住所情報を含むデジタルドキュメント（「空間ドキュメント」）を，直感的なインタフェースで扱うことのできるシステムとして，空間ドキュメント管理システム（SDMS: Spatial Document Management System）を提案し，実装したシステムについて説明を行った．SDMSは，ドラッグ&ドロップという簡単な操作で，ドキュメント中の場所情報を点として地図上に表示することができる．空間情報を扱うことのできるシステムとして，地理情報システムがあるが，定型書式を入力とするため，空間ドキュメントを直接扱うことは難しい．SDMSでは，非定型の一般的なデジタルドキュメントから，場所情報を抽出し，アドレスマッチング技術を利用してPOI(Point of Interest)を生成し，地図上に表示することができる．SDMSは，Java言語を用いて実装されており，HTML，Excel，TEXTなど多くのデジタルドキュメントを扱うことができる．SDMSは，日常的に利用できる人間中心型のツールとして開発を進めている．SDMSによって，多くの一般的なユーザが，簡単に，デジタルドキュメントを，空間的に整理し，地図化する環境が整備されていくと考えている．

謝辞

本研究は，厚生労働省科学研究費補助金（健康科学総合研究事業）「地理及び社会状況を加味した地域分析方法の開発に関する研究（代表：浅見泰司）」の支援を受けている．

文　献

[1] Google, Google Maps API Version 2 Documentation, http://www.google.com/apis/maps/documentation/

[2] Yahoo!JAPAN, Yahoo!地図情報, http://map.yahoo.co.jp

[3] GeoRSS, http://www.georss.org/

[4] Microformats-geo, http://microformats.org/wiki/geo

[5] P.A. Langley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind, Geographic Information Systems and Science, John Willy&Sons, 2001.

[6] 東京大学空間情報科学研究センター, CSV アドレスマッチングサービス, http://spat.csis.u-tokyo.ac.jp/cgi-bin/geocode.cgi

[7] E. Amitay, N. Har'el, R. Sivan, and A. Soffer, "Web-a-where: geotagging web content", SIGIR 2004, pp.273-280, 2004.

[8] 横路誠司，高橋克巳，三浦伸幸，島健一，"位置指向の情報の収集，構造化および検索手法"，情報処理学会論文誌，Vol.41，No.7，pp.1987-1998，2000.

[9] 長屋 務，森本泰貴，藤本典幸，出原 博，萩原兼一，"Google Maps API を応用したロボット型施設検索システムの試作"，データ工学ワークショップ2006，5B-i6，2006

[10] 国土交通省国土計画局国土情報整備室，街区レベル位置参照情報ダウンロードサービス，http://nlftp.mlit.go.jp/isj/.

[11] T. Sagara, M. Arikawa, and M. Sakauchi, "Spatial Document Management System Using Spatial Data Fusion", IIWAS2001, pp.399-409, 2001.

[12] 東京大学空間情報科学研究センター, CSIS シンプルジオコーディング実験（街区レベル位置参照情報），http://pc035.tkl.iis.u-tokyo.ac.jp/~sagara/geocode/modules/simple-geocode1/

[13] 国立保健医療科学院, 健康危機管理支援情報システム H-CRISIS, http://h-crisis.niph.go.jp/hcrisis/index.jsp