世界水と気候のネットワークに関するデータベース構築

Database Strategy based on World Water and Climate Network

熊谷道夫・石川俊之・焦春萌・早川和秀

要約

WWCN(World Water and Climate Network)は、2003年3月に開催された第3回世界水フォーラムを契機として設立された、陸水の研究者を中心としたNGOで、近年の気候変動によって急激に変化している世界の湖沼にかかわるデータや情報を交換し、修復に向けての貢献を行うことを目的としている。我々は、平成18年度に古今書院から「世界の湖沼と地球環境」という書籍を出版すると共に、米国カリフォルニア大学デービス校と共同で、未来型データネットワークの構築にかかわる基本設計（SODA）を行った。今後は、このシステムを展開することによって、平成18年10月に設立されたタホ湖環境科学研究センターとも協力して国際的な研究協力を実施してゆく予定である。

1. はじめに

現在、大量のデータが世界中で時々刻々と作り出されているが、多くの組織が直面する課題は、安全で柔軟なデータ形式を保ちながら、インターネット上で如何に早く簡単にデータを提供できるか、という点である。また、これらのデータはインターネットを通じてアクセスでき、ユーザーが利用している解析ソフトで分析可能なファイル形式でダウンロードできることが望ましい。

SODA（Semantic Online Data Access）は、「セマンティックウェブ」を意識して作成されたオンラインアクセス可能なデータ管理システムである。SODAは、リレーショナルデータベース(RDB)と連動するジェネリッククエリー（汎用検索）インターフェイスをインターネット上で提供することにより上記問題を解決し、様々な統計ソフト等で読み込むことが可能なファイル形式で出力できる機能を有している。SODAはまた、データを容易にSODAリポジトリ（格納場所）に加えることができるような、テンプレート型アップロード機能を備えている。そのため、もしも複数のデータが（データベース化に必要な各種パラメーターを含めて）SODAリポジトリに加えられていれば、ユーザーは、ジェネリッククエリーインターフェイスを通してデータの上位集合(スーパーセット)へアクセスし、各種パラメーターを用いてSODAからファイルを取得することが可能である。

（注）「セマンティックウェブ」は，ウェブの発明者であるバーナーズ＝リー(Tim Berners-Lee)によって提唱された枠組みであり，「ウェブに記述される情報に明確な意味の定義を与え，コンピューターと人間とがうまく協力して作業できるように」「現在のウェブを拡張したもの」である（Berners-Lee, T. et al.）。イメージはウェブの設計当初からあったとされるが，直接的な提唱は1998年頃であり，2001年の*Scientific American*誌に掲載された論文“Semantic Web”で世界的に注目され，用語としても定着するようになった。この論文の邦訳題は「自分で推論する未来型ウェブ」だが，その最終目標は，知的エージェントソフトウェアが人の代わりに問題解決のためのデータ収集・判断・評価を行ってくれるという未来図にある。

SODAを構成するプログラムは、ジェネリッククエリーインターフェイスをウェブ上で提供し、分散型の科学計測データベースへのアクセスを実現する「プラグイン型」モジュールである。SODAは、ウェブサイトポータル（例えば、DrupalやJoomla!）と連動しており、本質的に異なったデータ源どうしの橋渡し役をはたしている。その結果、ユーザーは、閲覧やダウンロード目的でデータにアクセスすることが可能である。また、新規のデータ情報は、対話型ウェブインターフェイスかあるいは自動収録機能のいずれかの方法で登録することができる。

SODAの中心的な役割を果たす登録データベースには、分散型データ源に関するメタデータが格納されている。結果として、複数のデータ源はそれぞれの内容を比較することが可能で、データが同じ「セマンティクス」を持つ場合には、それらの結果が結合される。初期設定では、誰でもSODAにアクセスすることができる。もしも必要であれば、コンテンツマネージメントシステム（CMS）に

よって、特定のデータへのアクセスを非公開化する事も可能である。ほとんどのCMSは、ユーザーに付与された“ロール”（権限）とログイン証明に基づいて、特定のページへのアクセスを制限する機能を有しており、SODAがデータを閲覧しダウンロードするページに制限を加える事ができる。この処理はCMSによって実現される機能であり、SODAシステムの枠外であるから、本論ではこれ以上これらの機能について言及しない。

2. メタデータプリミティブ

セマンティクスは、フィールド名やデータ型およびそれらの関係を含んだデータセットに関連付けられたメタデータから引き出される。SODAはまた、プリミティブ値と呼ばれる各値に意味を加えた個別のフィールドに対して、値を割り当てる。各プリミティブは、統制語（意味が明確に定義された語、Controlled Vocabulary）として、SODAの中央サーバー上に保存されている。そしてそれらは、全てのSODAのサイトに提供される。この統制語は、特定の研究分野を示す「レルム（保護・管理対象領域の論理単位）」に分割される。各プリミティブが関連づけられるのは、単位分類子、記述、理論的な高低値などである。

（注）「メタデータ」とは「データに関する（構造化された）データ」と定義され、その例としては、図書館や博物館・美術館で用いられる所蔵資料の目録や索引などが挙げられる。

プリミティブの例として水温を挙げると、その単位は摂氏かもしれないし華氏かもしれない。その単位が分かれば、他のプリミティブのように相互変換ができ、更には明示的にこの変換がなされるモジュールが構築されるだろう。

プリミティブはコレクション（レルム）として保存されるので、新しいレルムは、どのような特殊な研究分野に対してもいつでも付け加えることが出来る。レルムは、単純にいえば、データセットをより高次のカテゴリーに位置づける機能であり、これによって同じレルム由来のデータセットを簡単に比較できるようになる。レルムの例として、水質に関する全てのパラメーター類型を調べる陸水学的なものをあげることができる。これは、水質を理解するための全ての可能なパラメーターのスーパーセット（上位集合）と、様々な形で関連付けられた計測単位を含んでいる。

図書館や博物館・美術館では、対象とするもののメタデータを調べるのに、the Dublin Core(http://www.dublincore.org)のメタデータ要素集合をよく使用している。The Dublin Core Metadata Initiative(DCMA)は、より高度な知的情報発見システムを可能にする情報源を記述するために、相互運用可能なメタデータ標準と目的に特化したメタデータ語彙の普及を推進している。

SODAは、データセットを調べるためにDublin Coreと同じ要素集合を使用し、これらの数値をデータベーステーブルに保存し、さまざまな出力要求に応える機能を包含している。以下は、Dublin Coreで使用される15のメタデータ要素の表であり、SODAは、たとえ各フィールドが（SODAが扱う）各データセットの代替にならない場合でも、それらを使用する。

1. dc:title ： 情報源に与えられる名前
2. dc:creator ： 情報源の内容を作成する最終責任者
3. dc:subject ： 情報源の内容に関する主題(統制語彙)
4. dc:description: 情報源の内容説明 （アブストラクトやイメージデータの説明など）
5. dc:publisher ： 情報源を現在の形で利用可能にした主体（出版社や大学など）
6. dc:contributor：(著者ではないが)情報源の内容に貢献した人や組織
7. dc:date ： 情報源のライフサイクルにおけるイベントに関する日付
8. dc:type ： 情報源の内容についての種類またはジャンル
9. dc:format ： 情報源の物理的または電子的な取り決め（データ形式）
10. dc:identifier： 与えられた内容に含まれる情報源を一意に識別するための番号や名前
11. dc:source ： 派生した資源から元の資源への参照
12. dc:language ： 情報源の内容を記述している言語
13. dc:relation ： 関連する情報源に対する参照
14. dc:coverage ： 情報源の内容についての範囲や領域
15. dc:right ： 情報源の中や全体にわたって保持されている権利に関する情報

SODAシステムでは、Dublin Core以外に、測定単位や理論的な数値の高低、および空間的記述を含んだ他のメタデータを調べる。データ辞書とメタデータの多くは、他のウェブサービスがアクセスして情報収集が可能なpublic directoryの中にあるRDF（Resource Description Framework）ファイルに記述される。我々は、遠隔サイトがこの方法でデータを共有できるようなサービスを展開することを計画しており、SODAモジュールを用

いると、遠隔サイトにあるデータはそのまま保持した状態でそのデータ源をSODAのライブラリーに持ち込むことができる。データの問い合わせ要求が発生した時のみ、問い合わせられたデータが要求者のワークステーションに転送される。

クエリー（問い合わせ）インターフェイスは、研究者がさまざまなデータ源ライブラリーからデータを選択できるようにし、更にあるパラメーターの単位を（例えば摂氏を華氏に）変更できるようにしている。また、クエリーインターフェイスでは、ほとんどの表計算や統計パッケージで利用可能なコンマ区切りテキストファイル（.csv）の形式でデータを提供することができる。

3. データ出力

SODAにおけるデータの出力方法は2種ある。一つは端末画面上に表示するもので、ユーザーがデータの確認を視覚的に行うことができる。もう一つは、コンマ区切りのテキストファイルとして提供するもので、ユーザーのコンピューター上にダウンロードすることができる。ユーザーは、このテキストファイルの出力形式を選択できて、どのカラムを表示するのか、データがどんな計測単位なのか（この機能は現在改良中）、上部にヘッダーを表示させるかどうか、日付範囲の指定などを指定することができる。このテキストファイルは(csv形式で提供されるため）、Excelのような表計算ソフトや、Rのような統計ソフト、及びOracleや Microsoft Accessのような他のデータベースソフトに取り込むことができる。

SODAは更に、異なるデータ源に対してデータを問い合わせ、これらの情報源を一つのファイルにまとめることができる。この結果、例えば琵琶湖のデータとタホ湖のデータを比較することが、将来的にはかなり容易になるだろう。

また、メタデータの出力形式については、幾つかの出力形式をサポートする必要がある。これらのファイルは、他のインターネット情報源との相互利用性を高め、結果として、データベースを離島のようにするものではなく、むしろ、統合化されたネットワークの一部のようにするものである。基本的なデータ構造やメタデータセマンティックスは、XML（Extensible Markup Language）か、よりその用途に特化したRDF(Resource Development Framework）で記述されるだろう。データマッピング（データの他形式への変換）方法は、Dublin Coreメタデータの仕様で作成され、データセットもこの方法で表現されることになる。

将来的に、我々は他のデータフォーマットの出力をサポートしていくだろう。オープンソースソフトウエアは、他の様々なプロジェクトから得られるコンポーネントを選択することによって、きわめて容易に機能性を統合化することを可能にする。また、SODAに良く似た他のプロジェクトがあり、これらの開発者と、情報源やアイデアを貯めておくと、幾らでも双方のプロジェクトを拡張することができる。オープンソースソフトウエアは、正にこの目的のためにソースコードの使用を許可しており、この目的に沿った素晴らしい製品を開発している多くの開発者がいる（例えば、Apache webserver）。

4. データベースの移植

データは、ウェブベースのインターフェイスを用いてSODAに転送される。SODAはデータファイルを受け取り、適宜必要な形式に変換した上で(適切に関連づけられたメタデータと共に）SODAリポジトリに転送する。

データをアップロードする際には、テンプレート形式のアップロードシステムを用いる。最初に、アップロードされるべきファイルの各フィールドとそれらに関連したパラメーターを指定しなければならないが、この一連のプロセスはテンプレートとして保存することができる。したがって、ユーザーが、次に同じパラメーターを持ったファイルをアップロードする際には、（既に登録済みの）対応するテンプレートを選択するだけで簡単にファイルをアップロードすることができる。

5. 「制約の無い（仕様が公開されて誰もが利用可能な）」技術標準（オープンスタンダード）

現在SODAで利用されているRDFフレームワークに加えて、本プロジェクトでは他のオープンスタンダードも用いる。SODAモジュールは、オープンソースプログラミング言語であるPHPで記述されている。SODAは、Oracle、MSSQL、DB2などのプロプライエタリデータベースとMySQLやPostgreSQLなどのオープンソースデータベースの両方をサポートしている。これは、オープンソースデータベースアブストラクションレイヤーADOdbを用いることで可能となる。

SODAがもう少し成熟した時点で、本計画ではオープンソースコミュニティへ向けてSODAをリリースすることになる。これには、本プロジェクト以外の様々なグループの人々にもSODAに興味を持って頂き、本プロジェクトに対して時間と様々な情報を提供して頂ければという期待が込められている。

（注）「RDF」とは，WWW上のリソースに関する情報を表すための言語である。ページタイトルや著者，ウェブ・ページの更新日，ウェブ・ドキュメントの著作権およびライセンス情報，ある共有資源に対する利用可能スケジュールなどのような，ウェブリソースに関するメタデータの表現を特に目的としており，

セマンティックウェブを実現するための技術的な構成要素の一つとなっている。

6. SODAの動作要件

6.1 ソフトウェア要件

SODAの動作には、ウェブスクリプト言語のPHPを走らせるために、Apacheのようなウェブサーバーが必要である。SODAはPHPコンポーネントの一つであり、データベースアブストラクションレイヤーを用いているので、データベースバックエンドをセットする必要がない。

6.2 ハードウエア要件

ハードウエア要件はSODAのセットアップ方法に依存する。以下の3つの場合が一般的なものである。

(1) クライアント専用

このオプションを用いて、ユーザーは、他の組織やサーバー上にある自分のデータを提供するエンティティに対して、全てのデータを送ることになるだろう。

SODAの中央サーバーは、データベースにインデックスをつけ、それをSODAのライブラリーに収納する。ユーザーは、クライアントワークステーション（のウェブブラウザー）を通して、自分のデータにアクセスできる。

本方式のマイナス面としては、ユーザーが自分でデータを完全に制御できない点である。もしも、データを管理する組織をユーザーが信頼できるなら、これは最も経済的な方法である。

(2) データ・ストレージ

SODAネットワークの一部になっているデータを提供する場合は、データベースサーバーが必要である。この場合はユーザー自身が自分のデータを管理し続けるため、データの全てにわたって完全な品質管理ができるだろう。ただし、ユーザー側のデータベースサーバーはSODA中央サーバーから常にアクセス可能な状態にしなければならない。SODA中央サーバーはそれにインデックスをつけ、クエリーインターフェイスを通じてデータをユーザーに提供するだろう。

(3) データ・ストレージとSODAの中央サーバー

もしユーザーが、データストレージサーバーにSODA中央サーバーを加えるならば（実際同じサーバーの場合もあり得る）、提供したいデータソースにインデックスを付けるウェブサービスをユーザー側で運用することになる。ユーザーは、自分のデータ源を他のデータ源とネットワークを通じて結合することもできるし、他とは孤立させてユーザーの検索のみからデータを提供することもできる。

6.3 データ要件

現在のところ、モジュールの能力には若干の制約がある。データは、本来は一時的なものであるに違いない。

もしデータが異なるデータ源からのものと結合されているなら、それは同じサンプリング頻度を持つ必要がある。すなわち、現時点では、モジュールは、あるデータと他の頻度にオンザフライで合わせているデータを集約することはしない。この問題は将来解決したいと考えている。

7. まとめ

SODAは、データ・ストレージのために、特異的なイベント最適化フォーマットを使用している。このフォーマットは、データの集約や検索に加えて、データを利用してできることの柔軟性を最大限に引き出している。加えて、このフォーマットは、異なったサンプリング・ステーションから得られる、本質的に異なった均質のデータセットを結合するための最良の方法を提供する。なぜなら、その選択方法が、全てのデータ源にわたって首尾一貫しているからである。そのストレージは、データが標準的に規格化された方法よりも大きい必要はない、ということがデータに対する要件である。この比率は、今後きちんと決定される必要があり、それはシステムに導入されたデータベースがどのような規格化をしているかに依存しているけれども、恐らく20～30%増し程度にしかならないだろう。

RDFトリプルに見られるテーブルのように、SODAのテーブルは、「薄く長い」。SODAのテーブルは3つ以上のフィールドを有している（RDFトリプルのテーブルを作る）が、個々のデータ数値は、データベースの中では一行に保存される。この列に関連付けられたものとして、日付表記や特定の観測場所(位置コード)、数値形式(成分タイプ)があり、これらは、プリミティブバリュー、バリュークオリファイヤー、バリュー（数値）と呼ばれる。

（注）RDFはリソースの「関係」を主語、述語、目的語という3つの要素（トリプル）で表現し，これをRDFトリプルと呼ぶ。

日付表記は、日付範囲全体をより速く集計するために、整数値として付加的に保存される。データフィールドから数値を抽出するのにかかる時間およびその操作の連続的な発生のために、付加するのに必要なスペースは、演算にかかる時間節約とくらべて無視できる。これは特に大きなデータセットを検索する場合に顕著である。例を挙げると、日付データを整数値として保存することで、集計を含むすべての問い合わせに必要な“グループ毎の”演算を最適化することができる。もし時間の次元を持つ特定のデータ列に対してその数値が一定ならば、最小時間間隔は、最初の1ステップであると仮定できる。例えば、分を表す数値が全て同じ値で、時間を

表す数値がかなり変化しているならば、データは時間ごとに集められたものだろうと考えることができる。もちろん、これが必ずそうなるだろうというわけではないが、この例は、こういったテストを行う場合の1つの強い規準を与えている。

一般的に、データは特定の場所で収集あるいは採取されているはずである。すなわち、問題となるデータに対して（観測地点などの）ある空間情報が存在する。観測ステーションは、それ自身のメタデータセットを有しているため、レファランス(場所のID)のみが数値テーブルに保存されている。ステーションテーブルに入って検索することによってステーションに関する情報を取り出すことができる。空間成分は、対象とする空間成分が、地面から離れるか近づくかする時、少し扱いにくくなる。高度や深度のような三番目の次元は、緯度･経度の座標系として保存できなくなるので、こういった三番目の次元が必要となる。それゆえに、追加のフィールドが、ステーションテーブル（単位やデータ形式と同じように、この三番目の次元におけるデータを集めるのかどうかというブール型数値）と、観測点からの相対的な距離を表す数値テーブルの両方に付加される。これらの数値は元の状態で保存され、結果として、水深と高度は共に正の値を持ち、ステーションコードは、これを意味するインデックスを浮動小数点値で与える。

使用するモジュールは、ネバダ州のトラッキー川沿いで集められたセンサーデータに対してオンラインアクセスを提供するための、トラッキー川水質プログラムというプロジェクトにおいてテストされている。このプロジェクトではアップグレード版の方がすぐれてはいるが、現在は、初期の機能を有したバージョンが用いられている。トラッキー川プロジェクトは、SODAモジュールのためのたたき台として引き続き使われるだろう。

全てのデータ数値は、クオリファイヤー（もしくは品質保証コード)を入力するために利用可能なフィールドを有している。これは、かなり標準的なデータ要素であり、SODAはこれを維持するための努力を続けている。これらのクオリファイヤーコードを保存するために、追記テーブルがセットアップされる。これによって、そのテーブルへのアクセス時にデータ数値を返すことが可能になる。

我々は、琵琶湖環境科学研究センターが作成するネットワーク型のデータベースを、LBERI-DOS(Data Online System)と名づけ、図1のように位置づけている。これは、センターが収集し、提供するデータセットは、研究目的

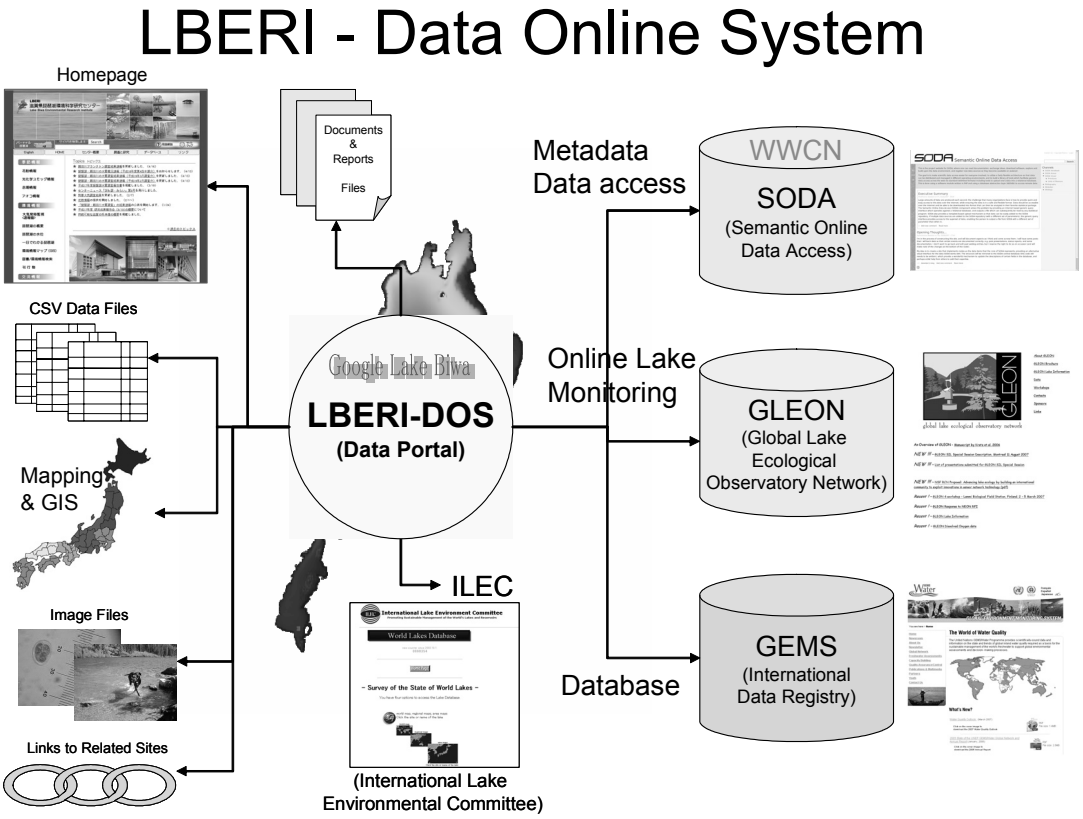

図1　琵琶湖環境科学研究センターが志向する次世代型データ検索システム

を優先し、データの質・情報源・コピーライト等を担保しようというものである。したがって、不特定多数のユーザーへのサービスは、他の機関と連携し提供することを想定している。これは、既存の施設なり組織と有効に協同することと、現存する人的・経費的資源を最大限効率化することを意図している。その意味で、ひとつの機関が占有してサービスを提供するという従来の方式とは大きく異なる点に独自性を有している。

付録

将来的に、SODAは、NetCDFを含むかなり広い範囲のデータフォーマットをサポートするだろう。NetCDFは、データを転送するのにいくつかのプラットフォームで使われている。そしていくつかのアプリケーションが、このフォーマットをネイティブに利用している。様々なツールを使えば、SODAが将来的にNetCDFをサポートする事は特に困難ではない。SODAシステムはかなりモジュール化されているので、他のプロジェクトの成果を一緒にしたり、機能を付け加えたりすることが、かなり容易である。それは、殆どが、2つのフォーマット間の往来(cross-walk)を作り出すことによって記述されてきた、いくつかのオープンソースの構成要素を統合化する問題である。

ウェブベースのアプリケーションは、現時点では、PHPで記述されている。PHPはオブジェクト指向型のウェブベースのスクリプト言語である。SODAはPHPの持つオブジェクト指向型の特徴が利用されており、従って必然的にUMLを用いたモデルを概説する必要がある。

UMLダイアグラム

このダイアグラムは、現在、未完成である。

(注) UML (Unified Modeling Language)
オブジェクト指向のソフトウェア開発における，プログラム設計図の統一表記法。Rational Software社のGrady Booch氏，James Rumbaugh氏，Ivar Jacobson氏の3人によって開発された。従来，オブジェクト指向設計の表記法は50以上の規格が乱立していたが，1997年11月にOMGによってUMLが標準として認定された。Microsoft社やIBM社，Oracle社，Unisys社などの大手企業が支持を表明している。

文献

Berners-Lee,T. et al. 村井純ほか訳 (2001) : 自分で推論する未来型ウェブ.日経サイエンス.31(8) : 54-65.

Berners-Lee,T. et al (2001) : The Semantic Web. Scientific American, 284(5) : 34-44.