

適切かつ高度なデータ処理環境を提供するための ウェブインターフェース “Web Processing Catalog” かぐや搭載スペクトルプロファイラデータを事例として

林 洋平^{*1} 山本幸生^{*1} 小川 佳子^{*2} 平田 成^{*2} 出村 裕英^{*2} 寺菌 淳也^{*2} 松永 恒雄^{*3}

“Web Processing Catalog” : Web interface for distribution of
the appropriate and advanced environment for the data processing

– A case study for the data from the Spectral Profiler on board Japanese satellite Kaguya

HAYASHI Yohei, YAMAMOTO Yukio^{*1}, OGAWA Yoshiko^{*2}, HIRATA Naru^{*2},
DEMURA Hirohide^{*2}, TERAZONO Junya^{*2}, and MATSUNAGA Tsuneo^{*3}

Abstract:

In the planetary science, many observation data are available via internet. To get some information from the observation data, a lot of steps of data processing are necessary. Either the data providers or users have to handle such processes. For the case where the providers are responsible, the data processing is a black box for the users usually, so the potential of the data could not be fully utilized. On the other hand, for the case the users themselves follow the processes, the data usability would be the problem, because the users need to know many technical things and they sometimes could not afford that. We focus on the importance of the relationship between the data and data processing. We propose a new protocol “Web Processing Catalog” for developing web interface to distributing the metadata and desirable programs for data processing. Finally, we also implement a preliminary system and conduct verification where we use the actual data from the spectral profiler onboard Japanese lunar orbiter, Kaguya.

keywords: Data Processing, Scientific Data, Metadata, Web Service, Web Processing Catalog, GEKKO, Kaguya (SELENE), Spectral Profiler

概要

惑星科学の分野ではウェブ上のサービスを通じて科学データが利用可能である。科学データから情報を得るためには目的に応じてデータ処理がなされる必要がある。このデータ処理はサービス提供者あるいはユーザ自らが行う必要がある。前者の場合、データ処理のプロセスはブラックボックスになりがちであり、データが本来もつ可能性を發揮しきれない可能性がある。後者では、データの利用には多くの専門知識を必要とする場合があり利用が阻まれる場合もある。そこで上記の問題を解決するため、データとデータ処理プログラムを関連付ける重要性を明らかにする。そして、ユーザへ適切かつ高度なデータ処理環境を提供することを目指し、両者を関連付けるメタデータおよびデータ処理プログラムを配信するウェブインターフェースを構築するためのプロトコル “Web Processing Catalog” を提案する。また、実際の月探査衛星「かぐや」に搭載されたスペクトルプロファイラ (SP) のデータを用いた試験的なシステムの実装を行い検証をおこなう。

1. はじめに

惑星科学の研究対象である太陽系の天体は望遠鏡や探査衛星によって様々な観測がなされ、多くの科学データが蓄積されている。本論における科学データは、可視化処理がなされていないデータを指し、様々なデータ処理により様々な情報が取

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*1} 宇宙航空研究開発機構宇宙科学研究所
(Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency)

^{*2} 会津大学先端情報科学研究センター

(Research Center for Advanced Information Science and Technology, The University of Aizu)

^{*3} 国立環境研究所 (National Institute for Environmental Studies)

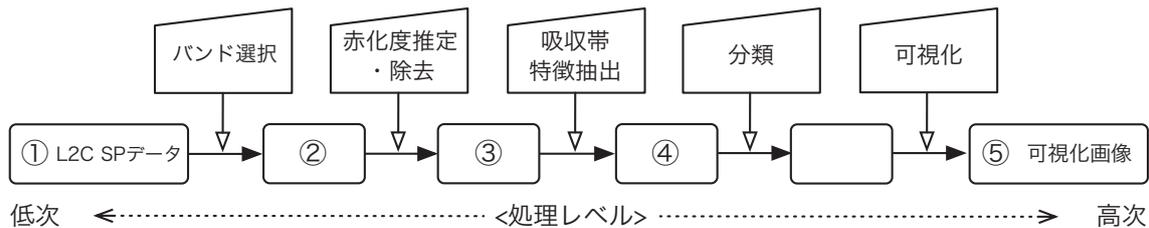


図 1 データ利用に至るまでのデータ処理プロセス

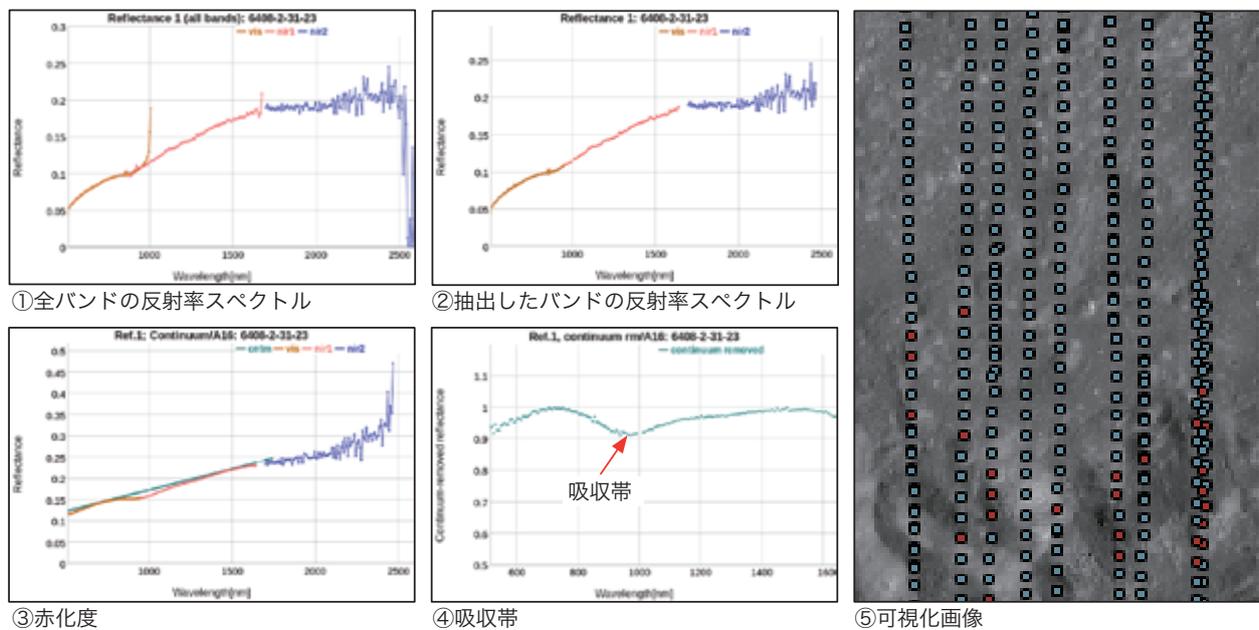


図 2 図 1①～⑤のデータについてそれぞれ可視化したグラフおよび画像

り出せる可能性を持っているものとする。科学データが得られている天体は、大気が薄い火星や地球の月のように変化が比較的少ない天体がある一方で、金星や木星のように濃い大気に覆われ刻一刻と変化する天体があり、イトカワのように球体ではない比較的小さな天体もある。大小・性質も様々な多くの惑星が観測の対象となっており、それぞれ観測対象特有の現象を明らかにするためには、様々な観測機器によってもたらされる科学データを活用する必要がある。地球観測で用いられていない観測機器を用いて観測が行われることもある。一般的に利用されていない最先端のセンサーあるいは特殊なセンサーのデータは、データの取り扱いが可能な人材に限られると推測される。

例えば、かぐやデータアーカイブ¹⁾で配信されているプロダクトレベル 2 (L2C) の SP データ (図 1 ①) の場合、鉦物分布情報を読み出すためには図 1 に示したように「バンド選択」、「赤化度推定・除去」、「吸収帯特徴抽出」、「分類」、「可視化」といったデータ処理が必要となる。これらの一連のデータ処理を科学データに適用することによって、図 2⑤に示した画像が得られる。データ処理により変化する実際のデータについてわかりやすくするため、図 1①～⑤のデータについてそれぞれ可視化したグラフを図 2 へ参考として掲載した。これらのグラフは「月光」(後述)で作成した。図 2④の「吸収帯」は、「赤化度推定・除去」を行ったデータにおける窪みの最深値を示す波長であり、矢印で示した部分になる。図 2⑤の「可視化画像」は、分類のデータ処理の結果に基づきグループごとに色を割当て、別途用意した Lunar Reconnaissance Orbiter 搭載の Wide Angle Camera (WAC)²⁾の画像上へ四角い点としてプロットしたものになる。図 ⑤の四角い点のひとつひとつが SP の観測点であり、四角い点が同色であればスペクトルが似通っていることを示している。このようなデータ処理はデータへの理解がなければ難しく、ノウハウの固まりであるといえる。ユーザが科学データを利用するためには、使用目的に応じてデータ処理を行い、画像やグラフなど可視化された状態にする必要がある。

このような状況に対して林ら³⁾はウェブアプリケーション「月光」⁴⁾を開発した。「月光」は、SP データをウェブブラウザから閲覧できる地理情報システム (GIS) である。SP は可視から近赤外までの波長域 (0.5～2.6 μm) で月面からの反射光を 296 バンドで連続的にとらえる分解能を有している⁵⁾⁶⁾。「月光」では、全 SP データ (L2C) を搭載しており、ユーザはオリジナルの生データをはじめ、「利用に適したバンドの抽出」や「ノイズ除去」、「赤化度推定・除去」などのデータ処理を行なったグラフを閲覧できる。現在、著者らは「月光」上で SP データの本格解析を可能にするためのデータ処理機能の拡張を検討している⁷⁾。現在の「月光」では上記に示した若干のデータ処理機能を備えているが、「月光」上でユー

ザ自らが用意したデータ処理プログラムを適用することはできない。逆に「月光」のデータ処理プログラムを「月光」以外で用いることもできない。著者らは「月光」の拡張にあたり、「月光」で完結しないオープンなシステムづくりを目指している。そこで、ウェブサービスにおいて適切かつ高度なデータ処理を実現するため、既存のウェブサービスで用いられている地図データや科学データの配信に用いられている代表的なプロトコルを検証しつつ、新たなプロトコルの提案を行う。

2. 各種プロトコルにおけるユーザによるデータとデータ処理への関与

多くの科学データはインターネットによって配信されている。配信側とユーザ側はインターネットによって隔てられており、データ配信のためのプロトコルを用いて配信が行われる。また、科学データはユーザによって利用されるまでの間に必ず何かしらのデータ処理が行なわれている。データ処理は、配信側、ユーザ側、あるいはその両者が関わっている。データ処理がどこで、どの程度おこなわれるかによって、ユーザのデータ処理への関わりが大きく変わってくる。本章ではモデル図を示しつつ、地図配信が可能なプロトコルを3種類とりあげ、科学データからデータ処理を経て、可視化画像へ至るまでのユーザのデータとデータ処理への関与について述べる。

2.1 モデル図の凡例

図3以降のモデル図の解説を行う。各モデル図の左側の「サービス提供側」としているブロックはサービス提供者のコンピュータ環境であり、右側の「ユーザ側」のブロックはユーザのコンピュータ環境になる。コンピュータ環境はサービス提供側とユーザ側でそれぞれひとつにまとめられているが、実際は分散環境であったり、サービス提供側がデータ処理を行っていない場合もある。しかしながら本議論の本質ではないため図のようにサービス提供側とユーザ側とに単純化して取り扱うこととする。

「角が取れた四角」は「データ」を、「台形」は「プログラム」を表している。可視化されたデータを閲覧するためにはGISアプリケーションなどのアプリケーションが必要だが、コンピュータ環境においてデータ処理以外のこれらのプログラムについては省略する。

サービス提供側とユーザ側にまたがっている「薄黒実線の矢印」は「インターネットでのデータ配信」を示す。水平に伸びている「黒く塗りつぶされた矢印」は、「矢印の始点につながっているデータから終点につながっているデータへの変換」を示している。垂直に伸びている「白抜きの矢印」は前記の変換に際して「データ処理の適用」を示している。各種「矢印」や「データ」「プログラム」について、実線の場合はユーザが操作できることを示し、点線の場合は、ユーザが操作できないことを表している。

2.2 Web Map Service

Web Map Service (WMS)⁸⁾ は、ユーザがインターネットを通じて地図画像を取得するインターフェースを定めた標準プロトコルである。地図画像は位置情報を持った画像のことである。地図画像は、本論が対象とする科学データから作成される可視化画像以外にも、ベクターデータによって作成される地質図や道路地図なども含む。アメリカ地質調査所⁹⁾ やNASAのジェット推進研究所 (JPL)¹⁰⁾ は、WMSを用いることにより月や火星を始め探査が行われた天体の衛星画像や地形図などを配信している。ユーザは、WMSにおいて定義されている「GetMap」リクエストを用いてデータリソース、座標系 (CRS) および空間的範囲を示したバウンディングボックス (BBOX) などの情報をサーバへ送ることによって、レスポンスとして地図画像を得ることができる。

ここから、データ配信にWMSが用いられている場合におけるデータ処理について検証する。図3(a)に示すように、サービス提供側のデータ処理にはユーザ側は関与できない。WMSを用いて可視化画像を行う場合、配信に先立ちサービス提供側でデータ処理が必要となる。データ処理が専門家の監修のもとで行われている場合、妥当性が保証されているといえる。

また、配信後のデータ処理に関して、WMSは基本的にはウェブブラウザで用いられることが想定されており、配信後の画像に対してデータ処理を行うことは想定されていない。WMS 1.3.0¹¹⁾ では、地図画像のフォーマットとして、JIF、PNG、JPEG、TIFFが示されている。多次元ラスタや科学データをそのまま画素値として割り当てた画像、特に符号なし整数型以外のデータ型の画像などは規格で想定している利用の範囲を越える。

ところでWMSには、「GetCapabilities」リクエストがあり、サービス提供者はメタデータを配信することができる。そこで「abstract」の項目を用い、データ処理に関することをユーザ側に伝えることは可能である。例えば、JPLのonMoon¹⁰⁾ は、「abstract」を通じてデータ処理に関する情報を自由記述文で配信している。

2.3 Web Coverage Service

Web Coverage Service (WCS)¹²⁾ は、ユーザがインターネットを通じて地理空間におけるカバレッジ (被覆) データを取得するインターフェースを定めた標準プロトコルである。WCSを用いて配信されるカバレッジデータは、ユーザ側でデータ処理を行うことができる可視化処理が行われていないデータである。ユーザは、WCSにおいて定義されている

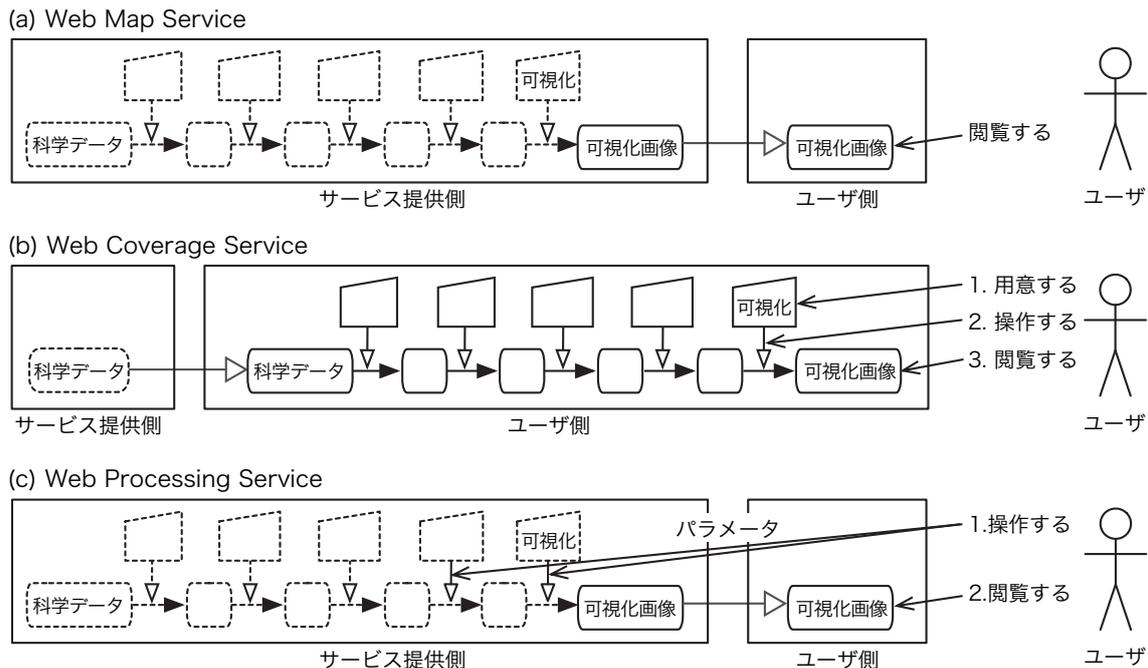


図 3 各プロトコルにおけるユーザとデータ処理の関係を示した模式図

「GetCoverage」リクエストを用いて CRS や BBOX などの情報に加え、それぞれのデータ特有の時間軸や特定の波長といった詳細な切り出しの情報などをサーバへ送ることによって、レスポンスとしてカバレッジデータを得ることができる。

ここから、データ配信に WCS が用いられている場合におけるデータ処理について検証する。WCS では配信用のデータフォーマットとして GeoTIFF, NetCDF, JPEG2000 などを明示しており、これらは科学データの格納に適している。図 3(b) に示すように、WCS で配信されるデータは可視化処理が行われていないため、ユーザがデータ処理に直接的に関与することになる。すなわち、ユーザは多くのデータ処理についてプログラミングをするか既存のプログラムを用意する必要があり、データに関する様々なノウハウがユーザには要求される。その一方で、様々なデータ処理を駆使することにより様々な可視化が可能であり、データのもつ可能性を十分に引き出せる。そのためデータ操作の自由度が高いといえる。その半面、ユーザ側のデータ処理の妥当性については保証されない。

2.4 Web Processing Service

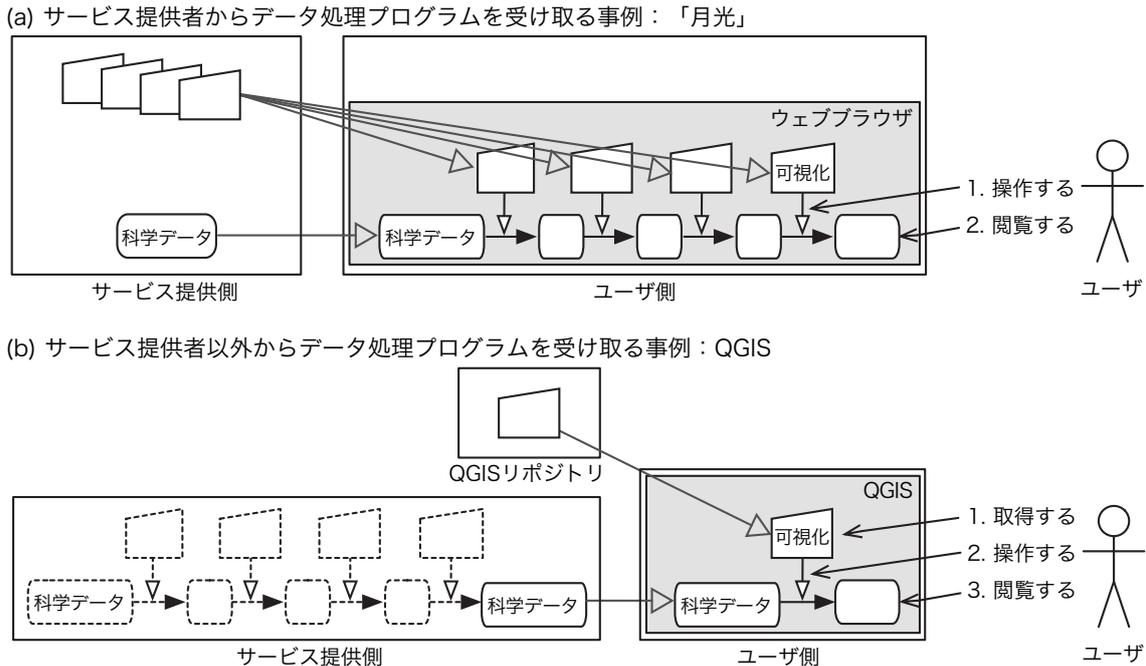
Web Processing Service (WPS)¹³⁾ は、サーバーのデータ処理をインターネットを通じて実施するインターフェースを定めた標準プロトコルである。ユーザは、WPS において定義されている「DescribeProcess」リクエストの実行によってデータ処理に必要なパラメータ値の情報を得られる。「Execute」リクエストを用いてパラメータを指定することによってデータ処理を実行できる。「GetResult」リクエストによってデータ処理結果を得ることができる。

ここから、データ配信に WPS が用いられている場合におけるデータ処理について検証する。図 3(c) に示すように、ユーザは可視化された地図画像をインターネットを通じて得られるという点では WMS と同様だが、上述のようにデータ処理の部分についてパラメータをサーバ側に渡すことによって関与できるという点が大きく異なる。だが、WPS を通じてサーバで実行することが可能なデータ処理は、あらかじめ配信側が用意した特定のデータ処理に限られる。図 3(c) の例では、配信側に 5 つのデータ処理があるが、そのうち右側 2 つのデータ処理にユーザが関与していることを表現している。その他のデータ処理にはユーザが関与できないことを示している。

WPS では、ユーザがデータに関するノウハウを持ち合わせていなくとも、ある程度データ処理に関与できる。図 3(b) の WCS のようにデータ処理の変更を自由にできるというわけではない。また図 3(a) の WMS のように、科学データから抽出可能な情報の自由度が一切ないという状態でもない。WPS では、制限付きではあるが動的にデータ処理が反映された可視化画像を得ることができる。

どのようなデータ処理が行われるかは、パラメータが設定できる部分に関してはおよその推測が立つので、データ処理の明示性はある程度は保たれる。とはいえ配信側のデータ処理をユーザ側で完全に再現することは困難である。またパラメータ以外の部分（図 3(c) の点線部分）はサービス提供者が明示していない限りは不明である。

なおデータ配信側でのデータ処理は、データ特性を知り尽くした専門家の監修のもと作成されており、妥当性が保証されているといえる。



2.5 既存のプロトコルの課題

本章では3つのプロトコルごとに、ユーザの側からみたデータ処理に注目した。データ処理が配信側かユーザ側のどちらで行われるかによって、データの持つポテンシャルと学習コストなどのユーザ負荷が変わり、両者はトレードオフの関係であるといえる。科学データの持つ可能性、すなわち情報を引き出すためにはユーザがデータ処理に強く関与する必要がある。WMSでは、データへの関与はできず、科学データのもつ一側面の情報のみ得られるが、ユーザのデータ処理への負荷が全くない。WCSでは、データへの関与ができ、科学データの多面的な情報を引き出せる可能性があるものの、ユーザのデータ処理への負担が高い。WPSでは、ユーザの負荷を抑えつつデータ処理への関与が可能だが、科学データから引き出せる情報は限定的である。

3. データ処理プログラムの配信事例

科学データの持つポテンシャルを引き出すためには、ユーザのデータに対する関与が不可欠である。一方で、科学データに適したデータ処理プログラムを準備し適用することについては、ユーザ側への負荷が高い。そこで、科学データに適した処理プログラムを配信すればユーザ側への負荷を下げつつ、データの持つポテンシャルを引き出せる可能性がでてくる。そこで、ユーザ側へデータ処理プログラムを配信している事例をみていくことにする。なお、ウェブGISにおいてデータ処理プログラムを配信するインターフェースを定めたプロトコルは現状では存在しない。配信側のデータ処理のプログラムをユーザに配信し、ユーザ側でデータ処理を反映できればデータの持つポテンシャルを保ちつつ、ユーザ負荷を下げるができる。その実現のために本章では利用可能な既存のデータ処理のプログラムについて検証していく。

3.1 サービス提供者からデータ処理プログラムを受け取る事例：「月光」

先述の「月光」はデータ処理プログラムを受け取れる。月光の起動とともに、データ処理プログラムはサーバからダウンロードされ、ユーザの指示によってデータ処理プログラムが科学データに適用され、グラフという形でデータの可視化が行われる。「月光」では、図4(a)のように、データ処理プログラムと科学データは同一の配信者となっている。データにデータ処理が結び付けられており、データとデータ処理との関係は緊密である。

一方で、データ処理の利用は月光のサービスでのみに限られる。月光のデータ処理はJavaScriptでコーディングされているが、データのインターフェースなど様々な処理と一緒にまとめられた月光ライブラリとして配信されている。各データ処理を個別に取り出し、反映させることは難しい。月光以外のデータに月光ライブラリのデータ処理を反映させることはできない。

また、「月光」はスペクトルの可視化・閲覧を念頭に作られており、本論文執筆時点では、図1の「吸収帯検出」、その特徴に応じた「分類」、分類結果の「可視化」といったデータ処理を行うことができず、図1⑤で示した可視化画像を生成することができない。図1③のデータはダウンロード可能であるため、このデータを用いてその後の処理は別途行う必要がある。

	データとプログラムの 配信者	データに対する妥当な データ処理の適用	データ処理の組み換え	他のアプリケーション でのプログラム利用	他のプログラムの適用
月光	同じ	◎	×	×	×
QGIS	違う	△	×	×	○ プラグインとして
新方式	サービスとして 分けるべき	◎	◎	◎	◎

表 1 データ処理プログラムの比較表

3.2 データ提供者以外からデータ処理プログラムを受け取る事例：QGIS のプラグイン

QGIS¹⁴⁾ はフリーで利用できるデスクトップ型 GIS アプリケーションである。QGIS には標準の機能以外にも、GIS で一般的に用いられる処理について、多くのプラグインが有志により開発されている。それらのプラグインは QGIS リポジトリからダウンロードすることによって利用できる。

QGIS のプラグインは、図 4(b) に示したとおり、データの配信者とデータ処理プログラムの配信者は異なる。このためデータとデータ処理との関係は希薄である。ユーザがあるデータ処理を本来適用すべきでないデータに反映させてしまう可能性が残る。また、提供されているプラグインは配信されるデータの処理レベルが画像データのような処理レベルが高次である場合に限定される。さらに、プラグインは QGIS でのみの利用に限られ、他のアプリケーションでの再利用はできない。

3.3 既存の利用可能なデータ処理プログラムの課題

上記のとおりユーザが利用可能なデータ処理プログラムの実例はあるが、それぞれのサービスには課題がある。表 1 にまとめを示す。

まず第 1 の課題は、データとデータ処理との関係が、ある意味、ユーザの曖昧な信頼で成り立っているという点である。データに詳しい者がデータとデータ処理の関係を保証することが必要である。データとデータ処理の配信元が同一であれば、組み合わせはまず間違いないといえる。あるいは、データ処理プログラムの配信者が、特定のデータに対して対応していると明示していれば、あやまったデータ処理の適用が防げる。

第 2 の課題は、データ処理が配信されたとしても、「月光」のように特定のアプリケーションでのみの利用に限定されると、ユーザ自ら用意したプログラムの適用ができない、プログラムの再利用ができないといった問題が生じる点である。データ処理は特定のサービスやアプリケーションから独立であるべきであると考えられる。

4. 提案するプロトコル

これまでの検討を軸にして、既存の方式の課題を解決するための基本概念を整理し、再考すると以下にまとめられる。

1. データとデータ処理は関連付けられるべきである。
2. データ処理はモジュールとして組み合わせ可能な粒度が望ましい。
3. データ処理プログラムは特定のアプリケーションから自由であるべきである。
4. データ処理の組み合わせは、パッケージとしてまとめられるべきである。
5. 配信側の一連のデータ処理はユーザ自らの操作を可能とするためパッケージとして明示するべきである。
6. パッケージはデータ処理のガイドラインという位置づけであり、ユーザによるパッケージ内容の変更を可能にすべきである。

事項以降では、これらの基本概念について検討をおこなう。

4.1 基本概念 1：データとデータ処理プログラムの関係

前章で指摘したとおり、既存のデータ処理プログラムの配信では、データとデータ処理プログラムの関係は明示されているわけではなく、ユーザによる曖昧な信頼に限られていた。しかしながら、信頼に頼ると、プログラムの配信者が想定していなかったプログラムが誤って利用される可能性がある。これは、可視化された画像やグラフについて、それぞれの図がどのように作られているかプログラムレベルで明示されていないために生じる。

図 5 に示したように、データとデータ処理の関係性を明示的に示したメタデータを配信すれば、他者の検証が可能になり、より適切なデータへのデータ処理の適用が可能になるといえる。パッケージが対象とするデータとデータ処理によって得られるアウトプットを明示することが重要となる。

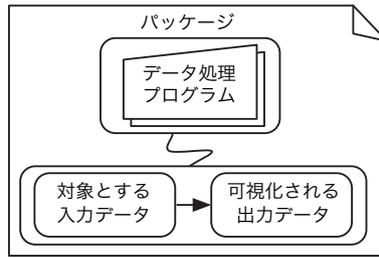


図 5 データとデータ処理の関係を示したメタデータの概念図

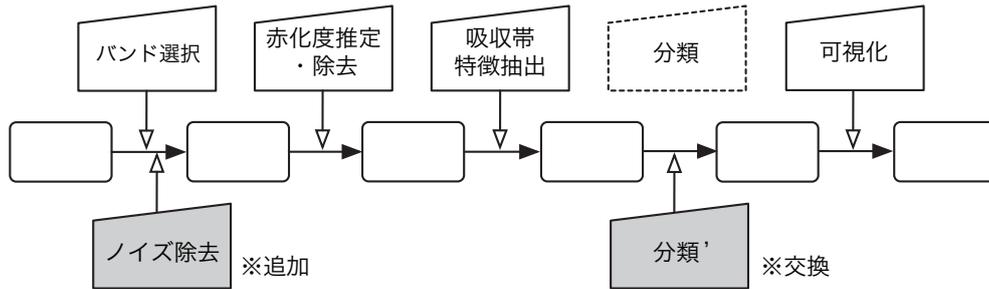


図 6 ユーザによるデータ処理への関与

4.2 基本概念2：望ましいデータ処理プログラムの粒度

データ処理プログラムは単に配信されれば良いというわけではなく、配信処理の構成自体に工夫が必要である。

図1で示したデータ処理について、例えばただ単にひとつかたまりのプログラムとして配信したならば、ユーザのデータ処理への関与は、単にそのひとつかたまりのデータ処理を実行するかしないかの選択肢に限られてしまう。一方、データ処理の粒度でプログラムを配信すれば、ユーザはデータ処理の粒度でデータ処理に関与できる可能性が生まれる。

前者も後者も得られる結果は同じだが、後者ではユーザがデータ処理に関与することによって得られる結果に変化を加えることができる点異なる。

図6の灰色で塗りつぶしたデータ処理は図1からの変更であり、新たに「ノイズ除去」を追加したり、「分類」を「分類'」に交換している。適切な粒度でデータ処理のプログラムを配信すれば、様々なデータ処理を自由に組み立てられる。ユーザの関与の仕方得られる可視化データは多様に変化する。このデータ処理のプログラムは実際には数値計算の関数によって構成される。

4.3 基本概念3：オープンなデータ処理プログラム

データ処理プログラムは、他のアプリケーションからも実行可能にするためオープンにするべきである。

特定のアプリケーションに組み込んであるデータ処理は、そのアプリケーションでのみの利用に限られ、オープンでない。対話型プログラムやGUIアプリケーションが該当する。一方で、OSのターミナル上で動作するコマンドラインや実行可能なライブラリは、ユーザが直接実行することができることはもちろん、他のアプリケーションからも実行可能であり、オープンである。

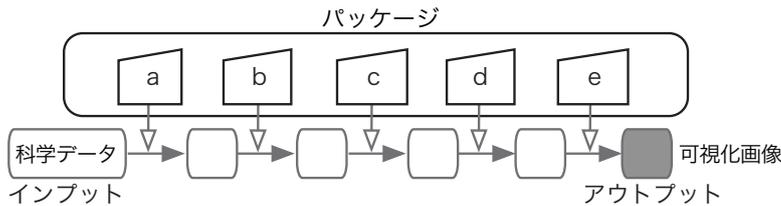
コマンドラインのデータ処理は、ユーザにとっては馴染みがないかもしれない。そこで、オープンなデータ処理プログラムに対して、GUIでラッピングする利用が考えられる。この方法であれば、データ処理のユーザビリティを保ちつつ、オープンなデータ処理を担保できる。参考になるのはGISアプリケーションのGRASS GIS¹⁵⁾である。GRASS GISはコマンドラインでの操作が基本だが、クロスプラットフォームなGUIツールキットのwxPythonを用いたGUI環境を公式に提供している。また、GUIベースのGISアプリケーションであるQGISから、GRASS GISの機能呼び出すこともできる。

4.4 基本概念4：一連のデータ処理プログラムのパッケージ

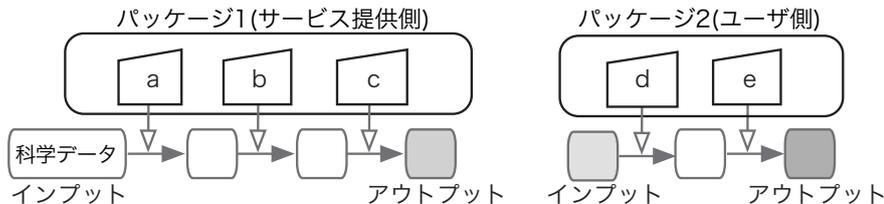
4.2のような粒度で構成されたデータ処理プログラムがあるとして、このプログラムだけでは実用的ではない。図1に示したとおりプログラムを連続的に操作することによって科学データから可視化された情報を得ることができる。このことからデータ処理プロセスを個々に配布することはもちろん、図7(a)に示したようにデータ処理のフローをパッケージとして配信することも必要である。

その際に、パッケージとしてまとめられたデータ処理のフローは図7(b)のとおりサービス提供側とユーザ側のそれぞれで利用が可能である。サービス提供側は、いかにデータ処理がなされたかを明示する目的でパッケージを利用できる。詳細は

(a) 科学データから可視化までの一連のプログラムを含んだパッケージ



(b) サービス提供側とユーザ側とに分割されたプログラムのパッケージ



(c) プログラム群のパッケージの利用例

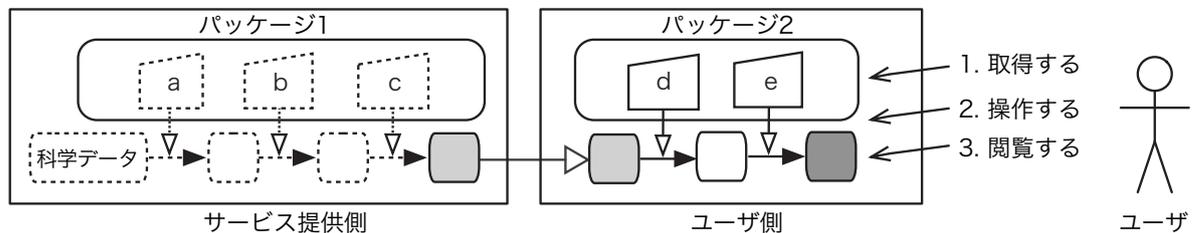


図 7 パッケージ化されたデータ処理とその適用

4.5 で述べる．ユーザは、受信したデータに対してデータ処理を適用する目的でパッケージを利用できる．詳細は 4.6 で述べる．この場合、図 7(c) に示したように、配信側とユーザ側の両者にわたって、データ処理を明示的に示すことが可能となる．

4.5 基本概念5：配信されるデータのリバース・エンジニアリング

配信されるデータがどのように作成されたかを示すことは、配信データの検証およびデータリテラシーにおいて重要である．どのようなユーザであっても初めて遭遇するデータに対しては未知の部分が多い．配布されるデータが高次であったとしても、データのリバース・エンジニアリングが可能であれば、そこをきっかけにデータに対するオペレーションに詳しくなることが可能である．

ユーザは、どのようにデータ処理がなされているかを確認するところからデータ処理の関与を始めることができる．次に、上記に述べたデータ処理の組み換えができるようになるだろう．さらに、上級者になれば特定のデータ処理に対してプログラミングを行い、そのプログラムを適用することが可能である．

4.6 基本概念6：ユーザによるデータ処理の組み換え

パッケージはそのままデータに対して適用してもよいが、ガイドライン的な位置づけとして利用することもできる．図 6 に示したように、ユーザはパッケージ中に示されている内容を参考にしつつ、データ処理をひとつずつ追いかけることができる．また、ユーザは必要に応じてデータ処理プログラムの追加や交換などの操作を自由に行うことができる．ユーザにとってデータ処理の敷居がさがることにつながる．

4.7 新方式の可能性

上記の基本概念を踏まえてシステムの実装を行うならば次に示す可能性が見えてくる．

第一に、データに対してどのような操作をするべきかわからないユーザであったとしても、可視化済みのデータを閲覧するのと変わらない体験を得ることができる．

第二に、ユーザの能力次第で、データのもつポテンシャルを引き出すことが可能となる．配信されているデータ処理に加えて、ユーザ自らプログラミングしたデータ処理プログラムも組み合わせた処理フローが可能である．データ処理に長けたユーザは自らが必要と考えるデータ処理のみの開発に集中することができる．

第三に、データ処理をオープンソース化し共有化することによって、ユーザはデータ処理をソースコードのレベルで理解することが可能となる．また、様々な外的な評価を受けることによって、データ処理の信頼性を高めることができる．

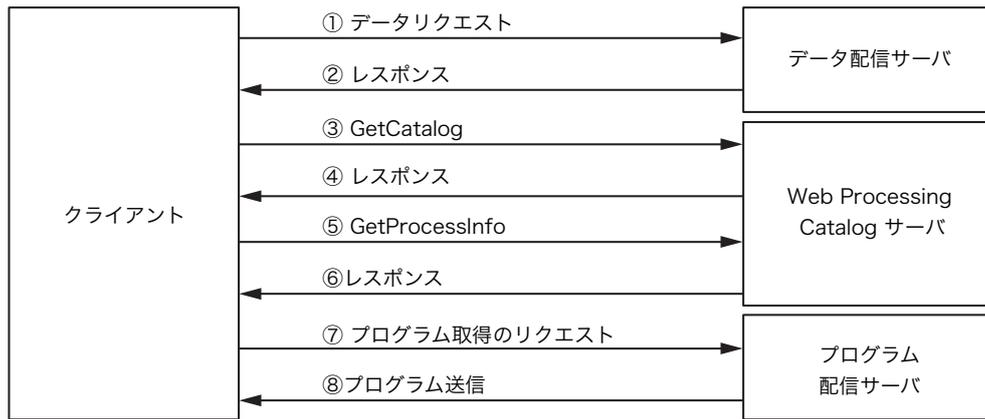


図 8 仮実装したサービスのシーケンス

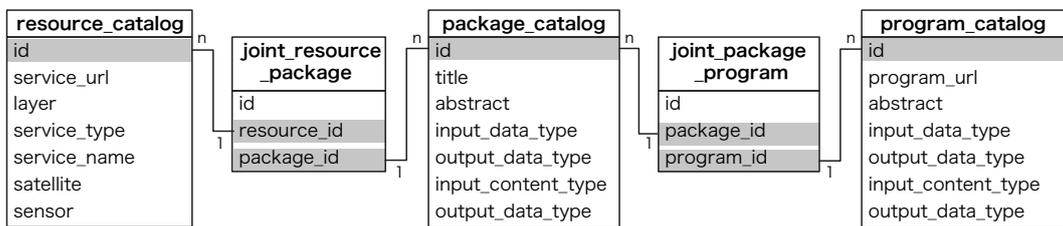


図 9 WPC メタデータを格納するデータベースの構造

本論の事例では、ひとつのデータリソースに対するデータ処理について検討をおこなってきた。実際の一連のデータ処理では複数のデータの組み合わせによって最終的に閲覧されるデータの生成が多く行われる。その場合は無限のデータ処理の組み合わせが考えられ、プロセス共有の重要性も高まるといえる。

5. 試験的な実装

本研究では前章までの議論を踏まえ、試験的にデータとデータ処理を結びつけるシステムの試験的な構築を行った。システムの実装にあたり、既存の利用できる情報システムは手を入れずそのまま利用することを心がけた。実用化に至った場合、既存の情報システムをそのまま活かすことによって、本研究に基づいたシステムの導入が可能となる。

5.1 システムの概要

本論では、データとデータ処理の関係性を示すメタデータの配信サービスを担うプロトコルを、仮に Web Processing Catalog(WPC) と呼ぶこととする。また、WPC を用いて配信するメタデータは WPC メタデータ、WPC メタデータを配信するサーバは WPC サーバと呼ぶこととする。

仮実装したシステムにおけるクライアントとサーバのサービスフローを図 8 に示す。仮構築を行ったシステムは、データ配信サーバ、WPC サーバ、プログラム配信サーバ 3つのサーバから構成されており、これら 3つのサーバが提供するサービスはそれぞれ独立している。データ配信サーバは WMS や WFS の通信が可能な既存の MapServer を利用した。また、プログラム配信サーバは HTTP の通信が可能な既存の Apache を利用した。WPC サーバについては、既存のものがなかったため、今回新たに実装を行った。WPC サーバとクライアントとのデータ通信は、WMS や WPS 同様、地理情報の配信で多く用いられる REST を採用した。

WPC メタデータは、WPC サーバ上のリレーショナル・データベース SQLite で管理しており、そのデータ構造を図 9 に示した。図 9 は、図 8 のサービスフローで必要となるデータの順序にならい、左から右へテーブルを配置した。テーブル「resource.catalog」には、配信されているデータの情報源に関する情報を格納している。テーブル「package.catalog」には、プログラム群で構成されるパッケージに関する情報を格納している。パッケージのプログラムを動作させるにあたり必要なデータのインプットとアウトプットの情報を含んでいる。テーブル「program.catalog」には、プログラムのメタデータを格納している。プログラムの参照情報のみで、実際のプログラムコードは含んでいない。それぞれのテーブルは N:N で結合している。

また、図 10 で示したとおり、WPC メタデータは XML で記述した。これは機械可読を狙ったものであり、例えばクライアント側における WPC とのやりとりの自動化や、データ処理プログラムの自動取得、データ処理プログラムの自動実行など

(a) GetCatalog 図8のシーケンス③④

http://192.168.100.3/cgi-bin/wpc.cgi?REQUEST=GetCatalog&
Server=http://test_server/cgi-bin/mapserv?/moon/kaguya_sp_wfs.map&Layer=kaguya_sp

-<Web_Processing_Catalog version="0.0.0">

<pre> -<Service> <Request> GetCatalog </Request> <Provider> CAIST, The University of Aizu </Provider> </Service> </pre>	WPC情報
<pre> -<DataResource ResoucelID="1"> -<Server> http://test_server/cgi-bin/mapserv?/moon/kaguya_sp_wfs.map </Server> <Layer> kaguya_sp_wfs </Layer> <ServiceType> WFS </ServiceType> </DataResource> </pre>	データ配信 サーバURL レイヤー情報 指定した情報 源に関する 情報
<pre> -<PackageList> -<Package PackageID="1"> <Title> pickup_recommended_data </Title> <abstract> Pickup recommended data. </abstract> </Package> -<Package PackageID="2"> </pre>	パッケージ 概要 パッケージ群 情報

(b) GetProcessInfo 図8のシーケンス⑤⑥

http://192.168.100.3/cgi-bin/wpc.cgi?REQUEST=GetPackageInfo&PackageID=7

-<Web_Processing_Catalog version="0.0.0">

<pre> -<Service> <Request> GetPackageInfo </Request> <Provider> CAIST, The University of Aizu </Provider> </Service> </pre>	WPC情報
<pre> -<Package PackageID="7"> <Title> get_absorption_band </Title> -<Abstract> Detect the absorption bands and describe the parameters characterizing the band shapes. </Abstract> <Input_data_type> array[float] </Input_data_type> <Output_data_type> float </Output_data_type> <Input_content_type> wavelengths </Input_content_type> <Output_content_type> wavelength </Output_content_type> <Input_Resource_id> 1 </Input_Resource_id> </Package> </pre>	パッケージ 概要 入出力に 関する情報 パッケージに 関する情報
<pre> -<ProgramList> -<Program ProgramID="1"> -<Program_URL> http://test_server/resource/pickupRecommendedData.py </Program_URL> </Program> -<Program ProgramID="3"> -<Program_URL> http://test_server/resource/continuum.py </Program_URL> </Program> -<Program ProgramID="7"> </pre>	プログラム 情報 プログラム群 情報

図 10 Web Processing Catalog のメタデータとその構造

が可能になる。どのアプリケーションからでも利用可能とするためには、XML のデータ構造を明確にする必要があり、XML Schema などのスキーマ言語による定義が必要である。今回の実装は仮実装であるため、スキーマ言語は用いなかった。

図 2⑥「可視化画像」のスペクトル特徴量マップは、実装した WPC サーバから取得した WPC メタデータ (図 10) , および WPC メタデータを頼りに取得したデータ処理プログラムを用いて解析し作図した。

5.2 システムのシーケンス

ユーザはまずデータ配信サーバにデータのリクエストを出し ①, データを受け取る ②. これは既存の仕組みの部分であり、データファイルの受け取りに該当する。

そして、データ処理プログラムの取得に移る。まず取得したデータに対してどのようなデータ処理のプログラムが存在するか WPC サーバへ問い合わせる ③. 配信データの特徴は、図 10(a) で示した URL を用いて HTTP リクエストで行う。ここでは「REQUEST」「Server」「Layer」の項目を用いている。「REQUEST」では「GetCatalog」を指定しているが、これはデータ処理の存在を WPC サーバへ問い合わせる命令である。「Server」はデータを配信しているサービスの URL 「http://test_server/cgi-bin/mapserv?/moon/kaguya_sp-wfs.map」を、「Layer」はデータ配信サーバのレイヤ「kaguya_sp」を指定している。

もしデータに対応したデータ処理のプログラムがあれば、ユーザは図 10(a) のような XML で記述されたリストを受け取ることができる ④. 図 10 に示した XML は、Firefox により整形された XML になる。

「server」で指定した値はデータベースの「resource_catalog」テーブルの「service_url」、 「layer」で指定した値はデータベースの「resource_catalog」テーブルの「layer」と照合し、適合するデータレコードがあれば、「package_catalog」のレコードのデータを送り出す。

ユーザは図 10(a) のリストの中から希望するデータ処理群のパッケージを探し出し、WPC サーバへ問い合わせる ⑤. 問い合わせには、XML 中の「PackageID」を用い、図 10(b) で示した URL を用いて HTTP リクエストで行う。ここでは「REQUEST」「PackageID」の項目を用いている。「REQUEST」では「GetPackageInfo」を指定しているが、これはパッケージおよびパッケージに含まれるデータ処理プログラムの情報を得るため WPC サーバへ問い合わせる命令である。「PackageID」は、「7」を指定している。

すると、ユーザは図 10(b) のような XML で記述された情報を受け取ることができる。この XML には、パッケージの概要や、パッケージを適用する際に必要な入出力情報、パッケージのプログラムの配信先の情報が含まれている。

そこでユーザは XML からプログラムを取得できる URL を用いて、プログラム配信サーバへ問い合わせ ⑦, プログラムを受け取る ⑧ことができる。

その後、「GetPackageInfo」の情報に基づき、プログラムの実行を行うことになる。「GetPackageInfo」は XML で記述されており、ユーザ側のアプリケーションで XML をパースした後、アプリケーションプログラムの自動実行も可能である。

6. 課題

本研究ではデータ処理の配信の仕組みについて記述したが、実際のサービス提供に向けて解決しなければならない課題がある。

まず、データ処理のプログラム群の実装だが、例示したとおり試験的に SP のデータ処理について Python で実装したのみである。この仕組みが実用的になるためには、惑星探査で用いられる一般的なデータ処理プログラムの実装が必要である。仮実装では Python を用いたが、どのようなプログラミング言語を用いるべきか検討が必要である。プログラムの利用に関して、データの入出力のインターフェースも同時に検討が必要である。

また、データとデータ処理を関連付ける WPC メタデータについて、実際の惑星科学の状況に照らし合わせて、今回、仮実装したデータ構造でよいのか、あるいは項目は過不足ないかの検討が必要だろう。

さらに、ユーザはデータ処理の配信を受けることができユーザ自らデータ処理ができるとしたが、具体的な方法は明示していない。基本的にはデータ処理のフローに従ってデータ処理を適用すれば良いが、利便性向上のためデータ処理を担う GUI の実装が必要だろう。

謝辞

本研究は JSPS 科研費 26400458 および 25287114 の助成を受けたものである。

参考文献

- 1) 宇宙航空研究開発機構. かぐや (SELENE) データアーカイブ. <https://12db.selene.darts.isas.jaxa.jp/> (2016 年 1 月 21 日アクセス) .

- 2) School of Earth & Space Exploration, Arizona State University. Lunar Reconnaissance Orbiter Camera; WAC Global Morphologic Map. http://wms.lroc.asu.edu/lroc/view_rdr/WAC_GLOBAL (2015年10月1日アクセス) .
- 3) 林洋平, 小川佳子, 平田成, 寺園淳也, 出村裕英, 松永恒雄, 山本聡, 横田康弘, 大竹真紀子, 大嶽久志. かぐや搭載スペクトルプロファイラデータ分析のためのウェブ地理情報システム月光. 宇宙科学情報解析論文誌, Vol. 4, pp. 91-103, 2015.
- 4) 会津大学・先端情報科学研究センター. 月光. http://fructus.u-aizu.ac.jp/gekko_info/ (2015年10月1日アクセス) .
- 5) 宇宙航空研究開発機構. 月周回衛星「かぐや (SELENE)」 - 観測ミッション - TC,MI,SP. http://www.kaguya.jaxa.jp/ja/equipment/tc_j.htm (2014年5月1日アクセス) .
- 6) Matsunaga T., M. Ohtake, J. Haruyama, Y. Ogawa, R. Nakamura, Y. Yokota, T. Morota, C. Honda, M. Torii, M. Abe, T. Nimura, T. Hiroi, T. Arai, K. Saiki, H. Takeda, N. Hirata, S. Kodama, T. Sugihara, H. Demura, N. Asada, J. Terazono, and H. Otake. Discoveries on the lithology of lunar crater central peaks by selene spectral profiler. *Geophysical Research Letters*, Vol. 35, No. 23, 2008. DOI:10.1029/2008GL035868.
- 7) 杉本紘平, 林洋平, 小川佳子, 平田成, 寺園淳也, 出村裕英, 松永恒雄, 山本聡, 横田康弘, 大竹真紀子, 大嶽久志. かぐやスペクトルプロファイラデータを動的に解析するためのウェブアプリケーション開発. 2014年日本地球惑星科学連合大会, 2014年5月.
- 8) Open Geospatial Consortium. Web Map Service — OGC. <http://www.opengeospatial.org/standards/wms> (2016年1月24日アクセス) .
- 9) United States Geological Survey. Webservices - Astrodocs. <http://astrodocs.wr.usgs.gov/index.php/Webservices> (2016年1月21日アクセス) .
- 10) California Institute of Technology Jet Propulsion Laboratory. NewMoon, WMS Server for Moon data. <http://onmoon.jpl.nasa.gov/> (2016年1月21日アクセス) .
- 11) Open Geospatial Consortium. Opengis web map service (wms) implementation specification (version: 1.3.0), 2006. Reference number 06-042. http://portal.opengeospatial.org/files/?artifact_id=14416.
- 12) Open Geospatial Consortium. Web Coverage Service — OGC. <http://www.opengeospatial.org/standards/wcs> (2016年1月24日アクセス) .
- 13) Open Geospatial Consortium. Web Processing Service — OGC. <http://www.opengeospatial.org/standards/wps> (2016年1月24日アクセス) .
- 14) QGIS Development Team. QGIS; A Free Open Source Geographic Information System. <http://qgis.org/> (2016年1月28日アクセス) .
- 15) GRASS Development Team. GRASS GIS; Bringing advanced geospatial technologies to the world. <https://grass.osgeo.org/> (2015年10月1日アクセス) .