

情報資源の再構築を実現するシステム結合基盤の提案

小野永貴[†] 松村敦^{††} 宇陀則彦^{††}

[†]筑波大学図書館情報専門学群 ^{††}筑波大学大学院図書館情報メディア研究科

本研究は複数の情報資源システムが持つ機能を相互に結合し、新たな情報資源システムを再構築できる基盤構築を目的とする。本稿では特に、複数のプラグインを視覚的に結合することで、情報フローを作成できる機能結合プラットフォームについて報告する。

Proposal of System Connecting Infrastructure that aims to Restructure Information Resources.

Haruki ONO[†] Atsushi MATSUMURA^{††} Norihiko UDA^{††}

[†]School of Library and Information Science, University of Tsukuba

^{††}Graduate School of Library, Information and Media Studies, University of Tsukuba

In this research, we aim to develop an infrastructure system that restructures new information resource systems by combining various functions of information resource systems. In this paper, we describe the function connecting platform that generates information flow by combining functions on graphical interface.

1. はじめに

図書館の電子的情報サービスの中核である電子図書館システムは、単一のシステムとして利用されることが想定されているが、電子図書館システムだけでユーザの全ての情報要求を満たすことはできない。そのうえ、Web 上には多様な情報資源があるため、電子図書館は情報資源のうちの一つにすぎなくなった。同じ書籍の情報を得ることを目的としても、電子図書館以外にも横断検索システム、オンライン書店や電子書籍サービスなど、複数のシステムを利用できる。しかしながら、複数のシステムを使い分けるのは煩雑であり、シームレスに利用することができない。このように、ユーザが情報を得る際に利用する情報資源は多様化しているのに対し、利用される情報資源システムは他のシステムとの同時利用を想定していないケースが多く、ズレが生じていると考えられる。

そこで本研究では、このような問題を解決し、ユーザが効率的に複数の情報資源から情報を得ることができるシステムを目指す。このようなシステムを実現するにあたっては、次の2点を検討する必要がある。

1. 現在の情報資源システムは、入力形式や出力形式の統一は行われていない場合が多く、ユーザは情報の統合・変換を行う必要がある。
2. 統合・変換はユーザ自身の手で行わなけれ

ばならず、データ処理やプログラミングなどの高度なスキルが必要となる。

入出力の形式を統一したシステムの例としては、ExLibris 社の Metalib[1]などの横断検索システムがある。しかしこれは、複雑な統合方法や出力形式の変更などには対応できない。一方、情報の加工・統合をするプログラムの例として、フィードアグリゲータ Plagger[2]が存在する。これは Web 上の RSS 等のフィードを加工する工程を自由にカスタマイズできるが、自身で設定言語を記述する必要があり、一般ユーザが日常的に利用することは困難である。また、同じくフィードアグリゲータとして Yahoo!Pipes[3]があるが、これは Web 上のフィードやページ全てを対象にしているため、情報の種別や内容を特定した検索は難しい。このように、既存の情報資源システムをユーザが自由に組み合わせて使うのは容易ではない。

そこで本稿では、各情報資源への情報の入出力を統括し、ユーザが利用しやすい形に変換できるシステム構築基盤を提案する。具体的には、まず個別のシステムに分散した情報資源の各機能をプラグイン化し、機能間で流通されるデータをカプセル化する。そのうえで、自由に機能を組み合わせられるようにする。

2. システム結合基盤の提案

本研究で提案する基盤は、ユーザの目的や用途に合わせて情報資源の機能を自由に組み合わ

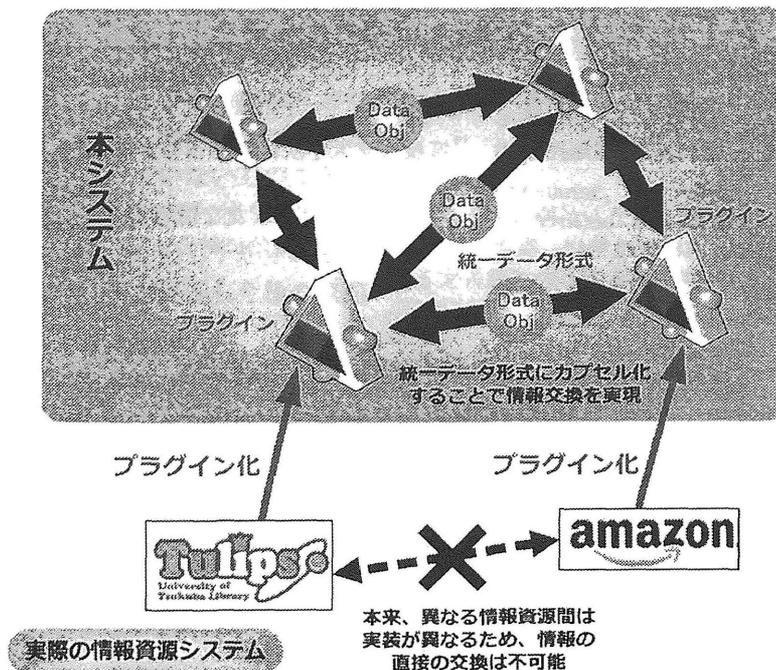


図1 システム結合基盤の概要

せ、目的とする形式で情報を得ることができるようにするものである。ユーザは各情報資源がもつ機能から目的に適したものを選択し、それらを結合するだけで、新たなシステムを再構築できる。そのためには、各情報資源システムの持つ機能の仕様や実装の差を基盤で吸収し、統一された形式と方法で情報資源へアクセスできることを保証する必要がある。

これは一般的に API (Application Programming Interface) と呼ばれる。例えば、REST と呼ばれる手法で HTTP により書誌情報を取得したり、図書館の新着図書 を RSS 形式で購読できるサービスが存在する。しかしこれらは、実際には各システムが持つデータを、別のシステムやソフトウェアから参照できるようにしたものであり、データの交換・流通が行われているに過ぎない。すなわち、データを他のシステムに提供したり、他のシステムから取り込むことにより、それぞれのシステムのユーザの利便性を図ることが目的であって、複数の情報資源を並行して利用するには向かない。よって、現状の API を直接ユーザが利用しても、システム同士の結合は困難である。

本基盤が目指すものは、複数の情報資源の機能を並行して扱う際に、その組み合わせや情報の流れを自在に組み替えることによって、各システムを結合し、ユーザが理想とする新しいシステム自体を構築できる仕組みである。

具体的には、まず個別のシステムに分散した情報資源の各機能を、本基盤上の一機能としてプラグイン化する。各システムへの機能の入出力を定義したプラグインを用意することで、ユーザは情報資源システムの実体を意識することなく、一元的に情報資源へアクセスできる。その結果、ユーザは複数システム間を何度も往復しなくてすむ。

次に、プラグイン間で相互に情報の受け渡しを行う仕組みを提供する。情報資源システムが異なると扱うデータも異なるため、データをカプセル化することで、データを共通化する。

本基盤の実現により、ユーザは様々な情報資源を組み合わせて利用することが容易になる。実現にあたっては、そこには 4 つの段階が存在する。

- 1) 様々な情報資源の入出力が基盤上で再現され、自由に呼び出せるようになる段階
- 2) 各情報資源の機能を自由に組み替え、自身の目的に合ったシステムを構築できるようになる段階
- 3) ユーザが組み合わせを保存し、新たなシステムとして再利用できる段階
- 4) 再利用できるだけでなく、ユーザが使われなくなった情報資源システムの一部として組み込める段階

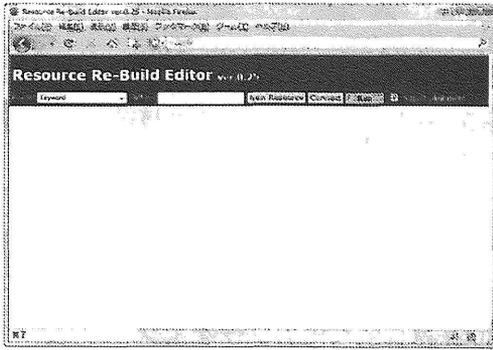


図 2 本システムの画面インタフェース

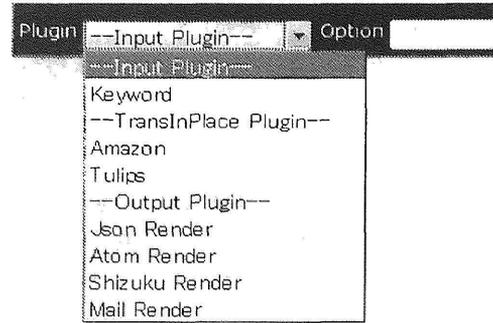


図 3 プラグインの選択ボックス

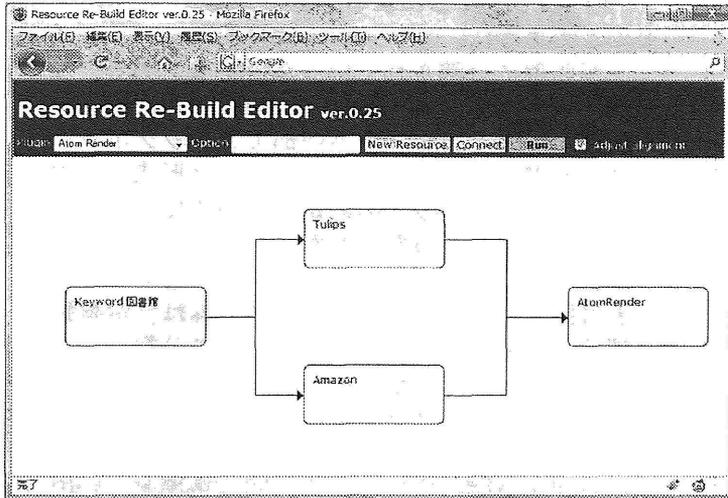


図 4 キャンバス上で描画する

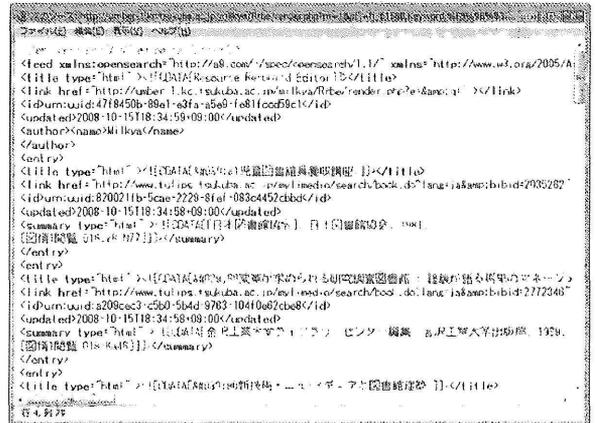


図 5 図 3 のフロー図の実行結果の Atom

本稿では、2段階目までの実装を行い、情報資源の機能の結合を実現する段階までの試作を行った。特にこだわったのは操作の簡便性で、直感的に組み合わせを選択できるプラットフォームを目指した。そこで、グラフィカルなインタフェースを導入し、直感的に操作することを可能にした。

3. 機能統合プラットフォーム

3.1 概要

本プラットフォームは、Web 上の様々な情報資源システムを対象とし、それらのシステムが持つ機能を一元的に利用できる形で抽出し、それらを柔軟に組み合わせられる環境を提供する。

本プラットフォームは、Web 上の各システムの機能をプラグインとして本プラットフォームの中に取り込み、あたかも本プラットフォーム上に各情報資源の機能のサブセットがあるかのように振る舞う。ユーザは本プラットフォーム上で複数の情報資源のプラグインを組み合わせ

ることで、一連の情報加工プロセスを実現できる。これらの操作は、プログラミング言語などをまったく知らなくても行える。

3.2 本プラットフォームの動作

図 2 は本プラットフォームの画面インタフェースである。本システムは画面上部に表示される、各操作を行うためのツールバーと、画面下部に表示される大きな空白の部分から構成される。ユーザは、画面下部の空白部分をキャンバスとみため、フロー図の描画を行うことで、情報資源のフローを構築できる。

本システムにはあらかじめ、Web 上の各種情報資源の機能を抽出したプラグインが取り込まれている。ユーザは画面上部のプラグイン選択ボックス(図 3)から、どのプラグインを使用するかを選択できる。プラグインを選択し、ボックス右側のボタンを押すと、プラグインを模した四角いボックスがキャンバス上に表示される。プラグインは入力プラグイン・中間処理プラグイン・出力プラグインに分類されて表示されて

おり、ユーザは自由にこれらを組み合わせて画面上に配置できる。

各プラグインをキャンパス上に配置し、一連の処理に必要なプラグインのボックスがキャンパス上に表示されたら、最後にボックス同士を矢印の線で結びつける。この操作も画面上部のツールバーで行うことが可能であり、ボックスの配置から線の付加まで全て視覚的に行える。例えば、「キーワードを入力し、筑波大学附属図書館 Tulips とオンライン書店 Amazon.co.jp で書籍検索を行い、その結果を Atom 形式で出力する」という処理のフローで図を描いた場合、キャンパスは図 4 のようになる。

処理に必要なプラグインのボックスが矢印の線で結びつけられ、フロー図が完成したら、画面上部の実行ボタンを押す。すると、このフロー図に従ってシステムが自動的に構築され、実行される。図 4 は、筑波大学附属図書館と Amazon.co.jp で検索を行った結果が結合された例であり、図 5 のように Atom 形式の XML に変換されて出力される。このように、キャンパスに描いた図がそのままシステムになるというのが本システムの特徴である。

本システムは、直列に接続するだけでなく、様々なつなぎ方に対応している。例えば、図 6 のように、1つの入力から2つの情報資源のプラグインに分岐させ、再び合流して出力プラグインにつながる描画を行った場合、複数の情報資源での処理結果が結合されて出力される。つまり複数の情報資源に対して同じ入力を行い、同時に検索を行うということを実現できる。

また、図 7 のように複数の情報資源のプラグインを直列につないだ場合、先に処理された結果をクエリとして受け取り、次のプラグインでの処理が行われる。つまり、情報資源間の絞り込み検索も実現できる。

さらに、本システムは出力形式も柔軟に対応できる。例えば図 8 のように複数の出力プラグインに分岐した場合、同じデータに対して複数の出力が行われ、例えば JSON 形式で結果を取得しながら、メールで同じ結果を受け取る、といった出力方法も簡単に構築できる。



図 6 分岐・合流接続

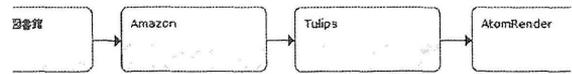


図 7 複数プラグインの直列接続

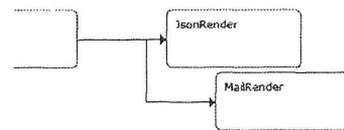


図 8 複合的な出力プラグイン

3.3 プラグインによる処理の分離

このように、本システムは目的別にプラグインが明確に分離されている点が特徴である。従来の情報資源システムは、データの入力形式が決まっており、システム内部でブラックボックスに処理が行われ、最後にきまった形式で出力が行われるという、入力から出力までが一体となったものであった。しかしそのような形態では、ユーザの個別な目的やニーズに対応することは困難である。そこで本システムは、情報資源の処理を行う各ステップを明確に分離し、プラグインとして処理を細分化することによって、ユーザの細かな情報行動のニーズに容易に対応できる。また、入力・中間処理・出力と目的別に処理を分離しているため、同じ処理を行い出力だけを変えるなどの、システムの部分的な変更にも柔軟に対応できる。

4. 本システムの動作原理

4.1 フロー図のシステムへの変換

本システムは、キャンパスに描かれた図に従って、フローの上流から下流へ情報を処理しながら流す。そのために、まずキャンパスに描かれたフロー図をデータ構造として解釈する必要がある。

本システムでは、JavaScript によって描かれたフロー図を文字列化し、サーバサイドのプログラムに送信する。サーバサイドでは、各プラグインをノードとする有向グラフでフロー図を表現する。生成された有向グラフのノードに対してポロジカルソートを行い、プラグインの処理順を決定し、1番目と決定されたプラグインから実行が始まる。処理順に従ってプラグインの処理が行われ、結果データがフローしてゆく。

4.2 プラグインから情報資源へのアクセス

上流からデータが流れてくると、プラグインはデータの変換処理を行う。上流からのデータをもとにクエリを生成し、各プラグインに設定

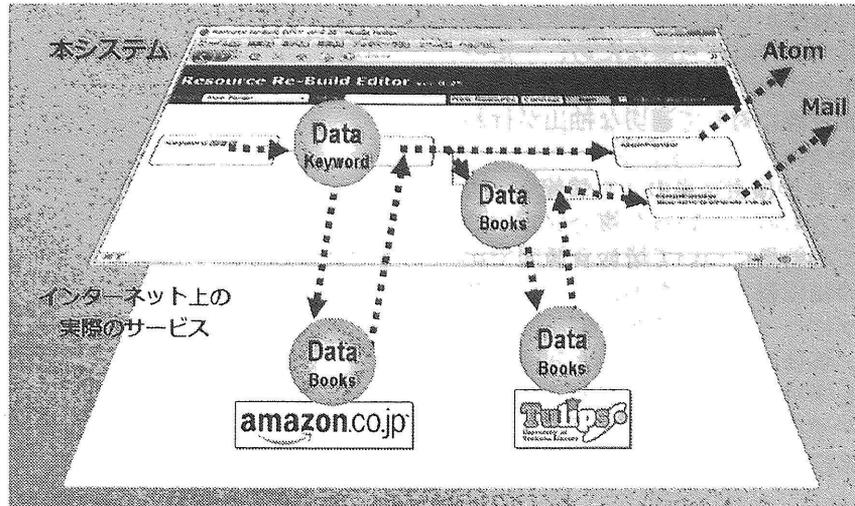


図 9 システム内部でのフロー図

された Web 上の実際の情報資源システムへ送信する。そのレスポンスを分析し、再び本システムのデータの形式にした上、情報資源の変換結果として下流のプラグインへと流す。なお、下流にデータを流す際、フロー図に分岐があればデータを複製して両方へと流し、フロー図に合流があればデータをマージする。

4.3 出力プラグインによる外部出力

全てのプラグインの変換処理が完了し、データが全て出力プラグインへと到達すると、最後に外部出力が行われる。本システムの内部形式から、各出力プラグインに設定された出力形式へとデータ形式が変換され、外部へ出力される。出力形式は、テキスト表示・HTML 表示といった画面表示以外にも、Atom (XML)・JSON といった汎用的なデータ交換形式や、設定されたユーザのメールアドレスに直接結果を送信するなど、様々な方法をとることが可能である。ユーザがどのように情報を使うかというニーズに合わせて、柔軟に結果の出力先を選択できる。

4.4 フローするデータ内容の制御

本システム上でフローしているデータは、グラフ上を流れる情報を抽象化したもので、これを“データオブジェクト”と呼ぶ。データオブジェクトは、データ内容を保持する領域があるが、中身についての制約はない。つまり、文字列や数値、配列やリストなどあらゆる構造を扱うことができる。その代わりに、そのデータがどのような種類のデータなのかを示す、“メディアタイプ”というヘッダ情報を保持しており、キーワード文字列・書籍 (ISBN) の集合・書誌情報などの種別を保持できる。

次に、各プラグインは、データオブジェクト

の入力部と出力部が存在し、どのようなメディアタイプを受け入れられるか、変換可能なパターンを保持している。上流のプラグインからデータオブジェクトの受け渡しがくると、まずデータオブジェクトのメディアタイプを確認し、受け入れ可能かどうか判定する。そして、受け入れ可能と判断された場合、データオブジェクトを受け入れ、情報資源システムへとアクセスし変換処理を行う。変換後のデータを本システムのデータオブジェクトに再び載せ、内容に従ったメディアタイプヘッダを付加した上で、下流のプラグインへと受け渡す。

このように、メディアタイプのチェック、データの変換、データオブジェクトの生成、メディアタイプ付与を順に繰り返すことにより、処理内容や結果のデータ構造が異なる複数の情報資源を統一的に処理することが可能となっている。

5. 考察

本システムの利点は、メディアタイプを付加したデータオブジェクトを統一データ形式としてフローさせることにより、制約を設けることなく、処理できるという点である。既存の情報資源システムの多くは、他のシステムとの連携や機能交換を想定していない場合が多いが、本システムは統一されたデータオブジェクトを用いることで、連携フローを実現している。このような連携が容易に実現できたのは、本基盤が内部で情報資源の実装差の吸収をシステム毎に行っているからである。

しかし、そのような実装ゆえ、あらゆる情報資源に対する処理の妥当性を保証することは困難である。その原因は以下の3つにある。

- 1) 複数の情報資源から得た結果が膨大な場合、全てを統合するのは不可能なため、一定の基準で情報を抽出する必要がある。しかし、全ての情報資源に対して適切な抽出が行われる保証はない。
- 2) 「情報資源 A の検索結果のうち情報資源 B にも存在するもの」といった複合フローの場合、全ての結果について情報資源間で比較を行うのは困難であるため、フローの結果がユーザにとって妥当である保証はない。
- 3) 現状では、各情報資源に対応した個別のプラグインを予め用意する必要があるため、全てのユーザを満足させるプラグインを用意できる保証はない。

これらの問題はすぐに解決するというわけにはいかないが、将来的には例えば、本基盤に対応するプラグインの設計自体を定義し公開するなど、いくつかの解決方法を検討している。

6. おわりに

本稿では、複数の情報資源のもつ機能がユーザの情報行動にあわせ再構築され提供されるシステムのモデルを提案し、そのためには機能結合のための基盤が必要であることを示した。その上、今回試作した基盤システムの機能と動作原理を説明し、データ内容を制約せず複数の情報資源の取り扱う情報を一元的に処理する手法を示した。

今後の課題は、処理対象の情報量と抽出範囲の検討、標準化への検討および再構築の妥当性の評価である。

参考文献

- [1] Ex Libris the bridge to knowledge, Overview.
<http://www.exlibrisgroup.com/category/MetaLib>
Overview, (参照 2008-09-16)
- [2] Plagger - Trac. <http://plagger.org/>, (参照
2008-09-16)
- [3] Pipes: Rewire the web. <http://pipes.yahoo.com/>,
(参照 2008-09-16)