



HOKKAIDO UNIVERSITY

Title	セマンティック・ウェブ上のリソースの探索的検索・統合・可視化分析フレームワーク
Author(s)	朴, 斌
Degree Grantor	北海道大学
Degree Name	博士(情報科学)
Dissertation Number	甲第11750号
Issue Date	2015-03-25
DOI	https://doi.org/10.14943/doctoral.k11750
Doc URL	https://hdl.handle.net/2115/58788
Type	doctoral thesis
File Information	Bin_Piao.pdf



セマンティック・ウェブ上のリソースの探索的検索・統合・可視化
分析フレームワーク

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF INFORMATION SCIENCE
TO THE GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY OF
HOKKAIDO UNIVERSITY

朴 斌

Division of Computer Science

序文

近年，各分野におけるセマンティック・ウェブ技術の普及が進むことによって，RDF(Resource Description Framework)[32]に基づいて記述されたデータやサービスの量が急激に増加することになり，今では，DBpedia[1]をはじめ様々な分野の約1000を超えるRDFデータセットがウェブ上に公開されるようになった[17]．これらのRDFデータセットはRDFクエリ言語であるSPARQL(SPARQL Protocol and RDF Query Language)[35]を用いて問い合わせることができる．これに併い，意味構造を持つこれらの大量データやサービスに対し，ただアクセスしてブラウジングするだけでなく，様々な種類のデータやサービスを統合利用して興味深いデータ集合やそれらの間の関係を検索・探索・分析しながら，新たな知見を発見したいという要求が高まってきている．

セマンティック・ウェブ上に集積された膨大なデータ集合や多種のサービスは，多様なデータを組み合わせることで，データの創発的な価値を生み出す可能性を高める一方，データ集合やサービス自体の理解を困難にしており，その多様な活用を難しくしている．セマンティック・ウェブを有効に扱うためには，セマンティック・ウェブ上のデータやサービスを自在に検索，統合，分析できることが重要である．しかし，それを可能にするには以下の4つの問題点を解決しなければならない．

(1) 検索要求定義や分析を行うためのシナリオの想定・作成が困難

セマンティック・ウェブ上の情報を利用して実世界の課題を解決する際，例えば「転勤先での3世代同居可能な新しい家を探したい」のように明示されている条

件だけでは課題の解決に至らず、通勤時間や、家賃、または家周辺にある教育施設、医療施設等のような様々な潜在的条件をも一緒に考慮し、分析する必要があるような曖昧な課題は、それを解決するのに必要とするあらゆる条件を予め想定した上で、適切な検索要求の定義や分析を行うためのシナリオを作成するのは困難である。

(2) 複雑な SPARQL 検索要求文のインタラクティブな定義が困難

適切な SPARQL 検索要求文を作成するには、検索対象 RDF データのスキーマ、つまり、どのようなクラスが存在し、それらはどのような意味的關係によって関連づけられているのかという構造を事前に把握し、それに基づいて検索要求文を作成する必要があるが、多くの RDF データセットは固定したスキーマを持っていない、巨大化しつつある各分野の RDF データに対して、そのスキーマの全体像を把握することは困難になりつつある。また、スキーマを知っていたとしても、SPARQL に関する深い知識を持たない一般ユーザには、論理演算を含む複雑な SPARQL 検索要求文を正確に素早く作成することは困難である。

(3) 多様なデータやサービスの組み合わせ統合が困難

セマンティック・ウェブにおける多様なデータやサービスの組み合わせ統合には、以下の4つの場合がある。

- a) 同一 RDF データセット中での多様なデータの組み合わせ統合
- b) 異なる RDF データセット間の多様なデータの組み合わせ統合
- c) 異なるサービス間の多様なデータの組み合わせ統合
- d) RDF データセットとサービス間の多様なデータの組み合わせ統合

ここで、a) と b) の場合は統合しようとするデータ集合のスキーマに関する知識が必要となり、特に b) の場合、使用されている語彙も異なることが多いことから、異なるデータ集合の組み合わせ統合は困難である。また、c) と d) の場合は、サービスに関する知識や高いプログラミング能力が必要となる。

(4) 複雑な構造をもつ膨大な RDF データに対する分析が困難

膨大なデータの分析において、多様な視点から探索的に可視化分析を行うことができる多重連携ビュー (Coordinated Multiple View) に関する研究が多くなされてきた。しかし、スキーマの全体像を事前に予想できない RDF データに対しては、従来の手法では適切な多重連携ビューを構築し、分析を行うことは困難である。

本論文では、上記の 4 つの問題点を解決するためにセマンティック・ウェブ上のリソースを探索的に検索、統合、可視化分析できる ESIASW (Exploratory Search, Integration, and Analysis of Semantic Web Resources) フレームワークを提案した。ESIASW フレームワークは、機能を持つ視覚的部品に対する直接操作を用いて、セマンティック・ウェブ上のリソースに対する検索、統合、分析を探索的に繰り返すことを可能にし、膨大なデータ集合やサービス自体に対する理解を深めていきながら、課題を解決するためのシナリオそのものを発見し、新たな知見発見に至る全過程を統括的に支援する枠組みである。

セマンティック・ウェブ上のリソースの検索において、ESIASW フレームワークでは、視覚的部品に対する直接操作によって、多視点から探索的に RDF データのスキーマを抽出・展開・可視化することを可能にしている。これにより、ユーザは、関心のある 1 つまたは複数のクラスをアクセス始点として、それと何らかの意味的關係を持つクラスを探索的に逐次抽出することによって、必要に応じてスキーマを部分的かつ探索的に展開して利用することができる。展開・可視化したスキーマに対し、ユーザは、それを直接操作することによって、複数の異なる視点からインタラクティブに検索条件を同時に指定したり変更することができる。また、視覚的部品に対する直接操作によるクラス間の論理演算を可能にすることで、必要なスキーマをユーザ自身が新しく定義し、それに基づいて論理演算を含む複雑な検索要求を定義することを可能にしている。

セマンティック・ウェブ上のリソースの組み合わせ統合において、ESIASW フレー

ムワークでは、同一作業空間内で、異なる複数の RDF データのスキーマを抽出・展開・可視化したり、必要なセマンティック・ウェブ・サービスを呼び出したりすることを可能にしている。ユーザは、抽出・展開・可視化された RDF データスキーマを意味的関連や論理的関連に基づいて1つのスキーマに連携統合して利用したり、異なるセマンティック・ウェブ・サービスを意味的関連に基づいて直接結合したり、RDF データスキーマとセマンティック・ウェブ・サービスを意味的関連に基づいて直接結合することで、多様なデータやサービスを自在に組み合わせ統合することができる。

セマンティック・ウェブ上のリソースの分析において、ESIASW フレームワークでは、RDF データのスキーマを探索的に抽出・展開しながら、それに基づいて多重連携ビューを動的に構築することを可能にする。ユーザは構築された多重連携ビューの上で複数のデータ集合をインタラクティブに選択することができる。ESIASW フレームワークでは、多重連携ビュー上の各選択操作に独自の色を割り振り、その結果を Brushing&Linking 技術と色の組み合わせ技術を用いて表すことによって、異なるデータ集合を相互比較したり、それらの間の相関関係を探し出したりすることを可能にする。

謝辞 本研究を遂行し学位論文をまとめるに当たり、多くのご支援とご指導を贈りました、指導教官である田中譲教授に深く感謝しております。時に応じて、厳しくご指導いただいたこと、またやさしく励ましてくださったことを通して、私自身の至らなさを実感することができたことは今後の努力の糧になるものであります。

本研究を遂行するに当たり、日々の研究だけでなく、日常の議論を通じて多くの知識や示唆をいただいた猪村元助教授には深く感謝しております。また、私の研究生活につきまして、多くの協力をいただいた知識メディア研究室の皆様には深く感謝しております。

最後に、これまで自分の思う道を進むことに対し、温かく見守りそして辛抱強く支援してくださった妻と両親に対しては深い感謝の意を表して謝辞といたします。

目次

序文	i
1 はじめに	1
1.1 背景	4
1.1.1 セマンティック・ウェブ上のリソースの検索	7
1.1.2 セマンティック・ウェブ上のリソースの連携統合	9
1.1.3 セマンティック・ウェブ上のリソースの分析	10
1.2 本研究の成果と意義	10
1.2.1 セマンティック・ウェブ上のリソースの探索的検索	12
1.2.2 セマンティック・ウェブ上のリソースの連携統合アクセス	13
1.2.3 セマンティック・ウェブ上のリソースの可視化分析	14
1.3 本論文の構造	14
2 基盤技術の概要	17
2.1 概要	17
2.2 セマンティック・ウェブ	17
2.2.1 RDF	18
2.2.2 RDF-Schema	20
2.2.3 OWL	21
2.2.4 SPARQL	22

目次

2.2.5	セマンティック・ウェブ・サービス	23
2.3	知識メディアアーキテクチャ	23
2.3.1	Webble World	25
2.3.2	部品間のスロット結合連携	25
3	ESIASW フレームワーク	29
3.1	RDF データモデルとその拡張	29
3.2	ESIASW フレームワークの概要	33
3.3	ESIASW フレームワークを用いた引っ越しの例	36
3.4	視覚的部品	39
3.4.1	ワークスペース部品	40
3.4.2	クラス部品	41
3.4.3	サービス部品	42
3.4.4	可視化分析部品	44
3.5	まとめ	45
4	セマンティック・ウェブ上のリソースの探索的検索	47
4.1	概要	47
4.2	多視点スキーマの探索的抽出・展開・可視化	49
4.3	スキーマに基づいた検索要求の定義	57
4.4	検索要求の評価	58
4.4.1	スキーマの探索的抽出に関する検索要求	59
4.4.2	スキーマに基づいた探索的検索	60
4.5	SPARQL への自動書き換え	62
5	セマンティック・ウェブ上のリソースの連携統合アクセス	69
5.1	概要	69
5.2	異なるスキーマ間の連携統合	70

目次

5.2.1	論理的関係による異なるスキーマ間の連携統合	70
5.2.2	意味関係に基づいて異なるスキーマ間の連携統合	72
5.3	セマンティック・ウェブ・サービス間の連携統合	74
5.4	スキーマとセマンティック・ウェブ・サービスの連携統合	79
6	セマンティック・ウェブ上のリソースの可視化分析	81
6.1	概要	81
6.2	多重連携ビューの動的構築	82
6.3	多重連携ビュー上のマルチ選択	86
7	関連研究との比較	91
7.1	セマンティック・ウェブ上のリソースの検索	92
7.2	セマンティック・ウェブ上のリソースの統合	93
7.3	セマンティック・ウェブ上のリソースの分析	93
7.4	ユーザビリティの評価	94
8	結論	99
	関連発表論文	103
	参考文献	i

図一覧

1.1	The Linking Open Data cloud diagram[4]	5
1.2	ESIASW フレームワーク基本的考え方	11
2.1	RDF グラフの例	19
2.2	RDF/XML の例	20
2.3	Webble world の画面コピー	24
2.4	Webble 間標準メッセージ	26
3.1	クラス間論理演算	31
3.2	クラスのプロパティ集合	31
3.3	RDF データとそのスキーマ	33
3.4	ESIASW フレームワークの概要	34
3.5	ESIASW フレームワークを用いた引っ越しの例 1	37
3.6	ESIASW フレームワークを用いた引っ越しの例 2	38
3.7	ワークスペース部品	40
3.8	クラス部品およびそのプロパティとオーバーラップクラスリスト	41
3.9	サービス部品の例	43
3.10	可視化分析部品の例	44
3.11	3 種類の視覚的オブジェクトとそれらの color space	44
4.1	ESIASW フレームワークを用いた探索的検索の例	48

図一覧

4.2	始点クラス定義の例	50
4.3	あるクラスから 1 つの関係プロパティを選択し, その値を用いて新しい抽象クラスを定義	51
4.4	クラス間論理演算による新しい抽象クラスの定義	52
4.5	オーバーラップクラスとの論理演算を用いたクラスの変更	53
4.6	スキーマ・トリプル・パスの例	56
4.7	スキーマに基づいた探索的検索の例	61
4.8	視覚的検索要求から SPARQL への書き換えの例	66
5.1	論理的関係に基づいた異なるスキーマの統合	71
5.2	異なるスキーマの組み合わせによる統合スキーマの定義	73
5.3	iServe におけるセマンティック・ウェブ・サービスの記述例	75
5.4	セマンティック・ウェブ・サービスの検索	76
5.5	セマンティック・ウェブ・サービス間の連携統合	77
5.6	スキーマとセマンティック・ウェブ・サービスの統合	78
6.1	可視化表現の定義	83
6.2	可視化分析部品を用いた多重連携ビューの構築の例	85
6.3	色の組み合わせ規則	87
6.4	マルチ選択の例	88
7.1	既存研究と本研究の比較	91

表一覧

2.1	RDF トリプルの集合	19
4.1	基本述語の定義	63
4.2	クラスに対する全ての書き換え規則	63
4.3	スキーマ・トリプル・パターンに対する全ての書き換え規則	64
4.4	基本述語から SPARQL 制約条件への書き換え規則	65
7.1	6つの検索タスク	95
7.2	検索要求の定量化	96
7.3	タスクを完了するための最少操作回数	96

第 1 章

はじめに

Introduction

近年、各分野におけるセマンティック・ウェブ技術の普及が進むことによって、RDF(Resource Description Framework)[32]に基づいて記述されたデータやサービスの量が急激に増加することになり、今では、DBpedia[1]をはじめ様々な分野の約 1000 を超える RDF データセットがウェブ上に公開されるようになった [17]. これらの RDF データセットは RDF クエリ言語である SPARQL(SPARQL Protocol and RDF Query Language)[35]を用いて問い合わせることができる. これに併い、意味構造を持つこれらの大量データやサービスに対し、ただアクセスしてブラウジングするだけでなく、様々な種類のデータやサービスを統合利用して興味深いデータ集合やそれらの間の関係を検索・探索・分析しながら、新たな知見を発見したいという要求が高まってきている.

セマンティック・ウェブ上に集積された膨大なデータ集合や多種のサービスは、多様なデータを組み合わせることで、データの創発的な価値を生み出す可能性を高める一方、データ集合やサービス自体の理解を困難にしており、その多様な活用を難しくしている. セマンティック・ウェブを有効に扱うためには、セマンティック・ウェブ上のデータやサービスを自在に検索、統合、分析できることが重要である. しかし、それを可能にするには以下の 4 つの問題点を解決しなければならない.

Chapter 1 INTRODUCTION

(1) 検索要求定義や分析を行うためのシナリオの想定・作成が困難

セマンティック・ウェブ上の情報を利用して実世界の課題を解決する際、例えば「転勤先での3世代同居可能な新しい家を探したい」のように明示されている条件だけでは課題の解決に至らず、通勤時間や、家賃、または家周辺にある教育施設、医療施設等のような様々な潜在的条件をも一緒に考慮し、分析する必要があるような曖昧な課題は、それを解決するのに必要とするあらゆる条件を予め想定した上で、適切な検索要求の定義や分析を行うためのシナリオを作成するのは困難である。

(2) 複雑な SPARQL 検索要求文のインタラクティブな定義が困難

適切な SPARQL 検索要求文を作成するには、検索対象 RDF データのスキーマ、つまり、どのようなクラスが存在し、それらはどのような意味的關係によって関連づけられているのかという構造を事前に把握し、それに基づいて検索要求文を作成する必要があるが、多くの RDF データセットは固定したスキーマを持っていない、巨大化しつつある各分野の RDF データに対して、そのスキーマの全体像を把握することは困難になりつつある。また、スキーマを知っていたとしても、SPARQL に関する深い知識を持たない一般ユーザには、論理演算を含む複雑な SPARQL 検索要求文を正確に素早く作成することは困難である。

(3) 多様なデータやサービスの組み合わせ統合が困難

セマンティック・ウェブにおける多様なデータやサービスの組み合わせ統合には、以下の4つの場合がある。

- a) 同一 RDF データセット中での多様なデータの組み合わせ統合
- b) 異なる RDF データセット間の多様なデータの組み合わせ統合
- c) 異なるサービス間の多様なデータの組み合わせ統合
- d) RDF データセットとサービス間の多様なデータの組み合わせ統合

ここで、a) と b) の場合は統合しようとするデータ集合のスキーマに関する知識

が必要となり、特に b) の場合、使用されている語彙も異なることが多いことから、異なるデータ集合の組み合わせ統合は困難である。また、c) と d) の場合は、サービスに関する知識や高いプログラミング能力が必要となる。

(4) 複雑な構造をもつ膨大な RDF データに対する分析が困難

膨大なデータの分析において、多様な視点から探索的に可視化分析を行うことができる多重連携ビュー (Coordinated Multiple View) に関する研究が多くなされてきた。しかし、スキーマの全体像を事前に予想できない RDF データに対しては、従来の手法では適切な多重連携ビューを構築し、分析を行うことは困難である。

本論文では、上記の 4 つの問題点を解決するためにセマンティック・ウェブ上のリソースを探索的に検索、統合、可視化分析できる ESIASW (Exploratory Search, Integration, and Visual Analysis of Semantic Web Resources) フレームワークを提案した。ESIASW フレームワークは、機能を持つ視覚的部品に対する直接操作を用いて、セマンティック・ウェブ上のリソースに対する検索、統合、分析を探索的に繰り返すことを可能にし、膨大なデータ集合やサービス自体に対する理解を深めていながら、課題を解決するためのシナリオそのものを発見し、新たな知見発見に至る全過程を統括的に支援する枠組みである。

セマンティック・ウェブ上のリソースの検索において、ESIASW フレームワークでは、視覚的部品に対する直接操作によって、多視点から探索的に RDF データのスキーマを抽出・展開・可視化することを可能にしている。これにより、ユーザは、関心のある 1 つまたは複数のクラスをアクセス始点として、それと何らかの意味的關係を持つクラスを探索的に逐次抽出することによって、必要に応じてスキーマを部分的かつ探索的に展開して利用することができる。展開・可視化したスキーマに対し、ユーザは、それを直接操作することによって、複数の異なる視点からインタラクティブに検索条件を同時に指定したり変更することができる。また、視覚的部品に対する直接操作によるクラス間の論理演算を可能にすることで、必要なスキーマをユーザ自身が新

Chapter 1 INTRODUCTION

しく定義し、それに基づいて論理演算を含む複雑な検索要求を定義することを可能にしている。

セマンティック・ウェブ上のリソースの組み合わせ統合において、ESIASW フレームワークでは、同一作業空間内で、異なる複数の RDF データのスキーマを抽出・展開・可視化したり、必要なセマンティック・ウェブ・サービスを呼び出したりすることを可能にしている。ユーザは、抽出・展開・可視化された RDF データスキーマを意味的関連や論理的関連に基づいて 1 つのスキーマに連携統合して利用したり、異なるセマンティック・ウェブ・サービスを意味的関連に基づいて直接結合したり、RDF データスキーマとセマンティック・ウェブ・サービスを意味的関連に基づいて直接結合することで、多様なデータやサービスを自在に組み合わせ統合することができる。

セマンティック・ウェブ上のリソースの分析において、ESIASW フレームワークでは、RDF データのスキーマを探索的に抽出・展開しながら、それに基づいて多重連携ビューを動的に構築することを可能にする。ユーザは構築された多重連携ビューの上で複数のデータ集合をインタラクティブに選択することができる。ESIASW フレームワークでは、各選択操作に独自の色を割り振り、その結果を Brushing&Linking 技術と色の組み合わせ技術を用いて表すことによって、異なるデータ集合を相互比較したり、それらの間の相関関係を探し出したりすることを可能にする。

1.1 背景

現在、ウェブ上には文書、音楽、映像だけでなく、モバイル・デバイスやセンサ、ソーシャル・ネットワーキングなどの普及で大量のデータが発生している。このような大量のデータを有効に利用するために、ウェブは「人間が読むための文書のウェブ」から、セマンティック・ウェブという「機械が理解できるデータのウェブ」へと向かっている。セマンティック・ウェブでは、ウェブ上のデータに機械処理可能なメタデータを付け加え、それらに意味構造を付与することによって、機械による意味関係に基づいた検索や推論のような知的な処理を可能にする。セマンティック・ウェブの標準化を

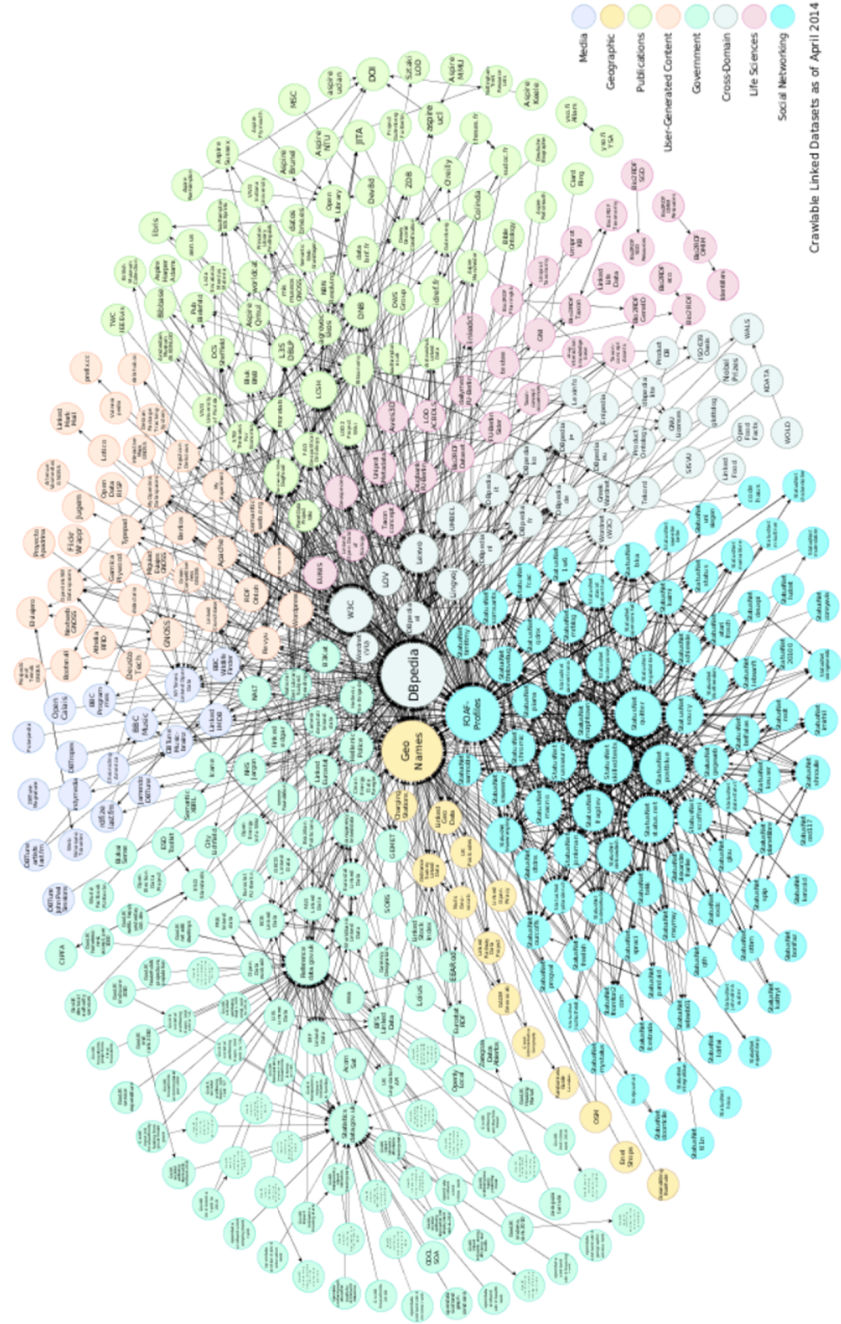


图 1.1: The Linking Open Data cloud diagram[4]

Chapter 1 INTRODUCTION

行っている W3C(World Wide Web Consortium) では、メタデータの記述言語として RDF(Resource Description Framework)[32] を勧告しており、RDF データの問い合わせ言語として SPARQL(SPARQL Protocol and RDF Query Language)[35] を勧告している。それ以外にも、RDF の語彙を記述するための RDF Schema[?] や、ウェブ上のオントロジーを記述するための OWL(Web Ontology Language)[34] などが W3C によってセマンティック・ウェブの標準技術として勧告されている。

近年、各分野におけるセマンティック・ウェブ技術の普及が進むことによって、ウェブ上でデータを公開・共有する新しい形式として、セマンティック・ウェブの要素技術を活用した LOD(Linked Open Data)[4] が開発された。今では、図 1.1 に示すように DBpedia[1] をはじめ、学術や政府関係等様々な分野の約 1000 を超える RDF データセットが LOD として公開されるようになった [17]。

セマンティック・ウェブでは、データだけでなく、ウェブサービスの記述にもメタデータを付加して、意味関係に基づいたウェブサービスの検索、連携、実行を可能にしようとしている。このようにウェブサービスをセマンティック・ウェブ・サービスと呼ぶ。セマンティック・ウェブ・サービスを支える基本的標準技術として、OWL-S(Web Ontology Language for Services)[31], SAWSDL(Semantic Annotations for WSDL)[33] などが挙げられており、これらの技術を用いてセマンティック・ウェブ・サービスの自動的発見や連携を試みた様々な研究 [27, 8, 14] が行われていた。しかし、セマンティック・ウェブ・サービスをより有効に扱うためには、セマンティック・ウェブ・サービスの記述そのものをも LOD として公開し、SPARQL を用いた自在に問い合わせできるようにすることが重要である。それを初めて可能にしたのが iServe[20] である。iServe では、アクセス可能な URI を用いて各ウェブサービスの識別を行い、RDF を用いて各サービスの記述を行っている。また、iServe では、SPARQL endpoint を通じて必要とするセマンティック・ウェブ・サービスを検索したり、既存のウェブ・サービスをセマンティック・ウェブ・サービスとして新しく登録したりすることを可能にしている。今まで iServe に登録されているウェブ・サービスの数は 2000 を超え

ている。

これに伴い、意味構造を持つこれらの大量データやサービスに対し、ただアクセスしてブラウジングするだけでなく、様々な種類のデータやサービスを統合利用して興味深いデータ集合やそれらの間の関係を検索・探索・分析しながら、新たな知見を発見したいという要求が高まってきている。しかし、巨大化しつつあるデータの量やサービスの種類は、データ集合やサービス自体の理解を困難にしており、それらを有効に扱うためには、セマンティック・ウェブ上のデータやサービスを自在に検索、統合、分析できることが重要である。また、セマンティック・ウェブ上のデータやサービスに対する検索、統合、分析を行う際、例えば、「日曜日にも診察可能な近所の病院を探したい」のように必要とする検索条件が課題の中で十分に記述されている課題とは違い、「転勤先での3世代同居可能な新しい家を探したい」のように明示されている条件だけでは課題の解決に至らず、通勤時間や、家賃、または家周辺にある教育施設、医療施設等のような様々な潜在的条件をも一緒に考慮し、分析する必要がある曖昧な課題に対しては、それを解決するのに必要とするあらゆる条件を予め想定した上で、適切な検索要求の定義や分析を行うことは困難であり、様々な検索条件に対する検索、統合、分析を自在に組み合わせ試行錯誤的に繰り返しながら、課題解決のためのシナリオそのものを発見していくことが重要である。

1.1.1 セマンティック・ウェブ上のリソースの検索

セマンティック・ウェブ上のデータやサービスを検索するためには、SPARQL 検索要求文を用いて各 RDF データセットの SPARQL endpoint へとアクセスする必要がある。しかし、SPARQL 検索要求文は冗長になることが多く、検索要求を変えながら連続的に検索を行う場合、手入力による検索要求文の連続的な作成・変更はユーザの多大な労力を必要とする。例えば、「日本の各野球チームの中で地元出身ではないピッチャーとその人の成績を調べたい」という SPARQL 検索要求文は以下のように記述

Chapter 1 INTRODUCTION

することができる.

```
PREFIX    dbpedia :< http : //dbpedia.org/resource/ >
PREFIX    dbpedia - owl :< http : //dbpedia.org/resource/Ontology/ >
PREFIX    p :< http : //dbpedia.org/resource/property/ >
PREFIX    foaf :< http : //xmlns.com/foaf/0.1/ >
SELECT    distinct ?baseballTeamName ?playerName ?value
WHERE     {?player rdf : type dbpedia - owl : JapaneseBaseballPlayer.
           ?player p : position dbpedia : Pitcher.
           ?player foaf : name ?playerName.
           ?player p : birthPlace ?city.
           ?player p : score ?score.
           MINUS{?player p : teams ?baseballTeam.
                 ?baseballTeam p : ballpark ?ballpark.
                 ?baseballTeam foaf : name ?baseballTeamName.
                 ?ballpark p : locationCity ?city.}
           }
```

(1.1)

構造化された RDF データを検索するための SPARQL 検索要求文は、検索要求文 (1.1) の *WHERE* 節に示すように、それ自身も構造化されている。このような構造化された SPARQL を用いて適切な検索要求文を作成するためには、検索対象である RDF データのスキーマを知っている必要がある。しかし、公開されている RDF データセットの多くは固定したスキーマの持たず、巨大化しつつある各分野の RDF データに対しては、そのスキーマの全体像を把握することは困難になりつつあり、検索要求定義に必要なスキーマは、ユーザが関心を持つクラスをアクセスの始点として、それと何らかの意味的関係を持つクラスを探索的に逐次抽出することによって、必要に応じて

部分的かつ探索的に展開して利用することが必要になりつつある。

また、各分野におけるセマンティック・ウェブ技術の普及が進むことによって、ユーザ層の多様化が顕著になりつつあり、スキーマに基づいた情報検索においては、元々定義されている既存のクラスとそれらの間の意味的關係を用いるだけでは、様々なユーザの多様な検索要求に対応することができなくなっている。OWL[34]では、より複雑な知識を記述するために、クラス間の論理演算による新たなクラスの定義を可能にしているが、OWLに関する知識を持たないユーザには、それを用いた記述により適切なクラスを定義したり、定義したクラスを即刻にスキーマに反映し、これに基づく検索要求を記述することは困難である。

1.1.2 セマンティック・ウェブ上のリソースの連携統合

セマンティック・ウェブ上に集積された膨大なデータ集合や多種のサービスは、多様なデータを組み合わせることで、データの創発的な価値を生み出すことを可能にする。セマンティック・ウェブにおける多様なデータやサービスの組み合わせ統合は、以下の4つの種類に分けられる。

- a) 同一 RDF データセット中での多様なデータの組み合わせ統合
- b) 異なる RDF データセット間の多様なデータの組み合わせ統合
- c) 異なるサービス間の多様なデータの組み合わせ統合
- d) RDF データセットとサービス間の多様なデータの組み合わせ統合

ここで、a) と b) の場合は統合しようとするデータ集合のスキーマに関する知識が必要となり、特に b) の場合、使用されている語彙も異なることが多く、異なるデータ集合の組み合わせ統合は困難である。セマンティック・ウェブでは、公開されている各 RDF データセット間のデータを相互関連づけ、多様なデータの組み合わせを可能にしようとしているが、データを自在に組み合わせするのに必要とする相互の関連関係を

全て定義することは不可能である．また，c) と d) の場合は，ウェブサービスに関する知識や高いプログラミング能力が必要となる．

1.1.3 セマンティック・ウェブ上のリソースの分析

セマンティック・ウェブ上のデータやサービスの分析を行う従来の研究の多くは [30, 18]，特定の RDF データセットの一部に対し，その中のリソース間の潜在的な意味的關係を探し出すデータ構造自体に対する分析を行っていた．しかし，近年急速に増加する様々な分野の RDF データは，それらを有効に扱うことによって実世界の課題解決に有用な新たな知見を発見したいという要求が高まってきている．膨大なデータの分析において，従来のデータベース分野では，多様な視点から探索的に可視化分析を行う研究が多くなされていた [19, 5, 11]．これらは，多様な視点から探索的に可視化できる多重連携ビュー技術を用いて，膨大なデータの色々な側面から可視化を行い，それを直接操作してデータのフィルタリングをしながら，探索的な可視化分析を行うことを可能にしている．しかし，これらの従来の手法は事前にスキーマを把握し，それに基づいてデータの分析を行う必要があり，スキーマの全体像を予想できない RDF データには適用することが困難である．

1.2 本研究の成果と意義

本論文の基本的な考え方は，図 1.2 に示すようにインタラクティブな可視化環境内で，プログラミング技術や SPARQL に関する深い知識を持たないユーザでも，セマンティック・ウェブ上のデータやサービスに対する探索的検索，連携統合，可視化分析を試行錯誤的に繰り返すことを可能にすることによって，データ集合やサービスの構造に対する認識を逐次深めていきながら，実世界の課題解決に至るまでの全過程を統括的に支援することである．これを，実現するためには，以下の機能が必要である．

- (1) 多視点スキーマの探索的抽出・展開・可視化
- (2) クラス間論理演算による新しいクラス定義

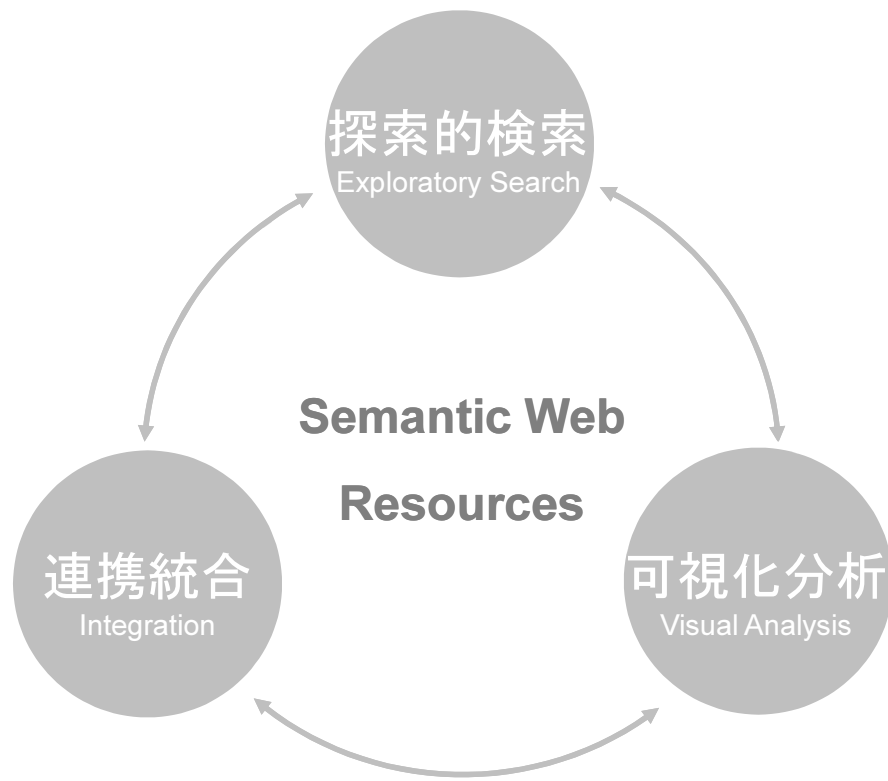


図 1.2: ESIASW フレームワーク基本的考え方

- (3) スキーマに基づいた視覚的検索要求定義
- (4) スキーマやセマンティック・ウェブ・サービスの連携統合
- (5) 多視点からの探索的可視化分析
- (6) 視覚的部品に対する直接操作を用いた (1)~(5) の実現

セマンティック・ウェブ上のリソースの探索的検索において、(1)の「多視点スキーマの探索的抽出・展開・可視化」により、ユーザは探索的に抽出・展開・可視化したスキーマに基づいて検索要求を定義したり、異なるスキーマを持つデータを統合したりすることが可能になる。スキーマを探索的に抽出・展開する際、既存クラスだけでなく、必要なクラスをユーザ自身が簡便に新しく定義し、定義した新たなクラスを用いて新し

Chapter 1 INTRODUCTION

いスキーマを即時定義できるようにするためには、視覚的部品に対する直接操作による (2) の「クラス間論理演算による新しいクラス定義」機能が必要となる。SPARQL に関する知識を持たないユーザでも、必要な検索要求を自身で簡単に作成できるようにするためには、(3) の「視覚的検索要求定義」機能が必要となる。セマンティック・ウェブ上のリソースの連携統合において、異なるスキーマを持つ複数のデータ集合を統合したり、データ集合とセマンティック・ウェブを統合したりするためには、(4) の「異なるスキーマやセマンティック・ウェブ・サービスの組み合わせ統合」機能が必要となる。セマンティック・ウェブ上のリソースの分析において、試行錯誤しながら興味深いデータ集合やそれらの間の関係を探索・分析するためには、(5) の「多視点からの探索的可視化分析」の機能が必要となる。プログラミング技術を持たないユーザでもインタラクティブに検索要求を変えながら、セマンティック・ウェブ情報を探索的に検索、連携統合、可視化分析できるようにするためには (6) の機能が必要となる。

本論文では、上記の (1)~(6) の機能を全て提供する ESIASW フレームワークを提案する [21, 22, 23]. ESIASW フレームワークでは、RDF データスキーマの各クラスを表すクラス部品、セマンティック・ウェブ・サービスをプロキシを作成可能なサービス部品、多様な視点から探索的に可視化分析できる多重連携ビューを動的作成可能な可視化分析部品の 3 つの視覚的部品を提供し、それらを直接操作することによって、セマンティック・ウェブ上のデータやサービスに対する探索的検索、統合、可視化分析の 3 つの操作をインタラクティブに繰り返しながら、新たな知見を発見する過程を統括的に支援する。本研究では、知識メディアアーキテクチャ[29] の Web バージョンである Webble World[16] 上で、ESIASW フレームワークに基くセマンティック・ウェブ・リソースの探索的検索・統合・可視化システムを実装した。

1.2.1 セマンティック・ウェブ上のリソースの探索的検索

セマンティック・ウェブ上のリソースの検索において、ESIASW フレームワークでは、クラス部品を直接操作することによって、そのクラスと何らかの意味的關係をも

つ他クラスを逐次抽出したり，検索要求の変数や制約条件を直接指定したり変更することを可能にした．これにより，スキーマに関する知識を持たないユーザでも，関心を持つ1つまたは複数のクラスをアクセスの始点として，その意味的關係を辿りながら，必要に応じてスキーマを部分的かつ探索的に展開して行きながら，複数の異なる視点から検索条件を同時に指定したり，変更することが可能にした．また，展開・可視化したスキーマ中の異なるクラスを直接組み合わせ論理演算することによって，論理演算を含む複雑な検索要求を定義するためのスキーマをユーザ自身が自在に定義することを可能にした．ESIASW フレームワークでは，これらの操作は全て自動的に SPARQL へと書き換えられ実行される．これにより，SPARQL やプログラミングに関する知識をもたないユーザでも，スキーマを探索的抽出・展開したり，それに基づいて検索要求をインタラクティブに定義し変更することができ，セマンティック・ウェブから必要な情報を探索的に探し出すことができる．

1.2.2 セマンティック・ウェブ上のリソースの連携統合アクセス

セマンティック・ウェブ上のリソースの統合アクセスにおいて，ESIASW フレームワークでは，同一作業空間内で，異なる複数の RDF データのスキーマを展開・可視化したり，サービス部品を用いて必要なセマンティック・ウェブ・サービスのプロキシを作成することを可能にする．サービス部品は，RDF に基づいたセマンティック・ウェブ・サービスの記述から，セマンティック・ウェブ・サービスの入力，出力などに関する情報を自動的に抽出し，それらを用いてセマンティック・ウェブ・サービスのプロキシを作成することができる．ユーザは展開・可視化された RDF データスキーマを意味的関連や論理的関連に基づいて1つのスキーマに統合して利用したり，異なるサービス部品を意味的関連に基づいて直接結合したり，RDF データスキーマとサービス部品を意味的関連に基づいて直接結合したりすることで，以下の4種類の組み合わせ統合を可能にする．

- a) 同一 RDF データセット中での多様なデータの組み合わせ統合

- b) 異なる RDF データセット間の多様なデータの組み合わせ統合
- c) 異なるサービス間の多様なデータの組み合わせ統合
- d) RDF データセットとサービス間の多様なデータの組み合わせ統合

これにより、RDF データのスキーマや、セマンティック・ウェブ・サービスに関する深い知識を持たないユーザでも、セマンティック・ウェブ上の多様なデータやサービスを自在に組み合わせ、統合利用することが可能になる。

1.2.3 セマンティック・ウェブ上のリソースの可視化分析

ESIASW フレームワークにおいて、可視化分析部品は、多種のビューと、マルチ選択、Brushing&Linking、カラーコーディネーションなどの3つコーディネーション規則を提供する。ユーザは可視化分析部品を展開された RDF データスキーマまたは、サービス部品に直接結合して、多様な視点から探索的に可視化分析ができる多重連携ビューを動的に構築しながら、その上で複数のデータ集合をインタラクティブに選択することを可能にする。ESIASW フレームワークでは、多重連携ビュー上の各選択操作に独自の色を割り振り、その結果を Brushing&Linking 技術とカラーコーディネーション技術を用いて表すことによって、異なるデータ集合を相互比較したり、それらの間の相関関係を探し出したりすることを可能にする。

1.3 本論文の構造

本論文では、2章でセマンティック・ウェブやその基盤技術について紹介し、3章で ESIAWS フレームワークする。4章では ESIAWS フレームワークを用いたセマンティック・ウェブ上のデータやサービス探索的検索について説明する。5章では ESIAWS フレームワークを用いたセマンティック・ウェブ上のデータやサービス組み合わせ統合について説明する。6章では ESIAWS フレームワークを用いたセマンティック・ウェブ上のデータやサービス探索的可視化分析について説明する。7章では提案フレーム

Chapter 1 INTRODUCTION

ワークのユーザビリティに関して既存の関連研究と比較しながら評価する．最後に，結論を 8 章で述べる．

第 2 章

基盤技術の概要

Enabling Technologies

2.1 概要

本章では、セマンティック・ウェブとそれを支える基盤技術について述べた後、インタラクティブな可視化環境を構築するのに用いる知識メディアアーキテクチャとそれに基づいて開発されたシステム Webble World[16] について述べる。

2.2 セマンティック・ウェブ

従来のウェブでは、ハイパーリンクを用いてウェブ上複数の文書を相互に関連づけ、結び付けている。しかし、ハイパーリンクは2つの文書を結び付けるだけで、それ自身は意味を示さない単純なリンクである。リンクの意味は、始点となる文書に書かれた情報を、人間がコンテキストとともに解析しなければならない。ウェブの創始者であるバーナーズ＝リーは、1994 に開催された第 1 回 WWW 国際会議において、人間にとって便利なハイパーリンクは機械が理解可能な情報がほとんどないことを指摘し、ウェブに「セマンティックス」を与える必要があると述べ、セマンティック・ウェブという概念を始めて提示した [2]。

セマンティック・ウェブは、ウェブ上の情報を機械的に処理するための枠組みである。

セマンティック・ウェブでは、HTML 文書を人工知能的に解析して内容を機械に理解させるのではなく、機械的処理可能なメタデータを HTML 文書に付与することで、文書の説明するためのデータを機械処理可能な形で別に用意する。セマンティック・ウェブの標準化を行っている W3C(World Wide Web Consortium) では、メタデータの記述言語として RDF(Resource Description Framework)[32] を勧告しており、RDF データの問い合わせ言語として SPARQL(SPARQL Protocol and RDF Query Language)[35] を勧告している。それ以外にも、RDF の語彙を記述するための RDF Schema[36] や、ウェブ上のオントロジーを記述するための OWL(Web Ontology Language)[34] などが W3C によって標準技術として勧告されている。

2.2.1 RDF

RDF(Resource Description Framework) はウェブ上のリソースを記述するための枠組みであり、セマンティック・ウェブを実現するための基盤技術として W3C により勧告されている。RDF では、リソースの概念をウェブ上で識別できる事物と一般化することによって、ウェブ上の文書だけでなく、人物や書籍、イベントなどの実世界の事物まで表現することが可能である。RDF における全てのリソースは URI(Uniform Resource Identifier) 参照によって識別される。RDF は機械処理可能で、論理的な関係を表現できるデータモデルとして、RDF トリプルという主語 (subject)、述語 (predicate)、目的語 (object) の三つ組みの集合を用いる。RDF トリプルの集合を RDF グラフと呼ぶ。主語は、URI で示されたリソースか、URI を持たず直接参照できない空白ノードのどちらかであり、目的語は文字列かリソースか空白ノードのいずれかである。述語は URI で示され、主語と目的語との意味関係を示す。表 2.1 に私 (eg:me) の国籍や所属について記述した RDF データを示す。これは、図 2.1 に示すように有向グラフとして表すことができる。RDF データは様々な構文によって表すことができるが、ウェブでのデータの交換のためには XML による表現がもっとも適用範囲が広く、重要である。図 2.2 に表 2.1 に示す RDF データの RDF/XML 構文を示す。

表 2.1: RDF トリプルの集合

主語	述語	目的語
eg:me	eg:nationality	eg:China
eg:me	eg:name	"Bin Piao"
eg:me	eg:affiliation	
eg:China	eg:capital	eg:Beijing
	eg:name	"Meme media laboratory"
	eg:tel	011-706-****

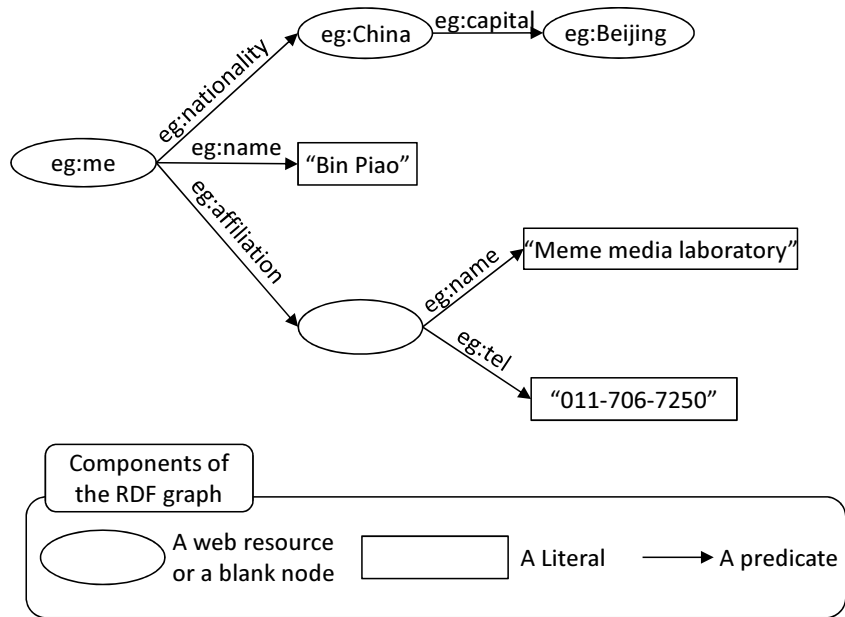


図 2.1: RDF グラフの例

```
1:    <rdf:RDF
2:        xmlns:eg=http://example.org>
3:    <rdf:Description rdf:about="http://example.org/me">
4:        <eg:nationality>
5:            <rdf:Description rdf:about="http://example.org/China">
6:                <eg:capital>
7:                    <rdf:Description rdf:about="http://example.org/Beijing"/>
8:                </eg:capital>
9:            </rdf:Description>
10:        </eg:nationality>
11:        <eg:name>Bin Piao</eg:name>
12:        <eg:affiliation>
13:            <rdf:Description>
14:                <eg:name>Meme Media Laboratory</eg:name>
15:                <eg:tel>011-706-7250</eg:name>
16:            </rdf:Description>
17:        </eg:affiliation>
18:    </rdf:Description>
19: </rdf:RDF>
```

図 2.2: RDF/XML の例

2.2.2 RDF-Schema

RDF はリソースのプロパティやリソース間の関係を記述するモデルを提供するが、プロパティそのものについては直接定義することができない。W3C では、RDF で用いる語彙定義に、RDF-Schema(RDF Vocabulary Description Language) を勧告している。RDF-Schema は、個々のプロパティの定義だけでなく、プロパティどうしの関係や、クラスを定義する手段も提供する。

RDF-Schema では、同じ性質を有するリソースの集合をクラスと呼び、あるクラスのメンバは、そのクラスのインスタンスと呼ぶ。RDF-Schema では、クラスを表現す

る基本語彙として、`rdfs:Resource`, `rdfs:Class`, `rdf:Property` などの基本クラスを予め用意している。全てのものは `rdfs:Resource` のインスタンスであり、全てのクラスは `rdfs:Class` のインスタンスである。`rdf:Property` は RDF のプロパティを表すクラスである。

RDF-Schema では、プロパティが関係付けるリソースがどんなクラスのインスタンスかという情報を記述することができる。プロパティの主語となるクラスを定義域、目的語となるクラスを値域と呼び、それぞれ `rdfs:domain`, `rdfs:range` というプロパティを用いて表現する。

2.2.3 OWL

RDF-schema は RDF で用いる語彙の基本的な表現手段を提供するが、概念やリソースの関係を精密に示したり、推論の可能な論理を記述するには、より洗練された語彙を定義するウェブ・オントロジー言語が必要となる。ウェブのリソースを記述するオントロジー言語として 2004 に W3C によって勧告されたのが OWL(Web Ontology Language) である。

OWL では、`owl:Class`, `owl:Thing`, `owl:Nothing` の 3 つの基本クラスを提供するとともに、以下の 6 つのクラス表現を提供する。

- (1) URI 参照による単純クラス
- (2) 列挙された要素によって構成されるクラス
- (3) プロパティの制約を持つクラス
- (4) 2 つ以上のクラスの積集合により定義されるクラス
- (5) 2 つ以上のクラスの和集合により定義されるクラス
- (6) あるクラスに対する補集合により定義されるクラス

(2) の列挙された要素によって構成されるクラスの記述には, owl:oneOf プロパティが用いられ, (3) のプロパティの制約を持つクラスの記述には, owl:Restriction プロパティが用いられる. (4) の 2 つ以上のクラスの積集合により定義されるクラスの記述は, owl:intersectionOf プロパティが用いられ, (5) の 2 つ以上のクラスの和集合により定義されるクラスの記述は, owl:unionOf が用いられ, (6) のあるクラスに対する補集合により定義されるクラスの記述は, owl:complementOf プロパティが用いられる.

2.2.4 SPARQL

SPARQL(SPARQL Protocol and RDF Query Language)2008 年に W3C によって勧告された RDF クエリ言語である.

SPARQL には, 以下の 4 つのクエリ形式がある.

- (1) SELECT
- (2) CONSTRUCT
- (3) ASK
- (4) DESCRIBE

これらのクエリ形式は, パターン・マッチングのソリューションを用いて, 結果集合または RDF グラフを作成する. SELECT 形式は, 変数とそのバインディングを直接返し, CONSTRUCT 形式は, グラフ・テンプレートで指定された 1 つの RDF グラフを返す. ASK 形式は, クエリ・パターンにソリューションがあるか否かを返し, DESCRIBE 形式は, 資源に関する RDF データを含んだ 1 つの結果 RDF グラフを返す.

本論文で定義する検索要求は, 全て SPARQL の SELECT クエリ形式を用いる. SELECT クエリ形式の構文は以下のように記述できる.

$$\begin{array}{ll} \textit{SELECT} & \{Variables\} \\ \textit{WHERE} & \{Graph\ Patterns\} \end{array}$$

SELECT 文は、検索結果として返す SPARQL 変数を指定する。SPARQL 変数は、先頭に “?” を付けた文字列として表記する。WHERE 文は、あらゆる複雑な構造の RDF データをクエリとして記述するグラフ・パターンを指定する

2.2.5 セマンティック・ウェブ・サービス

従来のウェブサービスには、サービスの内容がどのようなものか、それらをどうやって検索し、組み合わせたらよいのかなど、サービスの意味を記述する方法が用意されていない。セマンティック・ウェブ・サービスは、ウェブサービスの記述にメタデータを付加することによって、ウェブサービスの自動的検索、連携、実行を可能する。バーナーズ＝リーによれば、セマンティック・ウェブ・サービスは、ウェブサービスによるインターネットを介したアプリケーション統合と、セマンティック・ウェブによる組織・団体の壁を越えたデータ統合を融合した分野の総称である。

セマンティック・ウェブ・サービスを支える基本的な標準技術として、OWL-S(Web Ontology Language for Services) や SAWSDL(Semantic Annotations for WSDL) 等が挙げられる。近年、iServe[20] をはじめウェブサービス記述を LOD として公開し、SPARQL を用いてセマンティック・ウェブ・サービスの登録や検索を可能する Linked Service[?] が開発されるとともに、ウェブサービスも、セマンティック・ウェブ上の 1 つのリソースとして公開・共有することができるようになった。

2.3 知識メディアアーキテクチャ

知識メディアアーキテクチャは、既存のメディアを組み合わせることで機能合成を行うことによって、新たな機能をもつメディアの定義を可能している。知識メディアアーキテクチャでは、文書、画像、音声などのあらゆるメディアと、実現されるあらゆるアプリケーションなどを機能を持つ視覚的部品として表現する。ユーザは直接操作により、画面上の部品を組み合わせることができ、スロット結合を通じて機能連携を行うことができる。知識メディアアーキテクチャでは、各部品は MVC(Model, View, Controller)

Chapter 2 ENABLING TECHNOLOGIES



図 2.3: Webble world の画面コピー

アーキテクチャを持ち、View と Controller 部分が定義する視覚的オブジェクトのみならず、Model 部分が定義する内部機構をも持つ。知識メディアアーキテクチャでは、部品の粒度には制限がなく、テキストのような単純なものから、複数の部品を合成した複合部品まで様々なものを扱うことができる。

2.3.1 Webble World

Webble World[16] は知識メディアアーキテクチャの考えに基づき、マイクロソフト.NET に準拠した RIA 開発用技術である Silverlight を用いて開発公開したウェブ版の新しい知識メディアシステムである。図 2.3 に Webble World の画面コピーを示す。Webble Wrld における基本視覚的部品は Webble と呼ばれる。単一 Webble や、いくつかの単一 Webble から合成された複合 Webble は XML ファイルとして表され、Webble World サーバに格納される。

2.3.2 部品間のスロット結合連携

直接操作によって組み合わせた複数の Webble は親子関係によって結びられる。Webble 間の機能連携は、図 2.4 に示しているように、以下に示す 3 種類の標準メッセージの受け渡しにより実現されている。

- Set:
指定された親 Webble のスロットに値を書き込む。
- Update:
親 Webble が子 Webble に対して状態の更新を伝える。
- Gimme:
update メッセージを受けて、指定された親 Webble のスロットの値を読み出す。

Webble 間の Set メッセージは、 $\langle \text{slotname} \rangle \langle \text{value} \rangle$ のように 2 つの引数を持つ。 $\langle \text{slotname} \rangle$ は親 Webble と結合しているスロットの名前を表し、 $\langle \text{value} \rangle$ はそのスロ

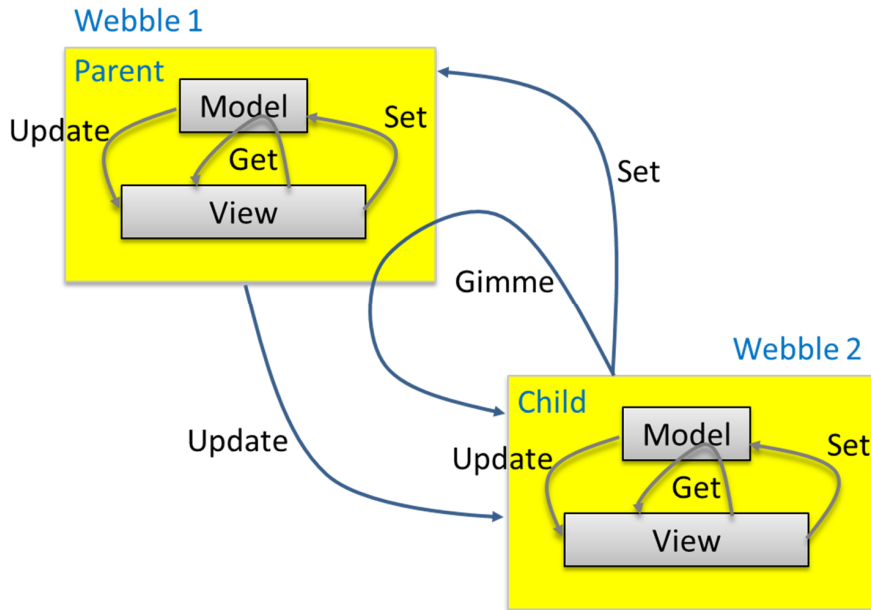


図 2.4: Webble 間標準メッセージ

トに書き込む値を表す。書き込まれた値は結合されている親 Webble のスロットに伝送される。親 Webble は内部状態が変更されると、子 Webble に対して Update メッセージを送る。Update を受けた子 Webble は、親 Webble に Gimme メッセージを送る。Gimme メッセージは、`<slotname>` のように 1 つの引数を持つ。ここで、`<slotname>` は結合している親 Webble のスロットの名前を表し、子 Webble はそのスロットから値を読みだし、結合している自身のスロットの値を変更する。

Webble の機能合成は、上で述べた 3 つのメッセージ `set`, `gimme`, `Update` によるスロット値の授受とそれに伴う内部状態の更新の伝搬によって行われる。これらのメッセージは各 Webble ごとに禁止することができ、メッセージを制御することにより機能合成の仕方を決めることができる。各 Webble はユーザからの操作もしくはスロット間のメッセージなどにより内部状態が変化した場合に以下のような処理を行う。

1. 親 Webble へ Set メッセージの送出が許可されていればそれを行い、子 Webble に対し状態の変化を伝えるために Update メッセージを送出する。

2. 子 Webble から Set メッセージを受け取った Webble は前述のメッセージ交換を行い、自分のスロットの値を更新する。このとき、状態が変化すれば1の動作を行う。
3. 親 Webble からの Update メッセージを受け取ると、Update メッセージの受け取りと Gimme メッセージの送出が許可されていれば親 Webble に Gimme メッセージを送る。これにより、親 Webble から読み出された値は自分のプライマリスロットに書き込まれる。またこのとき、状態が変化すれば1の動作を行う。

第 3 章

ESIASW フレームワーク

The ESIASW framework

本章では, RDF データモデルとその拡張について述べた後, 本論文で提案する ESIASW フレームワークの概要について述べ, 最後に ESIASW フレームワークで提供する 3 つの視覚的部品とそれらの機能について述べる.

3.1 RDF データモデルとその拡張

RDF では, URI の有限集合 I , blank node の有限集合 B , リテラルの有限集合 L に対し, 3 つ組 $(s, p, o) (\in (I \cup B) \times I \times (I \cup B \cup L))$ を RDF トリプルと呼ぶ. ここで, s は主語を, p は述語を, o は目的語を表す. RDF データセット $D (\subset (I \cup B) \times I \times (I \cup B \cup L))$ は RDF トリプルの集合である.

全ての $(I \cup B)$ の要素はリソースと呼ばれ, リソースはクラスと呼ばれる, 同じ性質を有するグループに分けられる. あるクラスのメンバは, そのクラスのインスタンスと呼ばれる. あるクラス c の全てのインスタンスの集合を $E_D(c)$ と記述する. 本論文では, 検索対象である RDF データセットの中で `rdfs:Class` または `owl:Class` のインスタンスとして既に定義されていた既存のクラスだけでなく, 同じ性質を有するリソース, つまり共通のプロパティ集合を個々のプロパティ集合の部分集合として含むよう

な複数のリソースの集合を，ユーザ自身が1つの新しいクラスとして抽象化することを可能にする．抽象化された新しいクラスを抽象クラスと呼び，あるリソースの集合 $R(\subset I \cup B)$ を1つの抽象クラスとして定義した場合， $abs(R)$ と記述する．新しく定義した抽象クラスはブランクノードで示されるクラスであり，URI 参照識別子を持たず，システムによって自動的に割り振られた ID による参照が必要となる．

RDF データにおいて，あるクラス c の各インスタンス $i(\in E_D(c))$ はクラス c 以外にも複数のクラスのインスタンスにもなることがある．本論文ではそれらのクラスを，クラス c のオーバーラップクラスと呼ぶ． $C_D(i)$ でリソース $i(\in I \cup B)$ の全てのクラスの集合を表すとき，クラス c の全てのオーバーラップクラスの集合を $OverC_D(c)$ と記述し，以下のように定義する．

$$OverC_D(c) \triangleq \bigcup_{i \in E_D(c)} C_D(i) \quad (3.1)$$

クラス間の論理演算は，論理演算子 AND($\wedge$)，OR($\vee$)，DIFFERENCE($-$) を用いて定義する．2つのクラス c_1 ， c_2 の間の論理演算を用いて新しく定義した抽象クラスを論理クラスと呼び， $c_1 \diamond c_2$ ，($\diamond \in \{\vee, \wedge, -\}$) のように記述する．論理クラスを各演算子に対して以下のように定義する．

$$c_1 \wedge c_2 \triangleq abs(E_D(c_1) \cap E_D(c_2)) \quad (3.2)$$

$$c_1 \vee c_2 \triangleq abs(E_D(c_1) \cup E_D(c_2)) \quad (3.3)$$

$$c_1 - c_2 \triangleq abs(E_D(c_1) - E_D(c_2)) \quad (3.4)$$

図 3.1 に示すようにクラス間論理演算は，クラスのインスタンス集合の積集合，和集合，差集合を抽象化したものである．

本論文におけるクラス間論理演算は，クラス部品に対する直接操作を用いて定義しており，全て自動的に SPARQL の論理演算へと変更されて実行される．これにより，ユーザはクラス間論理演算を表す OWL 文や複雑な SPARQL 検索要求文を記述しなくても，スクリーン上での直接操作によるクラス間論理演算を用いて必要なクラスを簡便に新しく定義することができる．また，クラス間論理演算を OWL 文ではなく，

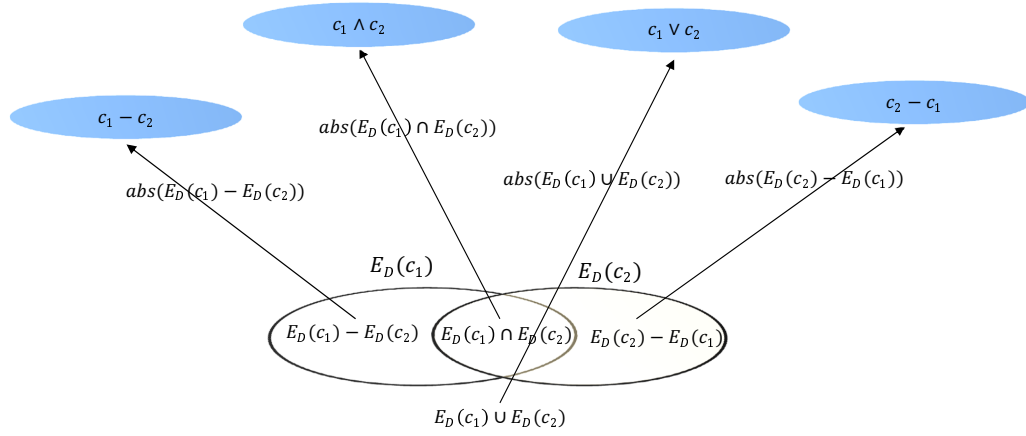


図 3.1: クラス間論理演算

SPARQL の論理演算へと変換して実行することによって、定義された論理クラスを用いて検索を行う際、OWL 文を用いて定義した論理クラスと検索対象である RDF データセットを事前に統合することなく、SPARQL への直接書き換えのみで実行することができる。

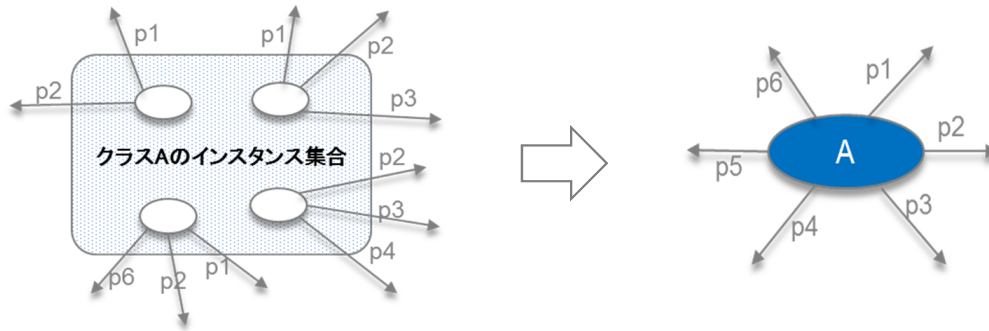


図 3.2: クラスのプロパティ集合

本論文では、図 3.2 に示すように、あるクラスの全てのインスタンスのプロパティ集合の和集合をそのクラスのプロパティ集合とする。クラスの各プロパティに対し、リソース間の関係を表すプロパティを関係プロパティとよび、リソースの属性を表すプロパティをアトリビュート・プロパティと呼ぶ。あるクラス c の全ての関係プロパティ

の集合を $ResP_D(c)$, 全てのアトリビュート・プロパティを $AttrP_D(c)$ と定義する.

$$ResP_D(c) \triangleq \{p | \exists i \in E_D(c) \exists r \in I \cup B ((i, p, r) \in D)\} \quad (3.5)$$

$$AttrP_D(c) \triangleq \{p | \exists i \in E_D(c) \exists o \in L ((i, p, o) \in D)\} \quad (3.6)$$

クラス c のあるプロパティ p に対し, p の主語となるクラス c のインスタンスの集合を $Subj(c, p)$, p の全ての目的語の集合を $Obj(c, p)$ と定義する.

$$Subj_D(c, p) \triangleq \{i | \exists o \in I \cup B \cup L (i \in E_D(c) \wedge (i, p, o) \in D)\} \quad (3.7)$$

$$Obj_D(c, p) \triangleq \{o | \exists i \in E_D(c) ((i, p, o) \in D)\} \quad (3.8)$$

ここで, 集合 $Subj_D(c, p)$ の要素数をクラス c における p の主語数と呼ぶ. また, 関係プロパティ $p \in ReP(c)$ に対し, 抽象クラス $abs(Obj_D(c, p))$ をクラス c の関係プロパティ p のプロパティ値クラスと呼び, $c.p$ と記述する.

本論文では, ある RDF データセット D の全体または一部のスキーマを, (sc, p, oc) のようなトリプル (以下では, これをスキーマトリプルという) の集合として定義し, S_D と記述する. ここで, sc と oc の各々は, リテラル, プロパティ, データ型のクラス (`rdfs:Literal`, `rdfs:Property`, `owl:ObjectProperty`, `owl:DatatypeProperty`, `rdfs:Datatype` の 5 つのクラス) と, それらの全てのサブクラスを除いた既存のクラスの 1 つ, ないしはユーザ自身が新しく定義した抽象クラスのいずれか 1 つを表し, 本論文では, sc を主語クラス, oc を目的語クラスと呼ぶ. p は,

$$p \in I \wedge \exists s \in E_D(sc) \exists o \in E_D(oc) ((s, p, o) \in D) \quad (3.9)$$

を満たすプロパティを表す. スキーマトリプルの例を図 3.3 に示す. スキーマトリプルにおける抽象クラスには, クラス間論理演算を用いて定義した論理クラスと, あるクラスの関係プロパティの値を用いて定義したプロパティ値クラスの 2 種類がある. 抽象クラスの定義において, OWL では, クラス間論理演算以外にも, プロパティ制約を持つ抽象クラス (`owl:Restriction`) と, 列挙される要素によって構成される抽象クラス (`owl:oneOf`) の 2 種類があるが, これらは, それぞれ 4.4.2 節で後述するように,

各クラスのプロパティに制約条件を付与して検索要求を定義することと、6.3 節で後述するように、検索されたクラスのインスタンスを制約条件として用いて新たな検索要求を定義することに相当するので、スキーマトリプルの定義の中にはこの 2 つのタイプの抽象クラスの定義は含めない。RDFS では、2 つのプロパティ `rdfs:domain` と `rdfs:range` を用いて、プロパティの主語クラスや目的語クラスを定義する機構を提供しているが、現在 Web 上に公開されている RDF データセットにおいては、プロパティの多くが `rdfs:domain` と `rdfs:range` の定義を持っていないことから、本論文では、スキーマトリプル中のプロパティを主語クラスと目的語クラスの両クラスのインスタンス間のプロパティを用いて定義するようにしている。

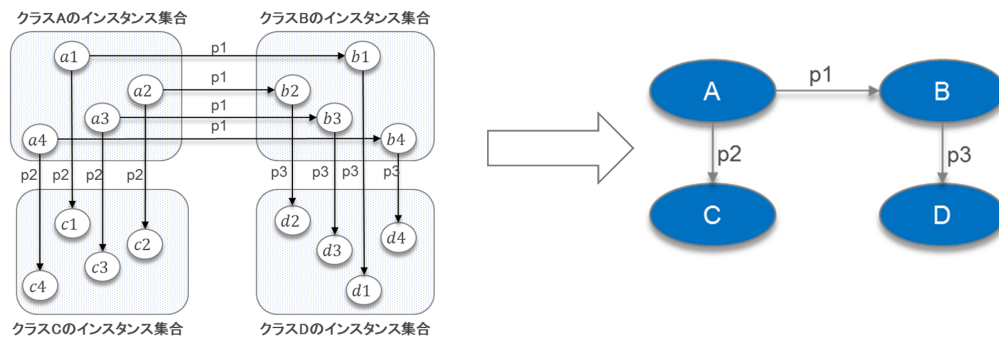


図 3.3: RDF データとそのスキーマ

3.2 ESIASW フレームワークの概要

ESIASW フレームワークは、機能を持つ視覚的部品に対する直接操作を用いて、以下の 6 つの機能を可能にすることで、セマンティック・ウェブ上のリソースに対する探索的検索、連携統合、可視化分析を支援し、その組み合わせ利用を可能にする枠組みである。ESIASW フレームワークは、膨大なデータ集合やサービスに対する探索的検索、連携統合、可視化分析を試行錯誤的に繰り返しながら、データ集合やサービス自体に対する理解を深めつつ、課題を解決するためのシナリオそのものを発見し、新たな知見発見に至る全過程を統括的に支援することができる。

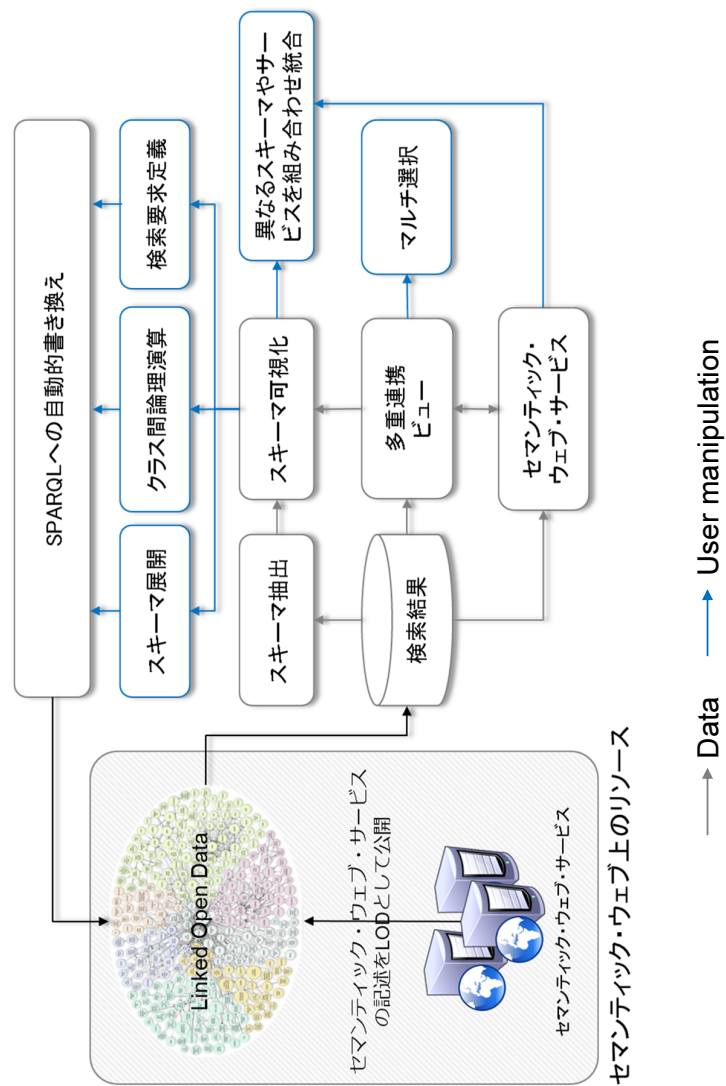


図 3.4: ESIASW フレームワークの概要

- (1) 多視点スキーマの探索的抽出・展開・可視化
- (2) スキーマに基づいた視覚的検索要求定義
- (3) クラス間論理演算による新しいクラス定義
- (4) スキーマやセマンティック・ウェブ・サービスの連携統合
- (5) 多視点からの探索的可視化分析
- (6) 視覚的部品に対する直接操作による (1)~(5) の実現

セマンティック・ウェブ上のリソースの探索的検索において、(1) の「多視点スキーマの探索的抽出・展開・可視化」により、ユーザは探索的に抽出・展開・可視化したスキーマに基づいて検索要求を定義したり、異なるスキーマを持つデータを統合したりすることが可能になる。スキーマを探索的に抽出・展開する際、既存クラスだけでなく、必要なクラスをユーザ自身が簡便に新しく定義し、定義した新たなクラスを用いて新しいスキーマを即時定義できるようにするためには、視覚的部品に対する直接操作による (2) の「クラス間論理演算による新しいクラス定義」機能が必要となる。SPARQL に関する知識を持たないユーザでも、必要な検索要求を自身で簡単に作成できるようにするためには、(3) の「視覚的検索要求定義」機能が必要となる。セマンティック・ウェブ上のリソースの連携統合において、異なるスキーマを持つ複数のデータ集合を統合したり、データ集合とセマンティック・ウェブを統合したりするためには、(4) の「異なるスキーマやセマンティック・ウェブ・サービスの組み合わせ統合」機能が必要となる。セマンティック・ウェブ上のリソースの分析において、試行錯誤しながら興味深いデータ集合やそれらの間の関係を探索・分析するためには、(5) の「多視点からの探索的可視化分析」の機能が必要となる。プログラミング技術を持たないユーザでもインタラクティブに検索要求を変えながら、セマンティック・ウェブ情報を探索的に検索、連携統合、可視化分析できるようにするためには (6) の機能が必要となる。

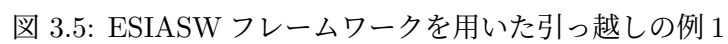
ESIASW フレームワークの概要を図 3.4 に示す。ESIASW フレームワークでは、任意の RDF データセットから、必要なスキーマを抽出し、可視化することができる。

ユーザは可視化されたスキーマを直接操作して、それをさらに展開したり、スキーマ中のクラスを直接組み合わせ論理演算することで新たなクラスを定義したり、スキーマに基づいて視覚的に検索要求を定義することができる。ESIASW フレームワークでは、これらの操作は、全て自動的に SPARQL へと書き換えられ実現される。また、検索したセマンティック・ウェブ・サービスの記述を用いて、そのサービスのプロキシの作成し、それを用いてサービスへのアクセスを可能にする。ユーザは、可視化されたスキーマや構築されたサービスのプロキシを直接組み合わせ連携することで、多様なデータ集合やサービスを組み合わせ統合することができる。また、ESIASW フレームワークでは、多重連携ビューを用いて検索結果やサービスのレスポンスを多様な視点から探索的分析することを可能にしている。

3.3 ESIASW フレームワークを用いた引っ越しの例

ESIASW フレームワークを用いて「転職先で新しい家を探したい」という課題を解決する例を図 3.5 と図 3.6 に示す。このように明示されている条件の他にも、ユーザは、家賃、通勤時間、家の広さ、子供が通える学校や塾、両親が利用できる病院、周辺の環境等を統合的に考慮する必要がある、最初から完全なシナリオを想定することは困難である。このような曖昧な課題を解決するためには、様々な条件を探索的に探し、精緻化しながら、課題を解決するためのシナリオそのものをも少しずつ発見して行く必要がある。

この課題において、ユーザは、まず初めに、東京にある手頃で広い物件を探そうとしている。この要求に対し、従来のウェブサービスを利用した場合、そこに登録されている各物件に対する記述を、人間が直接読んで理解しながら、相互に比較しなければならない。ESIASW フレームワークを用いた場合、ユーザは、図 3.5(1) に示すように、セマンティック・ウェブから「賃貸物件」というクラスを検索し、それを始点として、物件の情報を検索することができる。検索された「賃貸物件」クラスは、クラスのプロパティ集合を自動的に検索し、可視化する。ユーザは、可視化されたプロパ



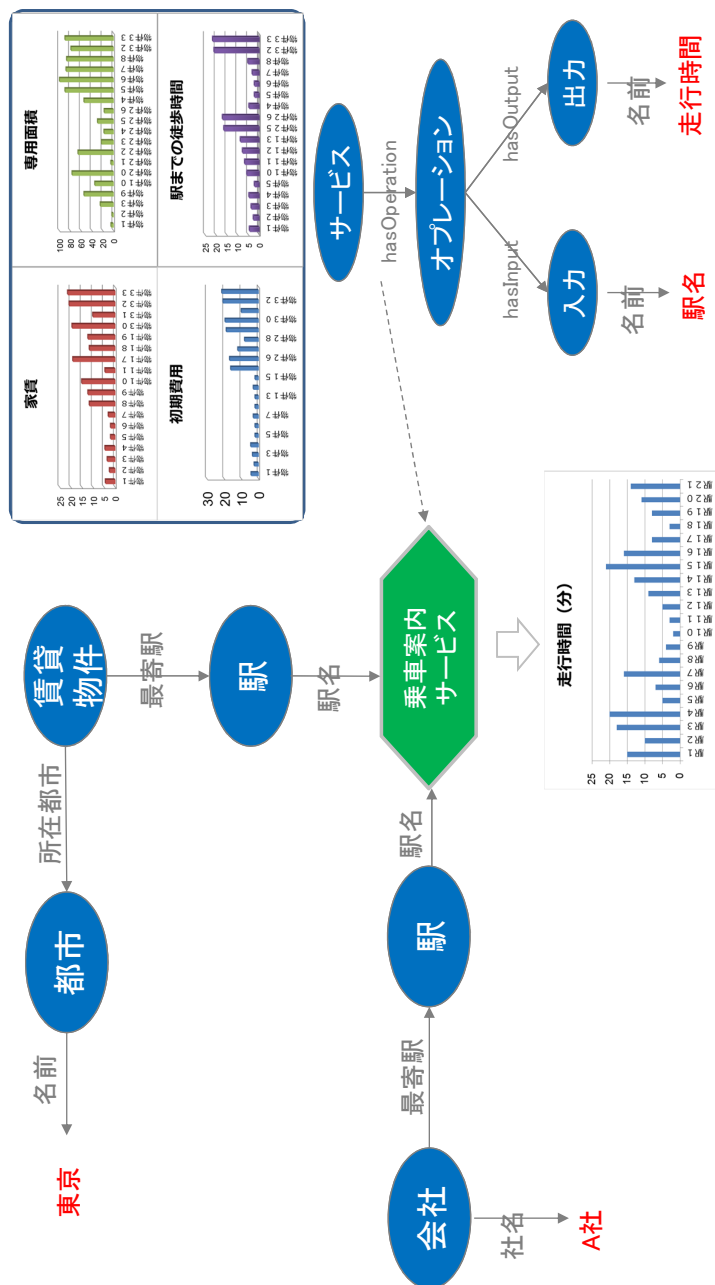


図 3.6: ESIASW フレームワークを用いた引っ越しの例 2

ティを直接操作することによって、関連付けられている他のクラスを抽出したり、そのプロパティの値を検索したりすることができる。この例では、図 3.5 の (2) と (4) に示すように、賃貸物件クラスのプロパティの中の「所在都市」というプロパティと「最寄駅」という 2 つのプロパティを用いて、賃貸物件の範囲を指定できる「都市」クラスと「駅」クラスを抽出している。ユーザは、抽出した「都市」クラスの「名前」プロパティに「東京」という制約条件を指定することによって、物件の地域を東京へと絞り込んだ後、「賃貸物件」クラスの「専用面積」、「初期費用」、「駅までの徒歩時間」、「家賃」の 4 つのプロパティと、「駅」クラスの「駅名」プロパティを直接操作し、その値を図 3.5 の (3) と (5) に示すように可視化することで、東京にある各駅からその周辺にある物件情報を比較しながら必要な物件候補を探し出すことができるようになる。

探し出した物件に対し、ユーザは、その中で会社までの通勤時間が短い物件を探そうとしている。このとき、図 3.6 に示すように、ユーザはまず会社クラスとその最寄駅クラスを抽出し、転職先の会社名を指定することによって、会社の最寄駅を検索することができる。次に、セマンティック・ウェブ・サービスの記述に関するスキーマを探索的に抽出し、入力記述に「駅名」という制約条件を、出力記述に「走行時間」という制約条件を付与することによって、両駅間の電車の走行時間を計算することができる「乗車案内」というサービスを検索することができる。ユーザは、賃貸物件の最寄駅名と会社の最寄駅名を検索された「乗車案内」サービスの 2 つの入力と直接結合し、出力である両駅間の走行時間を可視化することによって、通勤時間が短い物件を探し出すことができるようになる。

3.4 視覚的部品

ESIASW フレームワークでは、ワークスペース部品と、RDF データスキーマの各クラスを可視化するクラス部品、セマンティック・ウェブ・サービスのプロキシを作成可能なサービス部品、多重連携ビューを動的作成可能な可視化分析部品の 4 つの視覚的部品を提供し、それらを直接操作することによって、セマンティック・ウェブ上

のデータやサービスに対する探索的検索，連携統合，可視化分析と，その組み合わせ利用を可能にする．

3.4.1 ワークスペース部品

ワークスペース部品は，セマンティック・ウェブ上のリソースの探索検索，連携統合，可視化分析を行うための基盤部品である．ワークスペース部品では，図 3.7 に示すように，RDF データセットの選択機能や，キーワード検索を用いたクラスの検索機能を提供している．これらの機能を用いて，ユーザは検索対象とする RDF データセットから，既存のクラスを検索し，検索結果から，必要なクラスを選択して RDF データスキーマの探索的抽出・展開・可視化の始点クラスと定義ことができる．始点クラスから始め，探索的抽出・展開された各 RDF データのスキーマは，全てワークスペース部品の中に保持され，クラス部品によって可視化される．また，可視化されたスキーマに基づいて視覚的に定義した全ての検索要求は，全てワークスペース部品に伝送され自動的に SPARQL へと書き換える．



図 3.7: ワークスペース部品

3.4.2 クラス部品

スキーマ中の1つのクラス c を視覚的部品として部品化したクラス部品は、クラス c の関係プロパティの集合 $ResP_D(c)$ 、アトリビュート・プロパティの集合 $AttrP_D(c)$ 、オーバーラップクラスの集合 $OverC_D(c)$ を自動的に検索し、その結果を可視化する。検索された関係プロパティとアトリビュート・プロパティは、図 3.8 の (1) と (2) に示

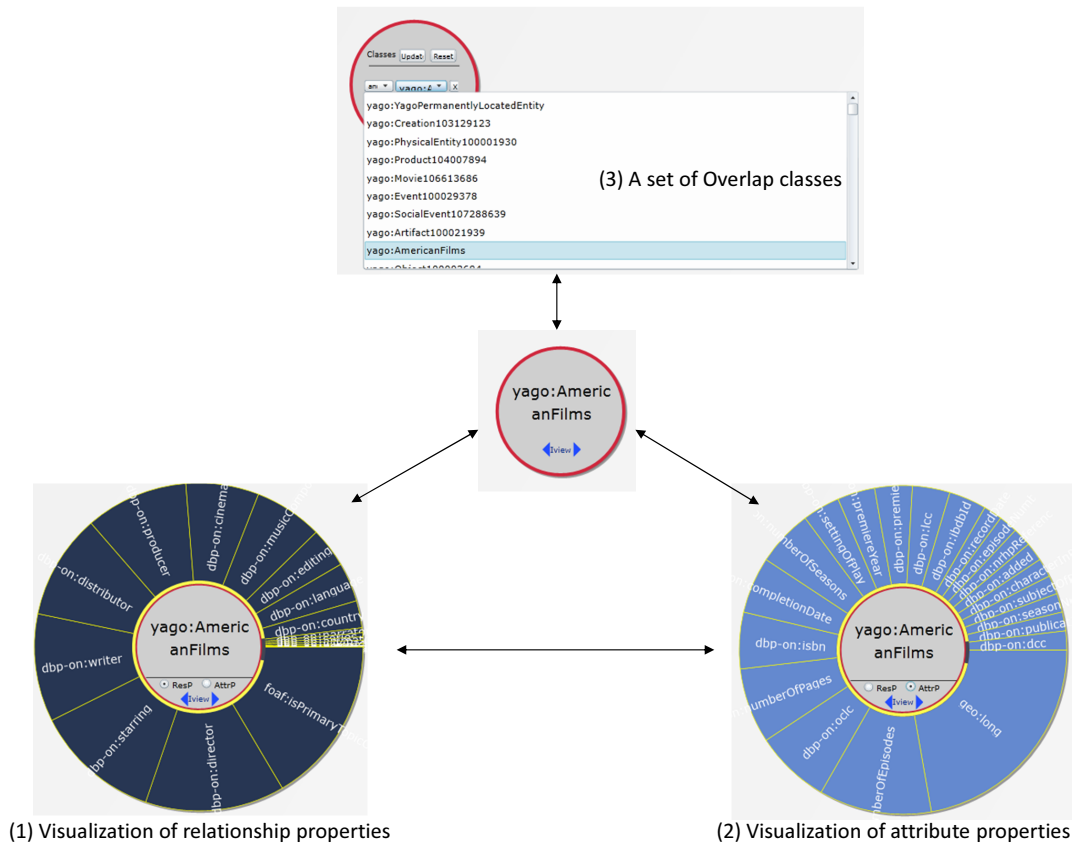


図 3.8: クラス部品およびそのプロパティとオーバーラップクラスリスト

すように2つの異なる色のドーナツグラフとしてクラス部品の周りに可視化される。これらは、クラス部品の真ん中にあるラジオボタンを選択することで交互に切り替えることができ、ユーザの指示によって表示・非表示を選択することができる。各ドーナツグラフの各スライスはそれぞれ一つのプロパティを表し、各スライスの内角の角

度はプロパティの主語数に比例している。ユーザはクラス部品がその周りに可視化する関係プロパティを直接操作することによって、このクラスと意味的關係を持つクラスを抽出することができる。また、クラス部品がその周りに可視化するアトリビュート・プロパティを直接操作することによって検索要求の変数や制約条件を視覚的に直接操作で定義することができる。各クラス部品には、円形のスライダーが付いており、これを操作して周辺にスライスとして表示するプロパティの数を制限することができる。検索されたクラス c の全てのオーバーラップクラスは、図 3.8 の (3) に示すようにドロップダウンリストとして表示される。ユーザは、クラス部品で表示される各クラスに対し、ドロップダウンリストを用いてこのクラスとオーバーラップするクラスを求め、そのリストから任意の 1 つを選択して、選択されたクラスと自身との間で論理演算を行うことによって、自身のクラスをこの新しいクラスへと制限することができる。また、異なるクラス部品を直接組み合わせで論理演算を適用することによって、新しい抽象クラスを定義することができる。

3.4.3 サービス部品

サービス部品は、図 3.9 に示すように指定されたセマンティック・ウェブ・サービスに対し、SPARQL を用いてそのサービスの記述から入力、出力、サービスリクエストのテンプレートなどの情報を自動的に抽出し、それらの用いて指定したセマンティック・ウェブ・サービスのプロキシを作成することができる。サービス部品は、入力または出力を他のサービス部品の入力または出力に直接結合したり、展開・可視化された RDF データスキーマ中のあるクラスのプロパティに結合することによって、異なるセマンティック・ウェブ・サービスを統合したり、セマンティック・ウェブ・サービスと任意の RDF データセットを組み合わせ統合することができる。サービス部品は、入力の値を、検索されたサービスリクエストのテンプレートの中のそれと対応する入力名と入れ替えることで、サービスへのリクエストを自動的に作成し、サービスへと問い合わせることができる。

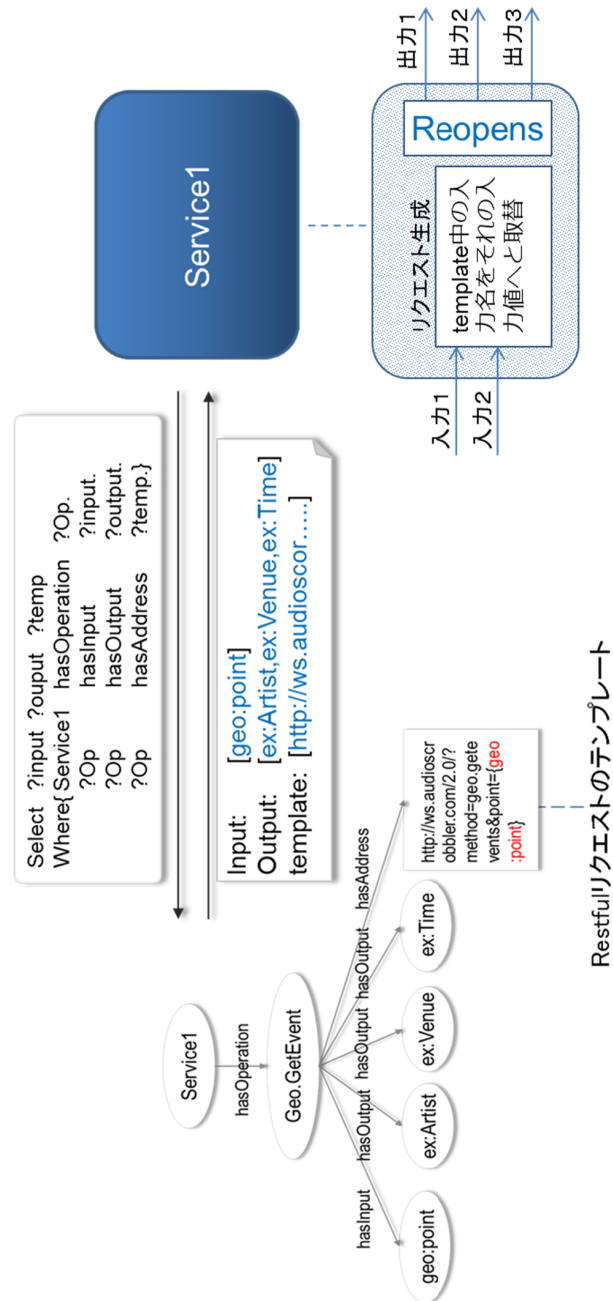


図 3.9: サービス部品の例

Chapter 3 THE ESIASW FRAMEWORK

3.4.4 可視化分析部品

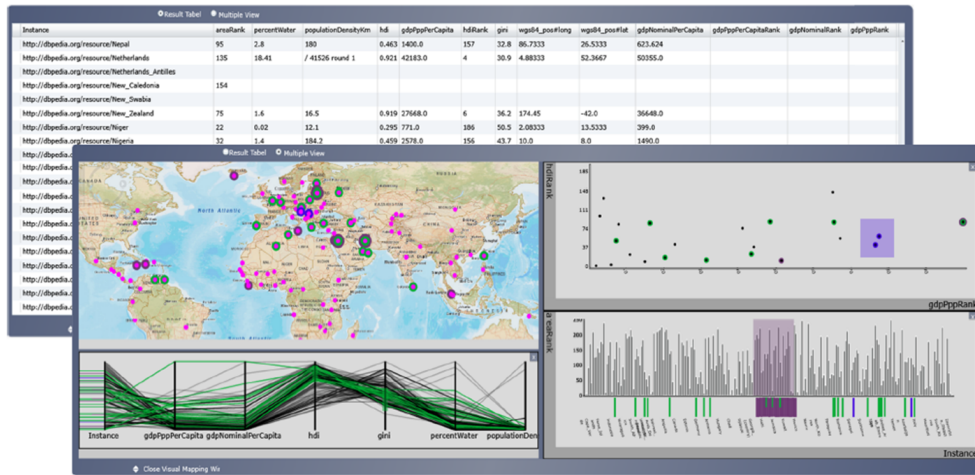


図 3.10: 可視化分析部品の例

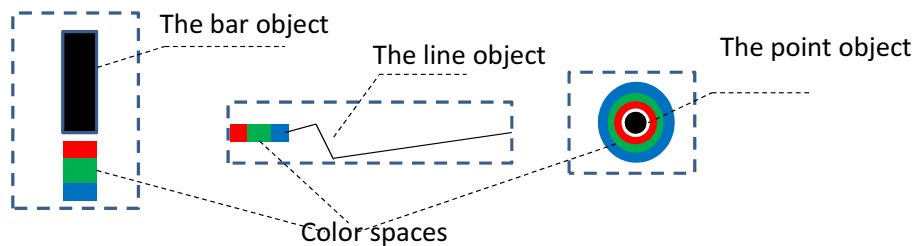


図 3.11: 3 種類の視覚的オブジェクトとそれらの color space

可視化分析部品では、ヒストグラムや散布図などの基本チャート、地図、パラレルコーディネートなどの多様なビューを提供しており、ユーザをこれらのビューを用いて必要な多重連携ビューを自在に構築することができる。図 3.10 に可視化分析部品の例を示す。この例では、GDP per capita, HDI, GINI, 位置などの国に関する情報を多重連携ビューとして可視化している。ESIASW フレームワークでは、構築された多重連携ビューをユーザが直接操作して、興味のある複数の視覚的オブジェクトの集合を同時に選択することができるマルチ選択を可能にする。可視化分析部品で提供する各ビューの中の視覚的オブジェクトは、図 3.11 に示すように、棒、線、点の 3 つの

種類に分類することができる。これらの各視覚的オブジェクトは、マルチ選択の際、各選択の結果を表すために、複数の色を順次に並べ表示できるカラーコード (color code) という属性持つ。

可視化分析部品で提供する各ビューは、単一クラスのプロパティ値、または複数のクラスのプロパティ値の組み合わせを可視化することができる。本論分では、各ビュー v で可視化されるクラスの集合を $C(v)$ と記述し、 $o(v)$ を用いて各ビュー v 中の任意の視覚的オブジェクトを記述する。各視覚的オブジェクト $o(v)$ は、クラス集合 $C(v)$ の要素である各クラスのインスタンスを保持する。 $r_o(c)$ を $o(v)$ の中に保持されているあるクラス $c(c \in C(v))$ のインスタンスとしたとき、各視覚的オブジェクト $o(v)$ に保持されているインスタンスの集合を $R_{o(v)}$ と定義する。

$$R_{o(v)} = \{r_o(c) | c \in C(v)\} \quad (3.10)$$

3.5 まとめ

ESIASW フレームワークでは、クラス部品に対する直接操作によって、それらと意味的關係を持つクラスを抽出したり、クラス間の論理演算を用いて新しいクラスを定義したりしながら、検索要求に応じたスキーマを多様な視点から探索的に抽出・展開することができる。展開された各スキーマはクラス部品をノードとし、それらの間の意味的關係を辺とするグラフとして可視化される。ユーザは、展開・可視化されたスキーマをスクリーン上で直接操作することによって検索要求をインタラクティブに定義しながら、セマンティック・ウェブ上のデータやサービスを検索することができる。また、同一作業空間内で、異なる複数の RDF データのスキーマを展開・可視化したり、必要なサービス部品を用いてセマンティック・ウェブ・サービスを呼び出したりすることを可能にしている。ユーザは展開・可視化された RDF データスキーマを意味的関連や論理的関連に基づいて1つのスキーマに統合して利用したり、異なるサービス部品を意味的関連に基づいて直接結合したり、RDF データスキーマとサービス部品を意味的関連に基づいて直接結合したりすることで、多様なデータやサービ

Chapter 3 THE ESIASW FRAMEWORK

スを自在に組み合わせ統合することを可能にする。また，視化分析部品を直接展開された RDF データスキーマまたは，サービス部品に結合して，多重連携ビューを動的に作成しながら，その上で複数のデータ集合をインタラクティブに選択し，その結果を Brushing&Linking 技術を用いて表すことによって，セマンティック・ウェブ上のリソースの探索的可視化分析を可能している。

第 4 章

セマンティック・ウェブ上のリソースの探索的検索

Exploratory Search of Semantic Web Resources

4.1 概要

セマンティック・ウェブ上のリソースの検索において、ESIASW フレームワークでは、クラス部品を直接操作することによって、そのクラスと何らかの意味的關係をもつ他クラスを逐次抽出したり、検索要求の変数や制約条件を直接指定したり変更することを可能にした。これにより、スキーマに関する知識を持たないユーザでも、関心を持つ 1 つまたは複数のクラスをアクセスの始点として、その意味的關係を辿りながら、必要に応じてスキーマを部分的かつ探索的に展開して行きながら、複数の異なる視点から検索条件を同時に指定したり、変更することが可能となる。また、展開・可視化したスキーマ中の異なるクラスを直接組み合わせ論理演算することによって、論理演算を含む複雑な検索要求を定義するためのスキーマをユーザ自身が自在に定義することが可能となる。ESIASW フレームワークでは、これらの操作は全て自動的に SPARQL へと書き換えられ実行される。これにより、SPARQL やプログラミングに関する知識をもたないユーザでも、スキーマを探索的抽出・展開したり、それに基づいて検索要求をインタラクティブに定義し変更することができ、セマンティック・ウェブから必要な情報を探索的に探し出すことができる。

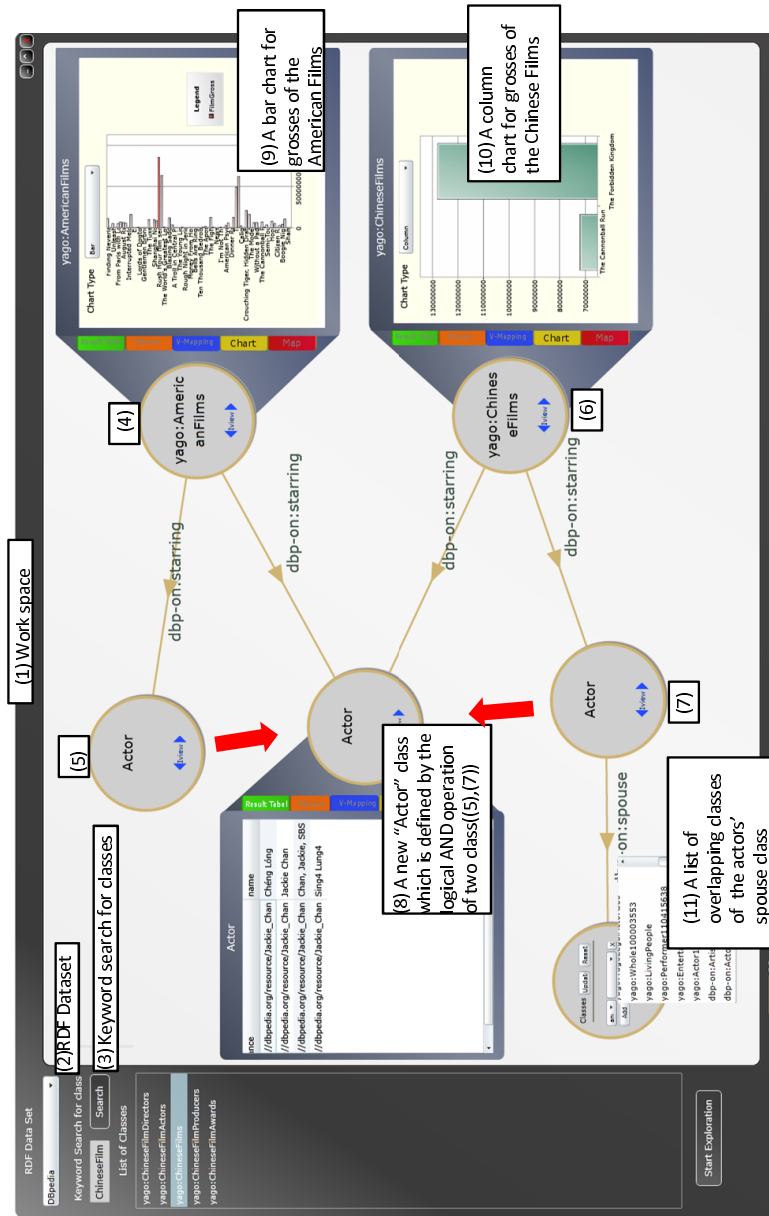


図 4.1: ESIASW フレームワークを用いた探索的検索の例

ESIASW フレームワークを用いたセマンティック・ウェブ上のリソースの探索的検索を、図 4.1 の例を用いて説明する。ハリウッドでアメリカ市場と中国市場をターゲットとした映画を制作しようとする場合、中国の視聴者にも親近感を与えるためには、アメリカ映画と中国映画の両方に出演した経験がある俳優たちを採用することが一つの重要なポイントになると考えられる。本フレームワークを用いることでユーザは、図 4.1 に示すように、画面上の直接操作によって、yago:AmericanFilms クラス (図 4.1 の (4)) と、その主演俳優たちを表す Actor クラス (図 4.1 の (5)), yago:ChineseFilms クラス (図 4.1 の (6)) とその主演俳優たちを表す Actor クラス (図 4.1 の (7)) を順々に抽出し、抽出した二つの Actor クラス間を直接組み合わせて論理積演算を適用することによって、両方の映画に出演したことがある俳優たちを表す新しい Actor クラス (図 4.1 の (8)) を簡単に定義することができる。

さらに、両方の映画に出演した経験者の中から、最も適切な人物を洗い出すために、図 4.1 の (9) と (10) に示すように、彼らが出演した過去の映画の興行収入のを検索して可視化し、その中から高い興行収入を得た映画を直接操作で選択 (図 4.1 の (9)) することによって、両方の映画に出演した最も人気のある俳優たちを探し出すことができる (図 4.1 の (8))。

4.2 多視点スキーマの探索的抽出・展開・可視化

ESIASW フレームワークでは、同じ作業空間内で、異なる複数のクラスを各々のアクセスの始点として、それぞれ別々にスキーマを探索的に抽出・展開することができる。展開された各スキーマの表示には、検索対象とする RDF データセット毎に自動的に異なる色が割り振られる (図 5.2)。

スキーマの探索的抽出・展開の始点となる各クラスは、指定した RDF データセット (図 4.2 の (1)) からキーワード検索 (図 4.2 の (2)) を用いて検索した既存のクラスの一つである。検索されたクラスのリストから必要なものを選択して、クリックすると、選択したクラスが始点クラスとして定義される (図 4.2 の (3))。定義された始点クラス

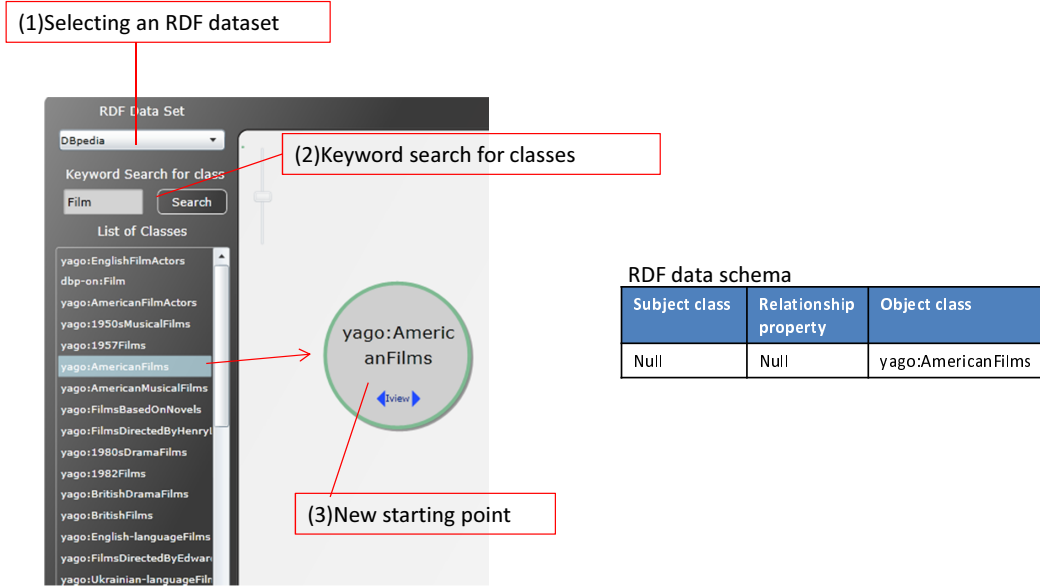


図 4.2: 始点クラス定義の例

は自動的に新しいクラス部品として部品化され、そのクラス部品を直接操作することによって、スキーマの探索的抽出・展開を始めることができる。

本フレームワークでは、スキーマ中のクラス c のクラス部品に対する以下の 3 種類の直接操作を用いてスキーマの探索的抽出・展開を行うことができる。

- (i) クラス c の関係プロパティ $p(\in ResP_D(c))$ のプロパティ値クラス $c.p$ を定義
- (ii) 異なる 2 つのクラス c_1, c_2 間の論理演算を用いて新しい論理クラス $c_1 \diamond c_2$, ($\diamond \in \{\wedge, \vee, -\}$) を定義
- (iii) クラス c を、そのオーバーラップクラスの一つ $c'(\in OverC_D(c))$ との論理演算を用いて定義した新しい論理クラス $c \diamond c'$, ($\diamond \in \{\wedge, \vee, -\}$) へと変更

ここで、操作 (i) は、図 4.3 に示すように、ユーザが、クラス c のクラス部品 (図 4.3 で示す Actor) から必要な一つの関係プロパティ (図 4.3 で示す dbp-on:spouse) を選択し、これをクリックすることによって簡単に実現できる。これによって定義された新しいプロパティ値クラス $c.p$ は、スキーマに追加され、自動的に新しいクラス部品とし

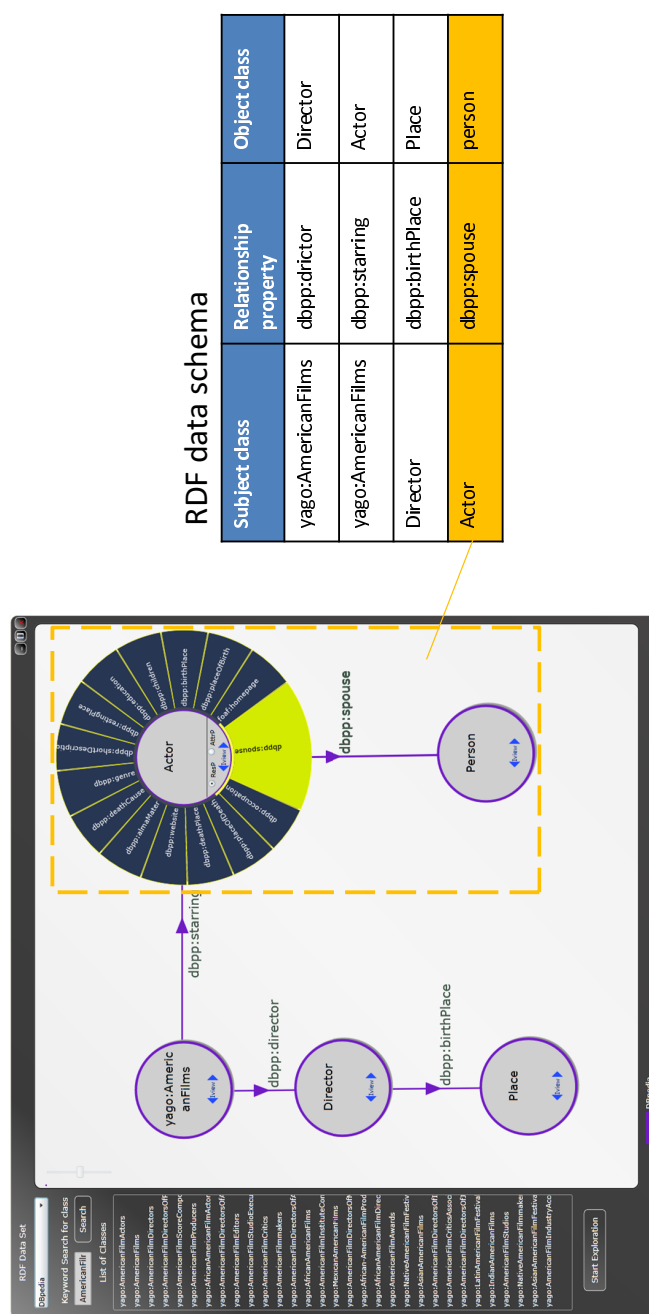


図 4.3: あるクラスから 1 つの関係プロパティを選択し、その値を用いて新しい抽象クラスを定義

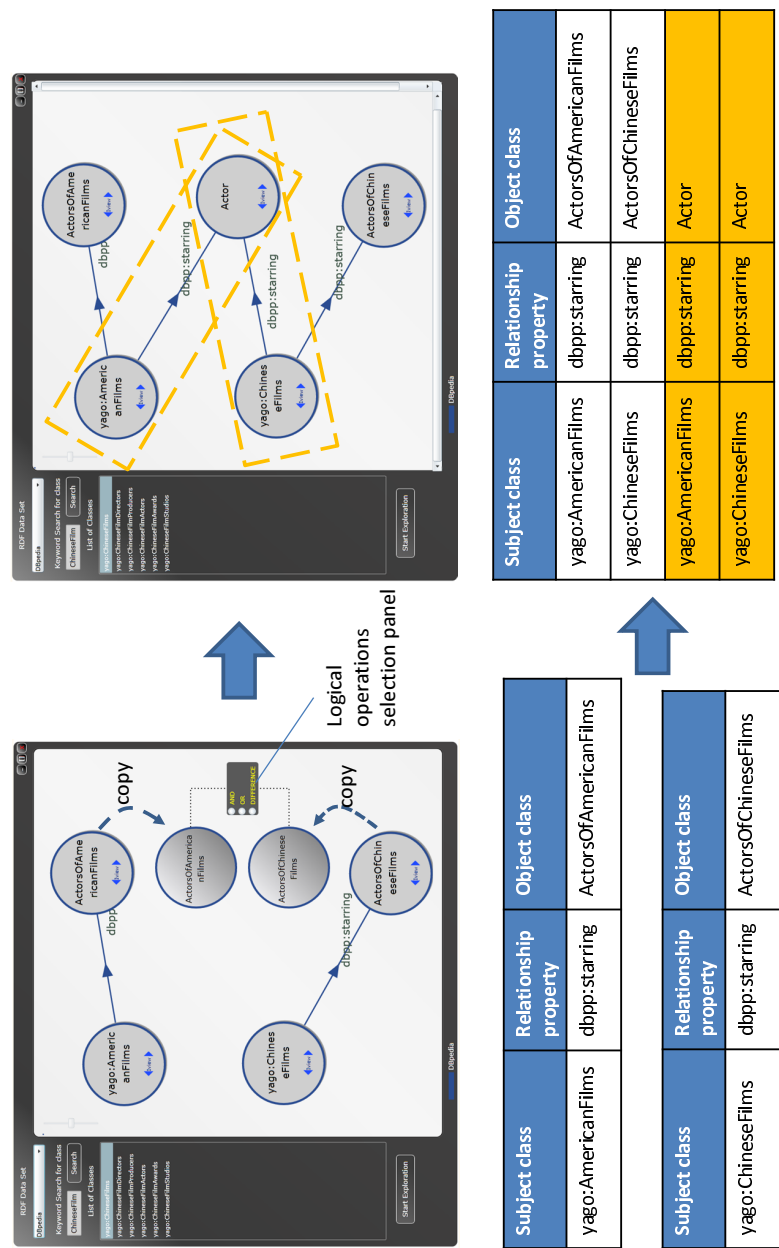


図 4.4: クラス間論理演算による新しい抽象クラスの定義

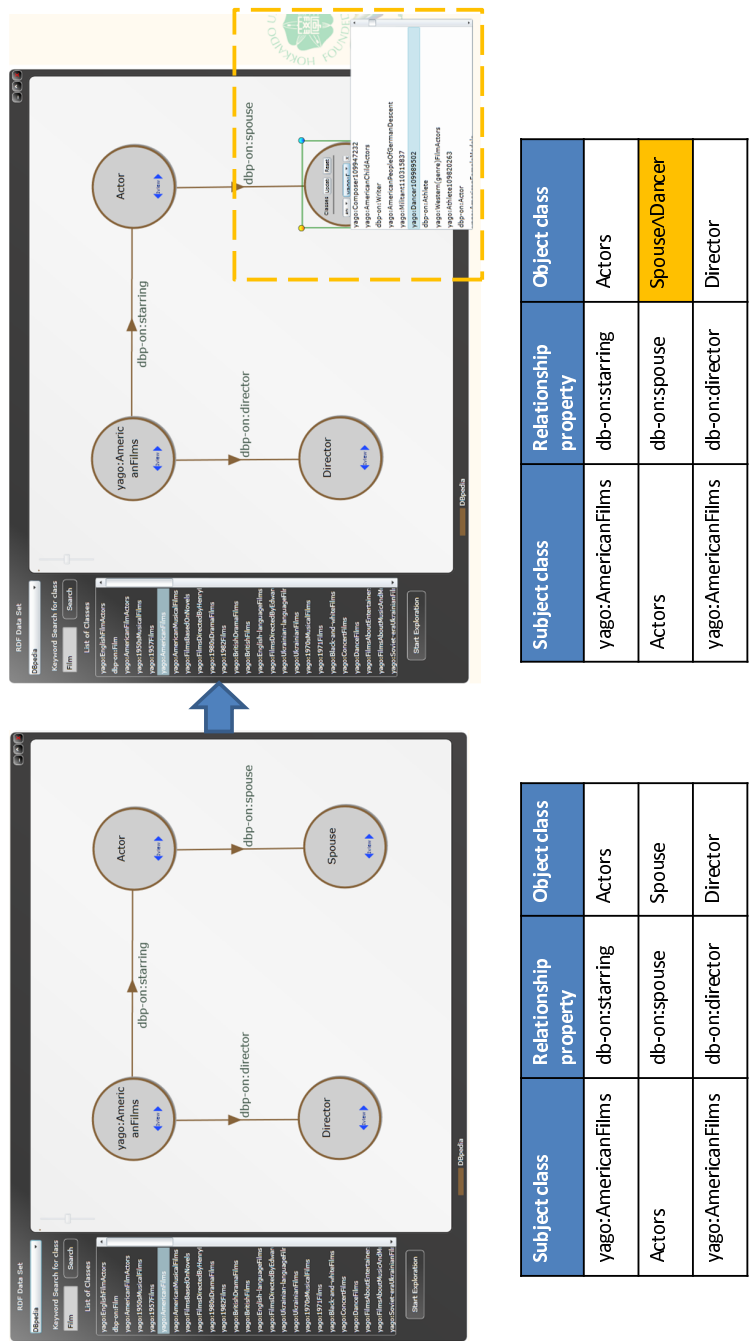


図 4.5: オーバーラップクラスとの論理演算を用いたクラスの変更

て部品化される (図 4.3 で示す Person). 新しく追加されたクラス部品の名前は, 既存のクラスの場合はそのクラスの名前を参照, ないしはユーザの直接入力が必要となる. 操作 (ii) は, 図 4.4 の例で示すように, スキーマの異なる 2 つのクラス c_1 , c_2 のクラス部品 (図 4.4 で示す ActorsOfAmericanFilms と ActorsOfChineseFilms) のコピーをドラッグして接近させると, 論理演算子の選択パネルがポップアップされ, その中から必要な論理演算子を選択することによってこの演算を実現することができる. これによって定義された新しい論理クラス $c_1 \diamond c_2$, ($\diamond \in \{\wedge, \vee, -\}$) は, スキーマに追加され, 自動的に新しいクラス部品として部品化される (図 4.4 で示す Actor). 操作 (iii) は, 図 4.5 に示すように, クラス部品のドロップダウンリストから必要なオーバーラップ・クラスと, 適切な論理演算子を選択することによって実現できる.

本論文では, 既に展開して得られているスキーマ S_D に対し, 新しい始点クラスとして定義されたクラスや上記の操作 (i) と (ii) によって新しく定義したクラス c を, スキーマ S_D に追加することによって展開される新しいスキーマを $add_{S_D}(c)$ と記述する. スキーマ S_D に追加したクラス c が新しい始点クラスであるとき, スキーマ $add_{S_D}(c)$ は

$$add_{S_D}(c) = S_D \cup \{(null, null, c)\} \quad (4.1)$$

と定義される. ここで, $null$ は空値を表す. このとき, S_D が空集合 ϕ である場合, $add_D(\phi, c)$ はクラス c を始点とする RDF データセット D の初期スキーマを表す. 図 4.2 のでは, 新しい始点クラスとしての yago:AmericanFilms クラスと, それを用いた初期スキーマの例を示している. スキーマ S_D に新たに追加したクラス c が S_D の中のクラス c' の関係プロパティ $p \in ResP_D(c')$ を用いて定義したプロパティ値クラス $c'.p$ であるときには, 新しいスキーマ $add_{S_D}(c'.p)$ は

$$add_{S_D}(c'.p) = S_D \cup \{(c', p, c'.p)\} \quad (4.2)$$

と定義される. 図 4.3 の例で示すように, Actor クラスの関係プロパティ dbpp:spouse をクリックして新たなクラス Person を抽出した場合, (Actor, dbpp:spouse, Person)

という新たなスキーマトリプルが既存のスキーマに追加される．スキーマ S_D に新たに追加したクラス c が S_D 中の異なる 2 つのクラス c_1, c_2 を論理演算して得た新たな論理クラス $c_1 \diamond c_2$ ($\diamond \in \{\vee, \wedge, -\}$) であるときには，新しいスキーマ $add_{S_D}(c_1 \diamond c_2)$ は

$$\begin{aligned} add_{S_D}(c_1 \diamond c_2) = & S_D \cup \text{modif}_{OT_{S_D}(c_1)}(c_1, c_1 \diamond c_2) \\ & \cup \text{modif}_{OT_{S_D}(c_2)}(c_2, c_1 \diamond c_2) \end{aligned} \quad (4.3)$$

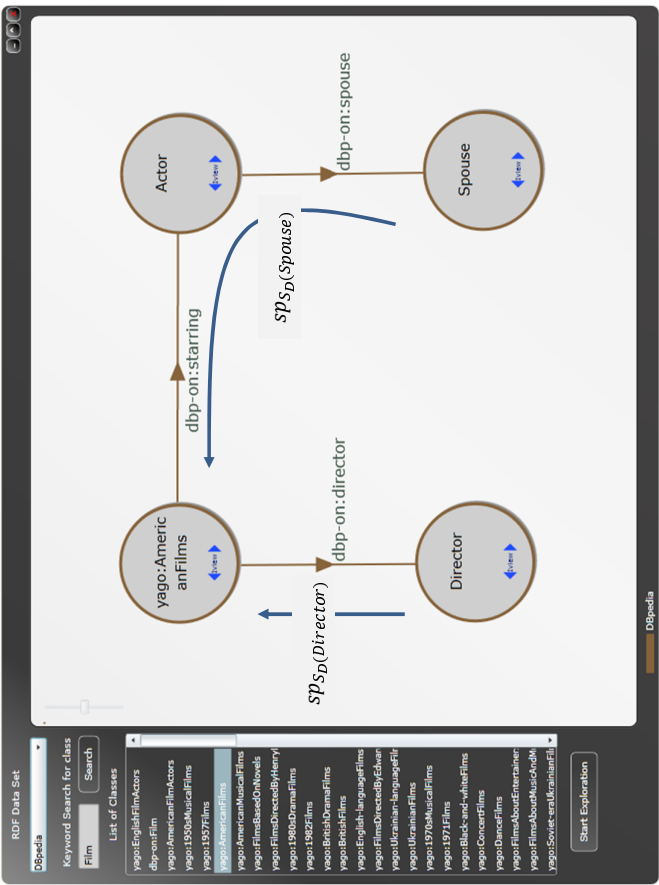
と定義される．ここで， $OT_{S_D}(c)$ はスキーマ S_D 中のクラス c を目的語クラスとするすべてのスキーマトリプルの集合

$$OT_{S_D}(c) = \{(c', p, c) \mid (c', p, c) \in S_D\} \quad (4.4)$$

を表し， $\text{modif}_{ST}(c, c')$ は

$$\begin{aligned} \text{modif}_{ST}(c, c') = & (ST - \{(c, p, o) \mid (c, p, o) \in ST\} \\ & - \{(s, p, c) \mid (s, p, c) \in ST\}) \\ & \cup \{(c', p, o) \mid (c, p, o) \in ST\} \\ & \cup \{(s, p, c') \mid (s, p, c) \in ST\} \end{aligned} \quad (4.5)$$

のようにスキーマトリプルの集合 ST 中のクラス名 c を全てクラス名 c' へと変更することを表す．図 4.4 の例で示すように，ActorsOfAmericanFilms クラスと ActorsOfChineseFilms クラスを論理積演算して，新たなクラス Actor を定義した場合，新たな 2 つのスキーマトリプル (yago:AmericanFilms, dbpp:starring, Actor) と (yago:ChineseFilms, dbpp:starring, Actor) が生成され，既存のスキーマに追加される．スキーマ S_D 中のクラス c のクラス部品に対し，クラス c を上記の操作 (iii) を用いて定義した新たな論理クラス $c \diamond c'$ ，($c' \in \text{Over}C_D(c)$, $\diamond \in \{\vee, \wedge, -\}$) へと変更して得た新たなスキーマは $\text{modif}_{S_D}(c, c \diamond c')$ と記述できる．図 4.5 の例では，Spouse クラスとそのオーバーラップクラスの 1 つである Dancer を論理積演算することを示している．その結果，元のスキーマの中 Spouse クラスは新しいクラス $\text{Spouse} \wedge \text{Dancer}$ と変更される．



スキーマ・トリプル・パス $sp_{SD}(\text{Director})$

Subject class	Relationship property	Object class
yago:AmericanFilms	db-on:director	Director

スキーマ・トリプル・パス $sp_{SD}(\text{Spouse})$

Subject class	Relationship property	Object class
yago:AmericanFilms	db-on:starring	Actors
Actors	db-on:spouse	Spouse

図 4.6: スキーマ・トリプル・パスの例

本論文では、スキーマ S_D の一つまたは複数の始点クラスから、あるクラス c までの全ての経路上のスキーマ・トリプルの集合を、クラス c のスキーマ・トリプル・パスと呼び、 $sp_{S_D}(c)$ と記述する。クラス c が始点クラスであるとき、クラス c のスキーマ・トリプル・パスは

$$sp_{S_D}(c) = \{(null, null, c)\}$$

と定義できる。 $SubC(ST)$ でスキーマトリプルの集合 ST の中の全ての主語クラスの集合を表すとき、始点クラス以外のクラス c のスキーマ・トリプル・パスは

$$sp_{S_D}(c) = OT_{S_D}(c) \cup \bigcup_{c' \in SubC(OT_{S_D}(c))} sp_{S_D}(c')$$

と定義することができる。スキーマ・トリプル・パスの例を図 4.6 に示す。

4.3 スキーマに基づいた検索要求の定義

$C(S_D)$ でスキーマ S_D の中の全てのクラスの集合を表し、 V で変数の集合を表すとき、ESIASW フレームワークでは、スキーマ S_D に対する基本検索要求を、変数の使用を許したトリプル $(c, p, o) (\in (C(S_D) \cup V \cup \{null\}) \times (I \cup V \cup \{null\}) \times (C(S_D) \cup V))$ (以下では、このようなトリプルをスキーマ・トリプル・パターンと呼ぶ) と、 c 及び o に対する述語 $R_1(c)$, $R_2(o)$ で表される 2 つの制約条件とを組み合わせた 3 つ組 $\langle (c, p, o), R_1(c), R_2(o) \rangle$ と、 (c, p, o) に現れる変数の中で結果を返すべき 1 つの変数を表す $v (\in V)$ との組

$$(v, \langle (c, p, o), R_1(c), R_2(o) \rangle) \quad (4.6)$$

で表す。この対において、変数の v のかわりに $null$ を用いた $(null, \langle (c, p, o), R_1(c), R_2(o) \rangle)$ も基本検索要求と定義する。 S_D 上の任意の検索要求は、このような基本検索要求を要素とする集合

$$\{(v_i, \langle (c_i, p_i, o_i), R_1^i(c_i), R_2^i(o_i) \rangle) | 1 \leq i \leq n\} \quad (4.7)$$

で表される。これを $Q(S_D)$ と記述することにする。

4.4 検索要求の評価

検索要求 $Q(S_D)$ の評価 $eval(Q(S_D))$ は

$$eval(Q(S_D)) \triangleq \{(v_1, v_2, \dots, v_n) \mid \bigwedge_i ((C_1(c_i, p_i, o_i) \vee C_2(c_i, p_i, o_i) \vee C_3(c_i, p_i, o_i) \vee C_4(c_i, p_i, o_i) \vee C_5(c_i, p_i, o_i)) \wedge R_1^i(c_i) \wedge R_2^i(o_i))\} \quad (4.8)$$

と定義される. $(v_1, v_2, \dots, v_n)$ 中の m 個 ($m < n$) の v_i が null のとき, $(v_1, v_2, \dots, v_n)$ は null 以外の変数をこの順に並べたリスト $(v_{i_1}, v_{i_2}, \dots, v_{i_{n-m}})$ を表すものとする. $C_1 \sim C_5$ は以下のように定義される.

- $C_1(c, p, o)$

$$\begin{aligned} & \text{iff } c \in C(S_D) \wedge o \in C(S_D) \wedge p \in I \wedge p \neq \text{rdf:type} \wedge ((\forall o_1, o_2 \in C(S_D) \\ & ((o \neq o_1 \vee o_2) \wedge (o \neq o_1 - o_2)) \wedge \exists a \in E_D(c) \exists b \in E_D(o) ((a, p, b) \in D)) \\ & \vee (\exists o_1, o_2 \in C(S_D) (o = o_1 - o_2) \wedge \exists a \in E_D(c) \exists b \in E_D(o) \\ & (((c, p, o_1) \in S_D \wedge (a, p, b) \in D) \vee ((c, p, o_2) \in S_D \wedge (a, p, b) \notin D)))) \quad (4.9) \end{aligned}$$

- $C_2(c, p, o)$

$$\text{iff } c \in C(S_D) \wedge o \in V \wedge \exists a \in E_D(c) ((a, p, o) \in D) \quad (4.10)$$

- $C_3(c, p, o)$

$$\begin{aligned} & \text{iff } c \in V \wedge o \in C(S_D) \wedge ((p \neq \text{rdf:type} \wedge \exists b \in E_D(o) \\ & ((c, p, b) \in D)) \vee (p = \text{rdf:type} \wedge c \in E_D(o))) \quad (4.11) \end{aligned}$$

- $C_4(c, p, o)$

$$\text{iff } c \in V \wedge o \in V \wedge (c, p, o) \in D \quad (4.12)$$

- $C_5(c, p, o)$

$$\text{iff } c = \text{null} \wedge p = \text{null} \wedge o \in C(S_D) \wedge o \in D \quad (4.13)$$

$C_1 \sim C_4$ は、検索要求 $Q(S_D)$ の要素である各基本検索要求中のスキーマ・トリプル・パターン (c, p, o) が満たす条件を、 c と o がそれぞれクラスか変数かいずれかであるかによって、4つの場合に分けて表している。 C_5 は新たな始点クラス o を追加する際に、スキーマに追加するスキーマ・トリプル $(\text{null}, \text{null}, o)$ を基本検索要求と見なす場合を考慮した条件である。 C_1 , つまり、 c も o もクラスの場合には、本フレームワークでは、 o がスキーマ中の異なる2つのクラスを用いて定義した論理和クラス、論理差クラス、それ以外に分けて (c, p, o) が満たす条件を表している。

ESIASW フレームワークでは、自由変数 x がスキーマ中のクラス、つまり $x \in C(S_D)$ であるとき、述語 $R_j^i(x)$ は恒等的に真と定義する。 x が変数、つまり $x \in V$ であるとき、述語 $R_j^i(x)$ は、表 4.1 に示す基本述語とその論理積として記述される。

4.4.1 スキーマの探索的抽出に関する検索要求

ある始点クラスから始めて、探索的に抽出・展開されたスキーマ中の各クラスに対し、ESIASW フレームワークでは、それらのインスタンス、プロパティ、プロパティ値、及びオーバーラップ・クラスの4つの要素を検索することが可能である。クラスのプロパティとオーバーラップ・クラスの検索は、3.2節で述べたように、クラスのクラス部品を生成する際にシステムが自動的に行う。クラス c の関係プロパティ、アトリビュート・プロパティ、およびオーバーラップ・クラスを検索するための検索要求 $Q(S_D)$ は、以下のように記述することができる。

- 関係プロパティ

$$\{(v_1, <(c, v_1, v_2), \text{true}, \text{isUri}(v_2) >)\} \cup \\ \{(\text{null}, <(c_i, p_i, o_i), \text{true}, \text{true}>)|(c_i, p_i, o_i) \in sp_{S_D}(c)\}$$

- アトリビュート・プロパティ

$$\{(v_1, < (c, v_1, v_2), \text{true}, \text{isLiteral}(v_2) >)\} \cup \\ \{(\text{null}, < (c_i, p_i, o_i), \text{true}, \text{true} >)|(c_i, p_i, o_i) \in sp_{S_D}(c)\}$$

- オーバーラップ・クラス

$$\{(v, < (c, \text{rdf:type}, v), \text{true}, \text{true} >)\} \cup \\ \{(\text{null}, < (c_i, p_i, o_i), \text{true}, \text{true} >)|(c_i, p_i, o_i) \in sp_{S_D}(c)\}$$

ここで、各検索要求の第2項は、クラス c のスキーマ・トリプル・パス $sp_{S_D}(c)$ の各要素を基本検索要求へと展開したものである。

4.4.2 スキーマに基づいた探索的検索

ESIASW フレームワークでは、上記の3つの要素の検索と、スキーマ中のクラスのインスタンスやプロパティ値の検索を相互に独立に行う。スキーマ S_D 中の1つまたは複数のクラスのインスタンスやプロパティ値を検索する場合、検索要求 $Q(S_D)$ は、 $S'_D(\subset S_D)$ を制約が課された各クラスのスキーマ・トリプル・パスの和集合とすると、

$$\{(v_i, < (c_i, p_i, o_i), R_1^i(c_i), R_2^i(o_i) >)|1 \leq i \leq k\} \\ \cup \{(\text{null}, < (c_j, p_j, o_j), \text{true}, \text{true} >)|(c_j, p_j, o_j) \in S'_D\} \quad (4.14)$$

と記述することができる。第1項はスキーマ中のクラスのインスタンスに対する制約を表す基本検索要求 $(v, < (v, \text{rdf:type}, c), \text{true}, \text{true} >)$ と、プロパティの値に対する制約を表す基本検索要求 $(v, < (c, p, v), \text{true}, R_2(v) >)$ とを要素とする集合を表し、第2項は、部分スキーマ $S'_D(\subset S_D)$ の各要素を基本検索要求へと展開したものである。

ESIASW フレームワークでは、クラスのクラス部品の周りに可視化されているアトリビュート・プロパティに対する直接操作によって、そのクラスのインスタンスやプロパティ値に対する制約を検索要求に追加ないし削除することができる。図4.7に示すようにクラス部品から必要なアトリビュート・プロパティを選択してクリックすると、検索要求の変数と制約条件を指定できるパネルが表示される。ユーザはパネル上の”Yes”または”No”と表示されたラジオボタンを選択することによって、選択したア

視覚的検索要求

```
{(v1 , <(c1, dbpp:budget,v1) , true , MoreThan(v1,200000)>),
(v2 , <(c2, foaf:givenName ,v2) , true , true >),
(v3 , <(c3, geo:lat ,v3) , true , true >),
(null , <(null , null , c1) , true , true>),
(null , <(c1,dbpp:starring,c2) ,true , true> ) ,
(null , <(c2,dbpp:birthPlace,c3) , true , true> )}
```

制約条件

スキーマ

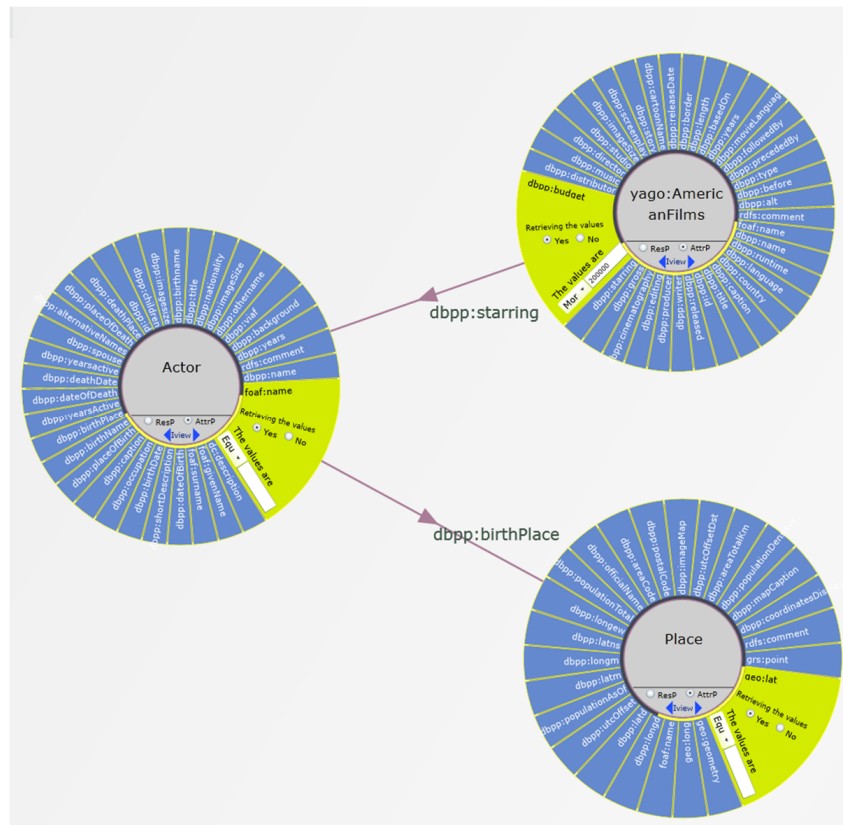


図 4.7: スキーマに基づいた探索的検索の例

トリビュート・プロパティの値を検索結果として返すべき変数として定義するかしないかを定めることができる。またパネル上のドロップダウンリストには、基本述語のリスト {Equal, Contains, MoreThan, LessThan, Between} が提供されており、ユーザはそこから必要な述語を選択し、引数を入力することによってプロパティ値に対する制約条件を指定することができる。ESIASW フレームワークでは、クラスのプロパティ値に対する制約条件を検索要求に追加するとき、そのインスタンスに対する制約をも自動的に追加される。図 4.7 の例では、アメリカ映画の中で興行収入が 2000 万を超えた映画に出演した俳優の名前と出生地を検索する場合の検索要求を指定している。図 4.7 で示すように、定義された検索要求の 1 行から 3 行までは、映画の興行収入、俳優の名前、出生地など値に対する制約を指定し、4 行から 6 行までは、展開したスキーマを基本検索要求に展開した結果を示している。

4.5 SPARQL への自動書き換え

ESIASW フレームワークでは、スキーマに基づいて作成された任意の検索要求は、全て SPARQL へと自動的に書き換えられて実行される。

スキーマ S_D に基づいて作成された検索要求 $Q(S_D)$ を SPARQL に書き換えるとき、検索要求 $Q(S_D)$ の要素となる各基本検索要求 $(v, < (c, p, o), R_1(c), R_2(o) >)$ の中の変数 v は、SPARQL 記述の SELECT 句の SPARQL 変数へと書き換えられ、3 つ組 $< (c, p, o), R_1(c), R_2(o) >$ は、SPARQL 記述の WHERE 句の制約条件へと書き換えられる。SPARQL 変数は、先頭に " ? " を付けた文字列として記述される。SPARQL 記述の WHERE 句の制約条件は、 V_{sparql} で SPARQL 変数の集合を表すとき、SPARQL 変数の使用を許したトリプル $(S, P, O) (\in (I \cup B \cup V_{sparql}) \times (I \cup V_{sparql}) \times (I \cup B \cup L \cup V_{sparql}))$ と、(.), UNION, OPTIONAL, MINUS などの演算子によって記述される RDF グラフ・パターンと、FILTER キーワードを用いて表す制約条件の集合の組み合わせとして記述される。

表 4.1: 基本述語の定義

基本述語	定義
IsUri(x)	$x \in I \cup B$ のときのみ真
IsLiteral(x)	$x \in L$ のときのみ真
Equal(x, y)	$x, y \in L$ で $x = y$ のときのみ真
Unequal(x, y)	$x, y \in L$ で $x \neq y$ のときのみ真
Contains(x, y)	$x, y \in L$ で x が y を含む文字列のときのみ真
MoreThan(x, y)	$x, y \in L$ で $x > y$ のときのみ真
LessThan(x, y)	$x, y \in L$ で $x < y$ のときのみ真
Between(x, y, z)	$x, y, z \in L$ で, $y < x < z$ のときのみ真
OneOf(x, Y)	$x \in I \cup L, Y \subset I \cup L$ で $x \in Y$ のときのみ真

表 4.2: クラスに対する全ての書き換え規則

	クラス c	$t_{class}(c, num(c))$
r1	c が既存のクラスである	$(?num(c) \text{ rdf:type } c.)$
r2	$c = c'.p$	-
r3	$c = c_1 \wedge c_2$	$t_{class}(c_1, num(c))$ AND $t_{class}(c_2, num(c))$
r4	$c = c_1 \vee c_2$	$\{t_{class}(c_1, num(c))\}$ UNION $\{t_{class}(c_2, num(c))\}$
r5	$c = c_1 - c_2$	$t_{class}(c_1, num(c))$ MINUS $\{t_{class}(c_2, n)\}$

表 4.3: スキーマ・トリプル・パターンに対する全ての書き換え規則

	スキーマ・トリプルパターン (c, p, o)	$t_{triple}((c, p, o))$
r6	$C_1(c, p, o) \wedge \forall o_1, o_2 \in C(S_D)((o \neq o_1 \vee o_2) \wedge (o \neq o_1 - o_2))$	$(?num(c) \ p \ ?num(o).) \text{AND} \ t_{class}(c, num(c)) \ \text{AND} \ t_{class}(o, num(o))$
r7	$C_1(c, p, o) \wedge \exists o_1, o_2 \in C(S_D)(o = o_1 \vee o_2) \wedge (\exists(c', p', o) \in S_D(C_1(c', p', o)))$	$\{(?num(c') \ p' \ ?num(o).) \text{AND} \ t_{class}(c', num(c')) \ \text{AND} \ t_{class}(o, num(o))\} \text{UNION} \{(?num(c) \ p \ ?num(o).) \text{AND} \ t_{class}(c, num(c))\}$
r8	$C_1(c, p, o) \wedge \exists o_1, o_2 \in C(S_D)((o = o_1 - o_2) \wedge (c, p, o_1) \in S_D)$	$(?num(c) \ p \ ?num(o).) \text{AND} \ t_{class}(c, num(c)) \text{AND} \ t_{class}(o_2, num(o))$
r9	$C_1(c, p, o) \wedge \exists o_1, o_2 \in C(S_D)(o = o_1 - o_2) \wedge (c, p, o_2) \in S_D$	$\text{MINUS}\{\text{AND}_{t \in sps_D} t_{triple}(t) \text{AND} (?num(c)p?num(o).) \text{AND} \ t_{class}(c, num(c)) \text{AND} \ t_{class}(o, num(o))\}$
r10	$C_2(c, p, o) \wedge p \in I$	$(?num(c) \ p \ ?ni_p.) \text{AND} t_{class}(c, ?num(c))$
r11	$C_2(c, p, o) \wedge p \in V$	$(?num(c) \ ?pro \ ?o.) \text{AND} t_{class}(c, ?num(c))$
r12	$C_3(c, p, o) \wedge p \in V$	$(?s \ ?pro \ ?num(o).) \text{AND} \ t_{class}(o, num(o))$
r13	$C_3(c, p, o) \wedge p \in I \wedge p \neq \text{rdf:type}$	$(?s \ p \ ?num(o).) \text{AND} \ t_{class}(o, num(o))$
r14	$C_3(c, p, o) \wedge p = \text{rdf:type}$	$t_{class}(o, num(o))$
r15	$C_4(c, p, o) \wedge p \in I$	$(?s \ p \ ?o.)$
r16	$C_4(c, p, o) \wedge p \in V$	$(?s \ ?pro \ ?o.)$
r17	$C_5(c, p, o)$	$t_{class}(o, num(o))$

表 4.4: 基本述語から SPARQL 制約条件への書き換え規則

	基本述語 $R_j^i(x)$	$t_{predicate}(R_j^i(x))$
r18.	IsUri(x)	FILTER(IsUri(?n))
r19.	IsLiteral(x)	FILTER(IsLiteral(?n))
r20.	Equal(x, y)	FILTER(?n= y)
r21.	Unequal(x, y)	FILTER(?n! y)
r22.	Contains(x, y)	FILTER regex(str(?n), " y ")
r23.	MoreThan(x, y)	FILTER (?n > y)
r24.	LessThan(x, y)	FILTER (?n < y)
r25.	Between(x, y, z)	FILTER (?n > y & & ?n < z)
r26.	OneOf(x, Y)	FILTER (?n= y_1 ... ?n= y_n)
	$Y = \{y_1, \dots, y_n\}$	
r27.	true	-

ESIASW フレームワークでは、基本検索要求の中の変数 v が、クラス c のインスタンス、プロパティ、ある 1 つのプロパティ p の値、オーバーラップ・クラスの 4 つの要素のいずれかを表す変数であるとき、対応する各々の SPARQL 変数はそれぞれ " $?num(c)$ ", " $?pro$ ", " $?num(c)_p$ ", " $num(c)_type$ " と記述することができる。ここで、 $num(c)$ はスキーマを探索的に抽出・展開する際に、スキーマの中でクラス c が何番目に抽出されたかを示す番号を表す。

ESIASW フレームワークでは、基本検索要求の中のスキーマ・トリプル・パターン (c, p, o) は、表 4.3 に示す書き換え規則によって、RDF グラフ・パターンへと書き換えられる。その書き換え結果を $t_{triple}((c, p, o))$ と記述する。表 4.3 に現れる $t_{class}(c, num(c))$ は、スキーマ・トリプル・パターン中のクラスを表 4.2 に示す書き換え規則によって、RDF グラフ・パターンへと書き替えたものである。ここで、表 4.3 と表 4.2 の SPARQL 記述中の AND は、二つの RDF グラフ・パターンの直接連結を表す。($?n1 ?p1 ?n2$).AND($?n2$

視覚的検索要求

- | | |
|------|---|
| I. | $\{(v_1, \langle (c_1, \text{dbpp:budget}, v_1), \text{true}, \text{MoreThan}(v_1, 200000) \rangle),$ |
| II. | $(v_2, \langle (c_2, \text{foaf:givenName}, v_2), \text{true}, \text{true} \rangle),$ |
| III. | $(v_3, \langle (c_3, \text{geo:lat}, v_3), \text{true}, \text{true} \rangle),$ |
| IV. | $(\text{null}, \langle (\text{null}, \text{null}, c_1), \text{true}, \text{true} \rangle),$ |
| V. | $(\text{null}, \langle (c_1, \text{dbpp:starring}, c_2), \text{true}, \text{true} \rangle),$ |
| VI. | $(\text{null}, \langle (c_2, \text{dbpp:birthPlace}, c_3), \text{true}, \text{true} \rangle)\}$ |



SPARQL

SELECT ?n1_budget, ?n2_name, ?n3_lat	行Iの中の $v_1 \rightarrow ?n1_budget$ 行IIの中の $v_2 \rightarrow ?n2_name$ 行IIIの中の $v_3 \rightarrow ?n3_lat$
WHERE{ ?n1 dbpp:budget ?n1_budget. FILTER (?n1_budget>200000) ?n2 foaf:givenName ?n2_name. ?n2 geo:lat ?n2_name. ?n1 rdf:type yago:AmericanFilms. ?n1 dbpp:starring ?n2. ?n2 dbpp:birthPlace ?n3. }	行I: 表4.3の規則r10を参照 行I: 表4.4の規則r23を参照 行II: 表4.3の規則r10を参照 行III: 表4.3の規則r10を参照 行IV: 表4.3の規則r17を参照 行V: 表4.3の規則r6を参照 行VI: 表4.3の規則r6を参照

図 4.8: 視覚的検索要求から SPARQL への書き換えの例

?p2 ?n3.) は, 実際の SPARQL 記述の中では (?n1 ?p1 ?n2. ?n2 ?p2 ?n3.) と記述される.

基本検索要求中の述語 $R_i(x)$ は, 表 4.4 に示す書き換え規則によって, SPARQL 記述の制約条件へと書き換えられ, その書き換え結果を $t_{predicate}(R_j^i(x))$ と定義する. これにより, 基本検索要求の中の 3 つ組 $\langle (c, p, o), R_1(c), R_2(o) \rangle$ から SPARQL への書き換え $t_{bq}(\langle (c, p, o), R_1(c), R_2(o) \rangle)$ は

$$t_{bq}(\langle (c, p, o), R_1(c), R_2(o) \rangle) = t_{triple}((c, p, o))$$

$$\text{AND } t_{predicate}(R_1(c)) \text{ AND } t_{predicate}(R_2(o))$$

と定義される. ESIASW フレームワークでは, 検索要求 $Q(S_D)$ の中の各 3 つ組 $\langle$

$(c_i, p_i, o_i), R_1^i(c_i), R_2^i(o_i) >$ を全て SPARQL 記述の WHERE 句の制約へと書き換えたとき, その書き換え結果 $t(Q(S_D))$ は

$$t(Q(S_D)) = \text{AND}_i t_{bq}(<(c_i, p_i, o_i), R_1^i(c_i), R_2^i(o_i)>)$$

と定義される. 視覚的に指定した検索要求を SPARQL 記述へと自動的に書き換えた例を図 4.5 に示す.

第 5 章

セマンティック・ウェブ上のリソースの統合アクセス

Intgration of Semantic Web Resources

5.1 概要

セマンティック・ウェブ上に集積された膨大なデータ集合や多種のサービスは、多様なデータを組み合わせることで、データの創発的な価値を生み出すことを可能にする。セマンティック・ウェブにおける多様なデータやサービスの組み合わせ統合は、以下の4つの種類に分けられる。

- a) 同一 RDF データセット中での多様なデータの組み合わせ統合
- b) 異なる RDF データセット間の多様なデータの組み合わせ統合
- c) 異なるサービス間の多様なデータの組み合わせ統合
- d) RDF データセットとサービス間の多様なデータの組み合わせ統合

ここで、a) と b) の場合は統合しようとするデータ集合のスキーマに関する知識が必要となり、特に b) の場合、使用されている語彙も異なることが多く、異なるデータ集合の組み合わせ統合は困難である。また、c) と d) の場合は、サービスに関する知識や高いプログラミング能力が必要となる。ESIASW フレームワークでは、同一作業空間

内で、異なる複数の RDF データスキーマを展開・可視化したり、サービス部品を用いて必要なセマンティック・ウェブ・サービスのプロキシを作成することを可能にしている。ユーザは展開・可視化された RDF データスキーマを意味的関連や論理的関連に基づいて 1 つのスキーマに統合して利用したり、異なるサービス部品を直接結合したり、RDF データスキーマとサービス部品を意味的関連に基づいて直接結合したりすることで、多様なデータやサービスを自在に組み合わせ統合することができる。

5.2 異なるスキーマ間の連携統合

ESIASW フレームワークでは、同一または異なる RDF データセットから探索的に抽出・展開・可視化した RDF データスキーマの連携統合を、クラス間の論理演算による連携統合と、同じ意味を持つクラスのアトリビュートプロパティを直接結合することによる連携統合の 2 種類に分ける。

5.2.1 論理的関係による異なるスキーマ間の連携統合

同一 RDF データセットから探索的に抽出・展開・可視化した異なる RDF データスキーマに対し、ユーザは、必要に応じて異なるスキーマの中のクラスを直接組み合わせ論理演算することによって、異なるスキーマを 1 つのスキーマとして統合利用することができる。

図 5.1 の例で示すように、スキーマの異なる 2 つのクラス c_1 , c_2 のクラス部品 (図 5.1 の (1) と (2) で示す Actor) のコピーをドラッグして接近させると、図 5.1 の (3) で示すように論理演算子の選択パネルがポップアップされ、その中から必要な論理演算子を選択することによってこの演算を実現することができる。両 RDF データスキーマ、定義された新しい論理クラス $c_1 \diamond c_2$, ($\diamond \in \{\wedge, \vee, -\}$) (図 5.1 の (4) で示す Actor) によって、1 つの RDF データスキーマとして統合される。

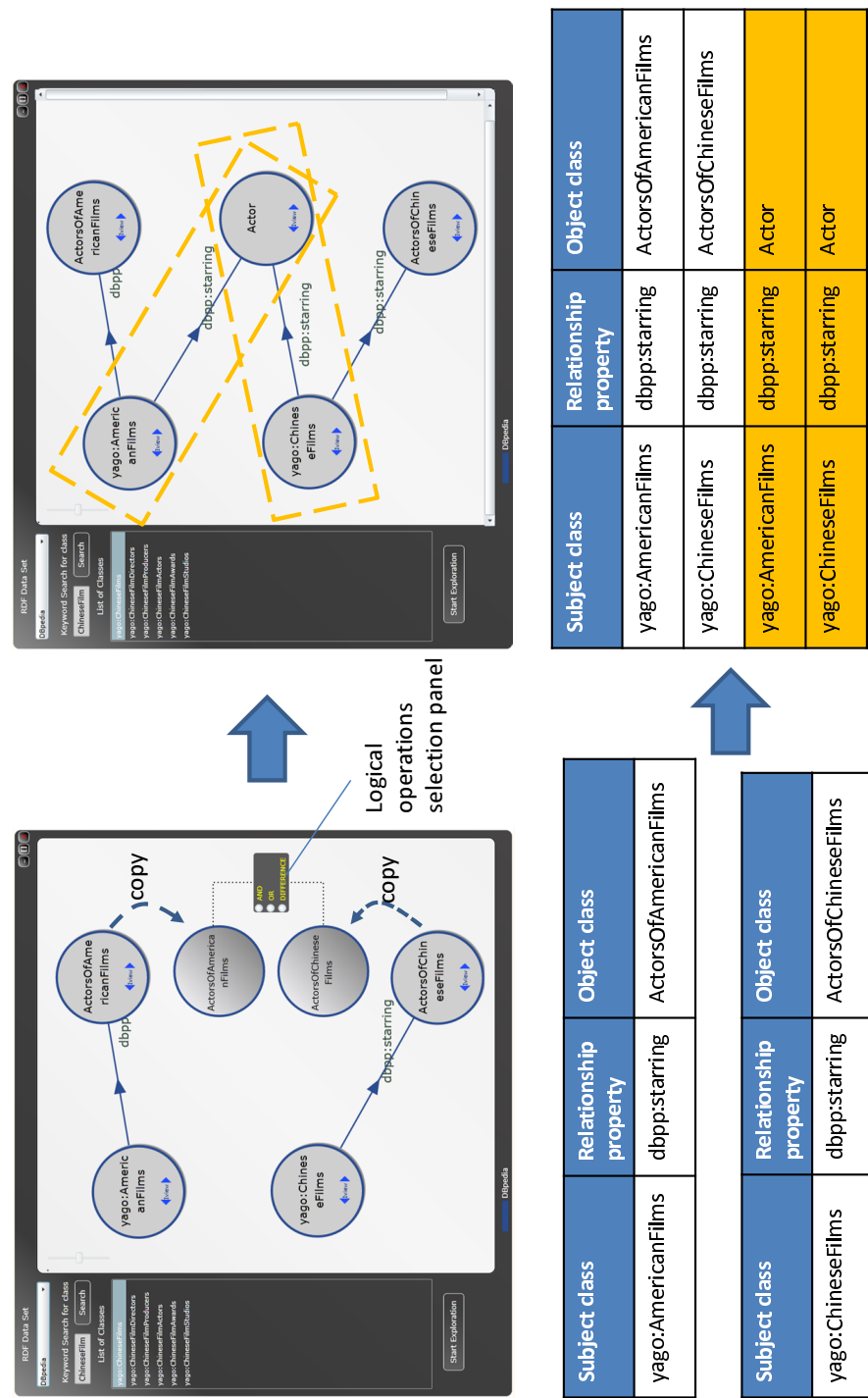


図 5.1: 論理的关系に基づいた異なるスキーマの統合

5.2.2 意味関係に基づいて異なるスキーマ間の連携統合

同一 RDF データセットから抽出・展開した異なる複数のスキーマに対し、ESIASW フレームワークでは、図 4.1 の例で示すように、異なるスキーマ中の 2 つのクラスの間に論理演算を適用することによって、これらのスキーマを統合することができる。しかし、検索対象とする RDF データセットが異なる場合には、使用されている語彙も異なることがあり、クラス間の論理演算が適用できなくなる。ESIASW フレームワークでは、異なる RDF データセットから抽出・展開した複数のスキーマに対し、図 5.2 の例で示すように、異なるスキーマ中の 2 つのクラスの対応するプロパティを相互に線で結んで結合することによって、両スキーマの指定したプロパティの間にプロパティ値を両方向に伝送できるチャンネルを構築することができる。異なるプロパティ間の相互結合の構築は、ユーザが両方のスキーマから結合したいクラスのアトリビュート・プロパティを選択して Shift+Click の操作をすることで実現できる。結合された両スキーマのいずれかを直接操作して、そのスキーマに基づいた新たな検索要求を定義すると、そのスキーマの中の結合操作を行ったクラスのアトリビュート・プロパティの値が自動的に検索され、構築されたチャンネルを通じてもう一方のスキーマの結合操作対象のクラスのアトリビュート・プロパティに伝送される。検索結果を受信したもう一方のスキーマのクラスは、それを結合に用いたアトリビュート・プロパティの制約条件として用いて新たな検索要求を定義することができる。図 5.2 の例では、DBpedia から抽出した yago:AmericanFilms クラスの映画のタイトルを表す”foaf:name”プロパティと、Linked Movie DataBase から抽出した film クラスの映画のタイトルを表す”dc:title”プロパティを結合し、yago:AmericanFilms クラスのインスタンスである映画の記述には含まれてない coustume_designer を検索しようとしている。図 5.2 の (1) で示すように、ユーザが DBpedia の yago:AmericanFilms クラスのあるインスタンスを選択すると、そのインスタンスの”foaf:name”プロパティの値が Linked Movie DataBase の film クラスに伝送され、”dc:title”プロパティの制約条件として、coustume_designer クラスのインスタンスの検索を行う (図 5.2 の (2))。

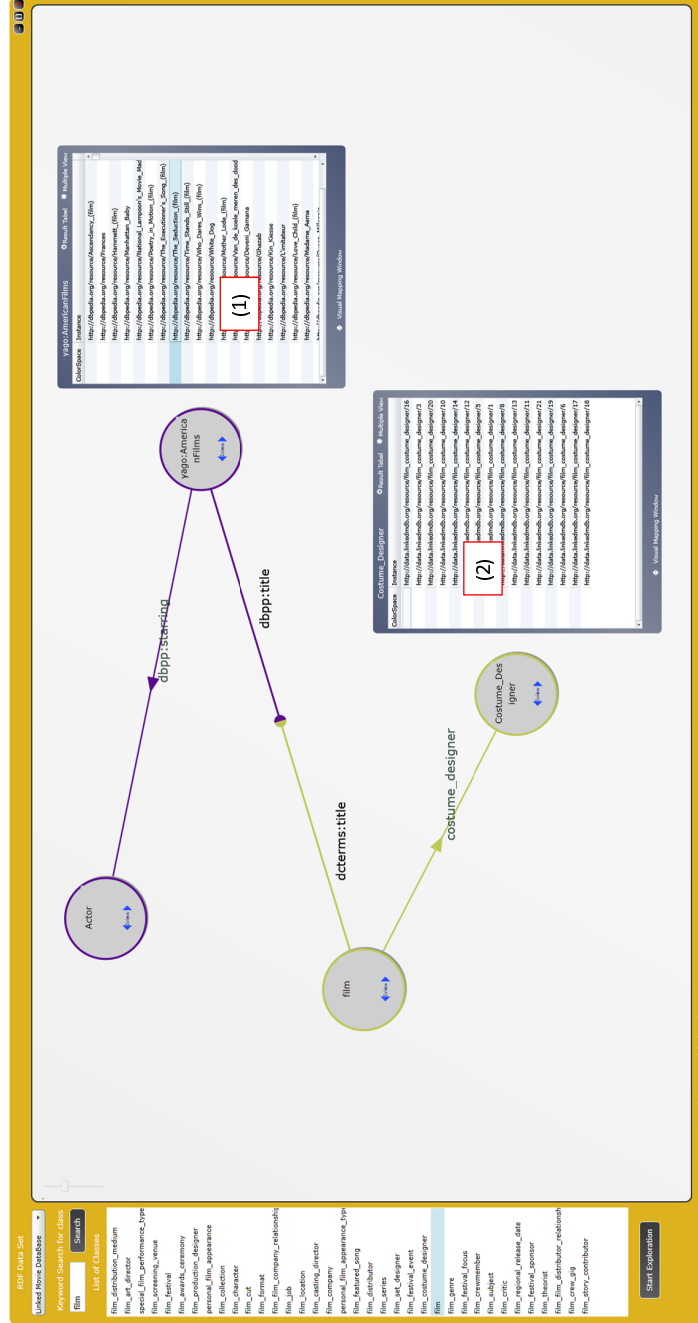


図 5.2: 異なるスキーマの組み合わせによる統合スキーマの定義

5.3 セマンティック・ウェブ・サービス間の連携統合

セマンティック・ウェブでは、データだけでなく、ウェブサービスの記述にもメタデータを付加して、機械による意味関係に基づいたウェブサービス検索、統合、実行を可能にしている。近年、ウェブ上に公開されているセマンティック・ウェブ・サービスの種類は急速に増加している。iServeでは、RDFに基づいて各サービスの記述を行い、その結果をLODとしてウェブ上に公開している。また、iServeでは、SPARQL endpointを提供することによって、SPARQLを用いて必要とするセマンティック・ウェブ・サービスを検索したり、新しいウェブサービスを登録したりすることを可能にしている。

図 5.3 に iServe におけるセマンティック・ウェブ・サービスの記述例を示す。この例で示すサービスは、LastFmEvents という指定した都市での開催予定の音楽イベントを検索するサービスである。LastFmEvents は図 5.3 に示すようにオペレーションに関する 1 つの記述 LastFmGetEventsOperation を持ち、LastFmGetEventsOperation はまた 3 つの入力に関する記述 Long, Lat, distance と、6 つの出力に関する記述 lat, long, Musical Event, Venue, Place, Date, Time を持つ。ESIASW フレームワークでは、図 5.4 に示すように iServe からセマンティック・ウェブ・サービスの記述に関する RDF データのスキーマを探索的に抽出・展開・可視化することができる。ユーザは Service クラスを始点としてセマンティック・ウェブ・サービスのオペレーションクラスや入力、出力クラスを順番に抽出することができる。図 5.4 に示すように、ユーザは抽出・展開したスキーマを用いて、サービスの入力に関する記述と出力に関する記述を直接指定することによって必要なサービスを検索することができる。この例では、入力を geo:long, geo:lat と指定し、出力を Musical Event と指定することによって、指定した場所で開催を予定している音楽イベントに関するサービス Geo.GetEvent という 1 つのサービスを発見している。発見されたセマンティック・ウェブ・サービスを識別する URI をクリックすると、新しいサービス部品が生成され、以下のような SPARQL 検索要求文を用いて選択したサービスの入力や出力等のパラメーターを自動的に検索

LastFmEvents		
	type	Service
	isDefinedBy	hrest.html
	label	LastFm Events
	source	hrest.html
	hasOperation	LastFmGetEventsOperation
LastFmGetEventsOperation		
	type	Operation
	label	geo.getEvents
	hasAddress	http://ws.audioscrobbler.com/2.0/? method=geo.getevents&distance={ d istance }&lat={ lat }&long={ long }
	hasInput	LastFmGetEventsInput
	hasMethod	GET
	hasOutput	LastFmGetEventsOutput
LastFmGetEventsInput		
	type	MessageContent
	loweringSchemaMapping	lastfm-events-get-lowering.xsparql
	modelReference	Long
	modelReference	lat
	modelReference	distance
LastFmGetEventsOutput		
	type	MessageContent
	liftingSchemaMapping	lastfm-events-get-lifting.xsparql
	modelReference	lat long
	modelReference	Musical Event
	modelReference	Venue
	modelReference	Place
	modelReference	Date
	modelReference	Time

図 5.3: iServe におけるセマンティック・ウェブ・サービスの記述例

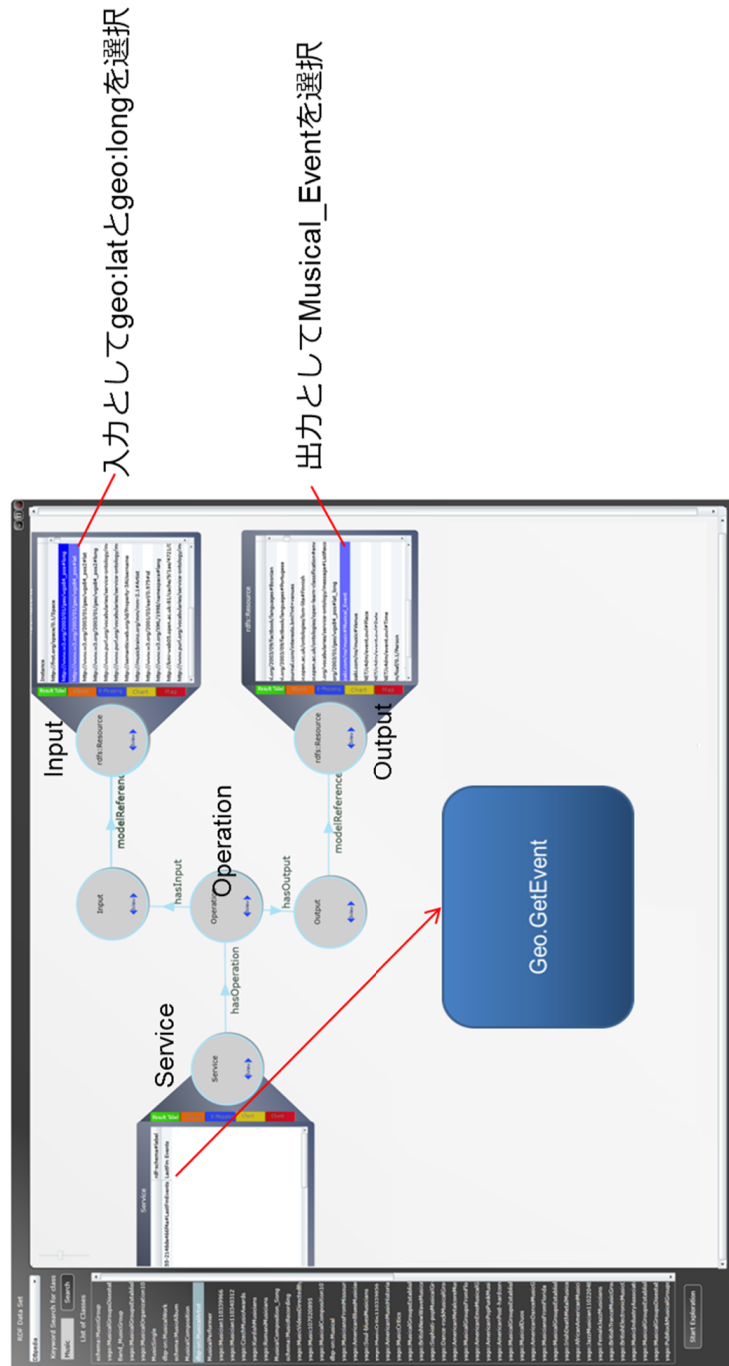


図 5.4: セマンティック・ウェブ・サービスの検索

する.

```

SELECT    distinct ?input ?output ?RestURITemplate
WHERE     {< URI >  _ : hasOperation  ?Operation.
           ?Operation _ : hasInput    ?in.
           ?Operation _ : hasOutput   ?out.
           ?in _ : modelReference  ?input.
           ?out _ : modelReference  ?output.
           ?Operation _ : hasAddress ?RestURITemplate.
        }

```

(5.1)

ここで、RestURITemplate はサービスリクエストのテンプレートを示す。

ESIASW フレームワークでは、図 5.5 に示すように、生成された複数のサービス部品の入出を直接連携することによって、異なるセマンティック・ウェブ・サービスを連携統合することができる。この例では、Geo.GetEvent サービスの Artist という出力を、GetAlbum サービスの同じ意味を持つ入力 Artist に結合することによって、指定した場所で開催される音楽イベントのアーティストのアルバムに関する情報の検索を可能にしている。

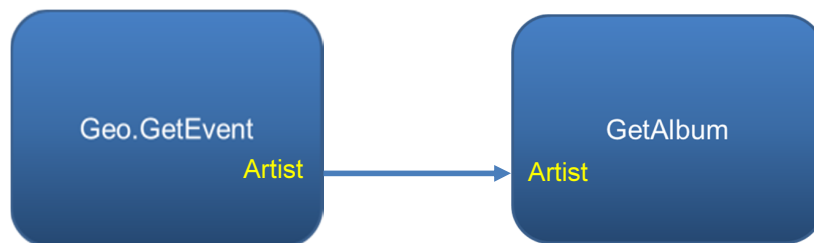


図 5.5: セマンティック・ウェブ・サービス間の連携統合

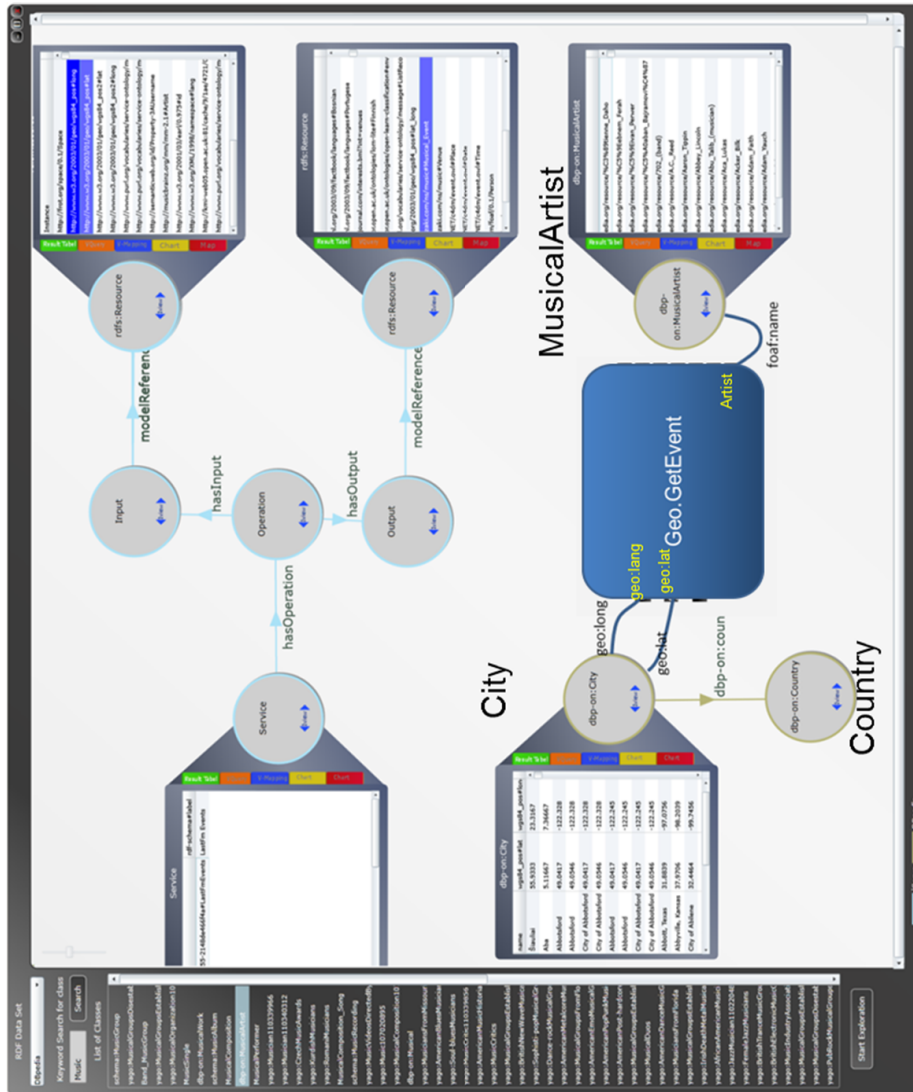


図 5.6: スキーマとセマンティック・ウェブ・サービスの統合

5.4 スキーマとセマンティック・ウェブ・サービスの連携統合

ESIASW フレームワークでは、サービス部品の入出力と展開された RDF データのスキーマのアトリビュートプロパティを直接結合することによって、セマンティック・ウェブ上のデータ集合とサービスを連携統合してアクセスすることができる。

図 5.6 にセマンティック・ウェブ・サービスと RDF データスキーマを連携統合する例を示す。この例では、図 5.4 の例で示した Geo.GetEvent サービスに対し、DBpedia から都市 (City) やその国 (Country) に関するスキーマを抽出し、スキーマの中の City クラスの 2 つアトリビュートプロパティ”geo:long”と”geo:lat”を Geo.GetEvent サービスの 2 つの入力”geo:long”と”geo:lat”に直接結合することで、検索した都市の経緯度を、Geo.GetEvent サービスの入力とすることができる。これにより、ユーザはスキーマ側で検索された都市のリストから興味のある都市を選択しながら、そこで開催予定となっている音楽イベントを簡単に検索することができる。さらに、DBpedia からミュージックアーティスト (MusicalArtist) クラスを抽出し、そのプロパティ”foaf:name”と、Geo.GetEvent サービスの出力”Artist”を直接結合することによって、開催する音楽イベントのミュージックアーティストに関する詳細な情報を検索することができる。

第 6 章

セマンティック・ウェブ上のリソースの可視化分析

Visual analysis of semantic web resources

6.1 概要

セマンティック・ウェブ上のデータやサービスの分析を行う従来の研究の多くは [30, 18], 特定の RDF データセットの一部に対し, その中のリソース間の潜在的な意味的关系を探し出すデータ構造自体に対する分析を行っていた. しかし, 近年急速に増加する様々な分野の RDF データは, それらを有効に扱うことによって実世界の課題解決に有用な新たな知見を発見したいという要求が高まってきている. 膨大なデータの分析において, 従来のデータベース分野では, 多様な視点から探索的に可視化分析を行う研究が多くなされていた [19, 5, 11]. これらは, 多様な視点から探索的に可視化できる多重連携ビュー技術を用いて, 膨大なデータの色々な側面から可視化を行い, それを直接操作することによって, データをフィルタリングしながら, 探索的な可視化分析を行うことを可能にしている. しかし, これらの従来の手法は事前にスキーマを把握し, それに基づいてデータの分析を行う必要があり, スキーマの全体像を予想できない RDF データには適用することが困難である.

ESIASW フレームワークでは, RDF データのスキーマを探索的に抽出・展開しながら, それに基づいて多重連携ビューを動的に構築することを可能にすることによって,

多重連携ビューを用いた多視点からの探索的可視化分析をセマンティック・ウェブ上のリソースの分析に適用することができた。ユーザは構築された多重連携ビューの上でマルチ選択、つまり、複数のデータ集合を同時に選択することができる。ESIASW フレームワークでは、多重連携ビュー上の各選択操作に独自の色を割り振り、その結果を Brushing&Linking 技術と色の組み合わせ技術を用いて表すことによって、異なるデータ集合を相互比較したり、それらの間の相関関係を探し出したりすることができる。

6.2 多重連携ビューの動的構築

多重連携ビューの構築において、各可視化分析部品は、図 6.1 に示すように可視化表現を定義するためのインタフェースを提供する。このインタフェースでは、図 6.1(1) に示す検索結果を基本チャートへとマッピングするためのパネル、図 6.1(2) に示す検索結果を地図上ポイントへとマッピングするためのパネル、図 6.1(3) に示す検索結果を平行座標へとマッピングするためのパネルの 3 種類の可視化表現定義パネルを提供する。これらの各パネルでは、制約が課されたアトリビュート・プロパティ名の集合をドロップダウンリストとして表しており、ユーザはそこから必要なものを直接選択することによって可視化表現の定義を行うことができる。基本チャートマッピングパネルにおいて、ユーザは、図 6.1(1) に示すように、“X_Axis”ドロップダウンリストと、“Y_Axis”ドロップダウンリストから任意の 2 つの属性を選択することによって、これらの値をヒストグラムや分散図の X 軸と Y 軸へとマッピングすることができる。地図マッピングパネルにおいて、ユーザは、図 6.1(2) に示すように、“Long”ドロップダウンリストと、“Lat”ドロップダウンリストから任意の 2 つの属性を選択することによって、検索された地名を地図上の対応するポイントへとマッピングすることができる。平行座標マッピングパネルにおいて、ユーザは、図 6.1(3) に示すように、“add new attribute”ボタンをクリックすることによって、必要な軸を定義するためのドロップダウンリストを逐次に追加し、追加されたドロップダウンリ

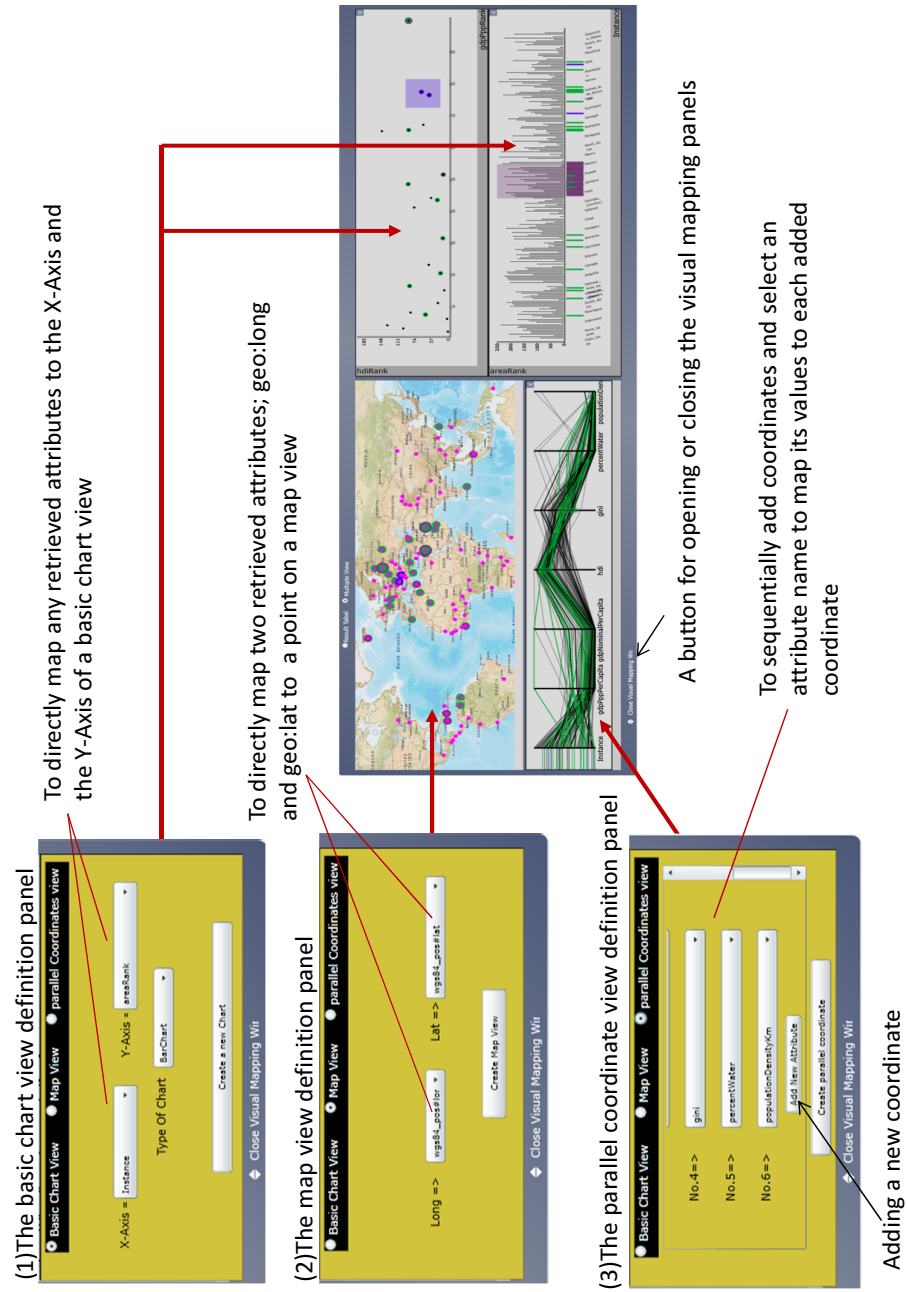


図 6.1: 可視化表現の定義

ストから必要な属性を選択することによって、その値を追加された軸にマッピングすることができる。

ESIASW フレームワークでは、探索的に抽出・展開・可視化された RDF データスキーマに基づいて作成された検索要求の評価結果を、ワークスペースで保持すると同時に、そのコピーをスキーマ中の対応する各クラス部品に分けて保持する。ESIASW フレームワークでは、可視化分析部品を探索的に抽出・展開・可視化した RDF データスキーマの中の各クラス部品と直接結合し、そのクラスのプロパティ値を用いて多重連携ビューを構築したり、またはワークスペース部品と直接結合し、スキーマの中の異なる複数のクラスのプロパティ値の組み合わせを用いて多重連携ビューを構築したりすることができる。これにより、ユーザは、スキーマの中のクラス間の意味的关系を参照しながら、各クラスのインスタンスの詳細について、相互比較したり、異なるクラスのプロパティの組み合わせから潜在的情報を発見したりすることができる。

可視化分析部品を用いて多重連携ビューを構築する例を 6.2 に示す。この例では、会社を表す Company クラス (図 6.2(a)) とその所在地を表す City クラス (図 6.2(b)) を含むスキーマを探索的に抽出・展開可視化し、それに基づいて検索した様々プロパティ値を用いて多重連携ビューを構築している。図 6.2 の (c) と (d) では、抽出・展開・可視化されたスキーマの中の各クラスに可視化分析部品を直接結合し、それらのプロパティ値を可視化している。図 6.2 の (c) では、Company クラスの営業利益 (Operating income), 当期純利益 (Net income), 売上高 (Revenues) の 3 つのプロパティ値を、それぞれ異なるヒストグラムビューとして可視化している。図 6.2 の (d) では、City クラスのインスタンスである各都市の位置を地図上に可視化し、人口をヒストグラムビューとして可視化し、毎月の平均気温をパラレルコーディネートビューとして可視化している。この場合、各ビュー v の中の視覚的オブジェクトに保持されているリソースの集合 $R_{o(v)}$ は、

$$R_{o(v)} = \{r_o(c)\} \quad (6.1)$$

のように単一要素の集合となる。ここで、 c は Company クラスか City クラスのいずれ

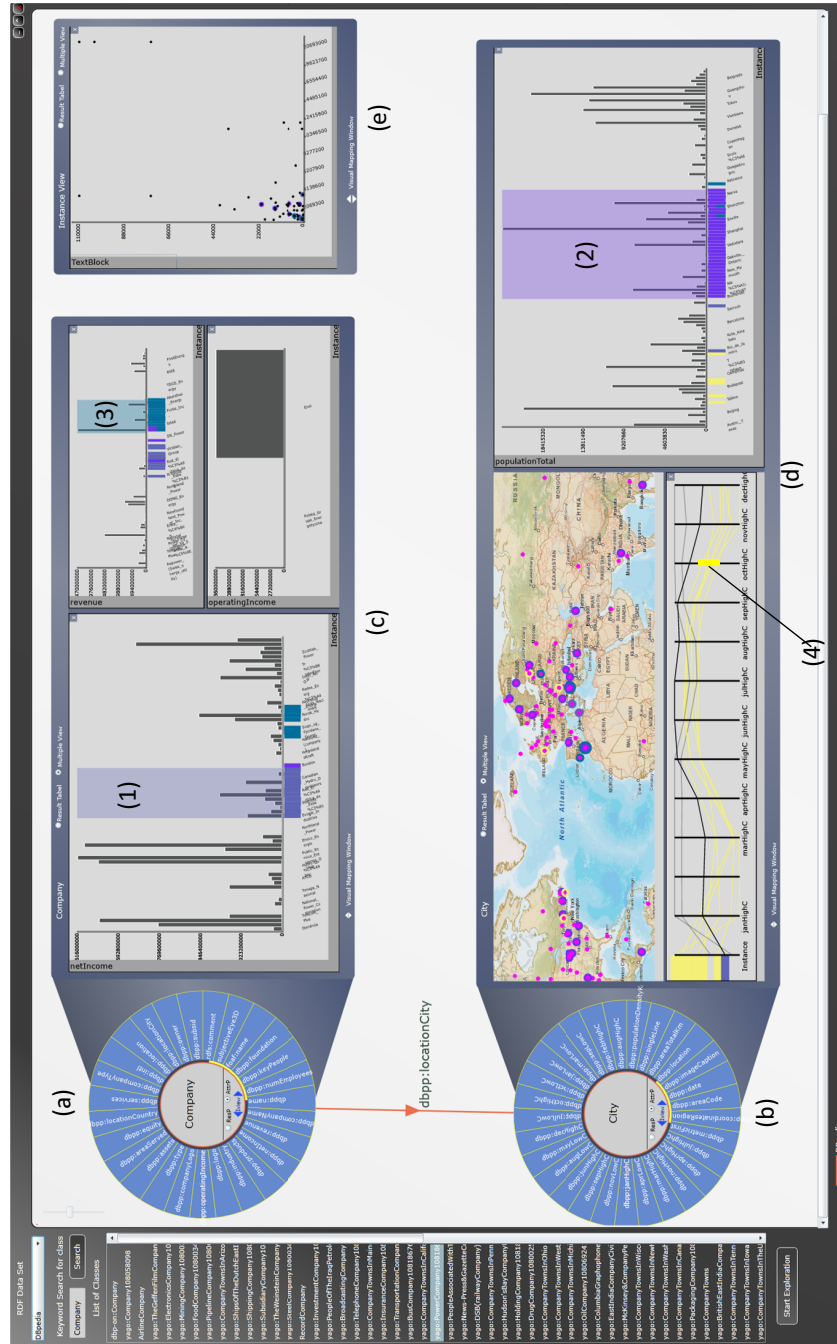


図 6.2: 可視化分析部品を用いた多重連携ビューの構築の例

れかの1つを表す.

図 6.2 の (e) では, 可視化分析部品をワークスペースと結合し, 検索結果の中の異なるクラスのプロパティの組み合わせを用いて構築した多重連携ビューの例を示す. この例では, Company クラスのインスタンスである各会社の従業員数 (Number of employees) プロパティの値を X 軸とし, City クラスのインスタンスである各都市の人口プロパティの値を Y 軸とする分散図ビューを構築している. これにより, 各都市で就職率の向上に対する各会社の貢献を調べることができる.

6.3 多重連携ビュー上のマルチ選択

ESIASW フレームワークでは, 構築された多重連携ビューの上で, 興味のある複数の視覚的オブジェクトの集合を同時に選択できるマルチ選択 (Multiple selection) を可能にしている. 各選択操作に対し, 可視化分析部品は選択された視覚的オブジェクトに保持されているリソース集合を結合しているクラス部品またはワークスペース部品に伝送する. クラス部品またはワークスペース部品は, 受け取ったリソースの集合を制約条件として, 新たな検索要求を定義する. 本論文では, 各ビュー v 上で直接選択した視覚的オブジェクトの集合を $O(v)$ と記述し, $O(v)$ の中の視覚的オブジェクトに保持されている全てのリソースの集合を $R_{O(v)}$ と定義する.

$$R_{O(v)} = \bigcup_{o(v) \in O(v)} R_{o(v)} \quad (6.2)$$

探索的抽出・展開・可視化された RDF データスキーマを S としたとき, 各ビュー v を直接操作することによって作成された検索要求 Q を以下のように定義する.

$$\begin{aligned} & SELECT \ distinct \ \bigwedge_{c \in S} ?num(c) \\ & WHERE \{ GP \wedge FILTER(\|_{r_o(c) \in R_{O(v)}} ?num(c) = r) \}, \end{aligned} \quad (6.3)$$

ここで, GP は RDF データスキーマ S を表すグラフパターンである. 定義された検索要求 Q の評価結果は, 制約条件を満たすスキーマの中の各クラスのインスタンス集

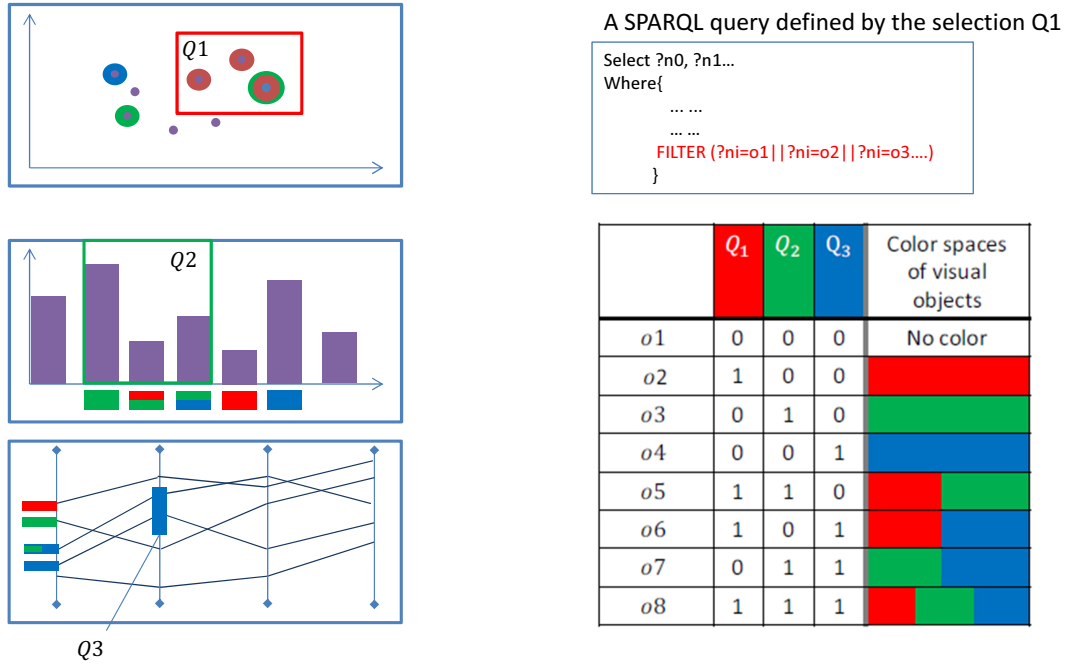


図 6.3: 色の組み合わせ規則

合の直積である。選択操作により定義された検索要求 Q の評価結果を $eval(Q)$ としたとき、任意のビュー v の中の任意の視覚的オブジェクト $o(v)$ に対し、視覚的オブジェクト $o(v)$ の中に保持しているリソース集合 $R_{o(v)}$ と $eval(Q)$ の積集合が空集合ではない場合、つまり $eval(Q) \cap R_{o(v)} \neq \phi$ が満たされると、視覚的オブジェクト $o(v)$ をこの選択操作の結果と呼ぶ。ESIASW フレームワークでは、各選択操作に独自の色を割り振り、その色を選択された視覚的オブジェクトまたは選択操作の結果となる視覚的オブジェクトの color code に順次に追加することで選択操作とその結果を表す。図 6.3 に多重連携ビューの上でのマルチ選択とそれに基づいた視覚的オブジェクトの color code の定義の例を示す。図 6.3 に示す $Q1$, $Q2$, $Q3$ は連携されている 3 つのビューで順次に行った選択操作であり、それぞれ赤色、緑色、青色を用いて識別する。図 6.3 の表の中の $o1 \sim o8$ は 8 つの視覚的オブジェクトを表し、数字 1 と 0 を用いて、その視覚的オブジェクトが選択操作の結果であるか非かを表す。図 6.3 の表で示すように、

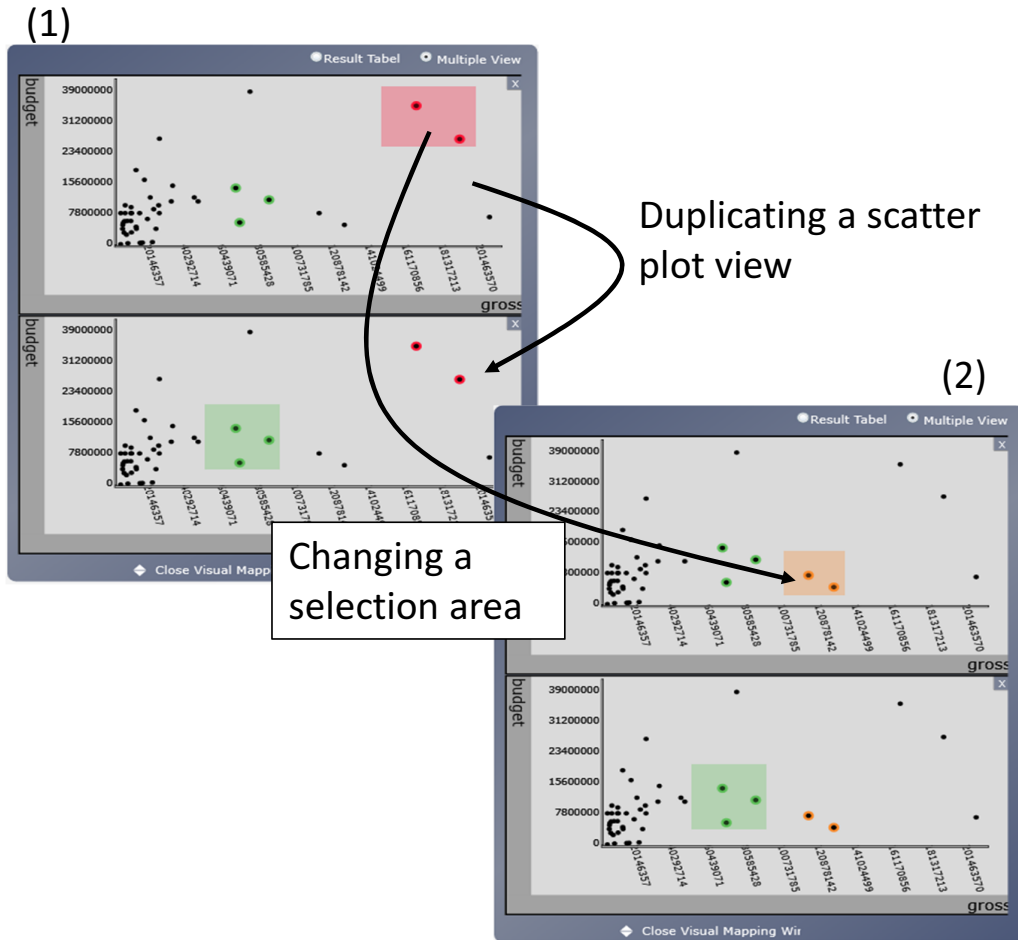


図 6.4: マルチ選択の例

ある視覚的オブジェクトが複数の選択操作の結果であるとき、その color code はそれらの選択操作に割り振られた色を順番に並べ表示したものとなる。これにより、ユーザは、多重連携ビューの上でのマルチ選択操作を用いて、データ集合を相互比較したり、それらの間の相関関係を探索的検索・分析することができる。

ESIASW フレームワークでは、多重連携ビューの上でマルチ選択を行う際、画面上の複雑さを削減するために、1つのビューで同時に複数の選択操作をすることを制限しており、同じビューの複製を用いて同一ビューの上での複数の選択を可能にしている。図 6.4 にビューの複製を用いて同一ビューに対する複数の選択操作の例を示す。こ

Chapter 6 EXPLORATORY VISUAL ANALYSIS OF SEMANTIC WEB RESOURCES

の例では、1つ分散図図を複製し、元の散布図と複製された散布図の中で、異なるエリアを選択することを示している。ビューの上で新たな選択操作を行うと、システムは自動的に元の選択とその色を削除し、新しい選択操作だけを表す。

第 7 章

関連研究との比較

Exploratory visual analysis of semantic web resources

本章では、まず提案する ESIASW フレームワークと既存の関連研究との比較を行い、実例を用いてユーザビリティの評価を行う。

セマンティック・ウェブに対する検索、統合、可視化分析に関する既存の関連研究と、その問題点を 1 章で示した 6 つの必須機能に関してまとめた結果を図 7.1 に示す。

	Mashpoint	MashQL	Visor	gFacet	Humboldt	Snap	PRISMA	KMVA	VLEPO	SWSEC	ESIASW
1. 多視点スキーマ探索的抽出展開	△	○	△	△	△	×	×	×	×	×	○
2. クラス間論理演算による新しいクラスの定義	×	×	×	×	×	×	×	×	×	×	○
3. 視覚的検索要求定義	△	○	△	○	○	△	△	△	△	△	○
4. 異なるスキーマやセマンティック・ウェブ・サービスの組み合わせ統合	×	△	×	×	×	×	×	×	△	△	○
5. 多視点からの探索的可視化分析	△	×	×	×	×	○	○	○	×	×	○
6. 1~5を探索的に繰り返すことを可能にすること	×	×	×	×	×	×	×	×	×	×	○

図 7.1: 既存研究と本研究の比較

7.1 セマンティック・ウェブ上のリソースの検索

Mashpoint[24] は、特定のセマンティック・ウェブ情報を可視化できるアプリケーションのライブラリーと、それらの間のピボット操作 (あるリソースの集合を中心にそれらの異なる属性を調べる操作) を用いてセマンティック・ウェブ情報を探索的に抽出・可視化するシステムである。このシステムは、1 章で挙げた 6 つの必要な機能の内、「多視点スキーマの探索的抽出・展開・可視化」と「クラス間論理演算による新しいクラス定義」の 2 つの機能は持っていない。さらに、Mashpoint でアクセス可能なセマンティック・ウェブ情報は、提供しているアプリケーションの種類によって制限されており、汎用フレームワークにはなっていない。MashQL[12] は、RDF データの探索的検索やマッシュアップが可能なシステムである。このシステムは、「多視点スキーマの探索的抽出・展開・可視化」と「視覚的検索要求定義」と「異なるスキーマの組み合わせ統合」の 3 つの機能を持つが、「クラス間論理演算による新しいクラス定義」と「多視点からの検索結果可視化」の 2 つの機能は持たない。

Visor[25] は、指定した複数の異なるクラス間の意味的関係を自動的検出し、可視化できるシステムである。このシステムは、「多視点スキーマの探索的抽出・展開・可視化」と「視覚的検索要求定義」の 2 つの機能のみを持つ。gFacet[10], mspace[28], Humboldt[15] は、ファセット検索を用いてセマンティック・ウェブ上の情報を探索するシステムである。これらは、「多視点スキーマの探索的抽出・展開・可視化」と「視覚的検索要求定義」の機能を持ち、さらに「検索結果を再利用した新しい検索要求定義」の機能も持つが、「クラス間論理演算による新しいクラス定義」と「異なるスキーマの組み合わせ統合」の 2 つの機能は持たない。

本研究以外の関連する既存研究では、RDF データのスキーマを探索的に抽出・展開する際に、クラス間の論理演算を用いて必要なクラスをスクリーン上での直接操作のみを用いてユーザ自身が新しく定義できるような支援機能の考えや仕組みは備わっていない。また、既存研究では、セマンティック・ウェブ上のリソースを探索的に検索することにより、展開されたスキーマ上の複数のリソースとそれらの属性の中から複

数のリソースや属性を任意に選び出し、これらの値を検索し、それらの間の相関関係などの関連性を可視化することはできない。

他にも、セマンティック・ウェブ上のリソースの検索を支援するシステムとして、RDF データのグラフを探索的に抽出・可視化する Tabulator[3]、PGV[6] 等があり、グラフィンタフェースによって SPARQL 検索要求の作成を支援する iSPARQL[13] や NITELIGHT[26] などがある。Tabulator と PGV は、個々のリソースの関係をすべて記述ことによって、繁雑になることが多く、スキーマを理解することが困難である。iSPARQL と NITELIGHT は、検索対象分野のスキーマに関する知識を必要とする。

7.2 セマンティック・ウェブ上のリソースの統合

セマンティック・ウェブ上のデータやサービスの組み合わせ統合を行う研究の多くは SWSEC[7], VLEPPO[9], 主に異なる RDF データセット間のデータの統合とセマンティック・ウェブ・サービスを自動的に連携することによるデータ統合の 2 種類に限られており、以下の 4 種類の組み合わせ統合の内、d) の RDF データセットとサービス間の多様なデータの組み合わせ統合の機能は持たない。

- a) 同一 RDF データセット中での多様なデータの組み合わせ統合
- b) 異なる RDF データセット間の多様なデータの組み合わせ統合
- c) 異なるサービス間の多様なデータの組み合わせ統合
- d) RDF データセットとサービス間の多様なデータの組み合わせ統合

7.3 セマンティック・ウェブ上のリソースの分析

セマンティック・ウェブ上のデータやサービスの分析を行う従来の研究の多くは [30, 18], 特定の RDF データセットの一部に対し、その中のリソース間の潜在的関連関係を探し出すデータ構造自体に対する分析を行っていた。しかし、近年急速に増加

する様々な分野の RDF データセットは、従来の単純の情報検索から、データやサービスを分析することによって実世界の課題解決に有用な新たな知見を発見したい要求が高まってきている。膨大なデータの分析において、従来のデータベース分野では、多様な視点から探索的に可視化分析を行う研究が多くなされていた [19, 5, 11]。これらは、多様な視点から探索的に可視化できる Coordinated Multiple View 技術を用いて、膨大なデータの色んな側面から可視化を行い、それを直接操作してデータのフィルタリングしながら、探索的な可視化分析を行うことを可能にしている。しかし、これらの従来の手法は事前にスキーマを把握し、それに基づいてデータの分析を行う必要があり、スキーマの全体像を予想できない RDF データには適用できない。

7.4 ユーザビリティの評価

本研究では、知識メディアアーキテクチャ[29]の Web バージョンである webble world[16]上で、ESIASW フレームワークに基くセマンティック・ウェブ・リソースの探索的検索・可視化システムを実装した。本論文では、表 7.1 に示す 6 つの検索タスクを完了するのに必要なスキーマの抽出・展開操作の数と、スキーマ中のクラスのプロパティに対する制約操作の数との、2 種類の操作数のそれぞれの最小回数をを用いて、このシステムのユーザビリティを既存の研究と比較をして評価した、その結果を表 7.3 に示す。本論文では、これらのタスクを完了するための各々の検索要求に対し、検索要求の作成のために展開が必要な (1) スキーマのサイズ、つまり、スキーマ中のクラスの数と、(2) スキーマを抽出・展開するために必要なクラス間の論理演算の回数、(3) スキーマの始点クラスの数、(4) 制約を課されたプロパティの数、(5) 検索対象とする RDF データセットの数は表 7.2 のようになっている。

表 7.2 と 7.3 を参照することで、検索タスク T1 と T2 のように、クラス間の論理演算の回数が 0 であるタスクを実現する場合、本フレームワークに基づいたシステムに必要な最少操作回数は既存の研究の操作数に等しくなることがわかる。

表 7.2 で示すように検索タスク T3~T5 は、クラス間論理演算による新しいクラス

表 7.1: 6つの検索タスク

T1	アメリカ映画のタイトルと興行収入を検索
T2	今年にリリースされたアメリカ映画のタイトルと、その映画の主演俳優と監督を検索
T3	自分が監督した映画の中に俳優として出演したことがある監督と、その映画のタイトルと興行収入を検索
T4	主演俳優と監督の出身校が同一である映画のタイトルと、その映画の主演俳優と監督の出身校を検索
T5	中国映画とアメリカ映画の両方に出演したことがある俳優の中で、中国人ではない人を探す.
T6	DBpedia 中の映画に対し、DBpedia には記述されてない映画の衣装デザイナーを Linkend Movie DataBase から検索

の定義を必要とする検索タスクである。クラス間論理演算機能を持たない既存の研究を用いて検索タスク T3~T5 を実現しようとしたとき、検索タスク T3 の場合は、映画クラスと、その監督と俳優のクラスをそれぞれ抽出して、映画クラスの全ての映画インスタンスに対し、ユーザ自身がそれらの監督と俳優を順番に比較しながら、監督が俳優でもある映画を探さなければならない。検索タスク T4 の場合も同様で、映画クラスと、その監督と俳優の出身校のクラスをそれぞれ抽出し、映画クラスの全ての映画インスタンスに対し、ユーザ自身がそれらの監督と俳優の出身校を比較しながら、同一出身校の監督と俳優を持つ映画を探さなければならない。これらのタスクを完了するために必要となるユーザ自身による比較回数は映画クラスのインスタンスの数に等しくなり、映画クラスのインスタンスの数が多かった場合、検索タスクを完了するのは極めて困難な作業となる。検索タスク T5 の場合には、中国映画のクラスとアメリカ映画のクラスを2つの始点として、それらの俳優と各俳優の国籍（ここでは、俳優の国籍が生まれた国であると仮定）を表すクラスをそれぞれ抽出して、俳優の国籍

表 7.2: 検索要求の定量化

	T1	T2	T3	T4	T5	T6
スキーマサイズ	1	3	2	4	4	3
クラス間論理演算回数	0	0	1	1	2	0
始点クラス数	1	1	1	1	2	2
制約を課されたプロパティ数	2	4	3	2	1	2
RDF データセット数	1	1	1	1	1	2

表 7.3: タスクを完了するための最少操作回数

	MashQL	mSpace	gFacet	Visor	ESIASW
T1	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
T2	(4,5)	(4,5)	(4,5)	-	(4,5)
T3	-	-	-	-	(7,3)
T4	-	-	-	-	(9,2)
T5	-	-	-	-	(12,1)
T6	(8,2)	-	-	-	(6,2)

(スキーマ抽出, 制約条件指定)

を表す両クラスのインスタンスの中から中国ではない国を全て探し出して選択し、その条件を満たす両俳優クラスのインスタンスを相互比較しながら、同一人物になっているケースを探さなければならない。これもデータの量が多い場合、上記の T3, T4 と同様に膨大な作業が必要となる。

本フレームワークに基づいたシステムを用いて検索タスク T3~T5 を実行する場合、タスクを完了するための最少操作数は表 7.3 に示すように、(始点クラス数 + スキーマサイズ + 論理演算回数 $\times 4$ + 制約操作の回数) 以下になるということがわかる。これにより、本システムは、論理クラスを含む複雑なスキーマの抽出・展開でも比較的

少ない操作回数で容易に行うことができるといえる。

表 7.3 に示すように、本フレームワークに基づくシステム以外では、既存の研究の中で MashQL だけが複数の RDF データセットの統合を必要とする検索タスク T6 を完了することができる。しかし、MashQL は、異なる RDF データセットから抽出したスキーマを直接結合することはできず、新たなスキーマを作成して異なる RDF データセットから抽出したスキーマに基づいて検索した結果を統合しなければならないため、スキーマ抽出の操作数が多くなる。

第 8 章

結論

Conclustions

本論文では、ESIASW フレームワークを提案した。ESIASW フレームワークは、セマンティック・ウェブ上に集積された膨大なデータ集合やサービスに対する探索的検索、連携統合、可視化分析を支援し、その組み合わせ利用を可能にする枠組みである。ESIASW フレームワークでは、RDF データスキーマの各クラス可視化するクラス部品、セマンティック・ウェブ・サービスのプロキシを作成するサービス部品、多重連携ビューを動的作成する可視化分析部品の 3 つの視覚的部品を提供し、それらを直接操作したり、相互結合することによって、セマンティック・ウェブやプログラミングに関する知識を持たないユーザでも、セマンティック・ウェブ上のリソースに対する探索的検索、連携統合、可視化分析の 3 つを自在に組み合わせ利用できる可視化環境を提供することができるようになった。

セマンティック・ウェブ上のリソースの検索において、ESIASW フレームワークでは、クラス部品を直接操作することによって、そのクラスと何らかの意味的關係をもつ他クラスを逐次抽出したり、検索要求の変数や制約条件を直接指定したり変更することを可能にした。これにより、スキーマに関する知識を持たないユーザでも、関心を持つ 1 つまたは複数のクラスをアクセスの始点として、その意味的關係を辿りながら、必要に応じてスキーマを部分的かつ探索的に展開して行きながら、複数の異な

Chapter 8 CONCLUSIONS

る視点から検索条件を同時に指定したり、変更することが可能にした。また、展開・可視化したスキーマ中の異なるクラスを直接組み合わせ論理演算することによって、論理演算を含む複雑な検索要求を定義するためのスキーマをユーザ自身が自在に定義することを可能にした。ESIASW フレームワークでは、これらの操作は全て自動的に SPARQL へと書き換えられ実行される。これにより、SPARQL やプログラミングに関する知識をもたないユーザでも、スキーマを探索的抽出・展開したり、それに基づいて検索要求をインタラクティブに定義し変更することが可能になり、セマンティック・ウェブから必要な情報を探索的に探し出すことができるようになった。

セマンティック・ウェブ上のリソースの統合アクセスにおいて、ESIASW フレームワークでは、同一作業空間内で、異なる複数の RDF データのスキーマを展開・可視化したり、サービス部品を用いて必要なセマンティック・ウェブ・サービスのプロキシを作成することを可能にした。サービス部品は、RDF に基づいたセマンティック・ウェブ・サービスの記述から、セマンティック・ウェブ・サービスの入力、出力などに関する情報を自動的に抽出し、それらを用いてセマンティック・ウェブ・サービスのプロキシを作成することができる。ESIASW フレームワークでは、展開・可視化された RDF データスキーマを意味的関連や論理的関連に基づいて 1 つのスキーマに統合して利用したり、異なるサービス部品を意味的関連に基づいて直接結合したり、RDF データスキーマとサービス部品を意味的関連に基づいて直接結合したりすることで、以下の 4 種類の組み合わせ統合を可能にした。

- a) 同一 RDF データセット中での多様なデータの組み合わせ統合
- b) 異なる RDF データセット間の多様なデータの組み合わせ統合
- c) 異なるサービス間の多様なデータの組み合わせ統合
- d) RDF データセットとサービス間の多様なデータの組み合わせ統合

これにより、RDF データのスキーマや、セマンティック・ウェブ・サービスに関する深い知識を持たないユーザでも、セマンティック・ウェブ上の多様なデータやサービ

スを自在に組み合わせて、統合利用することが可能になった。

ESIASW フレームワークにおいて、可視化分析部品は、多種のビューと、マルチ選択、Brushing&Linking、カラーコーディネーションなどの3つコーディネーション規則を提供する。ユーザは可視化分析部品を展開された RDF データスキーマまたは、サービス部品に直接結合して、多様な視点から探索的に可視化分析ができる多重連携ビューを動的に構築しながら、その上で複数のデータ集合をインタラクティブに選択することが可能になった。ESIASW フレームワークでは、多重連携ビュー上の各選択操作に独自の色を割り振り、その結果を Brushing&Linking 技術とカラーコーディネーション技術を用いて表すことによって、異なるデータ集合を相互比較したり、それらの間の相関関係を探し出したりすることも可能にした。

関連発表論文

1. 朴斌, 田中譲. セマンティック・ウェブ上のリソースの探索的検索・可視化フレームワーク. 電子情報通信学会論文誌, Vol.J98-D,No.1,pages.193-205 Jan. 2015
2. Bin Piao and Yuzuru Tanaka. Interactive framework for visual exploratory search and integration of semantic web contents and services. In *Proceedings of 7th International Conference on Information Management, Innovation Management and Industrial Engineering*, pages.534-539, Oct. 2014.
3. Bin Piao and Yuzuru Tanaka. Interactive framework for visual exploratory search, integration, and visual analysis of semantic web resources. In *Proceedings of 16th International Conference on Information Integration and Web-based Applications & Services*, pages.200–206, Dec. 2014.

参考文献

- [1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*, pages 722–735, 2007.
- [2] T. Berners-Lee. Plenary at [www.geneva.94.](http://www.w3.org/Talks/WWW94Tim/)
<http://www.w3.org/Talks/WWW94Tim/>, 1994.
- [3] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction*, 2006.
- [4] R. Cyganiak and A. Jentzsch. The linking open data cloud diagram. <http://lod-cloud.net/>, 2014.
- [5] A. L. da Silva Kauer, B. S. Meiguins, R. M. C. do Carmo, M. de Brito Garcia, and A. S. G. Meiguins. An information visualization tool with multiple coordinated views for network traffic analysis. In *Proceedings of 12th International Conference on Information Visualisation, IV 2008*, pages 151–156, 2008.

参考文献

- [6] L. Deligiannidis, K. Kochut, and A. P. Sheth. Rdf data exploration and visualization. In *Proceedings of the ACM First Workshop on CyberInfrastructure: Information in eScience*, pages 39–46, 2007.
- [7] Y. Gamha, N. Bennacer, G. Vidal-Naquet, B. el Ayeb, and L. B. Romdhane. A framework for the semantic composition of web services handling user constraints. In *Proceedings of 2008 IEEE International Conference on Web Services (ICWS 2008)*, pages 228–237, 2008.
- [8] O. Hatzi, D. Vrakas, M. Nikolaidou, N. Bassiliades, D. Anagnostopoulos, and I. P. Vlahavas. An integrated approach to automated semantic web service composition through planning. *IEEE T. Services Computing*, 5(3):319–332, 2012.
- [9] O. Hatzi, D. Vrakas, M. Nikolaidou, N. Bassiliades, D. Anagnostopoulos, and I. P. Vlahavas. An integrated approach to automated semantic web service composition through planning. *IEEE T. Services Computing*, 5(3):319–332, 2012.
- [10] P. Heim and J. Ziegler. Faceted visual exploration of semantic data. In *Proceedings of Human Aspects of Visualization*, pages 58–75, 2011.
- [11] J. Huo. Kmvql: a visual query interface based on karnaugh map. In *Proceedings of the working conference on Advanced Visual Interfaces, AVI 2008*, pages 243–250, 2008.
- [12] M. Jarrar and M. D. Dikaiakos. Querying the data web: The mashql approach. *IEEE Internet Computing*, 14(3):58–67, 2010.
- [13] C. Kiefer, A. Bernstein, and M. Stocker. The fundamentals of isparql: A virtual triple approach for similarity-based semantic web tasks. In *Proceedings*

- of The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*, pages 295–309, 2007.
- [14] M. Klusch, B. Fries, and K. Sycara. Automated semantic web service discovery with owls-mx. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '06*, pages 915–922. ACM, 2006.
 - [15] G. Kobilarov and I. Dickinson. Humboldt: Exploring linked data. In *Proceedings of the WWW2008 Workshop on Linked Data on the Web, LDOW 2008*, 2008.
 - [16] M. Kuwahara and Y. Tanaka. Advanced ”webble” application development directly in the browser by utilizing the full power of meme media customization and event management capabilities. In *Proceedings of 2012 IEEE International Conference on Multimedia and Expo Workshops*, pages 211–216, 2012.
 - [17] LOD. State of the LOD cloud 2014. <http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/>, 2014.
 - [18] M. A. Martínez-Prieto, M. A. Gallego, and J. D. Fernández. Exchange and consumption of huge RDF data. In *Proceedings of The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012*, pages 437–452, 2012.
 - [19] C. North and B. Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proceedings of Advanced Visual Interfaces*, pages 128–135, 2000.
 - [20] C. Pedrinaci and J. Domingue. Toward the next wave of services: Linked services for the web of data. *J. UCS*, 16(13):1694–1719, 2010.

参考文献

- [21] B. Piao and Y. Tanaka. Interactive framework for visual exploratory search and integration of semantic web contents and services. In *Proceedings of 7th International Conference on Information Management, Innovation Management and Industrial Engineering*, pages 534–539, 2014.
- [22] B. Piao and Y. Tanaka. Interactive framework for visual exploratory search, integration, and visual analysis of semantic web resources. In *Proceedings of 16th International Conference on Information Integration and Web-based Applications and Services*, pages 200–206, 12 2014.
- [23] B. Piao and Y. Tanaka. Interactive framework for exploratory search and visualization of semantic web resources. *Transactions of IEICE(D)*, 1.J98-D(1), 2015.
- [24] I. O. Popov, M. M. C. Schraefel, G. Correndo, W. Hall, and N. Shadbolt. Interacting with the web of data through a web of inter-connected lenses. In *Proceedings of WWW2012 Workshop on Linked Data on the Web*, 2012.
- [25] I. O. Popov, M. M. C. Schraefel, W. Hall, and N. Shadbolt. Connecting the dots: A multi-pivot approach to data exploration. In *Proceedings of The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference*, pages 553–568, 2011.
- [26] A. Russell and P. R. Smart. Nitelight: A graphical editor for sparql queries. In *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008)*, 2008.
- [27] D.-H. Shin, K.-H. Lee, and T. Suda. Automated generation of composite web services based on functional semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):332 – 343, 2009.

- [28] D. A. Smith and N. R. Shadbolt. Facetontology: Expressive descriptions of facets in the semantic web. In *Proceedings of Semantic Technology, Second Joint International Conference, JIST 2012*, pages 223–238, 2012.
- [29] Y. Tanaka. *Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources*. Wiley-IEEE Press, 2003.
- [30] P. Teufel and G. Lackner. Rdf data analysis with activation patterns. In *Proceedings of I-KNOW '10 : 10th International Conference on Knowledge Management and Knowledge Technologies*, pages 41–49, 2010.
- [31] W3C. Owl web ontology language for services. <http://www.w3.org/Submission/2004/07/>, 2004.
- [32] W3C. Resource description framework. <http://www.w3.org/TR/rdf-primer/>, 2004.
- [33] W3C. Semantic annotations for wsdl, <http://www.w3.org/Submission/2004/07/> 2004.
- [34] W3C. Web ontology language. <http://www.w3.org/TR/owl-features/>, 2004.
- [35] W3C. SPARQL query language for RDF. <http://www.w3.org/TR/rdf-sparql-query/>, 2008.
- [36] W3C. RDF vocabulary description language 1.1. <http://www.w3.org/TR/rdf-schema/>, 2014.