

このドキュメントは

Conceptual Graphs and the Semantic Web

<http://www.w3.org/DesignIssues/CG.html>

の和訳です。

この文書には和訳上の誤りがありえます。

内容の保証はいたしかねますので、必ず W3C Web サイトの正式版文書を参照して下さい。

Tim Berners-Lee

Initially created: 2001/01/06, last change: \$Date: 2001/02/01 01:26:08 \$

ステータス: 個人的な閲覧の範囲に留める 編集ステータス: 初期ドラフト

[Up to Design Issues](#)

ウェブ・アーキテクチャーに対する考察

序文

幾度となく人々は私の研究を概念グラフと関連付けてきた。[Mary Keeler](#) は熟考の末、[a CG conference](#) においてセマンティックウェブが概念グラフとまさにどう一致するのか、あるいは逆に概念グラフがセマンティックウェブにどう一致するのかを示してくれた。しかしながら私にとって概観したその記事は、[Charles S. Peirce](#) のいくつかのアイデアを実装したいという熱烈な願望を除いては、CG が何であるかのよい参考とはならなかった、思うに、CG における例がハードロジックよりむしろ自然言語と関係をもつ傾向は、Swab に関し厳格な論理手続きへの傾向をもつ私のような人にとっては、CG 分野は何を意味するのか理解することをより困難にしていた。とにかく、2001/1/5 に私は John Sowa の "[the CG Standard](#)" へのポインタを見つけた。そして一通り拝読した。CG は -- それらの仕様が語っているように -- KIF と同等であることは明確である。それらは論理であり、円と矢印のダイアグラムで深さについて拡張されており、可視化を行う伝統的なものであるが、論理はそれでもやはり、セマンティックウェブと同じように、高階論理を含むそして -- ここではその相違点についていくつかコメントを述べたい。

概念グラフとセマンティックウェブ

簡潔に言えば、概念グラフ(CGs)は論理の閉仮説世界を記述するために用いられる論理言語である。それらは伝統的に平面上のグラフ表示に関する強調であるが、形式的なシリアル化が存在しており、一つは "Linear Form" で非常に [N3](#) と似通っており、そしてよりオフィシャルな CG Interchange Format(CGIF)がある。さまざまな賛否両論があるが、それらは基本的に KIF と同様な表現力を持ち -- それゆえにセマンティックウェブの基礎としてウェブ化されるしかない。ここでは、概念グラフと RDF に基づくセマンティックウェブの間の相違点と類似点について検討する。

"nonsemantic information" ([1], sec2)に関しては本稿では除外する。

CG のウェブ化

[webizing a language](#) の立場に立って、CGIF をもとにしたセマンティックウェブがどのようなものであるかを想像するために、どのように CGIF や LF に適用するかを見てみよう。

まず初めに我々が明らかにしなければならないことは、URI を持たせることによって、各々の概念と関係がファーストクラスのオブジェクトとなるように CG シンタックスを變形することである。シンタック

ス変形は単にキャラクタを URI 内に含めることのできるものであり、その結果として 任意の概念を参照、あるいは任意の関係が使用できるようになる。URI 空間を CG 識別子に写像する典型的な手法は、CGIF ドキュメントにおける URI を連結して、CGIF 識別子の URI を作ることであろう。そしてハッシュ記号と局所的な CG 識別子 -- URI タームにおける局所的な存在識別子を作ることであろう

頭の中で言語を web 化した結果、次なる疑問はどのようにそのようなセマンティックウェブの言語に写像するかということである。これは CG スペック[1]が KIF の写像を与えるという事実があるために単純化される。

型とクラス

CG と RDF は型の概念を共有している。CG は概念の世界と型に対する制限をもち、リレーションとリレーションの型とは別々になっている。それゆえ、CG を使ってリレーションに関する何かをリレーションにより記述することはできない。もし、真の双方向の写像が欲しいならば、CG は(初回の読み込みの際と思われるが)多かれ少なかれ -- メタレベルで表現するために -- 任意の RDF グラフへ具体化される必要がある。しかしながら、私見ではこれは有用ではないと思われる。CG の設計者たちはこのディスジャンクションを意図しており、それにより CG 概念型から RDF のクラスへ、そして CG リレーションからプロパティへ、CG リレーションからプロパティへ、そして CG リレーションタイプから rdf:プロパティのサブクラスである RDF クラスへの直接的で自然な写像となる。

セマンティックウェブの論理言語は普遍的であり、その内部では他のあらゆる言語の表現を許容していなければならない;しかしどの言語もそれ自身が普遍的であることを強いたりしない。

CG における集中型の思想

ナレッジベース(KB)における CG 概念は、集中型の見解を少数含んでいる。実際のところ CG には構造的な問題は存在しないが -- それらはウェブスケーリング要求なしに技術的な意思決定がなされたことによる。取り除くことは CG の理念に対しに何らダメージを与えるものではない。

- 閉じたナレッジベースの概念では、とりわけ全ての個々に独立な単一のカタログがある。ある KB は階層構造の型をもち、関係の階層を持ち、個人に対する中心的なカタログを持つ。階層構造は木構造ではなく、非環式のグラフであり、それゆえ閉じられた事実の上に問題を引き起こしたりはしない。KB は以下ものでなければならない。

- 概念のファクトは単一の型と関連付けられている。セマンティックウェブにおいては、あるものに関するオリジナルの製作者は、型を定義しなければならないかもしれないが、論理的にはステートメントは第三者によっても等しく型のアサーションを作ることができ、これらのステートメントとは rdf:type のステートメントであっても良い。

- 同一指示集合は単一のドミナント概念(dominant concept)でなければならない。

プロパティとリレーション

初読で判明した主要な違いは、RDF プロパティが常に二項的であり、CG リレーションは単項であることである。

RDF ベースモデル、そしてその論理的フレームワークの拡張である N3 メソッドは、基盤構造として支持されているようにみえるが、N-ary 形式が欠落していることが異なり、しかし N-adic リレーションにおける CG モデル中の弧の存在は、明示的に $n > 2$ の時には二項 RDF への自然な写像を示唆している。これは二つの形式が共存するように少しの張力を残している。

CG の世界はバイパタイトなグラフである -- ひとつは二つのリレーションと概念からなり、それらは別々のものである。RDF の世界では、モノからモノへとプロパティを介するリンクから成っており、それはプロパティとモノを別々のものとはしない。全てはモノである。

顕著な類似性

CG の働きと現在までのセマンティックウェブはいくつかの著しい類似性がある。両者ともに大部分は円と矢印のダイアグラムに示唆を得たものであり、LF と N3 にはいくつかのシンタックスの形式においてさえもそれが現れている。人々は、伝えなければならない何かのデータに対し、従来より円と矢印による表記を用いてきた。N3 において、たいへん単純にはあるが unicode へこれを適用してみる。

```
w3c:Michael >- org:member -> w3c:team .
```

John Sowa による例を見た際には、認識に対する確たる手ごたえを感じた。

```
[Go]-
```

```
(Agnt)->[Person: John]
```

```
(Dest)->[City: Boston]
```

```
(Inst)->[Bus].
```

N3 では以下のように思われる

```
@prefix : <#>.
```

```
[a :Go]
```

```
>- :agent -> [a :Person; = <#John>];
```

```
>- :dest -> [ a :City; = <#Boston>];
```

```
>- :inst -> [ a :Bus].
```

最終段階まで、注目すべき物となっている。両者のシンタックスは逆方向の矢印をもっており CG の LF では $a \leftarrow (p) \leftarrow b$, そして N3 では $a \leftarrow p \leftarrow b$ となる (参照: [RDF でも同様](#))

コンテキスト

“context”の概念は CG と N3 では大変似通ったものとなっており、両者の場合とも定式は引用を用いることで構成される。N3 では情報の集合をカプセル化する際の導入としてブレースが用いられ、それは集合として表される。[1]からの例として、“Tom believes that Mary wants to marry a sailor”:

```
[Person: Tom]<-(Expr)<-[Believe]->(Thme)-
```

```
  [Proposition: [Person: Mary *x]<-(Expr)<-[Want]->(Thme)-
```

```
    [Situation: [?x]<-(Agnt)<-[Marry]->(Thme)->[Sailor] ]].
```

N3 では、二項関係を RDF プロパティにマッピングし

```
<#Tom> a :Person; :believes [a :Proposition; = {
```

```
  <#Mary> a :Person; :wants [ a :Situation; = {
```

```
    <#Mary> :marriedTo [ a :Sailor ]
```

```
  ]]
```

```
}}].
```

(上記では、“=”は出来ないことを構成するための等価を表すステートメント、あるいは特徴の無いコンテキストをサブジェクトおよびステートメントのオブジェクトであるとするための N3 シンタックスである。) RDF では、独自のスタイルはしばしばモノの型として仮定されており、述語の範囲やドメインから導き出すことが出来る場合には、明示的に述べる必要はない。たとえば、あらゆる *believes* オブジェクトはプロポジションであるかも知れず、あらゆる *wants* オブジェクトはシチュエーションであるかも知れない。それゆえ上の具体例での N3 表現は次のようになる

```
<#Tom> :believes {
```

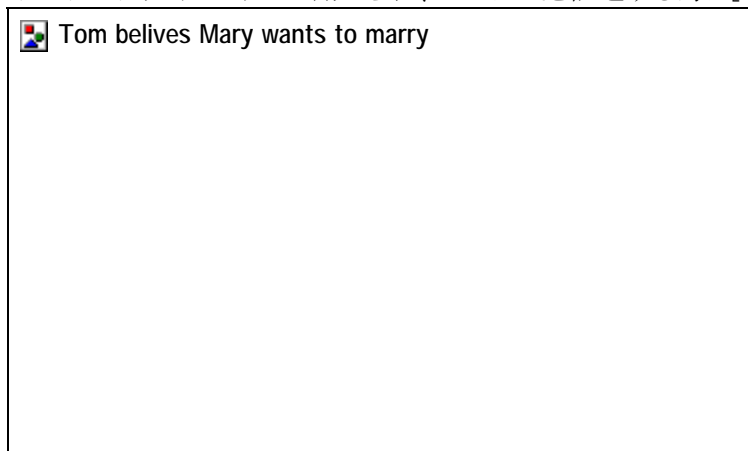
```
  <#Mary> :wants {
```

```
    <#Mary> :marriedTo [ a :Sailor ]
```

```
  }
```

```
}.
```

これが英語のセンテンスとして、そして多くの哲学と言語学(それは自然言語を表現しないと試みることで、多くの場合回避するものである)にとって、良いモデルであるかどうかという疑問は置いておく。CGの世界ではしばしばダイアグラムが用いられ、上の公式を記述する為に[1]から



N3において、私が描いた円-矢印ダイアグラムは[円]が結婚であった場合に、全称限量子があることを示すための四角形からの矢印を含んでいる。

このように適切な結果とはならない他の写像も多く存在する。CGからRDFへのひとつはCGのアーキをRDFプロパティに写像することであり、上記の例では次のようになる。

```
[ a :Belief;
  :expr <#Tom>;
  :thme: [ a Proposition; = {
    [ a :Want;
      :expr <#Mary>;
      :thme [ a :Situation; = {
        [ a :Marriage; :agent <#Mary>; :thme: [a :Sailor]]
      }
    ]
  }
]
```

英語では、これはおそらく次のようになる”There is a belief, experienced by Tom, that ”there is a want, felt by Mary, that there should be a situation: `Mary is married to a Sailor” ”.

限量子とラムダ

私はこの分野に関しては詳細な比較を行ったわけではない。N3とCFIFの両者とも存在限量子と全称限量子を持ち、全称限量子は仕様のもとに宣言されているが定義済み限量子として進展中である。両者ともにRDFのように、特徴のないノードからの存在限量子は暗黙の存在としている。CGIFにおいて私が解決し得なかった疑問は、”?x”と”*x”の用語を用いることで導入された限量子のスコープをどのように決定することができるのかということであった。[1]の中に説明があり、(私が思うに)全称限量子は存在限量子よりもより高次のスコープを持っているということであり -- N3でも同様である。N3では、モデルは量化された変数を、log:forAllまたはlog:forSomeのステートメントを用いることにより、直接そのスコープコンテキストにリンクさせなければならない。N3はそのようなラムダを持ってはいない。等価な定式をもつダブルの含意を与えることによって、新しいターム(プロパティもしくは関連したプロパティの小さな集合)を意味するダブルの含意を、式へのパラメータとして全称限量子の変数を用いることによって書き出すことができる。二つの設計が

直面している問題は、セマンティックウェブはオープンコンテキストにおいても動作し、コンテキスト外部が参照された場合には、ネストされた表現に対する意味を定義する。

結論

概念グラフは容易にセマンティックウェブと統合され、写像は明らかに直接的である。CGIF あるいは LF における CG を N3 へエクスポートすることは読者にとって適切な練習問題に思われる;-)。興味深くより挑戦的な課題は CG マシンを構成することであろう -- CG シンタックスを変更し -- 外部概念を参照する URI を含むグラフをインポートできるようなものである。CG においてリレーションタイプが概念ではないという問題は大きなものではない。多くのシステムが存在するからである -- とりわけオントロジー的なシステム -- それは類似の制限を持ち、それによって変換が可能になるようなものである。

CG の興味深いサブセットが存在する、“simple graph”と呼ばれており全てがひとつのコンテキストであり、否定や“定義済み限量子”が無いが、全称限量子を含み、これらは直接 RDF M&S 1.0 へ、もしくはブレース無しの N3 へ写像されるものである。

RDF のベースモデル、そしてその論理的なフレームワークを拡張した N3 メソッドは基本構造としてサポートされているように思われるが、N-ary 形式が欠けていることは不整合として目立っている。概して大きな一致があり、二つの技術は大変類似しており、うまくいけば容易に相互使用可能である。

付録: 用語の比較

用語の比較		
CG	RDF/N3	コメント
概念	リソース/ノード	
リレーション	プロパティ	RDF では、プロパティは二項的 :CG では リレーションは n-項的
タイプ	クラス	RDF では、“タイプ” はクラスでのプロパティのリソースのメンバシップを述べるプロパティである。
弧	プロパティ - N>3	CG における弧は二項の関係から n 項のモデルの為に用いられ、RDF と類似している。弧は三つ組ではなくペアとして考慮されている。それに関連する小さな整数を持っている。(cf XLink の役割、あるいは rdf:1, rdf:2 など)
コンテキスト	コンテキスト(N3のみ)	用語の著しい一致
coreference	daml:equivalent/ =	単純な機能のためのやや複雑な CG アーキテクチャーか? CG では同一指示のセット(等価なクラスの形式)が、単一な定義をする(“ドミナント”)概念を持っている。これは等価性を完全に対称なものとはしない。おそらく、同一指示のセットの名としてドミナントの概念を用いることは単に実用上有益であるからであろう。
actor	-	哲学的な違い: SWeb においては、実際の世界はモノとプロパティに直接リンクされる。モノは表現するための別のレイヤーをもっている。CG では「現実」へのリンクが CG の主要部とは別に提示される。

抽象構文	モデル	
識別子	URI	CGIF 識別子はケースセンシティブではない。URI と XML ID はそうである。(これは国際的な方法でケースインセンシティブなものを明記する上での困難の結果であった。)
知識ベース		型を自己内包することによる無矛盾性
線形形式	N3	人にとっての可読性のために導入されたシンタックス、両者ともに円と矢印のダイアグラムを必要とする。(→ agent → bus .) そして偶然にも [] の使用法が一致しており、. をデリミタとして用いる。
シグニチャ	レンジ, ドメイン	RDF のプロパティは二項だけであり、シグニチャはレンジとドメインである。より哲学的に重要なことには、ラムダ表現は単独で決定的なシグニチャを持っており、そこではプロパティに関して決定的でそれらの範囲とドメインについての第三者のステートメントが存在しても良い。

参照

[1] John F. Sowa, ed. "[Conceptual Graph Standard](#)", ["standard" であると思われるが確証はない。部分的には発展の余地があるとのラベル付けがされている].

[2] John F. Sowa, [Conceptual Graphs](#), web page

Related DesignIssues:

[Notation3](#)

[Webizing](#)

[Up to Design Issues](#)

[Tim BL](#)