

電気学会誌

2002

Vol.122

10

The Journal of The Institute of Electrical Engineers of Japan

特 集 ■ インターネットにおけるエージェント技術

- 解 説 ■ 地下1,000mの大規模揚水発電構想
- 十見百聞 ■ 伝統芸術の継承にITが一役
- 技術探索 ■ インパルス電圧計測技術の変遷
- 学生のページ ■ ナノスケール強磁性体とは

本会ホームページ <http://www.iee.or.jp>



電気学会誌



社団法人 電気学会
© Denki-gakkai 2002

〒102-0076 東京都千代田区五番町6-2
HOMAT HORIZONビル8F

特集 インターネットにおけるエージェント技術

エージェント技術は、インターネット上にユビキタスソリューションを実現する。安全でグローバルなwebサービスの提供、Semantic Webの実現などで注目されている。

チーフエディタ：浅川和雄（富士通研究所）

ゲストエディタ：雨宮真人（九州大学）

- | | | |
|----------|----------------------------|--|
| 1 | インターネットとエージェント技術 | 雨宮真人 668 |
| 2 | エージェントシティ | Jonathan DALE,
Steven WILLMOTT,
Bernard BURG 672 |
| 3 | Semantic Webにおけるアプリケーション統合 | James HENDLER,
Tim Berners-LEE,
Eric MILLER 676 |
| 4 | エージェントによるセキュアな電子コミュニティの構築 | 岡田 誠, 山崎重一郎 681 |
| 5 | エージェント技術に関する国際標準化活動 | 須栗裕樹 685 |

- | | |
|--------------------|----------------------------|
| 解説 | 和田文雄, 田中邦典 689 |
| 地下1,000mの大規模揚水発電構想 | |
| 随想 | 原 精二 661 |
| 私が期待する株主 | |
| 十見百聞 | 坪井雄一, 杉田晋哉, 阿部祥知 664 |
| 伝統芸術の継承にITが一役 | |
| 技術探索 | 原田達哉, 脇本隆之 693 |
| インパルス電圧計測技術の変遷 | |
| 学生のページ | 島田 寛 697 |
| ナノスケール強磁性体とは | |
| 支部のページ | 中国支部・北陸支部 702 |
| 学会だより | 708 |
| 編集後記 | 708 |
| 技術報告要旨 | 701 |
| 会告 | 709 |

* 表彰受賞候補者の推薦 * 大会・総合研究会 * 支部活動 * 国際会議・シンポジウム * 論文募集 * 求人 * 研究助成・表彰 * 共催・協賛・後援行事 * 研究会

The Journal of The Institute of Electrical Engineers of Japan (Denki-gakkai Shi)

SPECIAL ISSUE

Agent Technology on the Internet

- | | | |
|----------|---|--|
| 1 | Internet and Agent Technology | By Makoto AMAMIYA 668 |
| 2 | Agentcity | By Jonathan DALE, Steven WILLMOTT,
Bernard BURG 672 |
| 3 | Integrating Applications on the Semantic Web | By James HENDLER, Tim Brners-LEE,
Eric MILLER 676 |
| 4 | Secured Agent Community Framework with a VPC Platform | By Makoto OKADA,
Shigeichiro YAMAZAKI 681 |
| 5 | International Standardization Activities on Agent Systems | By Hiroki SUGURI 685 |

Large Scale Pumped Hydroelectric Storage Plan Utilizing the Underground of 1,000 meters Deep

Technical Review—By Fumio WADA,
Kuninori TANAKA 689

Shareholders I Expect to Be

Essay—By Seiji HARA 661

Information Technology Makes Traditional Arts Eternal

Pictures & Words—By Yuich TSUBOI, Shinya SUGITA,
Syochi ABE 664

Change of Impulse Voltage Measurement Technology

Technical Old & New—By Tatsuya HARADA,
Takayuki WAKIMOTO 693

What is “Nano-Scale Ferromagnets”?

Student Page—By Yutaka SHIMADA 697

Semantic Web における アプリケーション統合



Semantic Web, XML, RDF, Enterprise Application Integration
(企業間アプリケーション統合), e-commerce



James Heldler Tim Berners-Lee Eric Miller
翻訳：益岡 竜介

1. はじめに

Semantic Web は現在の Web の拡張である。この Semantic Web では、情報にはっきりと定義された意味が与えられ、コンピュータと人間がよりよく協調して仕事ができるようになる。これは Web に関するデータを定義し、リンクしておくことにより、そのデータを多様なアプリケーションから、より効果的な発見、自動化、統合、再利用に使おうという考えに基づいている。Web がその最も有用な姿に到達するためには、この Semantic Web に進化し、データを自動化されたツールと人間の両者ともが共有および処理できる、どこからでもアクセス可能なプラットホームを提供する必要がある。

Semantic Web は単に Web ページだけではなく、データベース、サービス、プログラム、センサ、PDA (Personal Digital Assistant)、さらには家庭用の電気製品までもが、Web 上のデータを使い、そして Web 上にデータを提供できるインフラを実現する。ソフトウェアエージェントはこの情報を使い、わくわくするような新しい方法で情報を検索し、フィルタし、そして準備することで、Web ユーザを手助けすることができる。Semantic Web の新しい言語は、Web 上の今までよりもかなり多くの情報を機械可読にすることで、このビジョンをより力強いものと

James Heldler Maryland 大学教授。Maryland 大学 Information and Network Dynamics 研究所における Semantic Web・エージェント関連 Director。US Air Force 科学諮問委員会メンバー。アメリカ人工知能学会 Fellow。前 DARPA 科学情報システム Chief Scientist。W3C セマンティックウェブ活動の主要メンバー。

Tim Berners-Lee 1990 年、CERN 研究所在任中に World Wide Web 考案。MIT の 3Com Founders Chair であり、W3C 活動を指揮している。2002 年日本国際賞受賞。

Eric Miller W3C セマンティックウェブ活動の主要メンバー。MIT 研究所研究員。W3C に参加するまでは、OCLC (Online Computer Library Center) Inc. の主任研究員であり、Dublin Core Metadata 活動の創設者の一人として Associate Director であった。

し、技術とツールの新しい時代を切り開いていくことを可能にする。

現在 Semantic Web の多くの要素はすでに存在しており、この技術のビジネスへの応用は日に日に明確になってきている。本稿で、我々は Semantic Web の近い将来の目標に着目し、その目標が Web ユーザにもたらし得る恩恵を概説する。

インターネットが国際的に確立し始めた 1989 年当時の文書システムの状態を考えてみよう。そのころは、遠隔システムから情報を引き出し参照するのは、まだ専門家の分野であった。もちろんインターネットは存在しており、遠隔システムへ接続すること自体は大抵可能であった。しかし、往々にしてそれらのシステムは互いに全く異なる情報アクセスのプロトコルを使っていた。例えば、あるシステムのデータベースを検索するには、そのシステムに telnet (あるいは何か別の遠隔ログインのプロトコル) を使ってアクセスし、特定の情報アクセスプロトコルを使う必要があったりした。関係する情報を見つけても、それをクリップボードにコピーし (あるいはたまたま手元にあった紙に走り書きし)、次にその情報を関連する文書に貼り付ける (あるいは再び打ち直す) 必要があった。したがって、得られる情報は関連の深さ、タイミングのよさ、あるいは正確さといった条件ではなく、往々にして入手の容易さによって決まっていた。

それが Web 以前の状況であった。Web 技術のある今日では、我々は情報を容易にそして継ぎ目なくリンクすることができます。現在、ほとんどの遠隔システムが Web サーバを持っている。そして、実のところ多種多様なコンピュータ上でいまだに 1989 年に動いていたソフトの新しいバージョンを使っているにもかかわらず、Web のインターフェースによりそれらが同じ継ぎ目のない首尾一貫した情報の世界の一部分となっている。日々使う Web 文書でコンテンツをリンクすることは以前に比べはるかに楽になった。

この一見想像を超えるような情報の世界の一方で、

Web アプリケーション間でコンテンツをやり取りするのは、まだまだ驚くほど難しい。もし機械が我々の使う Web コンテンツを理解できたなら、機械がそれらのコンテンツを多様な方法で使うことができる。私の PDA のカレンダープログラムが日付を理解できれば、予定が近づいた時に私にアラームを出すことができる。もし私の電子メールプログラムのアドレス帳が、ある情報が電話番号あるいは電子メールアドレスであると理解できれば、シングルクリックでその電子メールプログラムにその人との連絡の準備をさせることができる。日本にいるときに私のデジタル携帯電話に行き先を与えられたら、その携帯電話はそこへ行くのにどの電車に乗り、どれだけお金がかかるかを計算するプログラムにアクセスできる。

今日、そのような Web アプリケーションを使うための我々の能力は限られている。Web ページをいろいろ見ていく途中で、ある会合について書いてあるページに出くわしたとしてみよう。そのページには、その会合の時間、場所、その会合のまとめ役や参加者へのホームページを含む関連文書へのリンクがある。あなたはその会合に参加することを決め、“Register (参加)”ボタンをクリックする。この時点で、あなたはカレンダーの、会合の正確な日時のところにその予定が入力されていてほしいと思うであろう。あなたはデジタル携帯電話がその会合の住所をダウンロードし、会合の場所に間に合うように到着するための最適な電車の経路を計算してほしいと思うであろうし、連絡先管理ソフトがその会合が終わるまでは、関係者の連絡先情報を持っていてほしいと思うであろう。さらにそれらすべてを一度のマウスクリックで実現したいと思うであろう。

残念ながら、現在それらのことを実現することはできない。実際には我々は自分自身でカレンダーの相当する日時のところを探し、手間ひまをかけて会合の詳細を切り取って貼り付けなければならない。連絡先の詳細をそれぞれの参加者の Web ページからアドレス帳に、自分で住所の部分と電話番号を分けながら手でコピーしなければならない。さらには、デジタル携帯電話の小さなボタンを使って出発場所を入力し、会合場所を入力しなければならない。この状況は、Web 以前の文書システムの問題と全く同じである。端的に言えば、日常生活で使うデータに関してはまだ “Web 前期” 状態なのである。

Semantic Web はそれらの問題に対処し状況を改善できる技術をすでに提供している。我々はそれらの技術のうち、三つについて説明する—複数のデータベースの接続、異なる XML (Extensible Markup Language) DTD (Document Type Definition) あるいは異なるスキーマを

使うアプリケーション間のコンテンツ共有、そして姿を現しつつある重要な Semantic Web アプリケーションである Web サービスの発見と組み合わせを説明する。

2. データの Web-HTML を超えて—

前章では我々の個人的なデータをネットワークする際の困難さを挙げたが、ここでは企業にとってのインパクトを考えてみよう。あなたが、あなたの企業が依存している各種のデータ処理アプリケーションをつなごうとするときに、あるいはユーザが複数のデータベースからの情報を統合しようとするときに、やはり同じ “Web 前期” 状態にあることに気づく。在庫管理システムと会計システムの間にはある重複部分があり、もしそれらのシステムがつながれたら多くの再入力の必要がなくなり、再入力に伴うエラーも減るであろう。今日、あなたはプログラマを雇い、在庫管理システムからデータを取り出し、フォーマット変換をし、会計システムにロードする “glue code” (複数のシステムを結びつけるためのプログラム) を書いてもらう。顧客関係管理 (CRM: Customer Relationship Management) システムが受注管理システムからのデータを使えることに気づいたときも同様である—実際には会社はそのようにデータを使えるようになっていないがために損をしている。何度も何度も、特別目的のインターフェースが一つのシステムから別のシステムにデータを動かすために書かれる。もし数多くのアプリケーションが走っているなら、それらを結びつける組み合せの数は膨大になる。それらを結びつける作業は多くの特注プログラムを必要とし、結果として何人の高給プログラマによる膨大なプログラミング時間を要してしまう。

XML (<http://www.w3.org/XML/>) を使うことにより、状況は改善する。もし、すべてのアプリケーションが XML を使うようにできたら、プログラマは XML を扱うことさえ学べばよく、そうでない場合に使われる特殊な内部用フォーマットはすべて忘れることができる。それは、いくつかの “application glue” (結合プログラム) が XML の変換言語である XSLT (eXtensible Stylesheet Language Transformation, <http://www.w3.org/TR/xslt>) などの XML ツールを使って実現できることを意味する。しかし、厄介な問題は XML を使っても効率的なデータ交換の問題は消えてなくなることである。すべてのアプリケーションの組のうち、結びつける必要のある方向に関して実際にだれかが “XML to XML bridge” (異なる XML 間をつなぐもの) を作らなくてはいけない。それは XML ファイルを二つの異なるアプリケーションからとってきて、ただ単にそれらをくっつければいいというもの

ではない。(XML による) 問い合わせを XML 文書に対して行い、同時にその問い合わせにほかの文書からの制約を加えるためには、ただ問い合わせと制約をくっつけるだけではだめである。それは共通の要素を使ってデータを結合できる関係データベースにすべてが入っている状況とは異なる。

問題は、異なる関係データベースは異なるデータベーススキーマを使って作られており、それらのスキーマがはつきりと記述されていないことである。したがって、例えば〈CX 213〉と呼ばれる XML タグと別のデータベースの〈Income〉と呼ばれるフィールドとを直ちに関係付けることはできない。これに対して提案されている解決法は、スキーマをより明確にすることである。そして、それらのタグを共通の用語に対応付けることである。XML-Schema 言語 (<http://www.w3.org/XML/Schema>) により共通の利益に基づくグループがネットワーク上でスキーマを表現することが可能になる。一企業、あるいは特定の産業領域においてでも XML スキーマの標準を開発することができ、したがって共通の構造を使って彼らのコンテンツを表現できるようになる。残念ながら、時折このことは難しいことが判明する。そして、いろいろなユーザに対する大規模な語彙を開発することは、多分もっと難しいことがある。

ここでの問題は、前に述べたのと同じく「在庫管理システムと会計システムの間にはある重複部分がある」ことである。この「ある重複部分」はおおよそ完全なものではない。在庫管理システムは、例えば、すべての部品がユニークな番号を持つことを期待しているかも知れない——営業の人がそれについては譲らないだろう。ところが一方では、会計システムは一般消費者向けの部品を別のときに別の販売会社より購入することを受け入れるようなデザインになっている。したがって、在庫部品番号は一意に会計システムに対応付けられない。一般に、大きな複数の会社にまたがる(あるいは、競争的な産業内で)唯一のスキーマを定義することは複雑な作業であり、往々にして実りがない。同じデータを表す場合でも、本質的に仕事とデータに対する要望とが異なるために、異なるユーザは異なる視点にこだわる。小さな会社の中、あるいは協調的な産業内ではそういうしたこと(唯一のスキーマを定義すること)が確かに可能ではあるが、それでもサプライチェーン(需要供給関係のある企業たちがなす企業の連鎖)のどこかに同じスキーマを使わない、あるいはスキーマへの要望について完全に合意することができないユーザがいるために、厄介な問題はまず避けられない。

もし我々が異なるスキーマ間、あるいは異なるビジネス

の語彙を使うユーザ間を対応付けようとするなら、XML スキーマを超える能力が必要となる。異なるデータ構造間をマップするためにより強力なデータ表現が必要なことは、コンピューティングにおける自明の理である。例えば、関係データベースに使われる関係演算は、それ以前の(単純なファイル(flat file)による)データベースで使われていた表現のほとんどに比べより強力であり、それゆえその表現はそれ以前のアプローチ間の対応付けに標準的に使われるようになった。データベース間の複雑なマッピングや異種のデータベースにわたる問い合わせを行うために E-R (entity/relations) モデルやオブジェクトモデルなどのさらに強力な表現が必要になる。端的に言えば、より表現力の高い言語が相互運用性の層を一層ずつ上っていくことを可能にする。ちょうど古いデータベースシステムたちが首尾一貫した関係モデルを採用することにより突然それらの間に互換性が成立したように、現在の構造のない Web 上のデータあるいは XML スキーマによる定義でも、関係モデルを実質的に採用することにより、データモデル問題をより簡単に解決することができる。

この理由から、Semantic Web の基本的な構成要素は Resource Description Framework (RDF: <http://www.w3.org/RDF/>) になっている。二つのデータソースからの RDF による情報をくっつけるには、基本的にそれらのファイルたちを一つのファイルにつなげてしまえばよい。その際、同じ Universal Resource Indicators (URIs) に対応する用語たちを結合(join)するだけでよい。RDF ファイルに関する問い合わせをほかの RDF ファイルからの制約を含むことで拡張するには、その制約を問い合わせに付け加えてしまえばよい。XML は elements と attributes からなっているが、それらはファイルにどのようにものが書かれるかを示しているに過ぎない。RDF データは何ものかのあるプロパティ(性質)の値を表す記述からなる。そして、この記述はデータベースの表における一つのセルに正確に対応するものである。RDF の枠組みの中では、関係データベースにおいて使われている考えはすべて同様に使える。例えば、結合(join)やビューなどの演算は一般的なツールを使って簡単に書くことができる。

さて、これで企業でのアプリケーション統合はどうなるであろうか。各アプリケーションからの情報は最初から RDF で出力されるか、あるいは RDF に変換される。任意の問い合わせをこのデータのどのような組み合わせに対しても行うことができる。情報のフィルタも簡単に書くことができ、情報抽出、必要なデータの計算も変換を使って実現できる。次にこのデータはそれを必要とするほかのアプリケーションに入力される。基本的に、問題のサイズは

システムのサイズに対して線形となる。ちょうど Web サーバが、ほかの部分を混乱させることなく、Web の枠に組み込まれるように、新しい RDF アプリケーションも、システムのほかの部分を混乱させることなく、情報を提供し、そして使う。膨大な数の特注データインタフェースは一見魔法のように消え去る。文書同士のリンクのように、データが Web をつなぐことができるのである。

3. Semantic Web サービス

—プログラムとデータを結び付ける—

データベースが現在の Web に RDF なしでは簡単に統合できないことは、プログラムを Web に持ち込む際にも同様に当てはまる。プログラムは簡単に Web に統合されるように見えるかも知れない。結局のところリンクをクリックするとよく Java や Flash のプログラムがダウンロードされ、実行されるではないか。しかし残念ながら、このアプローチは多くの e-business アプリケーション、特に企業間 (b2b) アプリケーションではうまくいかない。他人のプログラムをロードして自分のマシン上で走らせるることは、他人のアプリケーションから情報を自分のアプリケーションを持ってくることとはかなり異なる。

販売会社から部品を買い、大きな運送会社に輸送を頼み、それらの部品をいくつかある生産拠点のうち輸送の時点で最も生産余力があるところに送ってもらおうと思っている企業の場合を考えてみよう。さらに、彼らはこの取り引きを Web 上で人の介在を最小限に抑えて処理したいと思っている。営業の人がオーダを入れると、サプライチェーンが動き出す！といった具合に。これはデータベースの時の問題と似ているが、データベースだけではなく、大抵は業務管理のためにいろいろ異なるプログラムが使われていて、問題はより難しくなっている。もっと悪いことに、これらのプログラムは特定目的のコンピュータ上、あるいはファイヤウォールで守られた中で動いているかも知れない。第一に解決すべき問題は、これらすべてのプログラムを Web 上でどのように一緒に動くようにするかである。すなわち、プロトコルとそれら各種プログラムが実現するサービスの記述を提供することである。

数多くの主要な企業がまさにこのことに一生懸命取り組んでいる。そして、その結果 “Web サービス” の市場は非常な勢いで拡大している。この市場は最近の Web 関連ビジネスでの最も成長の速い分野の一つである。例えば、ガートナーグループは近い将来「Web サービスを使うことにより、コスト削減につながり、IT プロジェクトは 30% の効率化を見るだろう」と予測している。Web サービスの市場規模予測は何十億ドルから始まり、そこからさらに

大きくなるとされている。

結果として、システムが何をするかを記述する手段を標準化するために、新しいプロトコルや言語が急速に開発されている。SOAP (Simple Object Access Protocol, <http://www.w3.org/TR/SOAP/>) と呼ばれる XML に基づくプロトコルが、プログラムから Web 上にあるほかのプログラムを呼び出すために開発された。さらには、新しい Web サービス記述言語や Web サービスアーキテクチャが現れつつある。これは World Wide Web Consortium の Web サービス活動 (<http://www.w3.org/2002/ws/>) での主要な中心課題となった。

Web サービスが広く使われるようになったとき、Semantic Web の技術は Web サービスをより有益なものとするだろう。Web 上では、多くのサービスプロバイダが彼らのサービスを膨大かつ多様なサービスユーザーに宣伝できる必要があるだろう。仲介したり、サービスユーザーとサービスプロバイダを自動的に引き合わせたりすることは難しく、その実現はデータベースのときの語彙の対応付けと同じような課題に直面している。現状では、サービスの入力、出力、ポートなど、サービスを互いに呼び出すための記述は実装されているが、サービスが何をするかの記述は “content” フィールドにまかされていて、そのフィールドは (XML ではあるが) 勝手に記述してよいことになっている。したがって、問題はデータベースの場合と同様である。“content” フィールドについて合意がなければ、多様なユーザコミュニティ間の数多くの異なる対応付けを用意する必要がある。事前に調整したグループ内ではある種の合意に到達できるかも知れないが、それでは異なるスキーマを使う外部のサービスプロバイダを見つけるのが難しくなる。また、サプライチェーンにわたって相当量の事前合意が必要になり、それはシステム硬直化の原因になりかねない。

データベースの場合と同じように、ここでも Semantic Web 言語の表現力の高さが助けになる。RDF の拡張である RDF-Schema (<http://www.w3.org/TR/rdf-schema/>) および新しく現れつつある Web オントロジ言語 (<http://www.w3.org/2001/sw/WebOnt/>) は、用語が互いにどのような関係にあるかを表現するための階層構造とシソーラス (類義語・反義語などの語彙索引) を構築することを可能とする。例えば、Web 上にスキーマを構築し、「輸送」というイベントがあること、「郵送」は「輸送」の一種であること、「翌日郵送」は「郵送」の一種であることなどの情報を表現できる。新しいサービスは単に自分自身の記述をシソーラスにくっつけるだけで、この中にリンクすることができる。それができるのは (データベ

ースのセクションで議論したように）くつつけた RDF 文書もそれ自体ちゃんとした RDF 文書になっているからである。

さらに言えば、サービス記述をほかのサービス記述に結びつける情報は、両者に共通する自然言語での用語があるという偶然に頼る必要はない。それは別のユーザあるいは開発者、別のソース、あるいはだれかの Web ページでたまたま見つけた事実であるにせよ、外部の情報源が対応付けの情報を与えることができる。それゆえ、「私の共著者が “lorry”（訳注：貨物自動車、トラック）と呼んでいるものはあなたが “truck” と呼んでいるものである」と私が言うことができ、それ以降、その RDF-schema の記述をくつければ “lorry” と “truck” の関係を見つけることができる。さらに RDF-schema や Web オントロジ言語などの新しい言語はより複雑な対応付けを表現し、くつつけることができる。したがって、“Nissan-Maxima” は自動車 (“automobile”) で、そのタイプは高級車 (“luxury”) で、その生産地は日本であることを表すことができる。そこに Nissan ディーラーのサービスの宣伝がリンクされれば、以上の性質のいざれでも Nissan ディーラーのサービスを探し出すことができる（訳注：例えば「日本製高級車のディーラーサービス」という検索で、その Nissan のディーラーサービスを見つけられるようになる）。

Semantic Web によってもたらされる Web サービスのそのほかの質の向上は、必要とするそのサービスがすぐに利用できない時に分かる。例えば、キャンディがいっぱい詰まったちょっとしたギフトボックスを売る企業が、チョコレートでできたハートを 100 グロス（1 グロス = 12 ダース = 144 個）とキャンディ棒（candy canes）を 100 グロス購入し、それら両方を名古屋にあるプラントに包装のために輸送したいとしよう。いくつものハートチョコレートの販売会社、いくつもの棒飴の販売会社、そしていくつもの運送会社を見つけることができる。しかし、どのサービスも一つだけでは、三つの要望をいっぺんに満たすことはできない。我々は、本質的に、三つのサービスから一つの新しいサービス（棒飴—ハートチョコレート—輸送サービス）を組み立てることになる。Semantic Web サービス記述の合成により、同じ用語を使うことについて事前に合意をしなくともこれらのサービス記述をまとめることができ

る。次に Semantic Web アプリケーションを使うことにより、効率的かつ効果的な方法でサービスを合成し、求めるゴールを実現する方法を検討することができる（例えば、チョコレートを輸送するには冷蔵しなくてはならないなど、チョコレートについての付加的な情報は効果的な輸送を確実にするのに役立つ）。複雑なサービス合成はまだまだ研究課題ではあるが、いろいろなサービスの入力と出力をマッチさせることによる基本的な合成は、今ある Semantic Web のツールを使って行うことがすでに可能である。

さて Semantic Web を作ることは、本当に特定の専門家たちのための未来的な研究ビジョンであろうか。すでに見てきたとおり、答えは “No” である。Semantic Web は World Wide Web のように、しっかりと確立された考えを元に、それらをインターネット上で相互運用可能な形で動くようにすることにより、徐々に大きくなっていくことができる。それは標準によって実現され、そしてそれらの標準こそが World Wide Web Consortium に関するすべてである。我々はデータに関する関係モデルやサービス合成の方法論を発明しようとしているわけではない。むしろ、我々は Web に、我々がすでに大体においてどのようにやるかを知っていることを持ち込もうとしている。すなわち、我々は分散システムの中でそれらが一緒にうまく動くようしている。そして、そのことを人がすべての接続を一つ一つ注意深く調整する必要なしで実現しようとしている。

端的に言えば、Semantic Web にはこの記事の中で議論した問題に取り組むための言語やツールがすでにある。会合についての Web ページをクリックすると、コンピュータがそれが予定であることに気づき、正しい情報を全部拾い上げ、内容を十分理解し、すべての適切なアプリケーションにその情報を送れるようになる日も近い。さらに、それらのアプリケーションは直接（Web サービスを使って）呼び出され、人間の介在はほとんど必要なくなるであろう。このようなデータとプログラムの統合のための市場は巨大であり、Semantic Web の技術を活用しようと選択する企業が最初に利益を受けるものになると信じている。

（平成 14 年 7 月 30 日受付）