14-E009

設計・製造データの活用と流通に関する調査研究

製品表現のための先進技術動向研究報告

平成１５年３月

財団法人日本情報処理開発協会
電子商取引推進センター

協力:電子商取引推進協議会

KEIRIN

この報告書は、（財）日本情報処理開発協会電子商取引推進センターが競輪の補助金を受けて、電子商取引推進協議会（ECOM）の協力を得て実施した事業の成果を取りまとめたものです。

ま　え　が　き

わが国の製造業は海外資本との提携や生産拠点の海外シフト、中小企業のデジタルデバイドなど産業構造的に大きな問題に直面しており、その中で今後は、いかにIT化によって市場又は顧客主導の製品開発や生産を行っているかが大きな課題となっている。

本事業では、製造業の競争力強化を踏まえ、ネットワークを活用したものつくり（B2B e-Engineering）をテーマとして取り上げ、そのあるべき姿を検討した。

（１）産業の視点、企業の視点から設計・製造における情報の活用と流通の取り組みを捉え、必要な要件を整理する。

（２）設計・製造データの流通のための製品表現のための先進的技術の動向と、海外における企業間協業の現場の事例。

本書「製品表現のための先進技術動向研究報告」は上記（２）についての報告書であり（１）については、報告書「B2B e-Engineeringに向けた検討結果の報告」を参照ください。

（２）に関して今年度は以下の活動を行った。

ａ：委員会アドホック部会を設け、情報表現のための先進的技術として、人にとってわかりやすい情報記述であり、機械可読形式の情報記述言語・XML 技術系の新しい流れであるセマンティック Web と STEP を融合することによる製品ライフサイクルに関わる情報処理の新しいビジョンを掲げ、製品ライフサイクルにおける視点を委員の専門の多様性に求め、それぞれの立場からのＸＭＬ技術系、セマンティックＷｅｂ、ＳＴＥＰ技術系などの分野研究を報告いただき検討した。

ｂ：海外（米国）企業事例調査活動を行った。

本報告書にて、（２）－ａは第１部に、（２）－ｂは第２部にまとめた。

なお、内容は、委員の方々に報告頂いた内容を事務局でまとめたもので、文責はすべて事務局にある。また、本報告書にある会社名、団体名、製品名、システム名などは、一般にその会社や開発元の登録商標である（本文中にはTM、(R)は明記していない)。

平成１５年３月

財団法人日本情報処理開発協会
電子商取引推進センター
電子商取引推進協議会

委員名簿（敬称略）

井越　昌紀	東京都立大学
綾　日天彦	三井造船（株）
太田　吉美	技術知識基盤構築機構
大野　邦夫	ﾄﾞｺﾓ･ｼｽﾃﾑｽﾞ(株)
岡田　　宏	日揮株式会社
児西　清義	ASP ｲﾝﾀﾞｽﾄﾘ･ｺﾝｿｰｼｱﾑ･ｼﾞｬﾊﾟﾝ
鈴木　亮平	（株）トヨタケーラム
橋田　浩一	産業技術総合研究所
森　　福瑞	日本総合システム(株)
綿貫　啓一	埼玉大学
河村　幸二	旭エンジニアリング(株)
秋山　雅弘	（株）アルモニコスコンサルティング
上石　幸拓	(株）アプストン技術開発
大貫　一志	ERP 研究推進ﾌｫｰﾗﾑ
小笠原　剛	TDK（株）
小倉　謙逸	日本アイ・ビー・エム（株）
坂井　佐千穂	セイコーエプソン(株)
佐々木　安夫	富士電機（株）
田中　敬昌	デジタルプロセス(株)
中野　宣政	三菱電機ﾒｶﾄﾛﾆｸｽｿﾌﾄｳｴｱ(株)
成子　由則	（株）トヨタケーラム
藤川　博巳	ﾌﾟﾛｾｽﾁｪｰﾝ研究所
松井　啓之	京都大学
三谷　　脩	（社）日本電気計測器工業会
宮崎　知明	(株）富士通総研
村上　憲也	ＮＴＴデータ先端技術(株)
山際　悦治	ソラン(株)

（事務局）

寺門　義孝

調　　敏行

吉川　忠克

目　　次

ーまえがき
ー委員名簿

□ 第１部：「セマンティックWebとSTEPの融合に向けて」……1

■ セマンティックWebとSTEPの融合に向けて……3

■ 知識情報処理からセマンティックWebへの期待…… 6

■ 企業間電子商取引の拡大とオープン化の課題……18
１．１　ユビキタスな時代……18
１．２　現状……19
１．３　ソリューションビジネス……20
１．４　ステークホルダー……22
１．５　スパイラル効果実現のために……23

■ アノテーションに基づくセマンティクウェブ……27

■ XMLとペトリネットグラフに基づく最適作業支援の一方法……33
１．はじめに……33
２．作業モデルの記述……33
３．実験……37
４．結論……39

■ 熟練技能伝承システムの構築 ……40
１．緒言……40
２．技能伝承法……40
３．技能伝承システムの構築……45
４．結言……50

■ メタデータ概念と記述……53
１．はじめに……53
２．メタとは何か……53
３．記述の理論とシステム展開……61
４．OMGにおけるメタデータ、メタモデルの展開…… 62
５．セマンティック・ウェブ……64
６．まとめ……65

■ セマンティックWebの課題と携帯電話から見た可能性……69
１．はじめに……69
２．セマンティックWebの概要とこれまでの経緯……69

３．セマンティックWebを取り巻く環境…… 73
４．携帯電話から見たWebの可能性…… 77
５．あとがき…… 79

■ Gellishについて…… 81

■ 制約自然言語の提案…… 89
１．はじめに…… 89
２．STEPの概要…… 90
３．Webの進化と技術知識基盤構築…… 104
４．STEPとWeb技術の融合…… 111
５．オントロジーとしての制約自然言語…… 116
６．おわりに…… 129

■ 仮説：STEPはセマンティックWebのキラーアプリになる…… 132
１．プロローグ…… 132
２．CADの進化とSTEP…… 132
３．Webの進化とセマンティックWeb…… 134
４．STEPはセマンティックWebのキラーアプリになる…… 135
５．X-STEPの提案…… 136
６．教育システムの刷新…… 138

■ 総括と今後の方針…… 141

□ 第２部：「プロダクト・データ活用企業事例」…… 143

■ ハーレーダビッドソン社（USA)…… 145

■ デーナ社（USA)…… 154

第１部：
「セマンティックWeb とSTEPの融合に向けて」

セマンティック Web と STEP の融合に向けて

主 査　井 越 昌 紀

STEP（Standard for the Exchange of Product model data）の開発が 1984 年 7 月に開始されてから早くも 19 年が経とうとしている。異なった CAD システムにおけるプロダクトモデルデータの交換のために、インタフェースとして STEP 仕様の交換フォーマットを提供する CAD ベンダーは着実に増加している。しかしながら、STEP のもともとの構想は、「製品のライフサイクルを通して、完全性、統合性を失うことなく、中間的な形式でコンピュータ化された製品モデルを構成する情報の記述を可能にする国際標準の確立」であり、単なるデータ交換仕様と捉えることではない。しかし、製品のライフサイクルを通して、情報の完全な記述をすることは難しい問題であった。現在の STEP の評価できる点は、特定の AP(Application Protocol)を開発するための方法ある AAM(Application Activity Model), ARM(Application Reference Model), AIM(Application Interpreted Model) の一連の流れによる開発方法論と、情報モデルを記述する言語 EXPRESS により、対象のスキーマとインスタンスを記述するオブジェクト指向記述方式を開発したことであると考えられる。しかしながら、情報モデルの記述で人に分かりやすい部分である ARM までと、テクニックを要する AIM へのマッピングの問題、およびかなりな知識を要求される割にはその効用が製品モデル交換に留まっていると考えられる STEP の世界の孤立性が評価を落としていると考えられる。

一方、情報のデジタル化が進む中で、計算機では CAD 等で扱う情報よりもずっと広い情報の記述が行われてきた。もともと、製造業等の STEP が扱うエンジニアリング分野においても、製品の企画、設計、製造、運用、保守、リサイクル、廃棄のライフサイクル全体を考えた場合、関連する情報は多種多様である。とくに、ライフサイクルを情報の流れとして考えると、その情報を総合的にコントロールするのは常に意思決定機能を持った人であるので、情報の流れのどの断面をとっても人とって分かりやすい情報の記述言語が必要である。人にとって分かりやすいとは、その情報が人に意味を伴って可読な形で保存されていることである。一方、機械にとっては情報のあいまいさを排除し、人が築き上げてきた問題解決の理論やモデルを記述しやすい情報の記述言語である必要である。この人にとっても機械にとっても可読である言語がマークアップ（タグ付け）言語であり、XML 技術系である。

タグ付け言語は、タグ、属性、属性値、およびコンテンツ部からなっている。コンテンツ（データ）部には、階層的にタグ付け言語が記述可能で、これが構造化された記述方式を提供している。タグ、属性、属性値は一般にコンテンツ（データ）のメタ情報であるメタコンテンツ（メタデータ）として見ることができる。XML は機械にとって可読であるが、理解可能と

は言えなかった。メタデータを利用することにより、機械にとっても理解できる情報の記述か可能となることを主張し、意味の体系をオントロジーとして形成し、ウェブの情報を計算可能な世界とすることを目指した技術がセマンティック Web と捉えることができる。

STEP の世界の孤立性を脱却するために、すなわち、STEP という情報の島をほかの製品ライフサイクルに関連する情報や人にも開放するために、STEP の分野にも XML 技術の利用が始められた。これが、情報交換のための規格 P28 と、UML 等のほかの言語との情報交換を目指す P25 である。しかしながら、この程度ではなかなか STEP の世界がライフサイクルにおけるほかの情報の流れと合流しない問題を抱えている。

本委員会の目的は、情報の島を結合する能力を持つと考えられる XML 技術系の新たな流れであるセマンティック Web（SW）と STEP を融合することにより、人の能力の柔軟性をさらに向上させ、人と機械とが協調して製品のライフサイクルに関わる情報処理の新しいビジョンを検討することである。本年度は、製品のライフサイクルにおける視点の多様性を委員の専門の多様性に求め、それぞれの立場から XML 技術系、セマンティック Web、STEP 技術系などについて意見を交換した。

児西委員の「知識情報処理からセマンティックＷｅｂへの期待」では、知識地図を描くことによって SW 技術の方向を欠落少なく議論できるとしている。

岡田委員の「企業間電子商取引の拡大とオープン化の課題」では、業務プロセスの中に STEP が位置づけられ、設計・製造データのオントロジーに基づく検索の効率化に期待している。

橋田委員の「アノテーションに基づくセマンティックウェブ」は、マルチメディア・コンテンツにアノテーションを付ける必要性と、そのためのオーサリングツールについて述べている。

井越委員の「ＸＭＬとペトリネットグラフに基づく最適作業支援方法」では、プロセスのペトリネットモデルと自然言語の融合によって、人と機械の協調が図れる一例を示している。

綿貫委員の「インターネット/マルチメディア技術を用いた熟練技能伝承システムの構築」では、暗黙知としてのノウハウやスキルの技術の伝承にマルチメディアを積極的に利用する試みが述べられている。

大野委員の「メタデータ概念と記述で」は、SW で重要な働きをするメタデータの概念について、メディアの歴史や意味に絡めて哲学的視点からメタ概念を解説している。

大野委員の「セマンティック Web の課題と携帯電話から見た可能性」では RDF や OWL などの SW の理論を解説し、かつての AI との違いと SW の可能性について述べている。

森委員の「Gellish について」では、オランダで開発された、STEP AP モデルを拡張して SW へ進化させていく Gellish について紹介している。

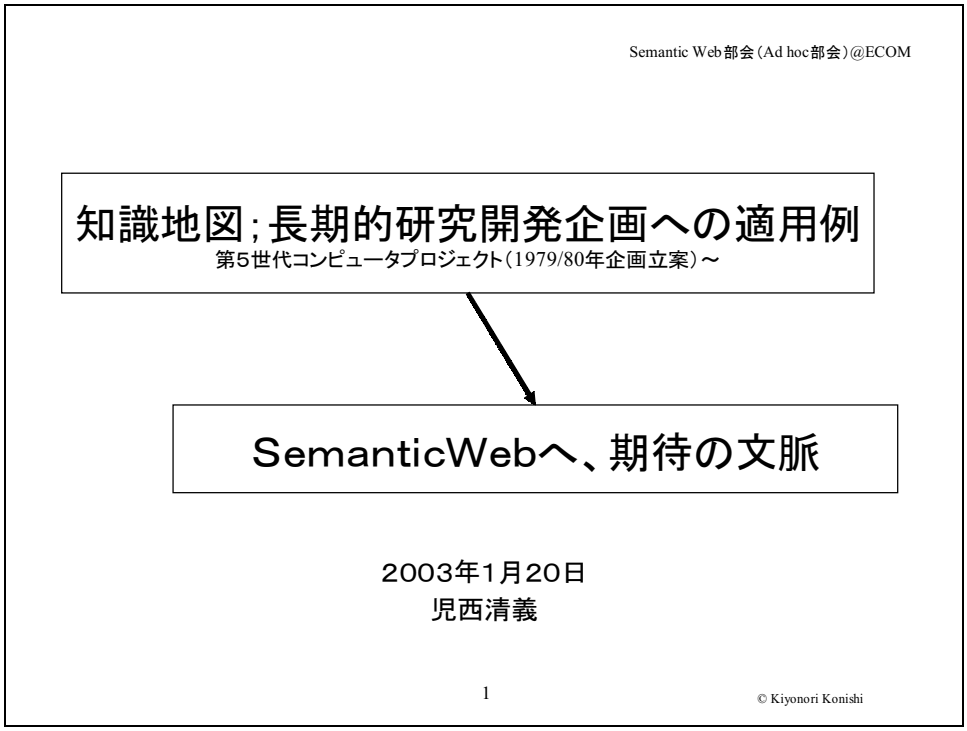
太田委員の「STEP と Semantic Web との融合を目指して」では、モデルの記述に適する GPM をベースとした制約自然言語の可能性について述べている。

綾委員の「仮説：STEP はセマンティック Web のキラーアプリになる」では、SW の提唱者 Tim Berners-Lee の階層モデルを参考にして、人と機械の協調を目指す新たなモデルの試案について述べている。

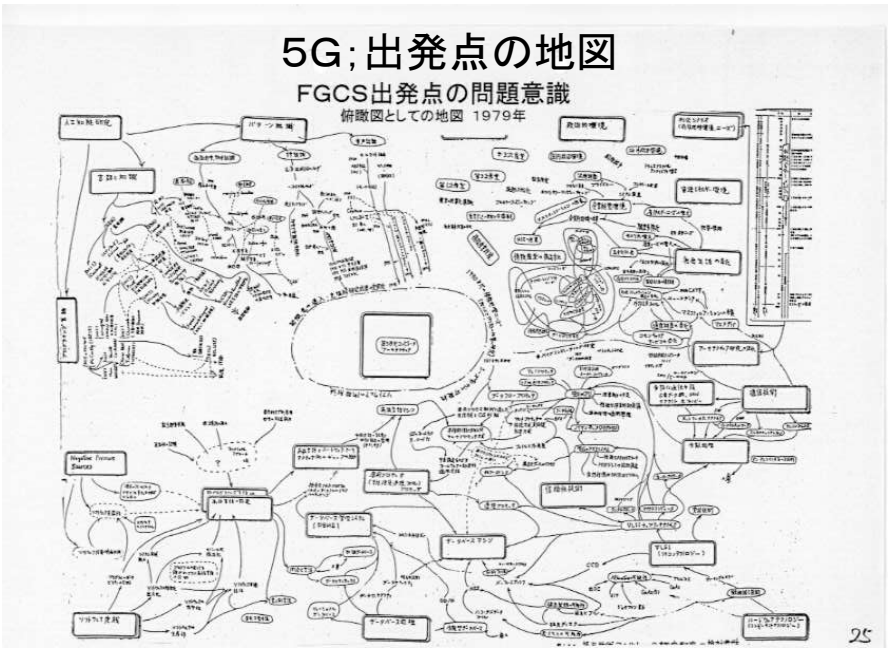
以上のように、本年度は各委員からのプレゼンテーションおよび討議を行い、綾委員の論文をベースに次年度も引き続き議論を展開し、融合のビジョンを検討することとした。

知識情報処理からセマンティックＷｅｂへの期待

委　員　　児　西　清　義



1979 年、1980 年というのは私がＮＴＴから通産省に出向して第 5 世代コンピュータプロジェクトを企画し、そのプロジェクトをプロデュースしていた時期である。このプロジェクトの立上げ過程で知識地図とシナリオプラニングという手法を考案した。本日は知識地図の説明よりもこの 20 年の間ライフワークとして節目節目で描き続けてきた知識地図を紹介しながら何故、今 Semantic Web に期待するものがあるのかを説明したい。



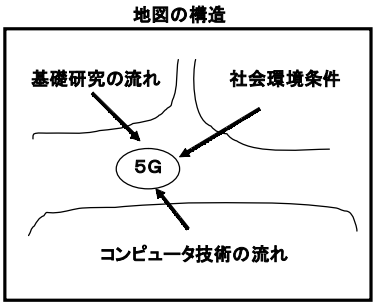
2

© Kiyonori Konishi

「５Ｇ；出発点の地図」は第 5 世代コンピュータのプロジェクトを企画する委員会の第１回の会合で私が委員の先生方にご説明した知識地図である。この地図では 10 年先の 1990 年代に使われる画期的なアーキテクチャのコンピュータを定義する、つまりプロジェクトの目標を定義するために検討しなければならないテーマを関連付けている。地図の左上の部分はパターン認識、知識と推論の言語などコンピュータ基礎理論の研究の流れを描いている。右上は 10 年先の社会環境のシナリオを考えるときに検討すべき事項の相関図である。地図の下半分がコンピュータアーキテクチャの新しい技術の芽やソフトウェアの問題などコンピュータシステム全体の技術を俯瞰するためのものである。この委員会では第 5 世代コンピュータの姿、開発目標を定義するために１年間をかけ、次の 1 年間で目標を実現するための研究開発プログラムを定義すると言う枠組みで進めた。この地図がその出発点である。

5G;地図(俯瞰図)の構造

- 非ノイマンコンピュータという革新的な目標→次々世代
  - 10年先;1980年→1990年代
- 地図の構造(右図)
  - 10年先の社会環境条件
  - 基礎研究の流れ
  - 現時点のコンピュータ技術の流れ
- 仮説;この3つの流れがぶつかり合う場所に10年先のコンピュータの姿があるはず
  - 3つの流れの遭遇点を議論

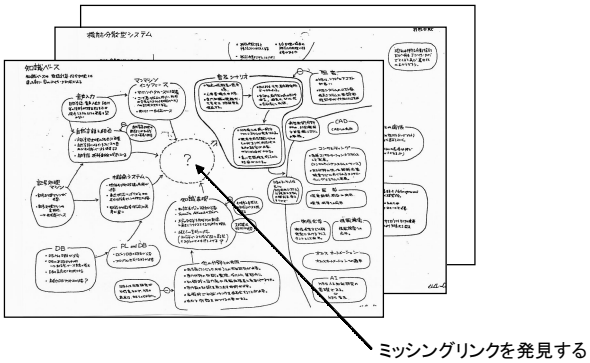


3

© Kiyonori Konishi

この知識地図の構造を簡単に分かりやすくしたものが「５Ｇ；地図（俯瞰図）の構造」である。この地図に表されているように知識地図を使ってブレーンストーミングを行うときに中間遭遇モデルという考え方(基本的なモチーフ)を多用する。様々な流れが出会う中間点(中間遭遇点）の姿を描くことが一つのポイントになる。

個別テーマ毎の地図

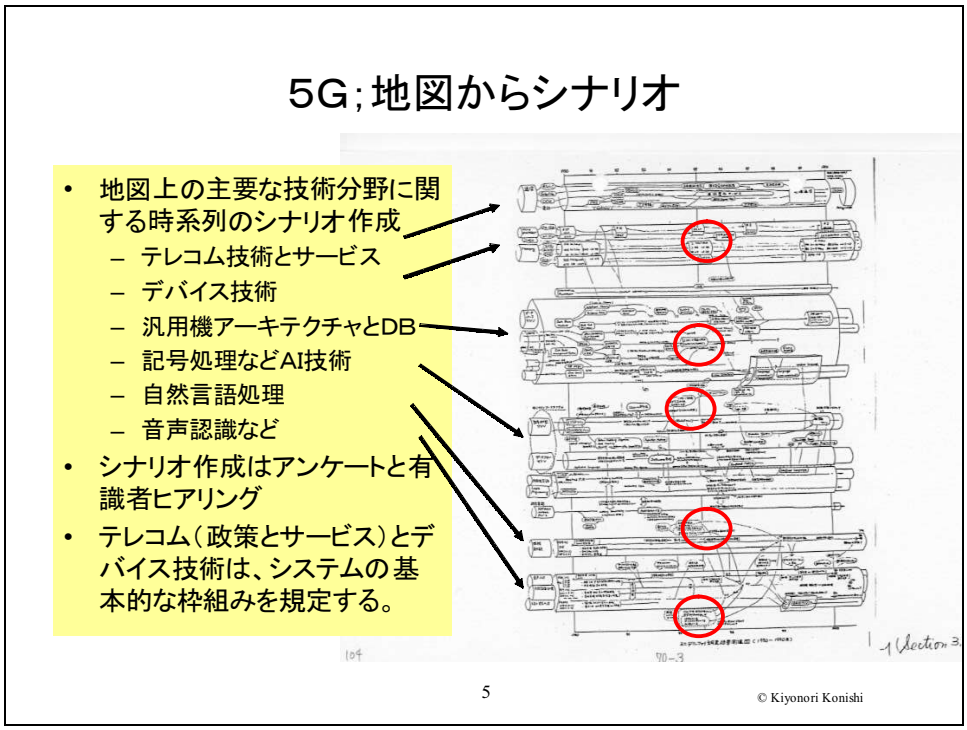


ミッシングリンクは研究開発課題

4

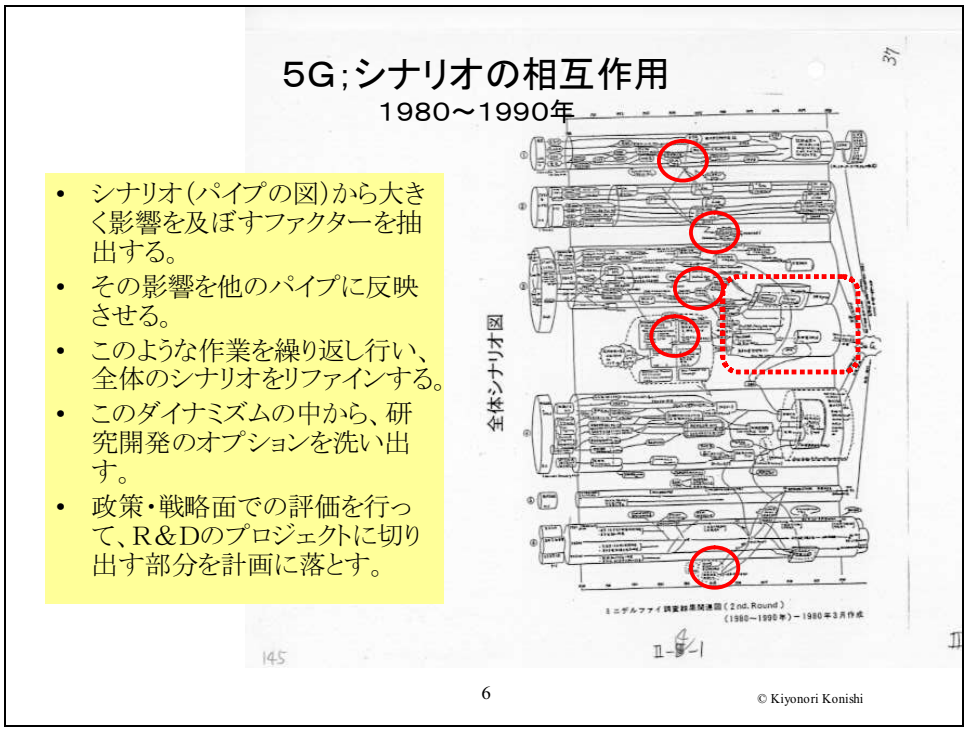
© Kiyonori Konishi

かくして委員会の活動が始まったが、ともすると委員会というのは勉強会になってしまう。具体的なプロジェクトをプロデュースする立場からはもっと精力的に情報を集めて企画作業に役立てることにしたい。そのような発想から委員会の体制に入っている先生方、若い研究者などの問題意識とビジョンを明確にしたいと考えた。この目的のためにアンケート調査を行った。このアンケートは「出発点の地図」上にあるキーワードに加えてその後の検討の中から浮かび上がってきたテーマを回答者が選択して、そのテーマに関して将来そのテーマがどのようになるのかを予測する形式のアンケートにした。この時点で委員会、分科会、その配下のWGなどを含めると 60 人から 70 人もの研究者が参加していた。その大量のアンケート結果をまとめる過程で定性的な記述回答の中からなんとかビジョンを導き出したいと考えて工夫したものが知識地図によるビジュアル表現である。この「個別テーマ毎の地図」はそのような個々のテーマに関する定性的な意見の広がりを表現している。このような地図を描きながらブレーンストームを行うときによく「論理がつながらない箇所（ミッシングリンク）」に出会う。リンクが繋がれば画期的なシステムが実現する。地図を描くときにこのような箇所をわざと空白のまま残すことにした。このような空白部分は研究開発企画の文脈の中では研究テーマが隠されている空白ということになる。この空白を繋げるプログラムが研究開発プロジェクトのシナリオということになる。



このような視点で知識地図を描き、次に 1990 年代に向けてのシナリオを作成する作業にとりかかった。その作業から導かれた最初のアウトプットが「５Ｇ；地図からシナリオ」である。ここでは電気通信、デバイス技術、コンピュータアーキテクチャ技術とデータベースシ

ステム、記号処理技術とアーキテクチャ、自然言語処理、パターン認識や音声合成技術などのシナリオをパイプの形で表現した。このパイプの左端が 1980 年、右端が 1990 年を表している。シナリオプラニングの視点から重要なポイントは、これら複数のシナリオ図を並列に並べて作業にとりかかることである。



複数の技術ドメインのシナリオを隣り合う形に並べて検討すると面白い形で議論が発展していく。つまり技術シナリオが相互に影響をしあうということに気づき、それら他の技術進歩の影響を別のの技術シナリオにフィードバックさせる形でシナリオを発展させていく。その結果が「５Ｇ；シナリオの相互作用」である。このシナリオでは、最初に描いた「５Ｇ；地図からシナリオ」に比べて大きく変化するダイナミックなシナリオが出来あがってきた。例えば、デバイス技術の予測から 1980 年代の半ばに実現されるマイクロプロセッサの処理能力が 1970 年代末の汎用小型コンピュータ並のパワーをもつことが分かった。その結果、汎用コンピュータのシナリオは 1985 年前後から大きく膨れ上がり、汎用大型の世界ばかりではなく、枝分かれして分散処理の世界と広がるというシナリオを描くことができた。

5G；政策判断＠1980年

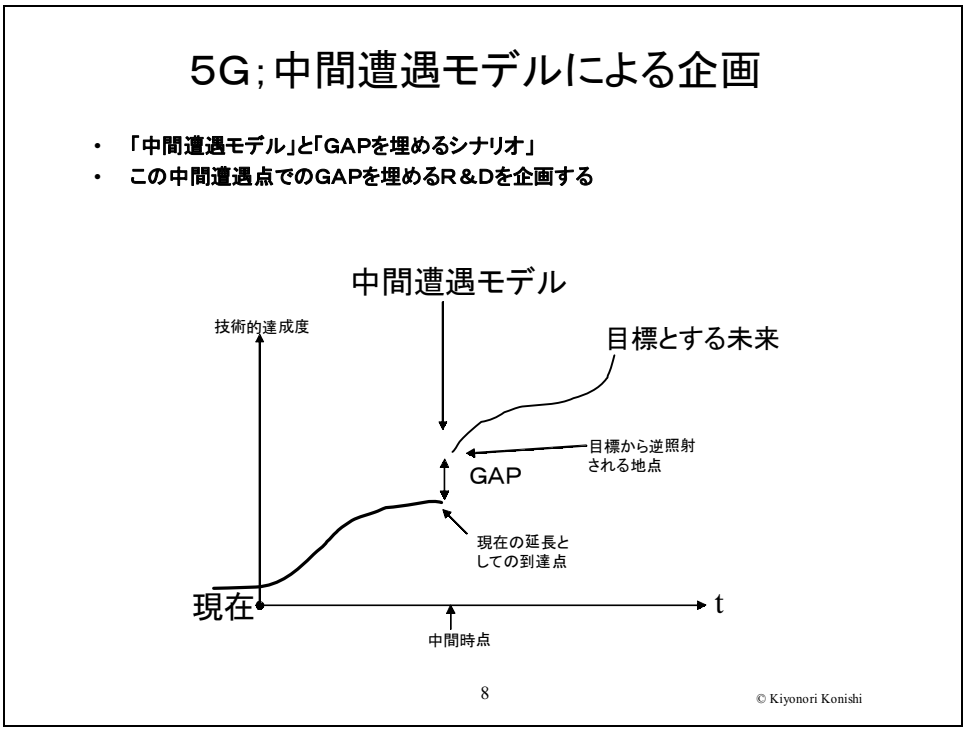
- 1980年代の半ばに大きな変化
  - 前頁のシナリオ図から自明
- 分岐点で考えられる方向
  - メインフレーム＋分散
  - 分散システム
  - 新しい応用分野(OA,CAE, etc.,)
  - 全く新しい応用
    - Multimedia processing
    - Knowledge information processing
    - Natural language
- 政策判断
  - 民間で出来るR＆Dは対象にしない
  - 最も挑戦的な研究対象（10年プロジェクト、政府予算）
  - 知識情報処理システムの開発へ
- 発想の転換・・・企画における「転-1」
  - 処理メカニズムの議論から処理する対象に目を向ける
    - 文字・数値の処理から知識、自然言語、マルチメディア
    - この処理対象の性質から処理系を考える
    - 文字・数値ではなく「知識」の処理へ
- 発想の転換・・・企画における「転-2」
  - 予測する未来から創りだす未来へ
    - 予測する部分シナリオを紡いで全体のシナリオを作成するのではなく、トップダウンに切り替える
    - 知識の処理というビジョンからアプローチする
  - **中間遭遇モデル**
    - 次頁の図

7

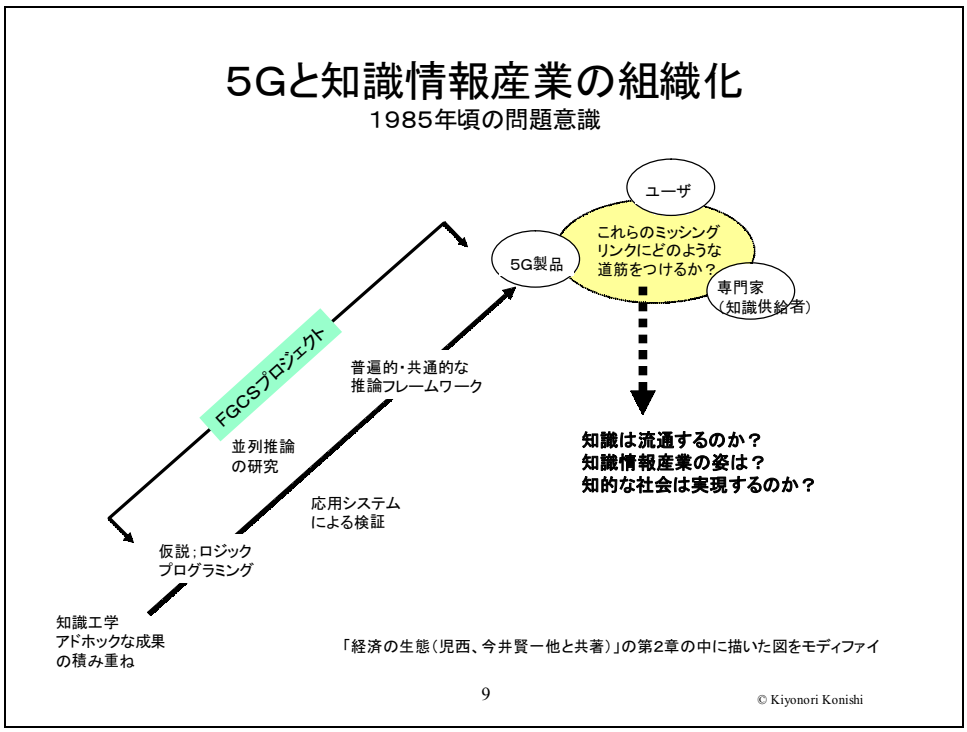
© Kiyonori Konishi

1990 年に向けてのシナリオを得ることが出来たので、次に第 5 世代コンピュータプロジェクトの目標をどこにするのかという政策判断が必要になった。そのときの政策判断のポイントを表すのが「５Ｇ；政策判断＠1980 年」である。ここで考えられるオプションは、①メインフレーム＋分散、②分散システム、③新しい応用分野（OA, CAE, etc.,）④全く新しい応用（マルチメディア、知識処理、自然言語処理）などであった。国家プロジェクトであるので民間で進められるものはその対象にはしない。もっとも挑戦的なテーマに取り組んで情報処理の新しい文化を創りたい。このような議論を経て知識情報処理の実現を狙う第 5 世代コンピュータプロジェクトがスタートした。

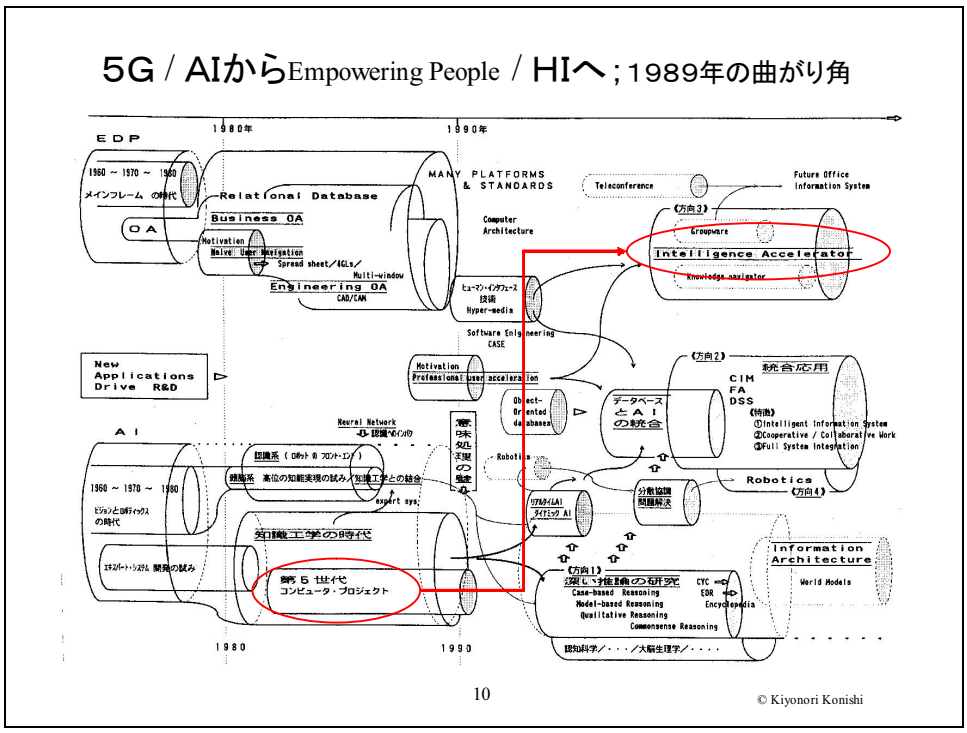
このプロセスを経験して企画というクリエイティブな作業に必要な発想の転換を経験した。発想の転換その 1 が「処理メカニズムの議論から処理する対象の性質に目を転じる」ことである。ともすると新しい技術の芽など処理するメカニズムの検討と議論の方が面白いのでそちら側からのアプローチしがちである。しかし、本質的なのは処理しようとする情報の性質と問題の性質である。第 5 世代の立案過程でも「文字・数値ではなく、知識へ」という転換を明確に意識してからはすっきりと研究開発プログラムの詳細化に取り組むことが出来るようになった。もう一つの発想の転換は、長期プロジェクトの立案においては「予測する未来から創り出す未来へ」という発想の転換である。ここからもう一つの中間遭遇モデルの考え方が導かれる。最初の中間遭遇モデルが知識地図上で複数の流れが出会い、合流するという空間的な中間遭遇点であったのに対して時間軸における中間遭遇点である。



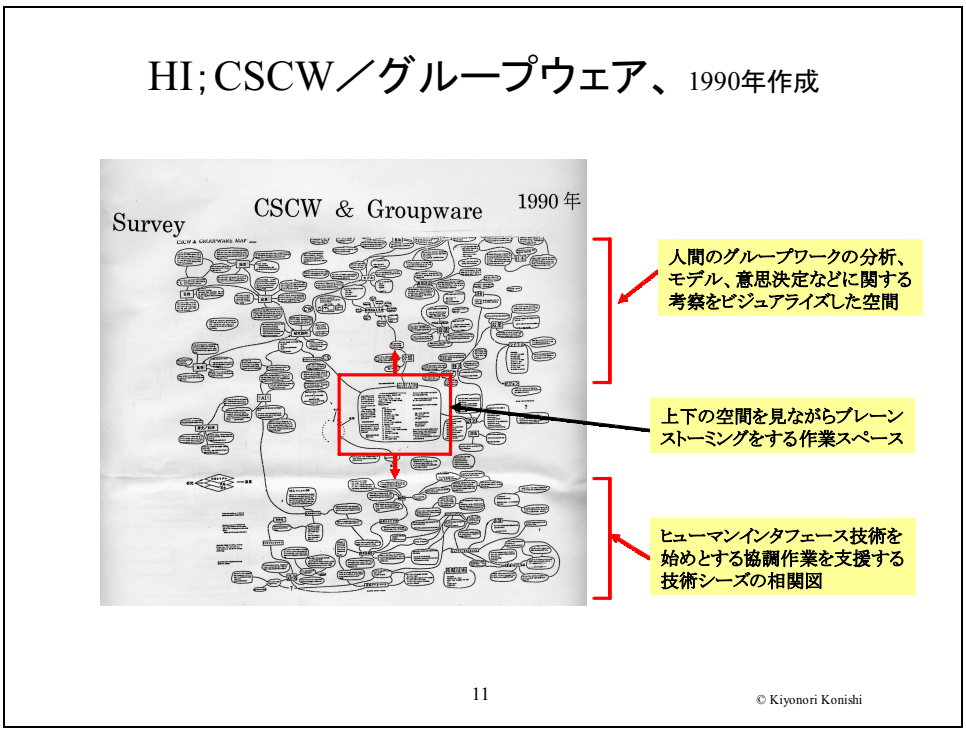
この考え方を「５Ｇ；中間遭遇モデルによる企画」に示す。この考え方のポイントは現在と「目標とする未来」の中間点を考える。この時にこの中間点を現在からの延長として予測するのではなく、目標とする未来から中間点に逆照射される点としてとらえることである。そうすると現在の延長線上から到達する地点とビジョンから逆照射される地点の間に当然のこととしてＧＡＰが存在することになる。このＧＡＰを埋めるシナリオが目標とする未来を実現するためのプログラムの第一歩ということになる。



このようにしてあのラディカルな国家プロジェクトをスタートさせたが、一つの気がかりが残った。それを示すのが「５Ｇと知識情報産業の組織化」である。というのは第 5 世代のアーキテクチャが実現されたとして、知識はどのように流通するのかという問いかけである。これについては、今井賢一先生や坂村健らと共著した「経済の生態」の中に書いた小論文「システム研究開発と組織」の中で問題を提起した。その内容をこの図に表現している。第 5 世代の延長の上で知識が流通して知識情報社会が発展し、それを支える知識情報産業が強くなって日本が強くなるということをめざした。ところが第 5 世代のフレームワークにその視点からの議論が欠けている。今回の Semantic Web の研究会に参加した理由はこの時からの問題意識の答えを探るためでもある。1990 年代から始まったインターネットの時代になってテキストという形態の知識が流通するようになってきた。この時代を経て Semantic Web の時代がくるとすれば知識はどのように流通しどのように創造的な世界になるのか、社会と産業の姿はどのように変化していくのか、90 年代を経て今やそれを議論できるだけの状況になってきたのではないか。



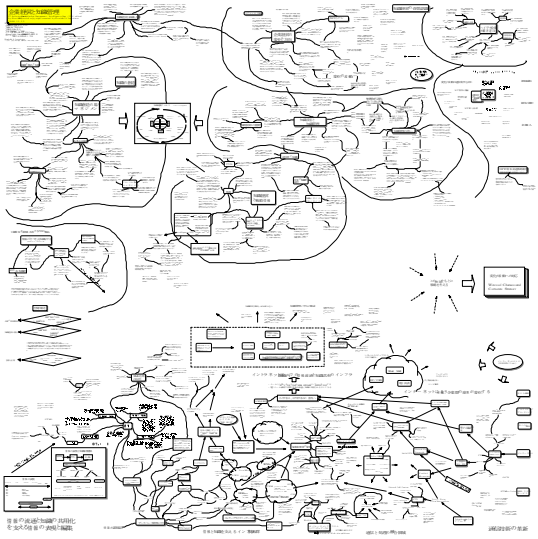
ベルリンの壁が崩壊した1989年という年は様々な分野で変化が始まった大きな節目であった。私自身、1980年代の後半はＮＴＴ内部の情報処理の政策や技術開発戦略に係わっており外の世界から少し距離を置いていた。ところがあるきっかけで外に目を向けたのが1989年である。情報処理の世界に関しては大きなパラダイムの変革が始まりつつある時期で、例えばそれまでのシステム工学、特に一つの目標を実現するためにすべてのサブシステムを動員する固いタイプのシステムから、「緩やかに結合する分散システム」の時代に変化する兆しが感じられるタイミングでもあった。第 5 世代で目標にした知識情報処理に関しても変化の兆しが出ていた。スタンフォード大学のファイゲンバウム教授とも議論したのであるが、知識情報処理は大きな壁に突き当たっているというのが実感になってきていた。コンピュータを賢くする方向の技術開発から人間の力をコンピュータで拡大するという方向へのパラダイムシフトが起きつつあった。実際に1990年に米国シアトルで開催されたＣＨＩコンファレンスでは「Empowering People 」というキャッチコピーが高らかに宣言されることになる。「５Ｇ／ＡＩから Empowering people / HI へ；1989年の曲り角」の図は、1989年の夏に米国の研究開発コミュニティの動向を調査したときに描いたシナリオである。実際に大きなシフトが起こる。ヒューマンインタフェースの進歩やアップル社ジョン・スカリーが提唱した Knowledge Navigator が目標とするビジョンに掲げられるようになっていく。



「ＨＩ；ＣＳＣＷ／グループウェア」は 1990 年にこの変化の性質を探るために理科大の溝口文雄教授と研究会をしてその結果をまとめた知識地図である。

この変化の中で私個人は当時までの知識情報処理技術とヒューマンインタフェースを指向する情報処理の技術の流れを統合する可能性を求めてオブジェクト指向データベースの技術とその実用化プロジェクトを立ち上げて進めることになる。米国 ACM/SIGMOD 議長をしていた人物とその製品を開発するベンチャー会社（Ｕｎｉ ＳＱＬ社）を米国テキサス州オースチンに設立してそのビジネスにのめり込んでいく。そのビジネスはしかし皮肉なことにインターネット上でのマルチメディア応用、イントラネット上での知識共有という応用分野で成功して、このパラダイムシフトの流れの上を進んでいくことになった。全く同じ時期に三井造船の綾日天彦氏はやはりオブジェクト指向技術を武器にこのシナリオ図の真中に描いてあるＣＩＭ／ＦＡなどエンジニアリングの世界で可能性を探るプロジェクトを経験された。綾委員も私も 1990 年代の 10 年間は同じオブジェクト指向技術を武器にして新しい道を切り拓こうとしてきた訳である。

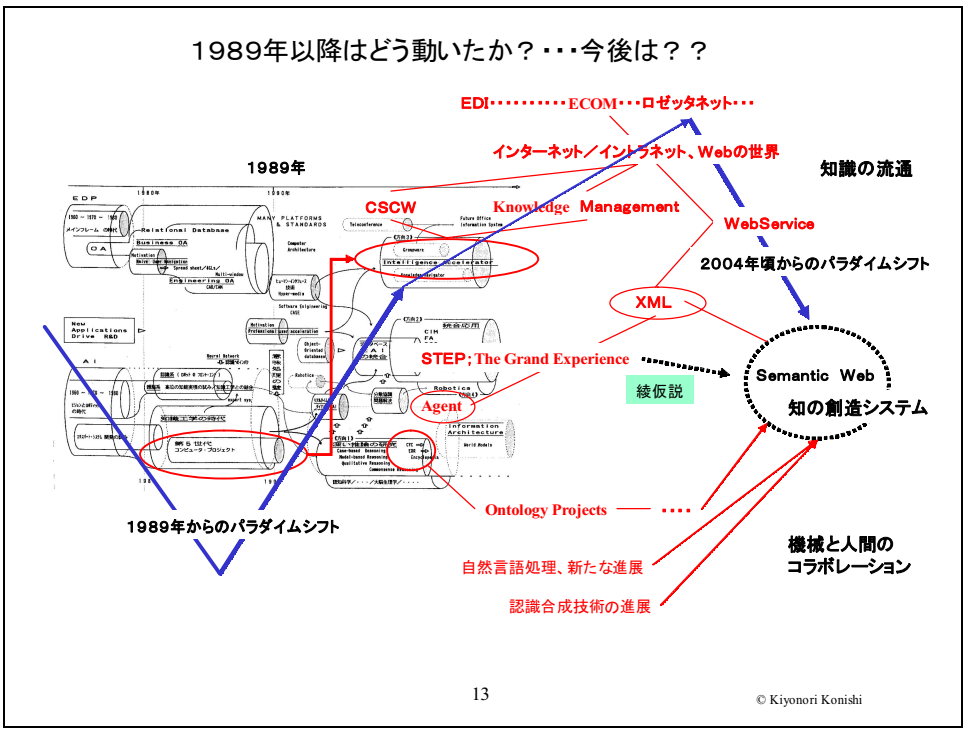
HI→KM；企業経営と知識管理、1998年作成



12

© Kiyonori Konishi

1998 年頃に知識工学ではなく知識経営、知識管理の視点からの議論が行われるようになってくる。これら経営学の視点からの議論とコンピュータ科学と技術、コンピュータ産業の製品を対比する目的で作成した知識地図が「ＨＩ→KM；企業経営と知識管理」である。ここでは経営学の視点と技術・製品の間にまだ大きなギャップがあるというのが結論であった。



さて、最後の「1999 年以降はどう動いたか？…今後は？？」のシナリオ図は現在の問題意識を表している。２０数年前に第 5 世代コンピュータプロジェクトを企画したときには、我々はあまりにも野心的であったかもしれない。しかし、この２０年間の技術の進歩と研究の進展をきちんと評価し、再びこの分野でプロジェクトを起動すればどの程度までの実現可能性があるのかを吟味すべきときにさしかかっているのではないか。認識合成技術は大きく進歩している。自然言語処理に関してもこの２０年間で辞書は格段に整備されてきた。２０年前には行われなかった統計的な手法も使われるようになってきている。エージェントモデルも現実のものになってきた。知識工学ではなく知識管理の視点からの多くの試みも実を結びつつある。インターネット・Ｗｅｂは巨大な知識ベースを形成する世界になってきた。一方,ビジネスに近い世界でもＥＤＩから始まりｅ-コマース、ロゼッタネット、Ｗｅｂサービスなどの進展も見られる。これらの動きの中でコンピュータと人間が共通に理解可能な言語であるＸＭＬも広く使われる方向である。一方、綾委員はこの１０年間取り組んできたエンジニアリングの中心的なテーマであったＳＴＥＰについて Semantic Web の枠組みとＳＴＥＰの枠組みを融合することにより新たな展望を拓こうとしている。このような全体の状況をみると、そろそろ自然言語と知識の流通の将来について地図が描ける時期にさしかかっていると感じられる。しかもそれは単に知識の流通ばかりではなく「知の創造システム」としてのビジョンが描けるのではないか。この作業を進めることが大切である。

企業間電子商取引の拡大とオープン化の課題

委　員　　岡　田　　宏

1. 企業間電子商取引の拡大とオープン化の課題

本章では、企業間電子商取引の拡大とオープン化の抱える課題について、とりわけ、STEP とセマンテック Web との観点から報告する。

1.1 ユビキタスな時代

ラテン語のユビキタス（Ubiquitous)、「至る所に偏在」という言葉を最初に聴いてはや 7 年、やっと、これで一才、歳をとれたのかもしれない。以来、「だれでも、いつでも、どこから、だれとでも、簡単に、意味のある情報を、確実にゲット」ということをキーワードに、「ユビキタス」に形容された世界が着実に定着してきているといえよう。街にでれば、携帯電話に代表される情報機器を片手に、電波の洪水の中から、情報取得や発信に没頭する光景に出くわさないことはないようになってきている。

携帯電話、爆発的なブームの中、その導入台数もうなぎのぼり、五千万台に届こうとしているともいわれている。独占的に供給されてきたアナログの固定式・単機能電話に代わり、いまや、寡占的競争下、移動体が基本で、カメラ付き、動画対応とますます多機能化し、会話の手段としてのみならず、アドレス帳、コンテンツのアーカイブ、マルチコンテンツ情報の取得、発信手段、さらには決済手段として、その重要度がますます充実してきている。これをもって、「ユビキタス」時代の本命とするのか、その後継が登場するのかは、大きく議論が分かれるところである。

他方、ネットワークインフラは着実に整備されてきており、携帯電話の席巻、ブロードバンド・インターネット社会の到来という表題も珍しくなくなってきている。「だれでも、いつでも、どこから、だれとでも、簡単に、意味のある情報を、確実にゲット」というユビキタスの利便性は、ビジネスの世界での企業間電子商取引の拡大とオープン化への寄与という観点で活かしきれているかというと、いささか誇張されすぎた結果の表現になっているかもしれない。とりわけ、設計・製造データの活用と流通という面では、その効果や評価についていささか冷ややかな目で評価されてしまうこともある。雨後の竹の子の如く萌芽した B2B や B2C の e-ビジネスモデルも、効果面への言及は低調に終わっている。e-ビジネスモデル自体、ここのところでワンサイクルし、ビジネスモデルとして継続できるものが勝ち組として残り、

やっと、その存在価値が認められるようになってきたものもあるといえよう。これに加えて、B2B2C という様に、商流のサプライチェーン全体を網羅したビジネスモデルも登場してきている。ものづくりのための設計・製造データの活用と流通、「デジマエ」、そして「動け日本」と、ここのところで、お家芸をどうするのかの議論が本格化し、施策もつぎつぎと展開されてきている。これに、e-Japan 構想の議論が追い風の如く吹き荒れているのが今日この頃である。

1.2 現状

ビジネスの世界において「なぜそうなのか」という問いかけに対し、日本の企業文化の特異性としての「文化」論が一つの模範回答となってきている。それは、「タテ社会の人間関係」、「日本的経営の系譜」とか「戦後日本の企業集団」で写像されているように、社員として入社するや否や、企業文化教育がなされ、一年も経てば立派な企業人となり、文化継承が永続的に行なわれてきている。そうした文化構造が根底にあり、その文化と融合するのに時間が要するという理屈である。「こういう文化があって、こういうプロセスとなっているから。」確かにそうかもしれないが、現状認識はそれだけで済ませていいものなのかいささか疑問である。

企業は、それぞれ独自の文化の中での仕事のやり方を踏襲しているが、そもそも、その１社だけで製品やサービスの提供を完結できるということはなく、文化の異なる企業間での取引が発生することになる。その異文化交流には取り決めが必要にあり、それが、「プロトコル」として定着することになる。このプロトコルを介して実際の取引が履行さるわけである。とわいえ、そこで使われる用語、メッセージ、契約内容といった面で、取引の商慣行が形成されているが、実態は、産業の垂直・水平統合といった力関係で、ある文化に基づく商慣行がディファクト標準ととして活用されてきている。また、業界によっては、CII シンタックスルールのように国内標準を定めたりしている。あるいは、外的要因により国際標準への対応が必須となってきたりもしている。ここにオープン化の萌芽がみられるが、いまだ全面活用という意味で文化として同化し、全面開花したわけではないといえよう。振り返ってみると、CALS が一大ブームを引き起こし、SGML が登壇した時代、JUAS で行った製造業における電子データの実態を調査した結果を見ても、業務で設計・製造データの活用と流通という観点で電子データを他社と交換しているところはまだまだ数が限られていた。そうした状況下で、SGML を活用することで、現実活用されているドキュメントベースの交換とそのドキュメントを生成するもとネタとしてのデータをどう対応つけ、整合性をとるかという議論がなされ、その回答は、DTD、特別な

意味のある＜Tag＞、そして文書の中に埋め込んだ DML を活用してドキュメントとそれを生成したデータとの間の整合性を維持しようというアイディアであった。

今日、種々の取引の履行にはコンピュータ処理が必須となっており、それを中核に基幹システムが形成されている。中でも、最初にコンピュータ化された最重要視されている財務会計処理系は、継続企業体にあって期間処理と損益計算と対照表の作成を行うものである。これをベースに企業文化論に基づく業務処理の大掛かりなしくみが構築されてきたということである。カスタムメイドの世界の産物であっても、複式簿記のアルゴリズムを具現化したものであり、発生した費用と実現した収益を把握できることにその導入メリットがある。これがメインフレームコンピュータをベースにした固有業務アプリケーションの歴史の一ページを飾ることになった。企業文化の電子化とともに、肥大化した固有業務アプリケーションからハードウェアとミドルウェアのオープンシステムをベースにした ERP や CRM に代表されるパッケージソフトウェアへの切り替えが大きな課題となってきている。基本的には、過去の資産とデータの継承に絡む議論であり、しかも手作りであるゆえに、電子商取引対応への対応を含む時代の変化へフレキシブルに、迅速に対応しずらくしてきている。また、コストがかかりすぎるその回収もおぼつかないうちに次世代の開発というややもするとコスト増大の悪循環に陥ってしまうことになる。

求められているのは、トレンドの紹介としての新たなる横文字の単の羅列ではなく、上述の状況を鑑みた、具体的なソルーションの提供のための提示にあるということである。横文字事例の翻訳に没頭し、しかもその内容が現実の業務とかけ離れたものであればあるほど、それ自体を理解し、企業文化に同化させることができるか否か判断するまで時間がかかってしまうことになる。華々しい海外での事例が日向部分とすると、その影に隠れた部分、つまり現状そのものに言及せず、またそれを把握（理解）せずに、単なる技術の紹介では、例えば、フリーソフトウェアであったともしても、普及すらおぼつかないことになってしまうということには留意すべきである。

1.3　ソルーションビジネス

アプリケーションの提供という面で見ても、言語で開発していたものがデータベースマネジメントシステムというパッケージをベースに開発されることが一般的となってきている。確かに市販のパッケージであるが、その上に構築されるアプリケ

ーション部分は、あいもかわらず、システムという名前のついた個別にプログラムの域を脱していないという指摘がある。その背景には、業務も形式知になっていないという文化にある。確かに、ワークフローのデザインという言葉に象徴されるように、コンピュータの中で業務フローを設計できるようになってきており、そのためのツールも販売され、導入事例も多々紹介されてきている。こんな状況の中で、例えば、「カンバン」のように一企業文化内の暗黙知が形式知化されて、それが標準マニュアルになって、公の刊行物になって、コンセプトとしてプレゼンテーションされ、アルファベットで記されたコード名とともに世の中に出て行くことになる。ある企業文化の中でしかわからなかったことが、こういう外部での知識化・体系化によって公知の事実化が行なわれ、世界各国で使われていくようになってくる、そういう時代になってきているといえよう。

ERP のパッケージとともに盛んに語られているのがカスタマイズをするべきか否かの議論、導入事例としてのある意味で大きな脚光を浴びてきている。そのカスタマイズする部分を、その企業のごとのカスタマイズでなく、日本文化としてのカスタマイズとし、それをもってアドオンのコンポーネントとしてしまうというもので、日本文化のベストプラクテスのパッケージ化である。これは、ERP ビジネスを逆手に取ったの展開であり、日本発のコンポーネントの輸出である。これまで企業の中で考案され、情報システム部門が中心となり、ベンダの実行部隊が構築していた時代には、考えられなかったような状況があるといえる。

IT ソルーションビジネスの勃興の背景として、データベースマネジメントシステムを用いた開発と汎用アプリケーションパッケージの登場、データモデリングの定着、ビジネスオブジェクトのライブラリー化、開発支援ツールの活用、ノウハウの形式知化、ERP パッケージ等の躍進、ERP パッケージ導入失敗からの体得効果の顕在化、ビジネスモデルのカタログ化がみられるようになってきた。やっと、「業務プログラムの開発」から「業務プロセス開発の時代」が訪れようとしてきている。こうした時代にもとめられるのは、業務プロセスそのものを開発し、その上で、最大の効果を得るために、情報技術の動向をとらまえ、どのような情報技術をどう活用していくかという視点であり、決して、技術ありきでのデザインではないということである。

他方、留意すべきは、大きなパラドックスチェンジが起ころうとしている時代であり、「考えているひま」がだんだんなくなってきているということである。もはや、行動を起さねばならないという「動け、日本!!」時代になってきているのも事実であ

る。投資という面から見ると、インフラの整備は着実に進行しているものの、そのインフラの活用という面では、必ずしも活用しきられていないということが言える。利用する側は、シビアな状況にあるといえる。これまでの投資案件を抱え、それにくわえて、e-対策のための追加投資が必要となってきている。しかも、短期での資金回収が前提にあり、ビジネスとしての成就には迅速な対応が望まれている。他方、一発勝負の新規参入者は、過去の資産がないだけ、アジルさをもって身軽に行動でき、しかも、ディファクト標準となっている過去の資産をコンポーネントとして活用でき、あるいはアライアンスを組むことも可能となっている。

インターネットのビジネスでの活用という点からみると、B2B2C のビジネスモデルの導入、ポータル間での言葉のハーモナイズ、XML ベースの EDI 化といった課題が語られてきている。B2B 分野での顕著な動向として以下のようなものがある：

・共通言語	W3C	XML 標準仕様
・相互運用基盤	OASIS	Web サービス標準仕様
・ビジネスプロセス	UN/EDIFACT	UN/EDIFACT 、ebXML 技術仕様
・取引契約	ROSETTANET	

これらは、すべて、ビジネスニーズに基づいた議論であり、その成就は、ビジネスの生死にかかわる問題であるという認識が浸透してきている。これは、技術が何かの議論ではなく、それを使って何ができるか、それで、メリットが十分説明できるのか、ある意味で、ビジネスの世界で概念が咀嚼され、その効果もふくめ、その概念が形式知として認知された結果である。同様の技術に係わる議論には、言語の話もあるし、開発方法論の適用もあるし、総合基盤の構築もあるし、ビジネスプロトコルの制度化もあるし、ビジネスモデルの提案もある。日本の企業の現実を鑑みながらこれらの議論を生かしきれるか尽くしきれるか、あるいは議論の結果を活かしきっていないということの判断が必要であり、認識にギャップがあるならば、それをどう埋めていくことができるかが大きな課題といえる。

1.4　ステークホルダー

産業界における Agile 化、e-化への要請の高揚にともないパラドックス：IT インフラの高度化 v.s. 民間での情報化投資の活用が議論されて、施策展開による打開の道が検討されている。IT 施策という面からみると、「政府」、「企業」と「IT ベンダ」がステークホルダーであり、その中で企業は、投資効果という尺度を用いて、判断し、

ビジネスプロセスを再設計すべきであること自体、十分理解している。その時に頼ろうとしている、いわゆる IT ベンダがその要求を十分理解し、ビジネスプロセスの再設計できる能力を疑問視している節がある。他方、IT ベンダの立場から見れば、業務用のプログラムは組めるが、ビジネスプロセスそのものの設計ができるかという一抹の不安がある。大手 IS 事業者は別にして、日本のソフトウェア業界を支えているごく平均的なところでは、プログラム開発が未だ生業であり、それから脱皮しようとして努力はするものの、いまだトンネルの先が見えない状況があるといえよう。受託開発した業務プログラムを汎用的に設計し、それをパッケージ化し、販売する。これはビジネスリスクとなっている。これまでは、提示された仕様をもとに設計し、プログラミングし、テストし、納品すればよかったものが、その大元となる「仕様を切って」というワンクッションが入ることにより、立場はまったく逆転してしまうことになる。すなわち、どんな仕様のもので、そもそもその仕様で業務ができるはずという前提であり、しかも、その仕様の実装を保証しなければならないことになる。これは、大きなリスクを背負い込む以外のなにものでもない。しかも、投下費用が確実に回収するギャランテーもない世界である。そもそも仕様を説明し、それがどんな業務上の効果を生むのか、体得していないゆえ、また、販売するという立場にたったこともないし、成約まで至ったことない。売ったこともないし、そういうビジネスでやったことがない。これでは、ビジネスモデルそのものが成立しないのである。

これまでの施策の評価をし、その信憑性を保証するシナリオはあるものの、それは、あくまでもシナリオであり、サクセスストリーとしてのクリデンシャルが問われている。成果物が真の成果を上げているのか、それ以前に、実際使われているのかが問題である。さらに、フォローアップ体制、これも大きな問題である。多くの開発は請負契約あるが、運用面は、報告を聴くだけということが多い。しかもその対象物が、成果物そのものとして要は、プログラムそのものであり、それをどう使うのか、それの新規性がどうであるのか、利益を実現するにはどういう仕組みにするのかという面がなおざりとなっている。例えば開発成果物を利したビジネスモデル開発そのものへの施策がありえるのか、これは大きな疑問である。すなわち、「暖簾」という無形財産そのものの価値の評価という点である。こういう観点からステークホルダーの抱えるそれぞれの利害というものを理解することは重要である。

1.5 スパイラル効果実現のために

真のユーザというのは、誰であるかという点、その姿がますます非常に見えにくいものとなってきている。一説によると、IT ベンダがパッケージとして世に送り出

されるもののユーザは、IT ベンダが主である。「ソルーション」を提供してくれるはずの IT ベンダに売り込みをかけずして、誰に売り込むのかという点を明確にする必要があるということである。IT ベンダが積極的に活用するスキームを構築し、その結果として新たなソルーションが提供される。もちろん、そのソルーションを活用することでそのユーザが利便を享受する。波及効果として、その企業の競争力が高まる、そういう仕組みが形成できるかどうかである。前述の投資効果、ビジネスプロセスの再構築、ソリューションベンダへの不安をどう克服できるか、また、ベンダ側としては、ビジネスプロセス設計への不安、萌芽市場参入リスク、利益構造への不安といった課題ももちろんあるが、こうした点を考慮したスキームつくりに一石を投じることは、施策への期待も含め、意義のあることではないかといえよう。もちろん、官民学のパイロットプロジェクトの不作傾向とか無形財産への援助面という課題はあるが、具体的なユーザの顔が見える提案をどう作れるか、技術紹介だけに終わらないで、現実にこの技術を使うとなにができるのか、その効果は何かということ、そこの議論がもう少し重要なんじゃないでしょうかということです。

留意すべきは「教訓」であり、そこから何を学ぶかである。そこで、設計・製造データの活用と流通の事例として CALS で一躍脚光を浴びた STEP の教訓に注目してみることにする。まず、復習の意味も含めて開発面から STEP、とりわけ STEP の規約体系の中でも Part200 番台として具体的な実装のための製品モデル部分（以下 STEP（AP）という）は何かをみてみることにする。

STEP（AP）は、ある工業製品をある視点からみたものは何かをデータモデルとして表現したものであるといえる。その STEP(AP)が業務で使えるかどうかを判断するには、まず、製品モデルとしての STEP（AP）が当該業務を対象としているか否かを判断することにある。STEP(AP)では、その STEP(AP)が対象とするスコープを定義している。次に、そのスコープを含む業務がどうなっているのかの全体をとらまえることになる。STEP（AP）では、スコープとなる業務が全体の業務の中でどの位置を占め、どんな業務から構成され、どのような入力、出力、指示、資源が関連するのかを抽象化しアクティビティモデルとして表現している。この部分を照査することで、当該業務との差異を確認することができる。留意すべきは、STEP（AP）では、視点が明確になっていることにあり、その視点から見たアクティビティとそこでやりとりされる情報であるということである。また、アクティビティモデルをベースに当該製品モデルに関連する言葉の定義と制約関係が体系的にまとめられている。これが、STEP（AP）の情報要件として定義されるものである。この部分を理解する

ことで、業務担当者が実務で使っている言葉との対応をとることが可能となる。この過程を通じて、製品モデルを記述する情報が当該業務で実際活用されているものを扱えるかどうかの判断をすることもできる。

情報要件は、されに抽象化されて、当該製品モデルを所与の視点から説明するためのデータモデルとして定義されている。STEP(AP)では、こうして定義されたデータモデルを前提に、STEP 全体の定義体を通して共通的に用いられる統合リソース群との対応関係が定義され、それがマッピングテーブルとしてまとめられている。統合リソース群とデータモデルとのマッピングを考慮することにより、STEP 製品モデルを定義しようとするものである。最終的には、この定義体に具体的な業務で実際に用いられているシステムをマップして設計・製造データの活用と流通を図ろうとするものである。なお、この STEP（AP）モデルは、あくまでも抽象化されているものであるゆえに、ここの実装を鑑みた適合性検証のための条件も設定されている。また、具体的な業務利用を想定したシナリオも当該 STEP(AP)の別添として用意されているものもある。

STEP(AP)での重要なコンセプトは、「抽象化」である。分析した個々の業務手順を抽象化したものがアクティビティモデルである。その目的は、図的表現を用いて視覚的にどんなアクティビティを対象にしているのかを表わそうとすることにある。しかし、この表現の限界は、時間(t)の概念が欠如していることであり、また、製品モデルの流通先や流通元ではなく、あくまでも活用する立場に固定された視点のみで記述されていることにある。実装を鑑みると、これでは、STEP（AP）を適用できる局面が限られてしまうことになりかねない。すなわち、実装時には、相互に関連するシステム群の中で、活用と流通がはかられるゆえに、これらとのやりとりを考慮することが必須となる。この点は、当該 STEP(AP)の別添で記述されたシナリオがあればある程度は補うことができるが、このシナリオを「ユーズケース」を用いて明確に定義することで大幅に改善されることになるといえる。ユーズケースを用いることにより、STEP（AP）が定義したある視点からとらまえられたモデルに関与する関与者の係わりも含めて外部から見た視点で把握することが可能となる。これに業務を支えるシステムを加味するためのシーケンス図を合わせることで、時間軸(t)の中で、関与者同士の情報のやり取りをベースにした実装を記述できる点は重要である。情報要件では、インスタンスとしての情報の格納庫のみに注目しているが、その情報の挙動をどうモデル化するかが課題であり、この点でクラス図は有効な手段となり、また時間軸(t)を考慮した具体的な挙動をシーケンスで表現することも実装には重要である。こうした道具立てで STEP(AP)を補うことにより、製品データモ

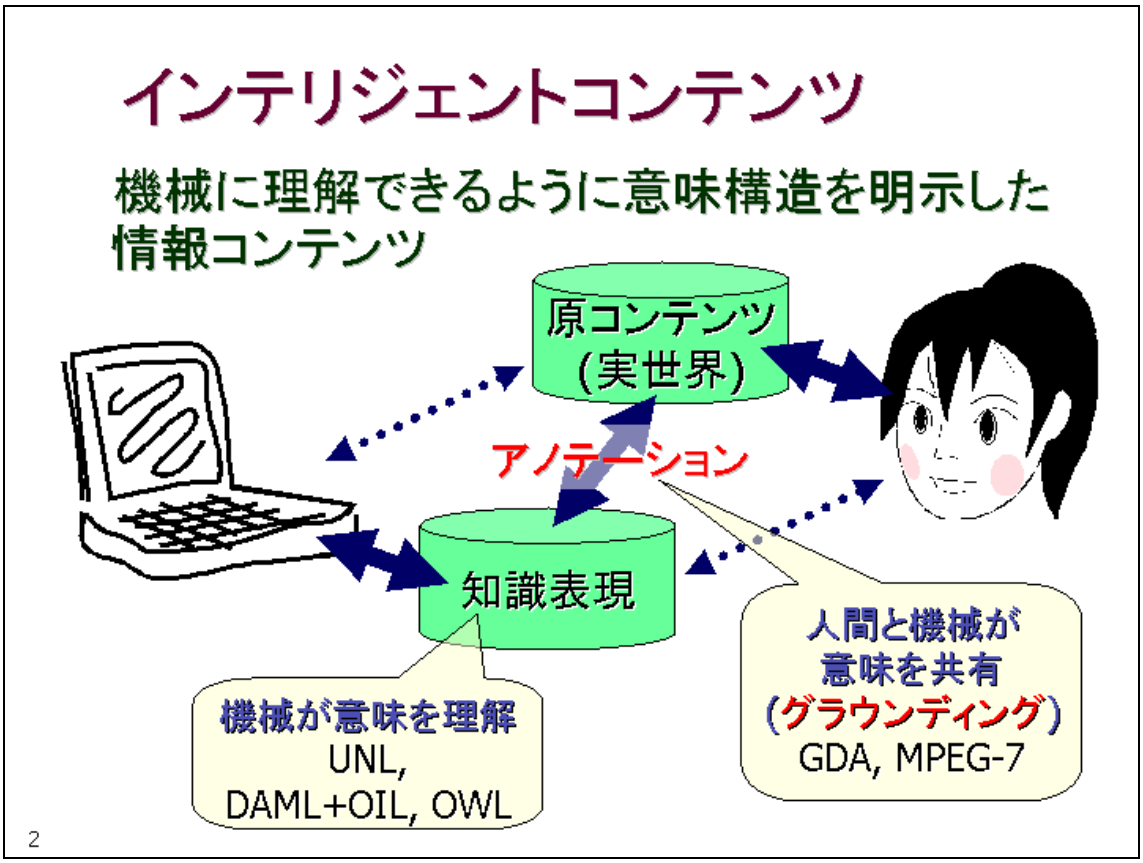
デルとしての定義を利用し、当該製品をある視点から見た論理構造として活用することが可能となりえる。それゆえ、設計・製造データの活用と流通のための実装を議論する時、STEP（AP）を用いる意義は、当該業務で活用されるシステム間でのデータを網羅的に定義するメタデータの役割にあるといえよう。

近年、XML にオントロジーに基づきメタデータを付して、検索の処理における論理性の保持と検索結果のトレーサビリティの向上を追求し、結果として検索精度をあげる提案がなされ、その活用効果に大きな期待が高揚してきている。その実装上での課題は、オントロジーの拠り所をどこに求めるかにあり、とりわけ、設計・製造データの活用と流通を鑑みた場合、この面で、STEP（AP）の論理データモデルと情報要件を含む成果を利する効果は十分あるといえよう。

以上

アノテーションに基づくセマンティックウェブ

委　員　　橋　田　　浩　一



狭い意味でのセマンティックｗｅｂというのは、機械が意味を理解できるようなコンテンツを知識表現で作っておいて機械にいろんなサービスをさせようという話ですけど、いろいろな問題があります。

一番大きなのは、いろいろな人がいろいろなオントロジーを作ったとすれば、それをどうやって結びつけて、より大きなサービスにしていくのか。あるオントロジーはそのコミュニティの中ではいいんですけれども、オントロジーの間を結び付けるのは結局人間がやらざるを得ない。それは自然言語とか、リアルワールドでのコミュニケーションの手段に帰着されるわけですから、やっぱり機械が意味を理解するだけではだめで、人間と機械がちゃんと意味を共有する必要があるわけです。

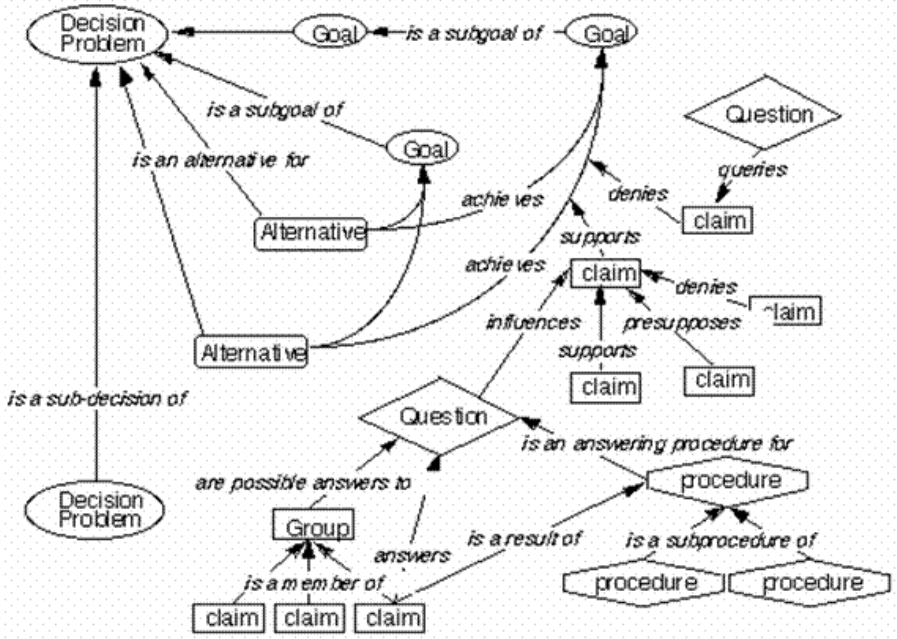
それが、ＧＤＡとかＭＰＥＧ－７のように、もとのコンテンツ、つまりビデオとか文章とかあるいはリアルワールドとかいうコンテンツにアノテーションを施すことによって、知識表現と結び付けると、結局は人間と機械の間でコンテンツの意味が共有される。たとえば、それによって別々のオントロジーが自動的に統合されるという話ではなくて、少なくとも人間を介在させながらリンクされて、人間を含むインタラクティブな形で使われるというようなことを考えています。

コミュニケーションの構造化

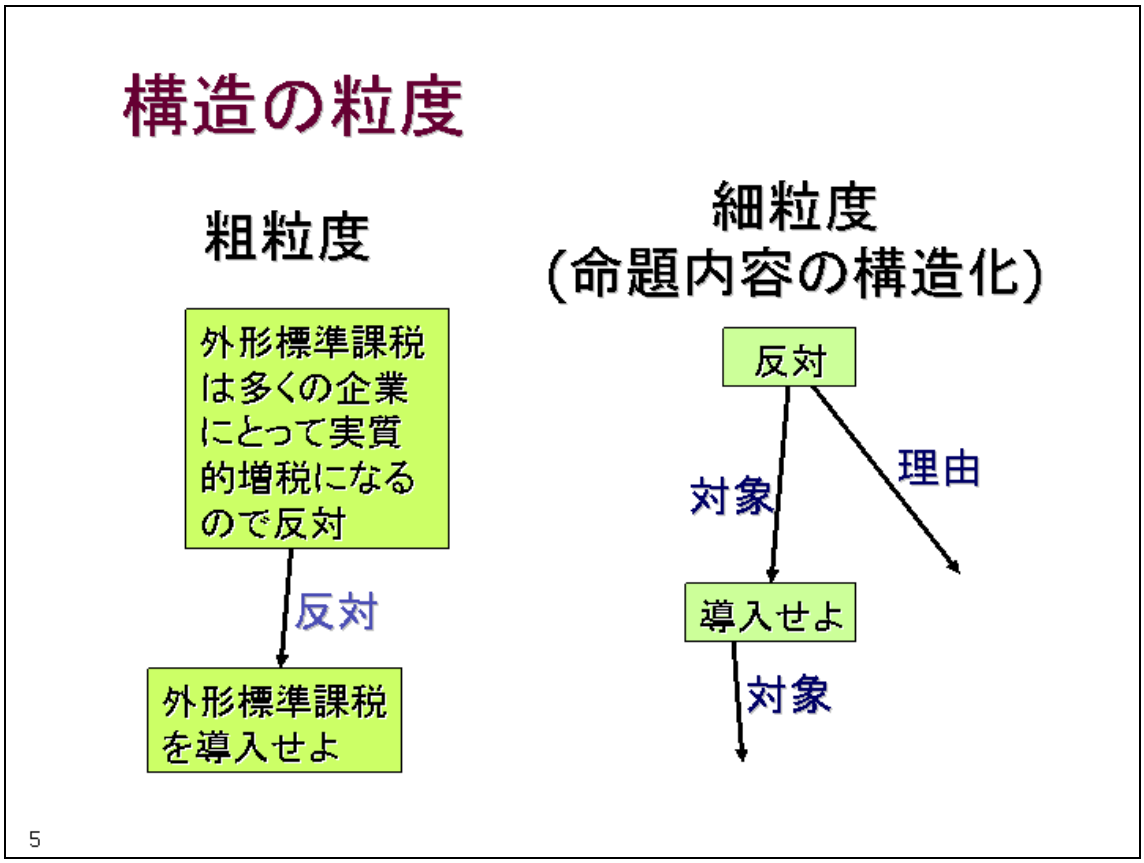
- Xanadu?, IBIS, Coordinator, Open Meeting, etc.
- 問題点
  - 認知的負荷
  - 曖昧性を許さない

3

IBIS、gIBIS、etc.



4



この前お話したのは、議論の構造化みたいな話があって、これがなかなか普及しないということでした。議論の中でこういう発言があったときに、それを別の発言とリンクで結んで一方が他方に対する反対意見になっているとかいう意味的な関係をリンクのラベルとして貼り付けるというようなことは今まで散々やられてきたということですが、そうじゃなくてもっと、粒度を細かくして構造化して、命題内容を機械にもわかるように明示したらいいんじゃないか。こういうふうにすると、議論の内容を要約するとか、あるいは検索するというようなことが高い精度でできるわけです。

　でも、おそらく構造化の粒度を細かくするだけではだめで、そもそも、議論するということ以前に一人一人がコンテンツを作るというか、文章を書くときの最初の段階から構造化が折り込まれているというようなことを考えてみました。

コンテンツを構造化しておけば検索がうまくできるとか言っても、その検索は他人がやるわけで、いきなり自分自身がうれしいわけじゃないですよね。だから、忙しくていちいちタグなんか付けてられないという話になってしまうわけですね。そこで、既存の文章なりビデオなり何なりがあって、それにあとでタグを付けるということではなくて、そもそもそういうコンテンツを作るときに、作るうえで楽だから、あるいはよりよいコンテンツができるからという理由で構造化するというような話を考えてみました。

オーサリング支援

- 従来のオーサリング支援技術
  - 線形…ワープロ、アウトラインエディタなど
    - 内容発想時の自由度が小さい
  - 非線形…KJ法、知識地図など
    - 文章生成機能なし
- セマンティックオーサリング
  - 非線形コンテンツ ＝ 命題内容 ＋ テキスト
  - 自分のための構造化
  - 発想支援 … 情報視覚化 ＋ インタラクション
    → コンテンツの品質向上
  - 文章化支援 … 自動的文章生成 ＋ インタラクショ
    → オーサリングの負荷軽減
  - 言語表現と意味構造との対応関係を保存
    → 自動的文章生成の品質向上

6

従来のオーサリングの支援というのは、１つは、リニアなオーサリングです。たとえば、アウトラインエディタというのがあったわけですけれども、それは最初からリニアなコンテンツを前提にしていますので、書きながらいろいろなかたちに変換するというのはなかなか難しくて、フォーマットを変えるぐらいですね。そういうことで、いろいろな視覚化みたいなことを通じて、発想を支援するとかというようふうなことはやりにくいわけです。

もう一つ、非線形のオーサリング、これはＫＪ法とか、Coordinator とかいうやり方やツールがあるわけですけれども、それらは最終的な文章の作成とは切り離されていて、たとえばＫＪ法でいっぱいカードを壁に貼ってマップを作っても、それがそのまま文章になるわけじゃない。文章を作成するうえで役には立ちますけれども文章作成というのはまた別の作業になります。だから、今までのやり方はコンテンツ作成のいろいろな側面を部分的にしかサポートしていない。それに対してここで提案したいのは、「セマンティックオーサリング」で、最初から命題内容を明示したネットワーク、あるいは RDF みたいな構造と、その構造のいろいろなところに断片的なビデオとかテキストとかがまとわり付いているというような、ハイブリッドなコンテンツをオーサリングするわけです。

それはあくまでも、人が検索しやすいようにとか、あるいは翻訳して人に見せるということではなくて、自分が今ここでやっている作業を簡単に、かつ高品質化するための方法です。それは１つには、そういうネットワーク状の構造をいろいろと変換して、いろいろ

な視覚化ができるわけです。それによって発想支援の効果を持たせることができます。そういうことで、ＫＪ法とか知識マップが果たしてきたような、内容そのものの発想を活性化するということから、コンテンツの品質向上が図れるというのが一つです。

もう一つは、命題内容を明示していますから、そこから自然言語処理のいわゆる文章生成技術を使えばそれなりの文章が生成できます。ただし、もちろんどんぴしゃりの文章をいきない生成するのは難しいのでインタラクティブに修正するわけですけれども、例えば、これじゃなくてこっちを先に言いたいとか、そこはもう少し詳しく言いたいとかいうようなことがインタラクティブに簡単にできます。よく文章を書くときに悩むのは、こういうことも言いたいし、ああいうことも言いたいんだけど、これを先に言っちゃうとこれに続かない。あるいは、こういう順番で言うと、なかなかオチに持っていくのが難しいとか、という厄介な組み合わせ問題になる。

文章の中の順番を入れ替えることによって、あることを先に言う場合と、逆に言う場合とでは代名詞の使い方だとかが違ってくる。あるいは、この言葉は先に使わないといけないとか、そういった構想上の調整をすごくやらなければいけなくて、そういうちょっとした入れ替えをやるというのはかなり億劫なわけです。こういうことを全く億劫でなくできるというようなことになれば、オーサリングの負担が減る。コンテンツの品質向上とオーサリングの負荷軽減ということがＭＳワードとか一太郎に比べてすぐれている。そういう認識が広まれれば、既存のワープロに代わってこういうセマンティックオーサリングが普及するということです。そうなれば、今後生産されるコンテンツはすべて意味的な構造化がなされていて、検索・要約、翻訳、何でもできる、というようなことになります。

命題内容だけではなくて、言語表現のデータが付随ているハイブリッドなコンテンツを蓄積していくことによって、こういういうことを言い表すにはどういう格好いい言い方があるかというようなことがわかるデータベースができて、それを使ってさらに品質の良い文章自動生成もできるようになるということです。

ＸＭＬとペトリネットグラフに基づく最適作業支援の一方法

委　員　　井　越　昌　紀

１．はじめに

利用されている多様な製品の保守や修理に関しては、常にその製品に対しての熟練技術者が対応できるとは限らない。今後は、海外でも利用できる作業環境を考慮し、かつ、大量な保守修理情報の中から、必要な部分の作業知識を作業者の熟練度に応じた適切な情報として配信できるような作業支援システムが望まれている。

製品を取り扱うためのマニュアルは、現在までに電子化・マルチメディア化されてきた。しかし機械製品の保守・修理などを行う際には、その作業の手順がモデル化されマニュアルに反映されている必要があり、従来の単純なマニュアルでは、作業者に作業の手順に沿って最適な指示を出すことが困難である点や、作業者が作業内容を理解できずに間違った作業を継続して行ってしまうという点において十分であるとは言えない。

本研究は、製品の保守や修理に対して、XML で記述されたモデルと、作業者の環境条件を入れることにより、作業者にとって最適な作業手順を自動生成し、その手順に対応した部分に自然言語を XML で記述しておくことにより、作業者にとって最適で分かりやすい作業手順を提示できることを示すことを目的としている。

本論文では､電気自動車のヘッドランプの交換を例にとって､環境を考慮した作業手順の発生と、それに対応する自然言語の作業指示の提示が成されることを示す。

２．作業モデルの記述

２．１AND/OR グラフの記述

ＡＮＤ/ＯＲグラフ[1]は、製品の構成を階層的に示したもので、全ての実現可能な分解・組立順序を表現する。しかしＡＮＤ/ＯＲグラフの問題点は、サブアセンブリは常に安定な状態でなければ成らないために、例えば２つの部品をネジなどの締結部品で組み立てる時には、３つの部品を同時に組み付ける表現になってしまうということである。つまり、ＡＮＤ/ＯＲグラフの中で２つ以上の部品と締結部品を組み合わせる時、その中での組立順序は曖昧なものになってしまうという問題点がある。図１に車のヘッドランプ部分の部品を示し、図２にそのＡＮＤ/ＯＲグラフを示す。

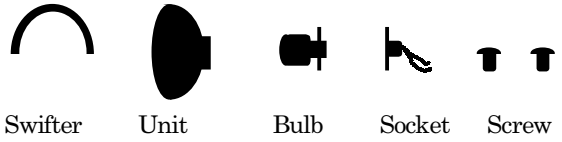


図１　ヘッドランプの部品

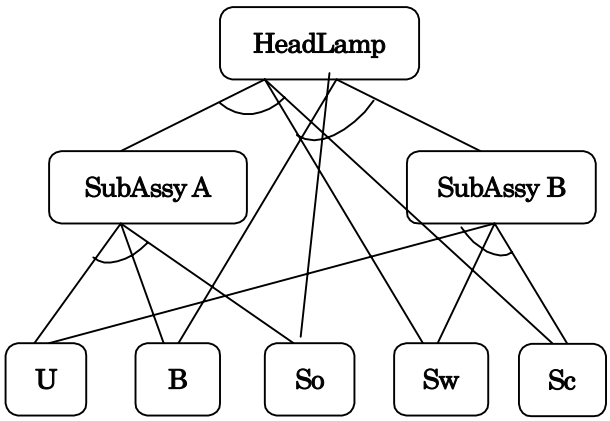


図２　ＡＮＤ/ＯＲグラフ

そこで本研究では、ＡＮＤ/ＯＲグラフに優先関係を付加することで部品間の組み付け順を明確にする。優先関係は“()”で表し、“()”で囲まれた作業に優先関係を持たせる。図３に図２のＡＮＤ/ＯＲの論理式とそれに優先関係を付加した論理式を示す。

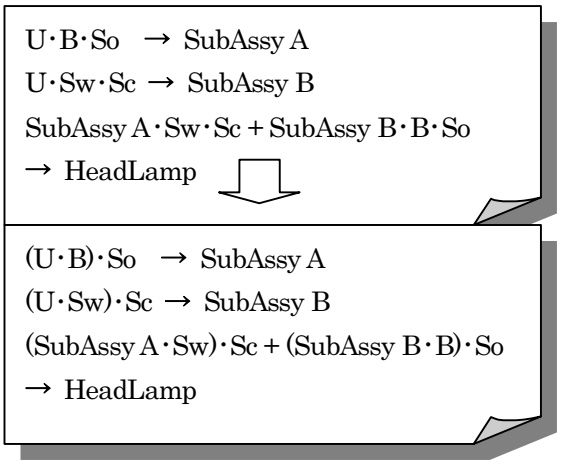


図３　優先順を考慮したＡＮＤ/ＯＲグラフの論理式

２．２環境条件を付加したペトリネットへの変換

3.1 で示した優先順位を付加したＡＮＤ/ＯＲグラフはプロセスモデルを表すペトリネットと見なせるためペトリネットグラフを用いて表現できる。ペトリネットグラフの中でプレースは各部品または締結部品を表し、トランジションはそれらの部品を組み立てる作業を表している。図４は、図２の AND/OR グラフをペトリネットに置き換え、環境条件として使用可能な工具を付加したものである。

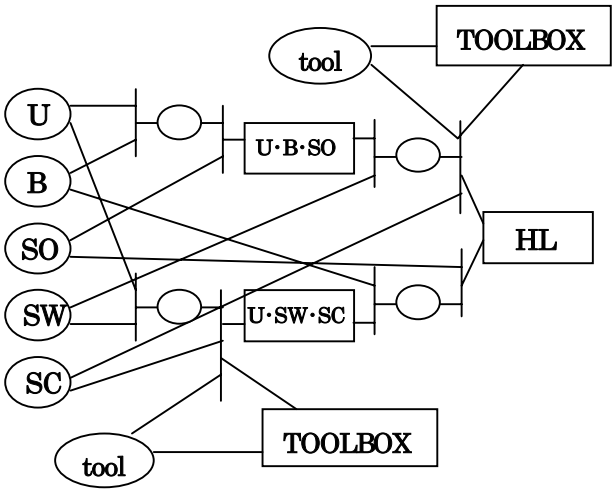


図４　ペトリネットグラフ

トランジションは、各作業についての作業の難易度を表す重みを持つ。各利用者によって作業条件が変わることから環境条件を考慮した重みを設定する必要がある。本研究では、現場で利用可能な工具情報を環境条件とする。

複数の工具で作業を行える場合、各工具によって作業の行いやすさに違いが生じることから作業者の使用する工具によって重みを変える必要がある。

また、図４には反映されていないが、実際に作業を行う場合、直前に使った工具と同じ工具で作業をする方が工具を変えるために工具箱から新しい工具を取り出すよりも効率的に作業を進めることができることから直前に使った工具との関係を重みとして反映させている。

このように環境条件として工具に関する上の２つの要素に応じた重みを設定していく。

本研究では、全ての重みの組合せについて最適解を求め、さらにその中で最も重みの総和の小さい組合せを最適解とする。

3.1　最適手順検索

環境条件からの重みとペトリネットグラフから利用者に最良な作業を示すために最適解を探索しなければならない。

本研究では、包原ら[2]によって研究された線形計画法を用いる。包原らは、ペトリネットグラフの状態遷移の数学モデルである状態方程式から実際の分解・組立手順に相当する解が状態方程式の基底解と１対１の対応になっていることを明らかにし、タスクの重みの総和を評価関数とする最適化問題が線形計画法で解けることを示した。

本研究の中では、環境条件によって変わる重みを評価関数とすることで環境条件に応じた最適解を導き出している。

２．３ＸＭＬ文書の記述

Web を介したシステムを実現するために製品構造を表現したＡＮＤ/ＯＲグラフをＸＭＬにより記述する。

作業者は全ての情報を得るのではなく、必要な情報だけを取り出すことを目的としている。ＸＭＬはデータ全体の構造化を行って、必要な情報だけを取り出すことを可能にする記述方法ある。図

５にＡＮＤ/ＯＲグラフを含んだ文書の一部を示す。

```
<trans num="18" weight="3">
<formal>UB111=((HL104*HL010)*HL012)*HL011</formal>
<natural>Ｍ４―２０トラスビスを取り外します。</natural>
<tool num="1">
  <tool num="2">3</tool>
  <tool num="3">2</tool>
  <tool num="4">1</tool>
</tool>
</trans>
<trans num="19" weight="1">
<formal>(HL104*HL010)*HL012</formal>
<natural>ヘッドランプリング の穴の上にあるＭ４ワッシャを
取り外します。</natural>
    ⋮
```

図５　ＸＭＬ化ＡＮＤ/ＯＲグラフ

以下で図５のＸＭＬ文書のタグとその属性の意味を説明する。

<trans>：トランジション（１つの作業を表す）

属性 num：トランジションの番号

属性 weight：作業の固定された重み

<formal>：優先関係を付加したＡＮＤ/ＯＲグラフの論理式（形式言語表現）

<natural>：作業を表す形式言語表現に対応した自然言語文。

<tool>：工具による作業のやりやすさを表す重み

属性 num：工具番号

また、ＸＭＬ化ＡＮＤ/ＯＲグラフの中で各部品と各アセンブリは、統一した記述を行っている。例えばヘッドランプアセンブリの１番目の部品は、“HL001”と表し、１ 番目のサブアセンブリは“HL101”と表す。

ＸＭＬはモデルの構造化を可能にするため、ＡＮＤ/ＯＲグラフの中から必要な部分だけを取り出すことができる。

本研究の中では、各アセンブリごとに記述したＡＮＤ/ＯＲグラフから作業に必要な部分のアセンブリを取り出す。また各アセンブリの他のアセンブリに対する関係についても構造化される。図６に今回考察した電気自動車とその各アセンブリとの関係を示す。

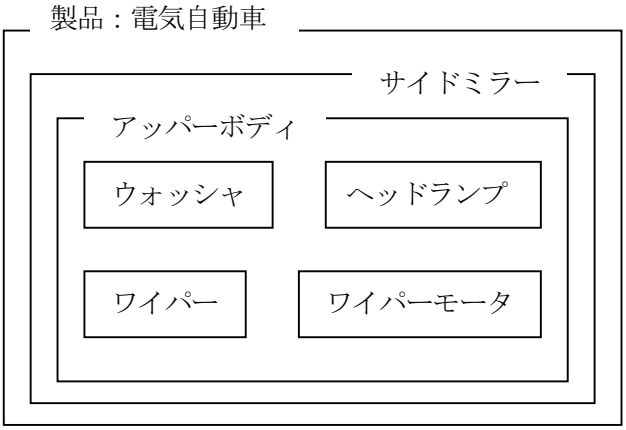


図６　製品の構造

図６から、例えばヘッドランプを修理するならば、初めにサイドミラーを外し、次にアッパーボディを外してからヘッドランプの分解を行う。

このように、全てのアセンブリとアセンブリの全ての部品は構造化されているので、必要なアセンブリを取り出した時にもその中で構造を持つことになる。

以上のようにして必要な部分だけを取り出したＸＭＬ文書からペトリネットグラフで考えた時のプレースとトランジションと工具情報からの重みを使って線形計画法により最適解を検索する。

３．実験

３．１実験システム

本研究では、ローカルな環境の作業者だけでなく遠隔地にいる作業者にも利用可能な作業支援システムを目指すため、サーバ側で動作するアプリケーションを作成することで Web を介したサーバ・クライアントシステムを作成する。

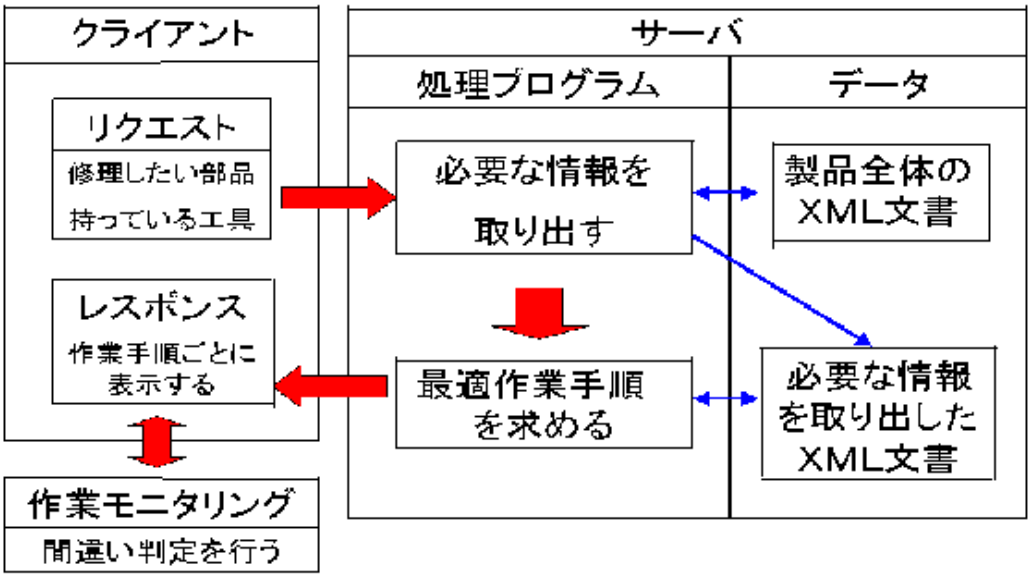


図７　処理手順

ここで、処理手順の中のモニタリングについて触れる。作業者が初心者であった場合には、示した

作業を理解できずに異なる作業を行ってしまうことが考えられる。そこで作業者に作業を示すだけでなく、作業者の行う作業が最適な作業と等しいかを判定することにより効率的で間違いの少ない作業が行えるようにする。本研究では、最適な作業を示すと同時にその作業を行う位置情報を簡単な作業位置認識システムに渡し、作業者の作業位置と比べることでモニタリングを行う。

３．２処理手順

本研究で作成した作業支援システムでは以下の順序で処理を行っていく。図７参照。

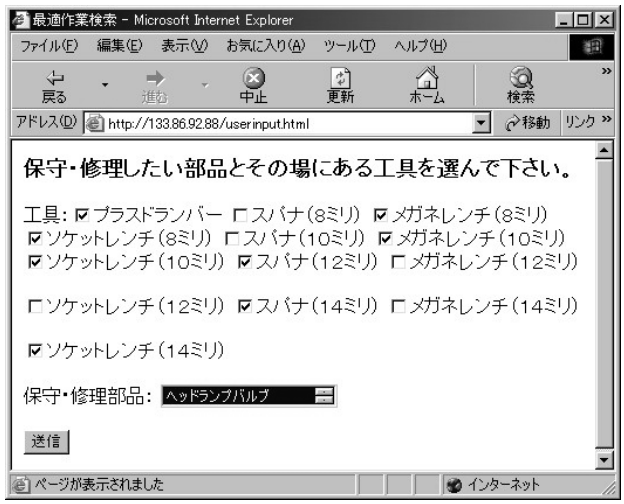
① 初めに、Web ページから作業者からのリクエストと工具の有無を環境情報として受け取る。

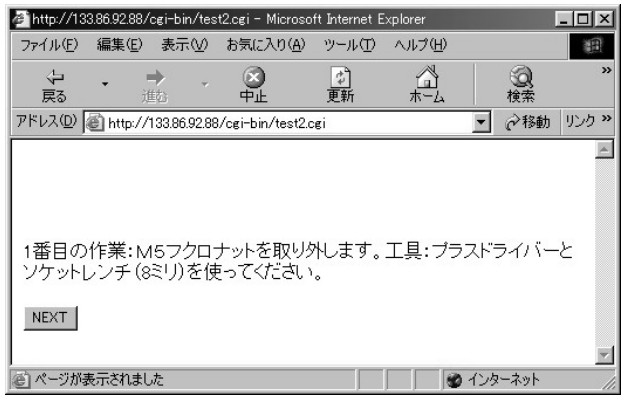
② 次に製品全体のＸＭＬから必要な部分を取り出し、環境情報と合わせて最適解検索の処理を行う。

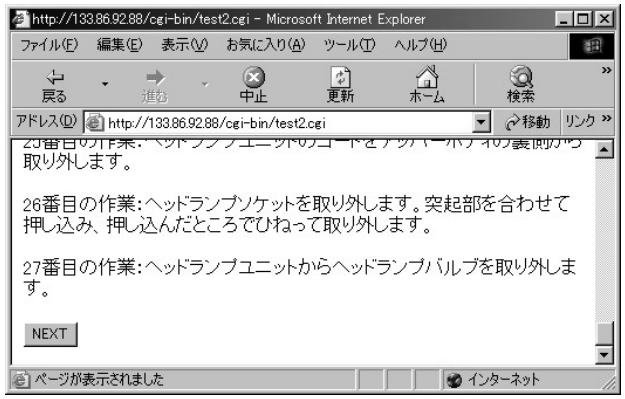
③ 最後に得られた解を作業の順に表示していく。

④ また、各作業の位置情報を作業位置認識システムに送り、作業者の作業位置が正しいか判定を行う。

以下にシステムの動作例を示す。対象とした作業は、ヘッドランプバルブの取り換え作業である。







４．結論

① 順序を持たせたＡＮＤ/ＯＲグラフを定義することで作業単位ごとの構造表現が可能になり、さらにＡＮＤ/ＯＲグラフを環境情報によって変わる重みを付加したペトリネットグラフに変換し、線形計画法によって解き、作業者にその時の最適手順を示すことができることを示した。

② サーバ側でＡＮＤ/ＯＲグラフをＸＭＬで記述することにより、クライアント側から要求された分解・組立の部分に関したＡＮＤ/ＯＲグラフおよびペトリネットグラフとそれに関連した自然言語で書かれた作業手順を作業者に示すことができることを示した。

③ 各作業を記述したＡＮＤ/ＯＲグラフに連携した位置情報から作業ごとにモニタリングを行うことで間違い判定を行い、利用者が間違いの少ない効率的な作業を行えることを示した。

謝辞：本研究は、常谷隆介君の修士の研究で行われたものであることを付記して感謝致します。
また、本報告書は、IMS/NGMS/KPAQ の 2001 年度報告書として提出したものを一部修正したものであることを付記します。

参考文献

【1】L. S. Homen de Mello and A. C. Sanderson: AND/OR Graph Representation of Assembly Plans, IEEE Transactions on Robotics and Automation, RA-6-2, 188/199 (1990)

【2】ペトリネットを用いた組立作業計画　－線形計画法による探索－　包原孝英　稲葉昭夫　川瀬俊夫　鈴木達也　大熊繁　計測自動制御学会論文集　Vol.29 No.4 461/469 (1993)

インターネット／マルチメディア技術を用いた
熟練技能伝承システムの構築

委 員　綿 貫 啓 一

１．緒　言

近年，製造業において，国際的な大競争時代の渦中にあり，生産拠点の海外移転など適地適産の傾向を強め，国内においては従来型の量産型垂直分業体制から，高精度・多品種極小ロットの生産体制への移行が進んでいる．この産業の空洞化の流れに加えて，長引く不況に起因する内需の不振から，さらに製造業を取り巻く環境が悪化し，生き残るにはこれまで以上に製品の質的，時間的，コスト的な条件が厳しくなることが予想される．特に大企業に比べ設計・生産環境，設備などの整備が遅れ，作業者の技能・技術が製品に大きく影響する中小企業にとっては，従来以上の熟練技能が不可欠であり，技術者・技能者の育成および技能伝承は重要な取り組みの一つとなっている．技術者・技能者の技術・技能伝承問題は重要であり，長期的な視点に立った適切な人材育成計画が必要であるが，多くの中小企業では，先に述べたような状況の変化に起因する，資金的，人的，時間的な制約が長期的な視点に立った人材育成，技能伝承を難しくしている．実際，過半数の中小企業では人材育成の手段として，仕事を通じた現場指導(OJT)やビデオライブラリ，技術ドキュメントなどを活用しているが，効率が悪く非常に時間がかかるために，OJT を中心とした従来の手法だけでは高度な技術・技能を体得した熟練技能者の技能継承が厳しくなってきている．製造業の現状および将来を考えると，今以上に熟練技能者の高齢化，産業構造の変化に伴う製造現場からの人材流出，ものづくりマインドの低下に加えていわゆる 3K 産業に対する意識の変化などによる若年就業者の減少が進むことで，これまで蓄積されてきた高度な技術・技能の伝承が一層困難になれば，技術・技能の衰退を招くことが予想されるために，企業にとって熟練技能の伝承は大きな問題になると考えられる．

そこで本研究では，高度な技能，高品質，短納期などが要求される単品鋳物の製造工程を事例に取り上げ，暗黙知と形式知の連携による知識表現法について提案し，XML（eXtensible Markable Language），SMIL（Synchronized Multimedia Integration Language），XVL（eXtensible Virtual world description Language）などを活用したインタラクティブな熟練技能伝承システムを構築し，その有用性について確認したので以下で報告する．

２．技能伝承法

２．１　従来の技能伝承手法の現状および問題点

現在、技能伝承においては，ビデオライブラリ，技術ドキュメントは資料作成に多くの労力や資金を伴う点や，作成後の使い勝手の悪さなどから，多くのメリットがあるにも関わらず活用されず，大多数の工場では従来どおり OJT のみで技能伝承が行われているというのが現状である．

また，これらの手法を知識変換の観点からまとめると，OJT やビデオライブラリでは暗黙知を暗黙知として伝えており，技術ドキュメントでは暗黙知を形式知に変換して伝達してい

る．非明示的な知識を非明示的な知識として，また非明示的な知識を情報が欠落した明示的な知識に変換して伝達するというこれらの手法を単独で用いることは，伝達される知識の質や量，効率が悪いと言える．

従来の技能伝承手法の問題点を以下にまとめる．

(1)非明示的な知識である暗黙知を，伝達しにくい形で伝えている．

(2)求める知識がなかなか探し出せない．

(3)技能伝承のための資料作成に，多大な労力と資金を必要とし，それらに見合うだけの成果が上がっていない．

(4)ビデオライブラリ，技術ドキュメントなど作成した資料を活用しきれていない．

(5)OJT，ビデオライブラリ，技術ドキュメントなど幾つかの手段があるにも関わらず，ほとんど OJT のみで行っているために，併せて活用することがほとんど無く，それぞれのメリットを生かしきれていない．

このように、従来の OJT および技術ドキュメントを主とした技能伝承法では、効率的な技能伝承が行えないのが現状である。

２．２　新たな技能伝承手法

これまで述べてきたように，これまで行われてきた技能伝承手法にはさまざまな問題点があると考えられる．これからの新たな熟練技能伝承システムに求められるものを次に示す．

(1)非明示的な知識である暗黙知を伝達しやすい形で提示する必要がある．技能伝承とは暗黙知を伝達することであり，技能伝承を目的とする熟練技能伝承システムでは，暗黙知を如何に伝達しやすい形で提示することができるかが問題になる．

(2)擬似的，仮想的な「場」の共有を可能にするために，「場」の共有を妨げる，時間的・空間的・規模的な制約を部分的でも克服する必要がある．「場」の共有が可能になれば，部分的ながらも技能の保存が可能になる．

(3)映像から獲得できる知識の質・量は，個人の知識，経験が大きく影響するために，個人の力量に依存しない提示方法が必要になる．つまり，映像に含まれる膨大な知識を明示的に提示することが求められる．

(4)膨大な知識の中からユーザが求める知識を容易に探し出すことが可能である．

(5)資料作成に多くの労力や資金を必要とせず，さらに過去に作成した資料を再び活用できるような拡張性を有している．

(6)従来のように，OJT のみで技能伝承を行うのではなく，今まであまり活用されなかったビデオライブラリ，技術ドキュメントなどを活用し，それぞれのメリットを生かすことで伝達される知識の質の向上および効率化を図る必要がある．

次に，本研究で提案する熟練技能伝承システムの特徴を示す．

(1)SMIL を用いたマルチメディア表現

本システム内では，技能は SMIL を用いて映像や映像に関連する文字情報，図などを同期的に再生し，並べて提示している．このように暗黙知を暗黙知と形式知が連携させることで，暗黙知を伝達しやすい形で提示できるため，ユーザが効率的に知識を獲得することができる．加えて，このように複数のクリップを同期的に再生させることで，多くの情報を伝えること

ができるために，本システムは時間的，空間的な制約を部分的に克服し，擬似的，仮想的な「場」の共有が可能になっている．

また，SMIL を用いることで先に述べたような複数クリップの同期的再生だけでなく映像内にリンクを貼ることができるために，映像に含まれる知識を明示的に表現できるために，個人の知識，経験に関係なく，映像から熟練技能者と同等の知識を得ることができる．加えて映像中の製品を XVL 形式のデータとして表現し，映像に関連付けているために，映像が表現しきれない情報も得ることができるようになっている．

そして，SMIL は普通のテキストエディタさえあればよく，簡単な文法なので，容易に作成することができる．

このようにビデオライブラリ，技術ドキュメントを組み合わせた本システムを活用して知識を獲得し，OJT を通じて知識を実際に経験していくことで，それぞれのメリットを生かした技能伝承が可能になる．

(2)インタラクティブ性

本システム内は，データは SMIL，XVL，XML などの活用できる形式になっているので，検索をかけることができる．合わせて，Java3D を用いた知識可視化アプレットが必要な情報までナビゲートするなど，ユーザが選択できるために，柔軟性の高い，効率的な知識獲得が可能になる．

(3)インターネットベース

システム内で用いられている SMIL，XVL，XML などのデータ形式はインターネット上での利用を前提としている．インターネット上での利用によって，「場」の共有における規模的な制約を克服でき，多くの人が利用することができるようになる．

２．３　技能伝承と知識創出

本研究で作成した熟練技能伝承システムは，先に述べたように SMIL などのマルチメディア技術を用いて熟練技能者が持つ熟練技能，経験，知識などの暗黙知を形式知と連携させて表現することで，暗黙知をより伝承・継承しやすい形式で伝達している．このようにすることで，効率の良い技能伝承が可能になるために，組織で熟練技能を共有できるようになり，そこから新たな知識を創造することも可能になる．そこで，本論文での暗黙知と形式知との連携について SECI モデルを用いて述べる．

ナレッジ・マネジメントの活動モデルの一つとして SECI モデルがある．これは，知識は４つの知識変換プロセス，(1)共同化（Socialization），(2)表出化（Externalization），(3)連結化（Combination），(4)内面化（Internalization）を通して暗黙地と形式知との間の相互作用から創造されるという概念である．SECI とは４つの知識変換プロセスの頭文字をとったものである．

各知識変換プロセスの定義は以下のとおりである．

(1)共同化：個人や集団が体験を共有することで各個人が暗黙知を認識するプロセス

(2)表出化：各個人に内在している暗黙知を言語化することで明示化され，集団で暗黙知を共有・統合し，新たな知識（形式知）を獲得するプロセス

(3)連結化：集団で共有している形式知同士を連結させ，組織的に増幅させることで，よ

り有益な形式知を獲得するプロセス

(4)内面化：集団が獲得した形式知をもとに，個人が行動することで，形式知を暗黙知として自身に内在化させるプロセス

これらの知識変換プロセスを通して，従来個人の持っている知識である暗黙知を形式知に変換することで知識は伝播しやすい形式になり，組織で知識を共有することが可能になる．このように，知識変換プロセスをエンドレスに繰り返すことを，「知識創造スパイラル」と呼ばれる．

２．４　熟練技能伝承システムの活用例

図１はマルチメディア技術を用いた熟練技能伝承システムの活用例と SECI モデルの知識変換プロセスを対応させたものである．

(1)共同化はユーザが OJT や現場作業を通じて，問題意識を明確化し，暗黙知を認識するプロセスである．

(2)表出化はユーザが熟練技能伝承システムを活用して，文章や技術データなどから形式知を獲得するプロセスである．

(3)連結化はユーザが映像などから，形式知と暗黙知が連携した形で提示される知識を獲得するプロセスである．

(4)内面化はユーザが技能伝承システムから得られた知識をもとに，行動・学習することで自分のものにしていくプロセスである．このシステムを活用しながら，知識を自分のものにしていくなかで，新たな知識創造が可能になり，また新たな暗黙知を獲得するようになる．

このように暗黙知を形式知と暗黙知が連携した形で提示することで，ユーザは効率良く知識を獲得することが可能になる．

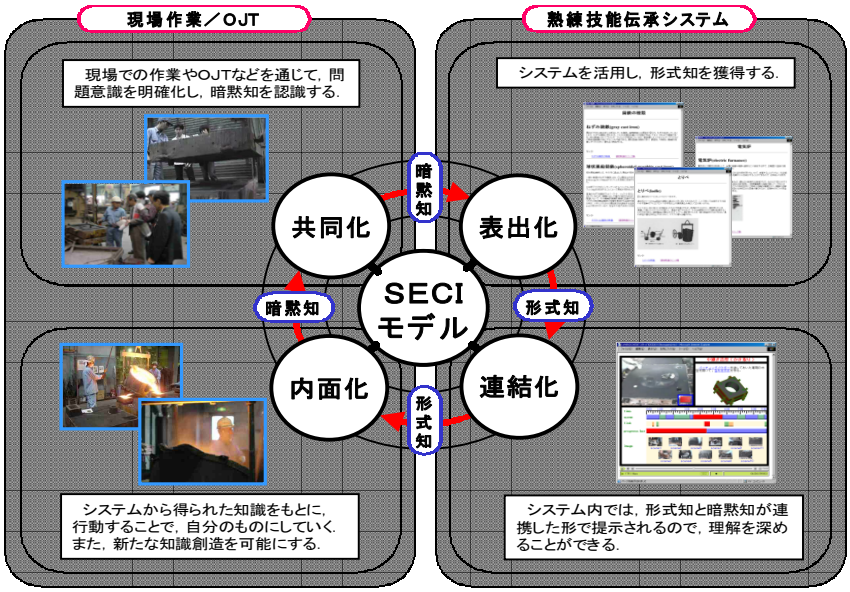


図１　熟練技能伝承システムの活用例

２．５　システム概要

図２に熟練技能伝承システムの概要を示す．本システムは，図２のように Web 上でのシステム利用を想定しているが，現時点では，スタンドアロン環境での動作しか確認をしていない．しかし，データ形式に XML，SMIL，XVL など Web 上での利用に向いている言語を採用しているために， Web 上での利用に切り替える際にも，大幅なシステムの変更は無いものと考えられる．

システムの利用としては，最初にユーザは検索やナビゲーションを使って，検索結果の一覧やナビゲーションをたどることで，目的とする知識にアクセスする．暗黙知に分類される知識であれば SMIL を使ったマルチメディア画面が提示され，形式知に分類される知識であれば XML を使った文書が提示される．共に，キーワードになるような知識にはリンクが貼られており，必要に応じてユーザが選択し，望む知識だけを獲得していく．また，映像では説明できない部分や文書では説明できない部分に関しては，互いに文書，映像を活用し連携させることで，それぞれのメリットを生かしている．そして，映像では表現できない部分については，３次元形状データやアニメーションを活用することで，映像の限界を補っている．

以上のように，暗黙知を暗黙知と形式知が連携させることで，ユーザに知識を伝播しやすい形で提示している．また，求める知識を容易に探し出すことが可能で，ユーザが選択しながら学ぶことができるなど，効率的に求める知識を獲得できるようになっている．

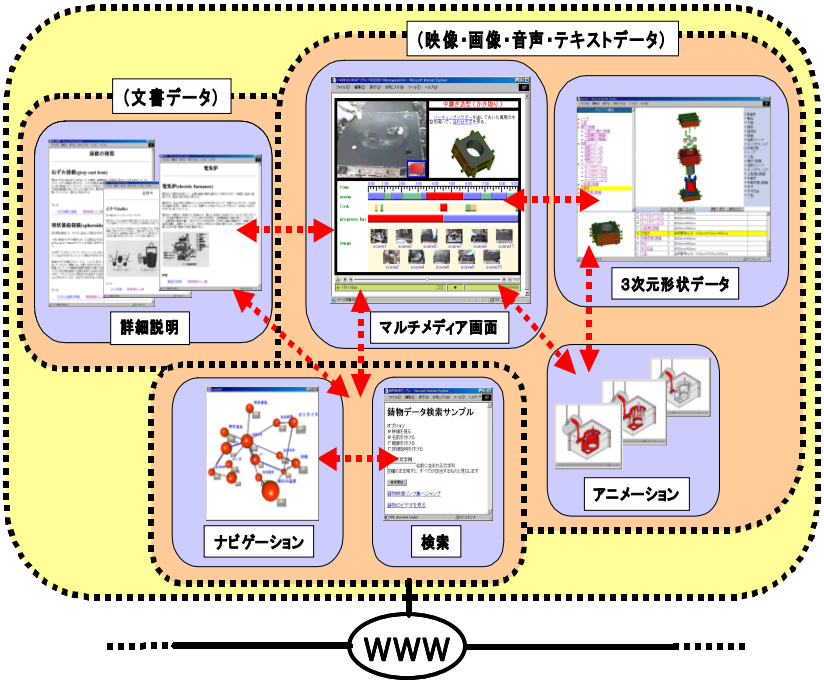


図２　熟練技能伝承システム概要

３．熟練技能伝承システムの構築

３．１　知識の体系化

システム作成に先立ち，熟練技能者へのインタビュを通して，鋳造に必要な知識の整理・分析・関連付けを行い，知識全体の体系化，および各知識間の関連付けを行っている．

知識の体系化については，各知識間の包含関係を調べ，概念ごとにまとめ，階層的に分類した．そして，各知識間での知識の重要度の評価を行った．体系化に際しては，各工程をひとまとまりと考え，工程ごとに，資材，道具，関連用語，技能などの概念をもとに知識の分類を行った．それは現在では昔のように職人が製品ごとに全工程を担当することはなく，工程ごとに担当する分業制が採用されているためである．

また，各知識間の関連付けについては，体系化した知識を基に，そして各知識間の関連および関連の強さについて分析し評価を行っている．

３．２　知識可視化アプレット

鋳造に必要な知識というのは非常に多岐にわたるために，上記のように知識を体系的に整理しただけでは不十分で，ユーザがシステムを活用する際には，要求される知識へのアクセスの容易さが重要になってくる．そのために本論文では図３に示すような，知識の概念，重要度，知識間の関連，関連の強さなどの情報をノードの大きさ，色，形状，リンクの色や太さなどで視覚的に表現するアプレットを Java3D を用いて作成している．なお，このアプレットには先に述べた体系化された知識が反映されている．

また，知識を表現するには知識全体の体系，および各知識の関連という２つの表現が必要と考え，知識の可視的な表現において，マクロな視点（知識全体の体系・階層の表示），ミクロな視点（知識間の関連の表示）を切り替えて表示できるようになっている．

このように知識を体系的に整理することで，知識を格納するデータベース内におけるデー

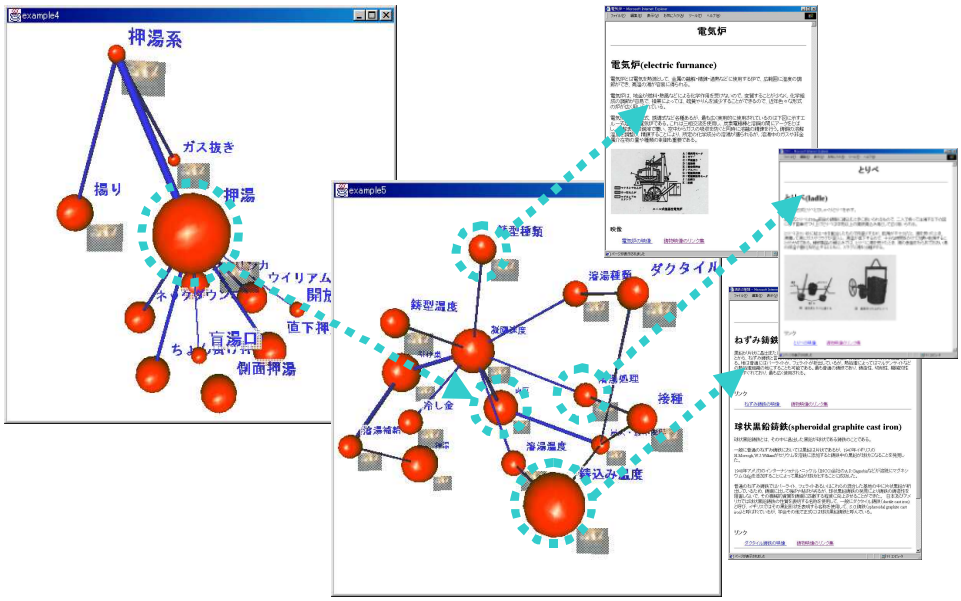


図３　知識可視化アプレットの仕様

タの関係の構造を定めることができるだけでなく，ユーザ，特に初心者にとって効率的な学習が可能になる．知識可視化アプレットと組み合わせることで，知識を可視的に表現できることから，全体像が把握しやすくなり，要求される知識や関連する知識を俯瞰的に捉えることができる．さらに，知識の重要度，および関連の強さに関しても明示的に表現されていることから，ユーザが要求される知識をより効率良く獲得することが可能になる．

３．３　映像情報の提示

映像を活用することで，技能，特に「動き」を記録・保存できるために，擬似的，仮想的な「場」の共有が可能になる．しかし，先に述べたように，従来の手法のように映像だけでは以下のようなさまざまな問題点が存在する．

(1)映像に含まれる膨大な知識，なかでも暗黙知が明示的に表現されていない

(2)撮影している空間を部分的にしか表現できない

(3)映像に含まれる場面などについての情報がない

そこで本システムにおいては，SMIL を活用し暗黙知と形式知を連携させて提示することで，これらの問題点を克服している．詳細については追って説明を加えていく．

図 4 は本システム内の映像情報の提示例である．映像は RealVideo ファイル（.rm），文章は RealText ファイル（.rt），画像は RealPix ファイル（.rp）にまとめられており，これらは全て SMIL によって記述され統合されている．図 4 の例では，1 つの RealVIdeo ファイル，それぞれ 7 つの RealText ファイル，RealPix ファイルから構成されている．

レイアウトとしては，上下に分けることができ，上部が映像中に含まれる知識（技能）の説明，下部が映像自体に関する情報を提示している．

図 4　映像情報の提示例

３．３．１　映像中の知識（暗黙知）の表現

図5は映像情報の提示例の上部を表している．映像中に含まれる知識を明示的に表現している．図5の左部はメインの映像を提示する部位でであり，右部は映像を説明する部位になっている．

「映像」部は，映像を提示する部位で従来のビデオライブラリはこの部位に相当する．

「場面タイトル」部で映像の現在の場面についてのタイトルを提示しており，後述するような映像に含まれる場面に関する情報の提示と併せて，ユーザが見ている映像にどのような場面が含まれているのか，また現在の映像が映像全体に対してどのような位置付けなのかといった，映像自体に関する情報の獲得を助けている．

「解説文」部では，映像の説明を提示する部位で，この文章の作成にあたっては熟練技能者の方に実際に映像を見てもらいながらインタビュを行い，その結果を反映させている．

「図」部では，映像が表現しきれない部分や解説文で説明しきれない部分を表現しており，全体像を把握しやすくすることで，理解を助けている．

「映像」部の中にある，「リンク表示」部では映像中に設定されたリンクの位置を表現している．映像中にリンクを設定することで，映像に含まれる知識を表現でき，より詳細な情報を提示することが可能になる．ところが，リンクの位置が表示されるわけではないので，別にリンクの位置を示す必要がある．しかし，映像中にリンクの位置を表示すれば，映像を隠すことになり理解を妨げてしまうために，映像中に右下部に影響を与えない大きさで領域を設定し，リンクの位置を表示している．映像の静止画像を縮小した画像をスライドショ形式で表示している．「リンク表示」部で色が塗られている部分と映像内の部分は縮小しているものの対応しており，映像中のリンク位置がわかるようになっている．

これまでに述べた，映像の解説やリンク設定の際には，全て熟練技能者の方に実際に映像を見てもらいながらインタビュを行っている．その中で，映像に含まれる知識を指摘してもらい，解説を加えてもらった結果を反映しており，これらは全て熟練技能者の観点からまとめられている．このような情報を映像と併せて提示することで，映像から獲得できる知識の質・量が個人の技量・力量に依存することなく，例え初心者でも映像から熟練技能者と同程度の知識を獲得することが可能になる．

また，映像中だけでなく解説文中にもリンクが設定されており，リンク先の詳細情報の活用については，ユーザが選択することができるようになっている．このようにユーザの選択にまかせることで，望む知識だけを獲得できるなど，柔軟性の高い効率的な学習が可能になる．

以上のように，暗黙知である技能を映像（暗黙知）と文章・図（形式知）などを連携させて表現することで，それぞれのメリットを生かすことができ，伝達する情報の質・効率が向上する．

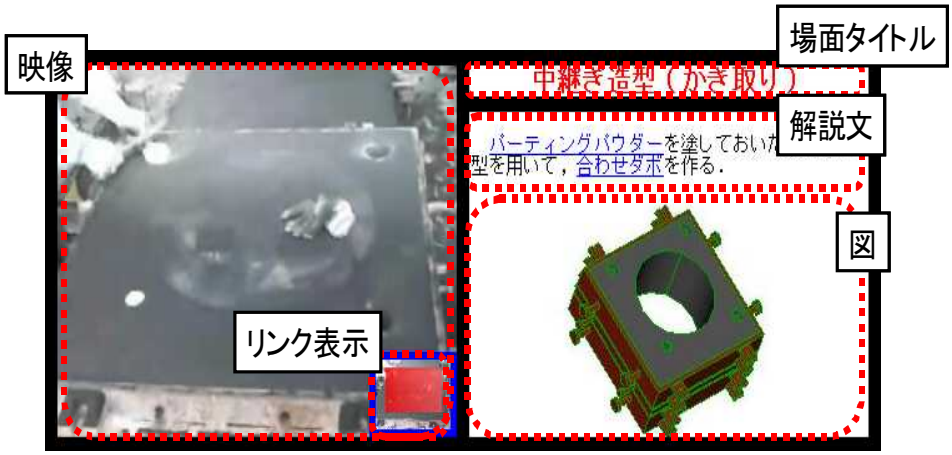


図５　映像情報の提示例（上部）

３．３．２　映像情報の表現

図６は映像情報の提示例の下部を表している．映像中に含まれる場面の展開やリンクが設定されている時間，現在の位置など映像自体の情報を，時間軸に対応させて視覚的に表現している．

「場面」部では，映像に含まれる場面およびそれぞれの時間，現在の場面などの情報が提示している．この部位では，ユーザに見ている映像にどのような場面が含まれていてどの程度の長さなのか，現在の場面が映像全体に対してどのような位置付けなのかといったような映像自体に関する情報を表現している．

「リンク」部では，映像中にリンクが設定されている時間帯の表示をしている．その時間帯は，映像にリンクを設定するだけの知識が存在していることを表しており，他の時間帯に比べて多くの知識を含んでいる．言い換えれば，映像の重要度が高い時間帯を表現している．

「現在位置」部では，現在の場面を時間軸に対応させた形で表現している．現在位置の時間に合わせて，対応する場面やリンクの色が変化することで，ユーザが現在の場面を把握やすくしている．

「画像」部では，各場面の静止画がサムネイル形式で表現されており，各場面の様子を画像で確認することができ，場面や展開の理解を助けている．

また，「場面」部の場面や「リンク」部のリンクは，マウスでクリックすると映像がその場面や時間に切り替わるようになっているなど，単に視覚的に表現しているだけでなく，映像の場面や時間と対応している．そのために，ユーザは望む映像にすぐアクセスすることができる．

以上のように，映像に関する情報を視覚的に表現することで，映像全体を俯瞰的に把握しやすくなり，映像の理解を助けている．また，ユーザは自身の望む映像にすぐにアクセスすることができるなど，映像が活用しやすい形になっている．

図 6　映像情報の提示例（下部）

３．４　形状情報の提示

図 7 は本システム内の形状情報の提示 例を示す．レイアウトとして，図 7 中の右上部，左下部にそれぞれ XVL ファイルの全体，部分を表示する「XVL プレーヤ」部，および左上部に部品の包含関係を表示する「アセンブリ構造表示」部，右下部に部品の詳細情報や一覧を表示する「部品表」部から構成されている．

「XVL プレーヤ（製品表示）」部は 3 次元形状データである XVL ファイルを表示する部位で，製品全体の 3 次元形状を表示する．この製品は映像と関連付けられており，映像内で扱われている製品を表している．先に述べたように，映像だけでは撮影した空間全体を表現することは不可能で，限られた空間しか表現することができない．しかし，理解のためには全体像の把握が必要で，このギャップを埋める必要がある．特に今回のシステムは単品鋳物を対象とするために，製品の形状が複雑で，工程が多いことなどから，製品の情報が不可欠であると言える．熟練技能者は映像を見ただけで，その製品の形状，方案，仕様などの情報を獲得できるなど，個人の技量・力量に負う部分が大きく，この部分も理解に差が出る要因となっている．そのために，映像内で扱われている製品の CAD データを作成し，XVL 形式に変換して，映像と関連付けることで熟練技能者でなくても，容易に全体像（製品形状）を把握できるために，映像の理解を助けることができる．また，製品形状が複雑であればあるほど効果を上げると考えられ，技能の保存の観点からも有効であると考えられる．

また，「XVL プレーヤ（製品表示）」部の隣にある，「状態操作ボタン」，「プレーヤ操作ボタン」でユーザはそれぞれ XVL ファイルの Group，Material などの操作，XVL プレーヤのフィット，移動，回転などの操作が可能になっており，製品形状をさまざまな角度から確認することができる．

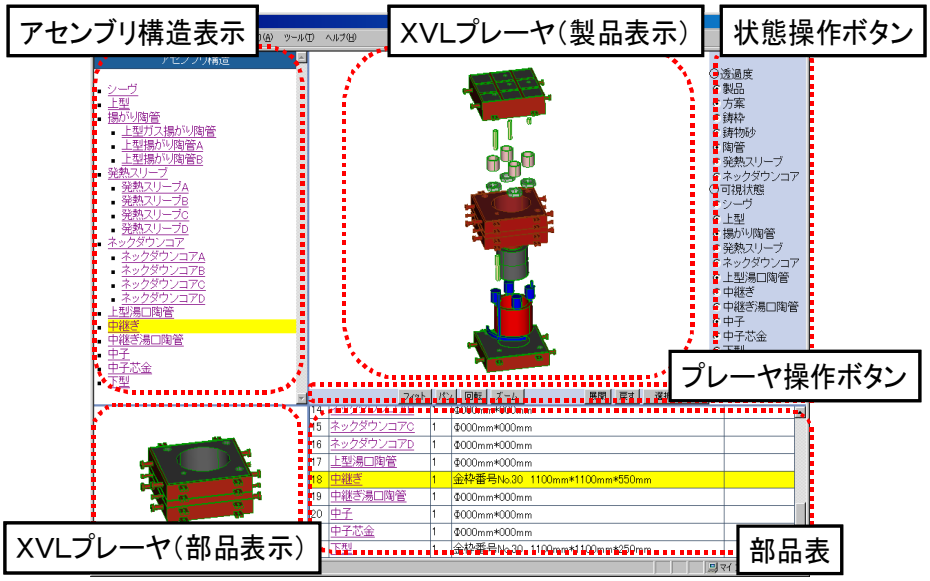


図 7　形状情報の提示例

「XVL プレーヤ（部品表示）」部は，他の部位で選択された部品を表示する部位である．このように製品表示用と部品表示用の XVL プレーヤをそれぞれ用意することで，個々の部品に関して全体を俯瞰的に把握しながら理解することができる．

「アセンブリ構造表示」部では，製品に使われている部品間の包含関係を表示し，部品間の関連を明示的に表現している．

「部品表」部では，製品に使われている部品の寸法や数量などの詳細情報や一覧を表示している．このような詳細情報を記録しておくことで，技能の保存にも有効であると考えられる．

4．結　言

本研究では，XML，SMIL，XVL などを活用したインタラクティブな熟練技能伝承システムのプロトタイプの作成を行った．本システムにおける暗黙知と形式知の連携による知識表現法およびインタラクティブなシステムについて，鋳造分野の熟練技能者に評価してもらい，一定の有用性を確認した．

文　献

1）綿貫啓一，岡崎元：SMIL/XML による熟練技能者の知識表現，日本機械学会 2001 年度年次大会講演論文集(IV)，No. 01-1，(2001)，pp. 57-58.

2）綿貫啓一，荒巻利彦，高橋悠介：インターネット／マルチメディア技術を用いた熟練技能伝承システムの構築，2002 年度精密工学会秋季大会講演論文集，B36，(2001)，pp. 69.

3）綿貫啓一，荒巻利彦，高橋悠介：マルチメディア技術を用いた単品鋳物製造のための熟練技能伝承システムの構築，日本機械学会関東支部ブロック合同講演会－2001 鳩山－講演論文集，No. 010-2，(2001)，pp. 99-100.

4）綿貫啓一，荒巻利彦：XML による 3 次元 CAD データの共有化，日本機械学会関東支部ブ

ロック合同講演会－2001 鳩山－講演論文集，No.010-2，(2001)，pp.251-252.
５）綿貫啓一，荒巻利彦，高橋悠介：インターネット／マルチメディア技術を用いた熟練技能伝承システムの構築，2001 年度精密工学会秋季大会講演論文集，B36，(2001)，pp.69.
６）綿貫啓一：インターネット／マルチメディア技術を用いた設計・生産支援のための熟練技能伝承システムの構築，溶接学会軽構造接合加工研究委員会講演資料，MP-290-01，(2001)，pp.1-6.
７）綿貫啓一：インターネット／マルチメディア技術を用いた熟練技能伝承システムの開発，埼玉県工科系大学における第 1 回合同シリーズ展要旨集，SA-12，(2001)，pp.6.
８）綿貫啓一：同期マルチメディア統合言語 SMIL と技能伝承，精密工学会ライフサイクルの支援のための XML 技術テキスト，(2001)，pp.32-40.
９）綿貫啓一：マルチメディア・インターネット技術を用いた鋳物熟練技能のデジタル化と技能伝承の試み，技能の科学化と科学の技能依存に関する研究会資料，(2001)，pp.1-5.
１０）綿貫啓一，荒巻利彦，高橋悠介：インターネット／マルチメディア技術を用いた熟練技能伝承システムの構築－暗黙知の獲得および表現法－，2002 年度精密工学会春季大会講演論文集，D67，(2002)，pp.186.
１１）綿貫啓一，荒巻利彦，高橋悠介：同期マルチメディア統合言語を利用した単品鋳物製品設計支援システムの構築，日本設計工学会，(2002)，pp.11-16.
１２）綿貫啓一，荒巻利彦，高橋悠介：マルチメディア技術を用いた熟練技能伝承システムにおける暗黙知と形式知の連携，日本機械学会関東支部ブロック合同講演会-2002 宇都宮-講演論文集，No.020-2，(2002)，pp.227-228.
１３）綿貫啓一，荒巻利彦，高橋悠介：インターネット／マルチメディア技術を用いた機械設計知識伝承システムの構築，日本機械学会 2002 年度年次大会講演論文集，Vol.V，No.02-1，(2002)，pp.63-64.
１４）綿貫啓一，荒巻利彦，高橋悠介：インターネット／マルチメディア技術を用いた熟練技能伝承システムの構築，2002 年度精密工学会秋季大会講演論文集，A32，(2002)，pp.22.
１５）綿貫啓一，荒巻利彦，高橋悠介：インターネット／マルチメディア技術を用いた単品鋳物製品設計支援システムの構築，日本機械学会第 12 回設計工学・システム部門講演会講演論文集，No.02-31，(2002)，pp.145-148.
１６）綿貫啓一：マルチメディア／インターネット用いた鋳物熟練技能伝承システムの構築，日本塑性加工学会 CAF 分科会製造業技能の技術科・共有・伝承と IT 研究会資料，(2002)，pp.1-5.
１７）綿貫啓一：インターネット／マルチメディア用いた鋳物熟練技能伝承システムの構築，埼玉大学 21 世紀総合研究機構プロジェクト発表会資料，(2002).
１８）綿貫啓一：マルチメディア／インターネット技術を用いた熟練技能伝承システムの開発，社団法人埼玉県経営者協会会報，(2002)，p.17.
１９）綿貫啓一：IT 技術を用いた鋳造熟練技能のデジタル化と技能伝承の試み，財団法人産業研究所，13-2，(2002)，pp.34-53.
２０）綿貫啓一：モノづくりの熟練技の”つぼ”をマルチメディアで伝承，商工ジャーナル 7 月号，(2002)，pp.60-63.

２１）綿貫啓一：技能伝承と知識創出，2003 年度精密工学会春季大会講演論文集(CD-ROM 論文集)，A73，(2003).
２２）綿貫啓一，荒巻利彦：インターネット／マルチメディア技術を用いた設計知識伝承システムの構築，2003 年度精密工学会春季大会講演論文集(CD-ROM 論文集)，A80，(2003).
２３）綿貫啓一：インターネット／マルチメディア技術を用いた熟練技能伝承システムの構築，ECOM Semantic Web 部会(Ad hoc 委員会)資料，(2003),pp.1-5.

メタデータ概念と記述

ドコモ・システムズ（株）　大野邦夫

1. はじめに

セマンティックWebを計算処理的な観点のアーキテクチャで採り上げられないかという課題があり、この報告はその問題への一つの考え方を示すものである。セマンティックWebは、XML層の上にRDFによるメタデータ層があり、その上にオントロジ層、ロジック層があり、意味的な処理を行うとされている。従って、XMLで記述されるRDFによるメタデータ層は基盤的な役割を果たしていると考えられる。

メタデータという語はデータに関するデータという意味で、図書館の書誌情報や地理情報などの分野ではかなり日常的に使われている。前者[1]については、コンテンツのメタデータを定めたDublin Core[2]があり、XMLによる記述が導入されてからは、急速に普及が進んでいる。地理情報におけるメタデータについても標準化の動きがあり、ISOの最終ドラフトになっている。（ISO/CD 19115.2）[3]

以上のコンテキストで使われている「メタデータ」という用語は、データベースのスキーマ的な意味に過ぎないが、実際はより深く広いい意味を持つ。要するに同一カテゴリの個々のデータが共通に持つ一段階抽象度の上がったレベルの情報を意味する。このような概念を一般的に「メタ」と呼ぶが、メタ概念はオブジェクトのクラス、構造体の構造、SGML/XMLのDTD、XMLにおけるXMLスキーマなどに含まれるものである。「それは単に集合の概念に過ぎないではないか」と思われる人もいるであろう。実はそのとおりであるが、特有の観点にフォーカスされる場合もある。

XMLのメタデータとしては、先に述べたとおりRDF[4]が取り上げられる。この場合は抽象度の上がった情報というよりは、ウェブで管理されるXMLコンテンツの管理情報的な役割を持つ。セマンティックWebの場合には、RDFは管理情報というよりは命題定義に近いものである。以上のようにメタデータという用語はコンテキストに依存して多様な意味を持ち得る。ここでは、メタ概念を歴史的、分析哲学的に考察し、それがセマンティックWebの構造に対してどのように関与し得るかを検討することとしたい。

2. メタとは何か

2.1 アリストテレスの形而上学

ところで、メタデータのメタとは何を意味するのであろうか。多くの人にとってメタという語は、アリストテレスの形而上学（メタフィジックス）に遡るのではあるまいか。アレクサンダー大王の家庭教師であったアリストテレスは、BC300頃にギリシャ哲学史として形而上学を執筆した。記述の中心はプラトンのイデア論の解説と批判であった。

2.2 メタの理論化：集合論

アリストテレスは、共通の述語を持つものは同一の性質を持つと考えたが、これは集合の概念に近いものである。プラトンのイデア論は、そのような共通の性質を階層的に積み上げ、その最高位の究極のメタをイデアとして位置づけたと思われる。このような抽象化のプロセスを厳密に体系化したのが近代の論理分析哲学であろう。次にライフサイクルに関与する可能性のある様々な情報メディアとメタデータについて検討してみよう

2.3 人類史における情報メディアの進展

表1は人類発祥から今日に至るまでの人類が用いてきた情報メディアを総括的にまとめたものであるが、ここではこれらのメディアの進展自体がメタ情報を指向する人類の基本的な性格であったことを論じたいと思う。

表1 人類が用いてきたコミュニケーション手段

適用時期	コミュニケーション手段
BC.1,000,000～	叫び声，ジェスチャ
BC.30,000～	洞窟壁画
BC.5,000～	象形文字
BC.3,000～	表意文字
BC.1,500～	表音文字
AD.1,000～	数式 代数学
AD.1,800～	近代論理学

2.3.1 叫び声，ジェスチャによる通信

言葉を持たなかった原始人類の基本的なコミュニケーション手段は、叫び声やゼスチャーによるものであったと考えられる[5]。その後、人類はより効率的なコミュニケーションを求めるようになった。例えば道具の作成と使用などにおける協調作業のためである。道具を作ると言うことは、それを使う目的があり、その目的を達成するプロセスが認識されていたことを物語る。おそらくは、仲間、動物、場所などを特定し、その対象へのアクションを指示するような簡単な言語が使われていたはずである。

このような原始的な言語については言語学の観点からの研究が行われている。ピジンやクレオールといった言語モデルで、言語の誕生モデルと言っても良いであろう。ピジンというレベルの言語は、指示のみで文法不在の言語である。赤ん坊が初めて言葉を覚えたときに使う会話のレベルと考えれば良い。クレオールは、ピジンを母語とした言語で、簡単な動詞や形容詞を伴ったものであるが、体系化された文法を持つには至っていないレベルである[6][7]。

人類が言語を持つに至るのは、これらのピジン、クレオールのレベルからさらに豊富な語彙と高度な文法を展開したからにほかならない。

2.3.2 洞窟壁画

BC.30,000年ころから、人類の芸術的な遺産として洞窟壁画が残されている。南アフリカの動物の洞窟壁画（29000年前）や、スペインのアルタミラ、フランスのラスコー洞窟の壁画（25000−13000年前）などが有名であるが、その目的は何らかの宗教的なセレモニーに使われたと考えられている。描かれている動物は、狩猟の対象であるものが殆どであるが、中には人間が混じっていたり、人間と動物が組み合わされたような架空の動物も描かれたりしていた[8][9]。

いずれにしろ、そのような絵画を通じて、集団を統括するための何らかのコミュニケーションを行っていたことは確実である。

2.3.3 象形文字

(1) シュメールの楔形文字

BC.10000年ころから、人類は象形文字を使いはじめたらしい。シュメールの楔形文字は、BC6000年頃が起源であると言われているが、これは帳簿をつけるための絵文字から発展したと言われている[10]。粘土板に穀物や家畜の数を記すために、簡単な絵を書いたのが発端の模様だが[11]、粘土板に絵を書くためには棒の先を尖らせるか楔形に加工する必要があった。その後、絵を描くよりは楔形の棒の先端を粘土板に押し付ける方が記録方法としては能率的なので、楔形文字が生まれたと思われる。

(2) エジプトのヒエログリフ

エジプトのヒエログリフ（聖刻文字）は、象形文字としては最もポピュラーな存在であろう。最近でもエジプトのヒエログリフに関連する書物が出版されている[12][13]。絵文字の種類は、当初（BC4000）は700種程度であったが、クレオパトラが在位した最後の頃（AD27）は5000種にも上っていたと言われる。ヒエログリフは、農業、医学、教育、法律分野など幅広い分野を記述している。そのために、必要な拡張が行われ、最後には5000種類の文字に発展した。

(3) 黄河文明の漢字

黄河文明の漢字も元は絵文字であったが、漢字の場合は体系的なルールを用いて字種を拡大し、わが国を含め現在まで用いられている。

2.3.4 表意文字

象形文字において、楔形文字とヒエログリフは消滅したが、漢字の世界だけが生き残り、その後表意文字へと発展する。象形文字が具体的な事物から出発したので、具体的な対象を指示表現するのに適合していたのに対し、表意文字は、より一般的、抽象的な表現を可能とせねばならなかった。そのために、新しい概念を表すためには新たな文字の追加か、既存文字の組み合わせを要し、結果的に新たな字種が追加され文字セットは膨大になった。また合成語としての熟語も定義されたのであるが、これらの字種や熟語を覚えるのは容易なことではない。

漢字を用いると言葉は通じなくても文字で理解可能というメリットがあることを指摘されるが、そのトレードオフとしての大量の問題点を抱えている。最大の問題は文字セットの記憶に時間を要することで、その結果識字率の向上は困難を極めるのである。日本に限っても、江戸時代末期から幾度となく漢字廃止の提案がなされてきたが、常に議論を呼びながら今日に至っている。

2.3.5 表音文字

アルファベットは、フェニキア文字がギリシャに普及したものである。表音文字は、アルファベットに端を発すると思われがちであるが、必ずしもそうとは言えない。アルファベット以前に、象形文字を用いた表音文字が用いられていたことが知られている。日本語の仮名も典型的な表音文字として知られている。日本語の仮名の場合は、当初は漢字の音を借りて万葉仮名のように漢字表記の仮名が使われはた。その後、音に対応していた漢字の簡略表現（ひらがな）や漢字の部分を用いて（カタカナ）表音文字である仮名が形成された。

アルファベットの場合もかなに近いプロセスを経て象形文字から形成された模様である。アルファベットの場合は、基本的には26文字を覚えれば読み書きが可能となる。楔形文字は600種、ヒエログリフが700−5000種、漢字が数万種もあることを考えると、これは革命的である。結果的に、アルファベットを採用した民族や国家における文字の普及、識字の向上に貢献に貢献した。

世界史をマクロに捉えると、西欧文明が他の文明を押さえて世界史の主流になったのは、文字体系としてアルファベットを採用したことに深く関係すると思われる。同様に、日本の仮名は日本人の識字率向上に貢献しており、非西欧で日本が先進諸国に比肩し得る地位を獲得できたのは仮名を発明したことが大きく貢献していると思われる。

2.3.6 代数学

数の概念、すなわち、和、差、積、（商）などは、世界史的に見ると、当初はエジプトで発達した。特にナイル川の氾濫を通じて1年が365日（4年に一度は366日）であることを明らかにし、農業における収穫の見積もりなどの必要性から測量技術が発達し、加減乗除の数式概念が形成された[14]。具体的な数による計算は人類の幅広い文明で知られていたが、数を抽象的に扱う変数の体系的概念（代数学）が誕生したのはAD9世紀のことであった。代数学の創始者は、アラビア人のモハンマド・ベン・ムサ・アル・フワーリズミと言われている。彼は830年頃に「アル・ジャンブルの書（Hisab al−jabr w'al−muqabalah）」を著し、これからalgebraという語が生まれた模様である[15]。

2.3.7 近代論理学

論理学は、ギリシャのアリストテレスが体系付けたことが知られている。これは古典論理学と呼ばれ、三段論法として知られているものであるが、自然言語における意味的な概念をベースとしている

ため厳密な体系とは言えない。近代論理学は、古典論理学とは異なり、集合概念と記号論理により構築された代数学的な手法により体系化された論理学である。

近代論理学により数学も再体系化された。フレーゲは集合論に基づく代数学の再構築を試みたが、バートランド・ラッセルのパラドックス（集合そのものを対象とする命題の本質的矛盾）により、暗礁に乗り上げてしまった。その救済のための理論が、「型の理論」であり、計算機科学におけるデータ型の雛型とされている[16]。

その後、ラッセルは、ホワイトヘッドとともに代数学を体系化し、プリンキピア・マセマティカ（”Principia Mathematica”）[17]を完成させた。

2.4 コンピュータが扱う情報メディアの進展

人類が扱う情報メディアは、産業革命後は電信、電話といった通信技術、さらに20世紀に入るとラジオやテレビといった放送技術、写真や蓄音機、映画といった記録・再生技術の進展で飛躍的な発展を遂げることになる。このような技術をベースにコンピュータが扱うメディアも徐々に拡大されてきた。その状況を表2に示す。以下はその解説である。

表2　コンピュータが扱うメディア

普及した年代	コンピュータが扱うメディア
1940年代	2進論理（ENIAC, 機械語）
1950年代	数字，数式計算（Fortran）
1960年代	英数字, 事務処理 (COBOL)
1970年代	漢字処理 (日本語ワープロ)
1980年代	GUI（アイコン，マウス）
1990年代	図形，画像
2000年代	映像，音声

2.4.1　2進論理による機械語システム

最初のコンピュータは、1946年に完成したENIAC（Electronic Numerical Integration & Calculation）であると言われるがこれに関しては異論がある[18]。とは言っても、ENIACがコンピュータとして実働した最初のマシンの一つであることに変わりはない。ENIACのプログラムはパッチボード結線によるものであり、チューリングが提唱した演算とデータを同一の言語体系で実現するコンピュータではない。真の意味のチューリングマシンは、1949年のEDSAC（Electronic Delay Storage Automatic Calculator）が最初の実用機である。これは蓄積プログラム方式による最初のコンピュータで、基本命令セット、サブルーチンなどと言った今日のコンピュータの基本的な機能を実現していた。

2.4.2　数字，数式計算（Fortran）

1950年代に入ると、コンピュータの素子の半導体化が実現し、センターの大型計算機の時代を迎える。最初に適用された分野は、科学技術計算分野であった。微分方程式の数値解法、有限要素法など、従来の手計算では不可能であった分野が開拓され、科学技術の進展に貢献した。具体的な適用分野としては、DoD（国防総省）の各種システム、製造業における設計分野、アカデミックな世界での応用などが主であった。

2.4.3　英数字（事務処理）

1960年代は、TSS（時分割処理）による事務処理分野へのコンピュータの応用が始まる。これに先立ち、英数字コード（ASCIIなど）が制定され、従来からコンピュータを導入してきた分野で使われ始め、それが事務処理分野に普及したと言える。ツールとしては、プログラム開発用のテキストエディタが普及し、それが一般の文書作成やオフィスのデータ作成に適用されたと考えることができ

る。その後、COBOL言語が登場し、官庁や企業における標準的な事務処理用のプログラム言語として定着した。

この時代の注目すべきエピソードとして、Unixの誕生が挙げられる。MITを中心に進められた先進的なプロジェクトとしてProject MACというものがあった。MACはMulti Access Computingの略である。その応用としてコンピュータユーティリティ構想が持ち上がり、そのためのシステムとして、巨大なTSSシステム用の計算機オペレーティングシステムであるMulticsが開発された[19]。コンピュータユーティリティ構想とは、コンピュータを、電気や水道のように日常の用途に使えるようにすることを狙ったシステムで、現在のインターネットによるEコマースや電子政府を当時の最新技術であったTSSにより実現しようとしたものであった。Multicsは、挑戦的な技術であったが、当時の市場ニーズや経済状況的に見て実現可能なレベルにはなく、技術的にも不完全であった。Multicsの技術を簡易化して、当時使われ始めたDECの小型のミニコンピュータ上で実現したOSが、Unixである。このOSはMulticsを単純化したことから、Multi−の代わりにUni−としたとのことである。

Unixには、edという行エディタが付属しており、これを活用するnroffやtroffといった文書作成用のマークアップ言語が開発された。これらのツールは、当初はプログラム開発に伴うドキュメント作成用であったが、後には汎用の文書作成ツールとして使われるようになる。一方、ミニコンレベルのコンピュータを用い、ゴルフボールプリンタやデイジーホイールプリンタを印刷端末とするワードプロセッサが出現する。安価な蓄積媒体としてフレキシブルディスクが出現したのも丁度この頃であった。

2.4.4 漢字処理（日本語処理）

1970年代に入ると、日本語処理システムが登場する。先ず漢字コードが制定され（JIS−6225, 6226）、大手計算機メーカ（IBM、富士通、日立、日本電気、電電公社）における漢字システムが開発された。さらに漢字を出力するためのワイヤドットプリンタやレーザビームプリンタが開発される。さらに1970年代で特筆されねばならない出来事があった。それはマイクロプロセッサの登場である。その結果、パーソナルコンピュータやワードプロセッサの価格性能比が継続的に向上する今日の時代を迎えることになった。この技術革新の波に乗り、1970年代の後半には日本語ワープロが登場した。東芝のJW−10と富士通のOASYSである。その後、日本の計算機メーカや家電メーカは雪崩を打って日本語ワープロを出荷するようになり、日本のオフィス文書を手書き文書からワープロ文書へと変えていった。

2.4.5 GUI

1980年代になると、Xerox PARC（パルアルト研究センタ）の成果であるGUI（アイコン，マウス）が、ワークステーションやパーソナルコンピュータに導入されるようになる。この技術の発端は、Xerox PARCで開発されたAltoマシンであった。このマシンを用い、Smalltalk, Interlisp−Dといったプログラム言語を用いてマルチウインドウシステムが開発され、アイコンとマウスを用いるGUIが導入された[20]。計算機間を高速でつなぐEthernetもこのプロジェクトの成果である。Smalltalkをベースとするオブジェクト指向技術と高速のLANは、今日のネットワーク文化を支える分散オブジェクト技術の発端となった[21][22]。

GUIを用いる実用システムは、Starワークステーションとして製品化された。この日本語版もJStarとして日本で販売されたがこのシステムの隆盛は長くは続かなかった。最大の理由は、価格対性能比にある。汎用のマイクロプロセッサを用いるSunやAppoloのようなワークステーションが出現し、そのOSとして用いられたUnix上のオフィス文書システムとしてのInterleafやXyvision、Framemakerといったライバル製品に市場を奪われていった。さらに、PARCの成果はパソコンの世界ではアップルのマッキントシュに引き継がれ、DTP（Desk Top Publishing）などによりオフィス文書の高品質化が推進された。このGUI技術の流れは、さらにアップルのライバルであるマイクロソフトのWindowsへも波及して今日に至っている。

2.4.6 図形，画像の時代

1990年代に入ると、メモリ価格の低価格化とオブジェクト技術の普及に伴い、図形（2D,3D）や画像（CG）システムが、従来の文書システムと連携して扱えるようになった。Adobe社からは、図形

データを処理するためのIllustraterや画像データを処理するためのPhotoshopが販売され、Kodak社から従来の銀塩フィルム写真をCDに格納するPhotoCDのような技術が提供され、コンピュータアーティストと言った新たな職種が生み出された。

さらにパソコンの低価格高性能化、インターネットの普及、カーナビやWEBにおける地図利用、デジタルカメラの普及などを通じて、図形，画像によるデジタルコンテンツは我々の生活の隅々にまで波及しつつある。

2.4.7 映像,音声,アニメの時代

2000年紀を迎え、ブロードバンドの掛け声とともに情報メディア的には映像,音声,アニメを活用する本格的なマルチメディアの時代へと突入した。ビジネス的には、放送、通信、情報処理がタイトに結びつき、単一の市場を形成しつつある。言い方を変えると、デジタルテレビとWEBが融合するということである。そのキーとなる技術はXMLである。

SMILにより、静的なコンテンツと映像がWEB上で融合するようになり、放送業界の表示規格であるBMLは、今後のWEBの規格であるXHTMLに極めて近いものである。両者の融合は極めて容易であり、両者を視野に置いたアプリケーションが今後ミドルウエアやビジネスロジックとして続々と誕生するであろう。

今後の必然的な流れであるウェブの高速化により、放送はインターネットに融合すると思われる。その段階になると、放送は有線に移行し、無線は個人や移動体を対象にしたモバイル応用に移行する可能性が強い。帯域拡大により、モバイルの世界でも映像やアニメが活用されるようになるであろう。

表**3**　コミュニケーションとコンピュータにおけるメディアの対比

コミュニケーション	コンピュータ
叫び声，ジェスチャ	映像，音声
洞窟壁画	図形，画像
象形文字	GUI
表意文字	漢字処理
表音文字	英数字, 事務処理
数式, 代数学	数字，数式計算
近代論理学	2進論理

2.5情報メディアと型

2.5.1 コミュニケーション vs コンピュータ

表3は、表1と表2を対応つけたものである。表2については、順序を上下逆にして示している。興味深いことに、人間社会のコミュニケーションの歴史とコンピュータ処理の歴史は、表3から分かるとおり、見事に逆向きに対応している[23]。これは偶然であろうか、それとも必然的な理由があるのだろうか。これには以下に述べるように必然的な理由があると考えられる。

2.5.2 コミュニケーションにおける抽象化の歴史

人間のコミュニケーションメディアの進展は，モデル化，記号化，抽象化（メタ化）の歴史であったと言える。人類は、叫び声やゼスチャーに端を発して言語を獲得し、さらに少ない符号で多くの意味内容を伝達するため文字を発明した。さらに普遍性，客観的真理の探究のために、数学や論理学といった考えるための考え方を体系化するに至った。この一連のプロセスは、人類の知的発展の経緯そのものと言えるであろう。

2.5.3 コンピュータにおける具体化の歴史

それに対し、コンピュータの歴史は，具体化，可視化，オブジェクト化の歴史であったと言える。これらは、メモリ・通信コストの低減により具体化された。人間にとって、論理的な思考に頼らないで，直接的に意味内容を把握する方が精神的には容易であり好まれる。論理学や数学を好む人は稀であり、長文の物語を長時間かけて読むよりは、同一内容の映画を短時間で鑑賞することを好むものであろう。コンピュータの世界にこの考え方を最初に持ち込んだのは、PARCのアランケイを中心とするLRGの人々であり、この思想はマルチメディアを推奨する多くのビジネスを生み出している。また、ヒューマン・コンピュータ・インタフェース分野においては、メリーランド大学においてシュナイダーマン教授のダイレクトマニピュレーションの考え方を生み出している。

2.5.4 対応の必然性

従って、表3の対応は偶然ではなく必然性が存在すると言えるであろう。要するに、抽象化を指向する人間の論理的な思考と、コンピュータにおける繹的なデータ導出のメカニズムが対応すると思われる。後者についてさらに具体的に述べると、型・クラスの定義と継承、これらの型やクラスに属するインスタンスの生成やそれに対するオペレーションなどが対応すると言えるであろう[24]。

2.6 記述の理論

2.6.1 帰納と演繹

上記の仮説においては、人類におけるコミュニケーションメディアの進展は抽象化というプロセスであり、一方コンピュータにおいては、具体化というプロセスであった。この両者は、論理的な観点からは、いわゆる「帰納」と「演繹」ということになる。

ところで、帰納、演繹という用語に関しては必ずしも明確な定義がなされているわけではない。演繹に関しては、「正当な前提の下に正当な論理を適用し正当な結論を導く過程」とでも言えるのであろうが、特に帰納に関しては考え方に幅があり種々の議論が存在する。ここでは「帰納的推論」について検討したホランド[25]らの定義に立脚し、「不確実な状況において知識を拡張する推論過程の全てを含む」という立場をとることとしたい。この考え方だと、動物の条件反射やバクテリアの環境適応までも帰納の範疇に含まれることになる。

種々の事象において、帰納的な手法により同一の性質、類似の性質を抽出する。これらは概念形成のプロセスと言えるであろう。それら概念に対して名前を付け、分類の基準にしてゆく。これこそ表意文字や新しい単語の誕生モデルであった。

このような帰納の考え方を体系化し、人間かから見た世界と言うものを定義する試みが宗教や哲学であろう。プラトンのイデア論、アリストテレスの形而上学などは、その古典的な体系であるが、そのような思想・哲学を、より厳密な数学的な視点で体系化しようとしたのが、2.3.7節で述べた論理数学を包含する近代の分析哲学であったと言える[26]。

2.6.2 構成的アプローチと分析的アプローチ

バートランド・ラッセルが、パラドックスの克服のために型の理論を提示したのは1906年のことであるが[27]、その問題の背景としての型の基本的な考え方は、1903年に出版された “The Principles of Mathematics”[28]の付録Bですでに記述されている。それによると、ラッセルは数に対して明確に型（集合）とそのメンバ（インスタンス）という考えを提示しており、現在のデータ型に近い考え方を説明している。“The Principles of Mathematics”は、ホワイトヘッドとの共著で有名なプリンキピア・マセマティカの陰に隠れてあまり知られてはいないが、この書を通じて彼が世界を論理で構築することに挑戦したことを窺い知ることができる。

それによると、ラッセルは人間の知性を数の理解から説明している。人間が世界を把握してゆく方法には、「構成的アプローチ」と「分析的アプローチ」があり、前者においては、数 ⇒ 無限 ⇒ 空間 ⇒ 物質 ⇒ 運動）といった展開があり、後者では、記号論理 ⇒ 記述の理論 ⇒ 型の理論といった展開が提案されている。

2.7 記号論理（Symbolic Logic）

2.7.1 命題論理（propositional logic）

記号論理におけり命題論理は、表4で示す論理演算を用いて命題を記述する。

表4　命題論理における演算と記号

演算	記号	意味
連言（conjunction）	∧，∩，　&	and
選言（disjunction ）	∨，∪	or
含意（implication）	⊃，⇒	if−then
等値（equivalence）	≡，⇔	equal
否定（negation）	～，¬	not

2.7.2　述語論理（predicate logic）

述語論理は、命題論理に表5に示す量化子（Quantifier）を適用した論理である。

表5　述語論理における量化子

量化子	記号	意味
全称記号（universal quantifier）	∀	all
存在記号（existential quantifier）	∃	some exist

第5世代コンピュータで検討されたプログラム言語であるPrologが用いたモデルである。

2.7.3　記述の理論（Theory of Description）

知識には直知（acquaintance）による知識と記述（description）による知識があり、一般に言葉は（固有名詞を含め）記述である。記述には不定的記述と確定的記述があるがこの理論が対象にするのは確定的記述である。記述の論理の功績は、無意味な命題?の扱いを、命題関数における値の不在として位置づけた点にある。たとえば、「黄金の山は山である」の記述は以下のように示される。

$\exists x(G(x) \wedge M(x) \wedge \forall y \forall z(G(y) \wedge M(y) \wedge G(z) \wedge M(z) \supset y=z) \wedge M(x))$

ただし、　G(x)：xは黄金である
　　M(x)：xは山である

記号論理学の世界は、以上のような命題論理、述語論理、記述の理論などを適用して、自然言語で記述される知識を集合演算に置き換えて表現することを試みた分野である。以前のPrologやLispによる知識表現は、それのコンピュータへの実装の試みであったがその結果は衆知のとおりである。人間の知識は論理的でないし、たとえ論理化できてもコンピュータに使いやすく実装するのは容易なことではない。

2.8 2つの方向性

“The Principles of Mathematics”は、以上のように世界の事象について記号論理（集合論）による体系化を試みたものであるが、検討の途上で有名なパラドックスに遭遇し、それ以上の論理的展開には失敗している。プリンキピア・マセマティカは、パラドックス克服後に試みられた世界記述の再挑戦と言えるが、世界を論理的に定義することに挑戦するラッセルの基本的なアイデアは、むしろ“The Principles of Mathematics”に記述されている。彼が提示した数を基本として展開してゆく型の考え方を構成的な方向と、分析的な方向に大別して展開させる考え方は、データをマルチメディアやシミュレーションに展開してゆく考え方と、メタデータ、メタモデル的なシステム構築を通じて意味的な世界を構築する考え方に対応するように思われる。

2.9 ヴィトゲンシュタインの論考

世界を論理的に定義することを試みた点に関し、ラッセル以上に有名なのはヴィトゲンシュタインであろう。彼の論考[29]は、極めて簡潔に世界を論理的に定義することを試みた。ラッセルは論考を賞賛し、その前書きまで書いている。ところがヴィトゲンシュタインはその後論理分析哲学的な考え方を放棄し、彼自身の著作である論考をも否定するようになる[30]。さらにプリンキピア・マセマティカをも批判するようになりラッセルとも対立するようになった[32]。さらにその後、ゲーテルの不完全性定理[31]などを通じて、論理分析哲学的な考え方は世の中から省みられなくなってしまった。

2.10　チューリングのアイデア

コンピュータの理論的な基盤としては、演算とデータを同一の言語体系で実現可能とすることを証明したチューリングの論文に遡るが、この論文は論理哲学と関係していた可能性がある[32]。この論文は、現在のノイマン型計算機の基礎をを与えるもので、その後、その実現モデルとしてのチューリングマシンが提案される[33]。このようにチューリングは現在のコンピュータの理論的なモデルを構築した。チューリングが、彼のアイデアを実現する上で、重要な経験を積んでいる。彼はヴィトゲンシュタインの講義における学生であり、ヴィトゲンシュタインとの講義中の質疑が後の彼のアイデアを創る上で重要な役割を演じたと言われている[32]。要するにチューリングマシンのアイデアは、ヴィトゲンシュタインとの議論が土台となっている模様なのである。さらに、ゲーデルは彼の不完全性理論を展開する上でチューリングの計算理論を参考にしたとのことである。近代論理学とコンピュータの誕生とはこのように深い関係があった。

3. 記述の理論とシステム展開

3.1型・クラスとメディアとの対応

表3の関係と、従来から使われてきた型やクラスとの対応関係を表6表7に示す。近代論理学、代数学、表音文字、表意文字などがコンピュータの進展に伴い確立してきたプログラム言語における boolean, int, float, char wcharなどの型と対応している様子がわかるであろう。象形文字がオブジェクト指向によるGUIシステムのicon−classに対応させているのは若干のこじつけであり、洞窟壁画や叫び声、ジェスチャの図形画像、マルチメディアに関しては、アプリケーション依存、実装依存の型でオブジェクトのクラスとして定義されているが、今後は幅広い分野で共通に認識される型として標準化されてゆく可能性はある。

このように型は、人間がコミュニケーションの過程で帰納的なプロセスを通じて共通に使える類型的な事物を抽象化（メタ化）して形成されてきた人類の遺産なのである。この経緯から、今後も各種領域の再利用可能な枠組みが着実に構築されてゆくと思われる。

とは言っても、その道は平坦ではない。例えば、基本型、拡張型については、各種プログラム言語で定義されてきたが、必ずしも統一されてはおらず、プログラム相互間におけるデータの相互運用性を損なっていた。CORBAのIDLは、それを克服するためにC, C++, Java, Ada, COBOL, Smalltalk, Comonlispなどの言語の型について、IDL型とのマッピングを定義したが、言語を超越したオブジェクトの相互運用という観点からは画期的な成果であったと言えるであろう[34]。

XMLスキーマが制定されたことにより、XMLコンテンツの世界でも上記プログラミング言語の世界と同様に本格的に型を導入する機会が訪れた。しかしながら、型の構成はプログラミング言語に較べると非常に複雑であり（例えば整数型だけで11種類もある）、初歩的な利用者から見て使いやすいとは言いがたいであろう。

表**6** 各種メディアと型・クラスとの関係

コミュニケーション	コンピュータ	型・クラス
叫び声，ジェスチャ	映像，音声	class
洞窟壁画	図形，画像	class
象形文字	GUI	icon−class
表意文字	漢字処理	wchar
表音文字	英数字, 事務処理	char
代数学	数字，数式計算	int, float
近代論理学	2進論理	boolean

3.2 マルチメディア的展開

構成的アプローチに対するシステム構築はオブジェクト指向的なアプローチにおける、拡張型の概念である。前項で述べたとおり表4の映像、音声、図形、画像、GUIなどにおける各種の拡張データ型は、個別システムのレベルで1990年代から着実に進展してきた。例えば、マルチメディアデータベースと言われたILLUSTRAは、上記の各種メディアに対するデータブレードを保有していた[35]。

以前、我々が開発したSGMLデータブレードは、これらの拡張型をドキュメントの要素として管理するために、設計されたマルチメディア・ドキュメント・データベース用のミドルウエアであった[36]。この設計思想は、マルチメディア・コンテンツのオーサリングと配布に関しては極めて妥当なものであったと思われるが、ILLUSTRAの開発の挫折により、中断せざるを得なかった。しかし、後に紹介するとおり、XMLの進展により、この設計思想に基づく開発を再開する機運が高まりつつあると考えている。

3.3 メタデータ・メタモデルの展開

人間の知的活動における帰納的なプロセスをコンピュータに実装する試みは、1960年代から所謂「人工知能」分野で行われてきたと言えるが、一部の興味ある分野[25][37]を除くとほとんど進展していないのが実情であろう。1980年代の知識工学やエキスパートシステムも、華々しい掛け声と宣伝が行われた割には、その成果たるや微々たるものであった。私自身、その関係者であったのだが、当時を反省すると、今後この種のアプローチは応用指向ではなく、論理学、分析哲学、言語学[38][39]、認知心理学[40]といった基礎的な研究とタイアップした企画が必要であろう。

4. OMGにおけるメタデータ、メタモデルの展開

4.1 OMGのメタデータアーキテクチャ

オブジェクト分析設計的なアプローチにおいては、以前から上記のメタデータやメタモデルと類似のプロセスを行っていた[41]。JavaやC++のようなオブジェクト指向プログラミング言語におけるクラス定義は、上記のプロセスをコンピュータ上で実現していると言えるであろう。UML（Unified Modeling Language）[42]をはじめとするオブジェクト分析設計手法は、以上の帰納に基づく抽象化プロセスを中核に、上流工程を含むシステム構築を誘導支援する方法論である[43]。この場合は、メタな世界の構築を人間が行い、コンピュータはそれを支援する役割に徹すると言えるであろう。

OMGは、以上の観点からメタデータ階層モデルを定義するMOFを制定し、MOFとUMLを系統的に用い、メタデータ、メタモデルのアーキテクチャをビジネスシステムに適用する検討を行っている。表7は、MOF（Meta Object Facilities）に基づく階層モデルである。

表7　OMGのメタデータ階層モデル

メタレベル	MOF用語	事例
M3	メタ・メタモデル	MOFモデル
M2	メタ・メタデータ メタモデル	UMLメタモデル CWMメタモデル
M1	メタデータ モデル	UMLモデル、ウェアハウスメタモデル
M0	データ	モデル化されたシスム、ウェアハウスDB

OMGは、この手法をXMLの世界に導入する試みを展開している。XMI（XML Metadata Interchange）[44][45]やMDA（Model Driven Architecture）[46][47][48]である。

この世界では、モデルとして記述された内容を、XMLのDTDやXMLスキーマ、さらにXMLインスタンスに自動的に変換するパラダイムの提供を試みるものであるが、以下具体的に説明する。

4.2 XMI

XMIは、XML技術を適用して汎用的なオブジェクト交換、変換機能を提供するものである。UML/MOFの規格を適用し、てモデルからXML DTD、さらにXMLインスタンスを 生成する。その概念を図1に示す。

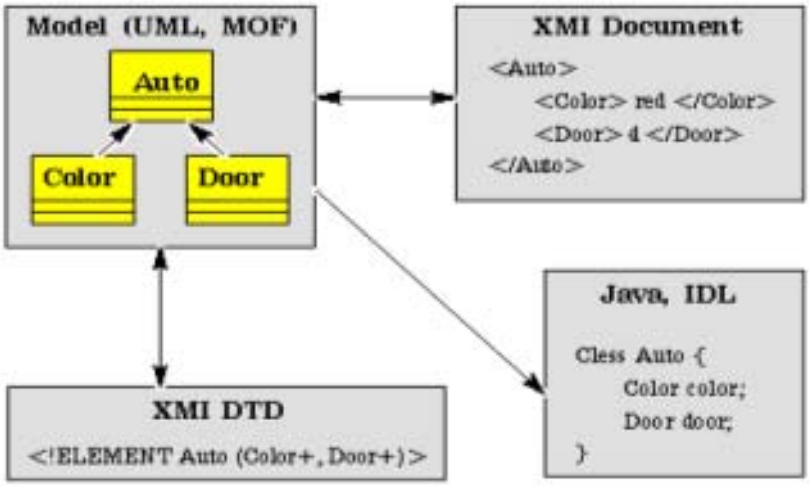


図１　**XMI**によるモデル変換の概念

XMIにより構造化されるメタデータ向けのDTDを生成する規則とインスタンスを生成する規則がXMI仕様の中核を構成する。XMI DTDはXMI文書の構文仕様と、XMI文書を、汎用的なXMLツールを用いて生成し検証する。

4.3 MDA(Model Driven Architecture)への拡張

XMIの機能を、単にツールとしてでなく、ビジネスモデルおレベルに適用しようとする挑戦的な試みがなされている。最近、各種システムへのUMLの導入が図られているので、UMLで記述されるビジネスモデルをXMIを用いてXMLのDTD、インスタンスに変換し、モデルのレベルから一気に実装のレベルのソリューションにまで結びつけてしまおうとするこのである。最近のウェブサービスなどに見られるように、企業のシステム構築は、XMLベースの分散オブジェクト化されつつあり、その上流工程にUMLを適用する動きが顕在化しているので戦略的には、妥当なものであろう。しかし、単純

なオブジェクトから企業システムに至るスXMIのケーラビリティの実現が技術的に実現可能かというと、膨大な課題が横たわっている。特にビジネスモデルは純粋に技術的に扱えるものではなく、情報の確認のためのライフサイクル管理や、ワークフロー管理といった人間や組織が関与する場面が多い。人間による確認や意思決定が関与する部分のモデルから実装への自動変換は不可能であろう。従って人間が介在しない部分のモジュールレベルの変換になると思われる。一方、ユーザインタフェースを含まないビジネスモデルが存在し得るのであろうか？MDAの実現の鍵は、自動変換よりは、人とコンピュータシステムとの的確な切り分けの方法論であり、人間の弱点をコンピュータが支援するモデルの提起であろう。

5. セマンティック・ウェブ

5.1 AIとの比較

セマンティック・ウェブは、UMLのアプローチとは別の観点からのメタ階層へのアプローチである。このアプローチは、XMLをベースとし、その上にメタデータ層としてのRDFとRDFスキーマ、その上にオントロジー層としてのDAML+OILやOWLがあり、意味的な用語の一元的な管理を行う。さらにその上に、ロジック層、プルーフ層が積み重ねられ、最上位にトラスト層が位置している。

セマンティックWebとかつてのAIとの比較が議論になることがある。かつてのAIにおいては、知識ベース構築ツール、推論エンジンなど、GUIベースのシステムが開発され、それを適用した数多くのシステムが構築された。それらの多くは実用に至らなかったが、その理由は技術的な理由よりは、保守運用などのシステムのライフサイクルに関連する要因であった。知識ベースを構築できても、それを維持管理する人間は構築者本人でなければならない場合が殆どであったのが実情である。それほどまでに知識ベースの構築・管理は高度なスキルを必要とするものであった。それ故に、「ナレッジ・エンジニア」なる用語が生まれ、知識ベースの構築・管理の専門家が今後の技術エリートとして注目を集めたこともあった。しかしそれらの期待は実現されなかった。

5.2 ルール推論モデルと決定木探索モデル

私もフレームとルールに基づいたシステムを検討したことはあったが[49]、実用的なシステムとしては、ルールを駆動させるよりはルールに基づいた決定木を生成し、それの探索を行わせる方がはるかに容易であり、そのようなシステムを提案し[50]構築したこともある[51][52]。その基本的な考え方は、当時のAIの基礎となった「プロダクション・ルール・システム」において、ルール競合を避ける場合の探索は、決定木で記述可能であり[50]、そのパラダイムでシステム構築を行う方が、システム構築、運用、維持管理が楽であるということであった。この考えは、先に述べたように、構築しても維持管理ができないと言われた当時のエキスパートシステムに対する一つの考え方であった。

5.3 状態空間の探索による帰納的推論モデル

一方、ルール競合におけるルール選択に帰納的なアプローチを取り入れ、帰納的な推論に挑戦した研究も行われていた[25]。人間の帰納的な推論を状態空間の遷移における多数の競合ルールの選択メカニズムに求め、ルール実行前後の状態と目標状態の距離を計算しつつ失敗をパラメタにフィードバックさせるというアプローチである。これは計算メカニズムが定義され、実装可能なのでそれなりの説得力がある。当時の計算機環境が、計算量に対して貧弱であったことを考慮すると、このアプローチは現状のWEB環境では可能性があると思う。

W3Cのセマンティック・ウェブが、上記のような計算モデルに基づく帰納的な推論を包含することにより、興味あるルールシステムを構築できる可能性はあると思う。その観点に立つとセマンティック・ウェブが受けている過去のAIと変わらない云々の批判は必ずしも的を得ていないと思う。真の困難な課題はそれ以上の階層と人間との接点、すなわち人間との役割の切り分けにあると思われるのである。

5.4 人間的要因

ルール層以上の問題はさておき、人間との接点について考えてみよう。セマンティック・ウェブにおいては、メタデータ、オントロジ、ルールなどがウェブ・コンテンツの付加的な情報として管理されることになるであろう。その場合、そのような付加的なデータを誰がどのように付与するかが問題である。

現在、HTMLでコンテンツを記述している人々がRDFベースのメタデータ、オントロジ、さらにルールなどを記述することはあり得ないであろう。然らば、かつてのナレッジエンジニアのような付加情報を作成する専門家集団が出現するのであろうか。これもありそうもない。

結局、コンテンツ作成者が意識することなく、そのコンテンツに相応しいメタデータ、オントロジ、ルールが自動生成されるようなメカニズムの出現を期待することになると思われる。しかしこれも現状の技術レベルでは容易ではない。コンテンツの作成、運用、廃棄といったライフサイクル的な観点からは、セマンティック・ウェブが離陸する可能性は低いと言わざるを得ないであろう。ただし、これは、一般向けのウェブ・コンテンツの観点からであって、特定の領域、特に軍事や防衛、宇宙開発などと言った分野は例外である。かつてのエキスパートシステムも特定の領域では現在も使われ続けているのである。

5.5 マルチメディア・コンテンツ・ウェブ

セマンティック・ウェブに関しては、以上述べたとおり大きな期待をするのは無理がある。上記のような帰納的推論の試みは存在するが、帰納に関する人類の理論は、プラトン、アリストテレスのギリシャ哲学以来、それほど大きな進歩を遂げているとは思えない。それに対し、２つの方向性のもう一つの方向性であるマルチメディア化を指向する構成的なアプローチの方は、メモリ価格の低下、CPUの性能向上、ネットワークの広帯域化と相俟って着実に実現の方向にある。特に最近は「ブロードバンド」のキーワードに押されて、この方向へのシステム展開が期待されている。従って、今後のウェブの利用に関しては、分析的なアプローチのセマンティック・ウェブよりは構成的アプローチのマルチメディア・コンテンツ・ウェブの方に分があると思われる。

なお、セマンティック・ウェブもマルチメディア・コンテンツ・ウェブもウェブの側面を語っているに過ぎず、実際のウェブは当然両者を含むものである。従って、構成的なアプローチと分析的なアプローチの両者が今後並行して進展することになると思われる。要するに拡張される型とそれに付随する系統的なメタデータ、それらの管理統合メカニズムの一体化である。従ってセマンティック・ウェブは、マルチメディア・コンテンツ・ウェブのメタデータに対してルールを適用して能動的な処理を行うことを意味することになると思われる。

6. まとめ

結局、帰納的なアプローチと演繹的なアプローチの両者を包含しながら今後のWebは進化すると思われる。演繹的なアプローチに関しては、ハードウエアの進歩に従い着実に進展し応用領域を広げてゆくであろう。一方、帰納的なアプローチは、5.3節で述べたような、状態空間における計算モデルを適用するような方法はあるが、人間の思考や知識獲得に比べると足元にも及ばないレベルにある。

コンピュータが処理する論理や計算は、明確なアルゴリズムを実行するだけであり、アルゴリズム自体の創造といったことは困難である。アルゴリズムの創造のためには、そのためのアルゴリズムが存在せなばならず、それをコンピュータに実行させるにはさらにそのアルゴリズムが要求されるというパターンである。現在のコンピュータにそれを期待するのは、困難というよりは無理であろう。

人間の場合は、アルゴリズム創造のためのアルゴリズムを創造しているわけでは無く、視覚、聴覚、触覚、嗅覚などの感覚器官が、外界の刺激に基づいて反応する生物としての生命活動の発展としての認識機能、さらにその繰り返しや学習にともなって獲得された知識である。その中で言葉や言語が大きな役割を果たすが、そのメカニズムは複雑である。視覚や聴覚を通じて感覚される外部世界は、映像や音声の世界であり、その認知機構は基本的には霊長類のような哺乳類と大きな変わりはないであろう。外部世界を分析し分類し、個物を認識し、人間は画像や図形を描けるようになり、個物を共通的に指示するIDとして言葉が生まれ、言葉の関係を記述する言語が生まれた。さらに言葉を記

録し伝達する手段として文字が生まれた。このあたりの経緯は、2.3節で述べた通りである。文字の活用に引き続き、数学や論理学に発展するが、この世界は明確な計算モデルとアルゴリズムの世界であり、コンピュータが得意とする分野である。

一方、人間の認知においては、文字よりは図形や画像、さらに動的な映像や音声といったメディアの方が好まれる。人間が言語・文字を通じて得る知識は、記述による知識と呼ばれる。それに対し視覚や聴覚から得られる知識は直示による知識と呼ばれ、脳内での記憶プロセスも異なる。脳内の記憶プロセスは、質感とも言うべき認知プロセスが関与し、この質感を得る過程は脳内のニューロンが関与している。このように質感を得る過程をクオリアが成立すると呼ぶ。

感覚から得られる知識には感覚的クオリアが関与し、記述から得られる知識には志向的クオリアが関与する。このように、人間の知識獲得プロセスには脳科学的なアプローチとの対応付けが検討されつつある。いずれにせよ、映像・音声、図形・画像といった感覚的・具体的なプロセスによる認知のほうが、文字や言語といった記述的・抽象的な情報による認知よりは容易である好まれる

他方、感覚では処理できない抽象的な概念に基づく思考は、感覚的クオリアが関与する余地はなく、志向的クオリアによる認識に依存せざるを得ない。これらは最終的には言語により達成される。

このような状況を考えると、人間の知識は、文字・言語というメディアを中心に、感覚を通じて認知し易い情報（映像・音声、図形・画像）とアルゴリズムで計算処理可能な厳密に定義可能な情報（論理学、数学）に階層化されることが分かる。

この階層をWebに適用すると、5.5節のマルチメディア・コンテンツWebが前者に、セマンティックWebが後者に対応することになる。XMLは、その中間の文字・言語Webということになるであろう。なお、この階層は、表3で述べた階層にも対応することは明白である。

結局、セマンティックWebが目指していることは、論理メディア、数学メディアを文字・言語メディアを代表するXMLで処理しようとする試みである。その観点からは、セマンティックWebを計算処理的な観点のアーキテクチャで採り上げようとするアプローチは正当であると言えるであろう。ただし、それを具体的に図示しコンセンサスを獲得するのは容易ではない。

文　献

[1]http://www.wul.waseda.ac.jp/PUBS/fumi/66/66−08.html

[2]http://dublincore.org/

[3]http://www.gsi.go.jp/REPORT/GIS−ISO/ISOGIS/summary/19115.html

[4]http://www.w3.org/TR/1999/REC−rdf−syntax−19990222/

Bryan L. Bell; “Managing the Intellectual Property Lifecycle”, Proc XML Europe’99 (1999)

[5]リチャード・リーキー,（岩本光雄訳,今西錦司監修）; “ 人類の起源 ” , 講談社, (1985)

[6]スティーブン・ピンカー,（椋田直子訳）; “ 言語を生み出す本能（上）” , NHKブックス#740, (1995)

[7]スティーブン・ピンカー,（椋田直子訳）; “ 言語を生み出す本能（下）” , NHKブックス#741, (1995)

[8]ジョルジュ・シャン,（矢島文夫監修）; “ 文字の歴史 ” , 創元社, (1990)

[9]ジョルジュ・シャン,（矢島文夫監修）; “ 記号の歴史 ” , 創元社, (1994)

[10]ドゥニ・ゲージ,（藤原正彦監修）; “ 数の歴史 ” ,創元社, (1998)

[11]室井; “バビロニアの数学”, 東大出版会 (2000)

[12]フィリップ・アーダ;”ヒエログリフを書こう！”, 翔泳社,（2000）

[13]マーク・コリア,ビル・マンリー;”ヒエログリフ解読法−古代エジプトの文字を読んでみよう”, Newton Press,（2000）

[14]高崎; “古代エジプトの数学”, 総合科学出版 (1977)

[15]W.S.アングラン、J.ランベグ（三宅訳）; “タレスの遺産−数学史と数学の基礎から”, シュプリンガー・フェアラーク東京, (1997)

[16]J.C.Cleaveland,（小林訳）; “ データ型序説 ” , 共立出版（1990）

[17]A・N・ホワイトヘッド, B・ラッセル（岡本等訳）; “ プリンキピアマテマティカ序論 ” , 哲学書房（1988）

[18]C.R.モレンホフ（最相等訳）; “ ENIAC神話の崩れた日 ” . 工業調査会（1994）

[19]ハワード・ラインゴールド（栗田・青木訳）; “ 思考のための道具 ” , パーソナルメディア(1987)

[20]L. Tesler; “ The Smalltalk Environment ” , Byte, Vol.6, No.8, (1981)

[21]D.Robson; “ Object−Oriented Software Systems ” , Byte, Vol.6, No.8, (1981)

[22]A.Goldberg, D.Robson; “ Smalltalk−80: The Language and Its Implementation ” , Addison Wesley, (1983)

[23]大野 ; “ テクニカルコミュニケーションと情報メディア ” , 情報処理学会第一回テクニカルコミュニケーションシンポジウム講演資料, （1993）

[24]P. Wegner; “ The Object−Oriented Classification Paradigm”, Research Directions in Object−Oriented Programming edited by Bruce Shriver & Peter Wegner, MIT Press, (1987)

[25]J.H.ホランドほか（市川ほか訳）; “INDUCTION − 推論・学習・発見の総合理論へ向けて”, 新曜社 81991)

[26]B.ラッセル（市井訳）; “ 西洋哲学史1 ” , みすず書房（1955）

[27] B.ラッセル（日高訳）; “ラッセル自叙伝−I”, 理想社, P.192, (1969)

[28] B. Russel; “The Principles of Mathematics(2nd Ed.)”, Rpotledge, (1937)

[29]ヴィトゲンシュタイン; “論理哲学論”, 中央公論世界の名著58−ラッセル・ヴィトゲンシュタイン・ホワイトヘッド編, 中央公論（1971）

[30]橋爪; “言語ゲームと社会理論 − ヴィトゲンシュタイン・ハート・ルーマン”, 勁草書房,

[31]廣瀬、横田; “ゲーデルの世界”, 海鳴社（1854）

[32]J.ライバー（今井訳）; “ 認知科学への招待：チューリングとウィトゲンシュタインを道しるべに ” , 新曜社（1994）

[33]H.H.ゴールドスタイン（末包等訳）; “ 計算機の歴史：パスカルからノイマンまで ” , 共立出版（1979）

[34]河込,中村,大野,飯島; “ 分散オブジェクトコンピューティング ” , 共立出版（1999）

[35]伊藤、太田、木下、仲山; “Datablade構築技法”, BNN, (1996)

[36]大野, 佐藤 ; ”ORDBによるマルチメディア・ドキュメントの管理”, 情報処理学会デジタルドキュメント研究会研究報告, DD7−5（1997.5）

[37]R.C.シャンク（渕、石崎訳）; “考えるコンピュータ”, ダイヤモンド社, (1985)

[38]吉田正幸 ; “ 分類学からの出発：プラトンからコンピュータへ ” , 中公新書#1148, 中央公論社（1993）

[39]中尾佐助 ; “ 分類の発想：思考のルールをつくる ” , 朝日選書#409, 朝日新聞社（1990）

[40]D.A.ノーマン（富田訳）; “ 認知心理学入門−学習と記憶 ” , 誠信書房(1984)

[41]J.マーチン等（三菱CC研究会訳）; “ オブジェクト指向方法論 ” , プレンティスホール・トッパン（1995）

[42]http://www.omg.org/technology/uml/index.htm

[43]オブジェクト指向研究会（今野監修）; “ オブジェクトモデリング表記法ガイド−UML 1.3 ” , ピアソン, (2000)

[44]OMG Doc. formal/00−11−02: XML Metadata Interchange (XMI) version 1.1

[45]大野; “ XMIによるメタデータの相互運用 ”, 電子情報通信学会「XMLとメタデータ記述」チュートリアル資料集, pp.18−58, (2001.4)

[46]http://www.omg.org/mda/index.htm

[47]和田周、“ モデル駆動型アーキテクチャ（MDA）に向けて ” オブジェクト・レポートVol.6,No6、創研プランニング（2001.3）

[48]Richard Soley; “ Model Driven Architecture ” , ftp://ftp.omg.org/pub/docs/omg/00−11−05.pdf

[49]K.Ohno; “An Expert System for Predicting the Relay Contact Arc Erosion”, Proc. International Conference on Electrical Contacts and Their Applications, Nagoya, Japan, (1986)

[50]大野; “決定木によるアプリケーションプログラム構築支援システム”, 信学技報, COMP90−84/SS90−30, (1990)

[51]K.Ohno; “Modeling Contact Erosion Using Object−Oriented Technology”, Trans IECE Japan Electron, Vol.E77 − C, No.10, (1994)

[52]大野; “公衆電話機保守支援システムの改良”, NTT技術ジャーナル, Vol.7, No.12, (1995)

セマンティックWebの課題と携帯電話から見た可能性

ドコモ・システムズ　大野邦夫

1. はじめに

セマンティックWebに関しては、時代のキーワードとして多様な議論が交わされているが、その多くは基礎研究段階で、ビジネスや産業での具体的・実用的な段階には至っていない。セマンティックWebを基礎研究として多様な視点から可能性を研究するのは結構なことであるが、それを実用時期やサービス提供時期が明確にされている具体的な開発に結び付けて議論することには注意する必要がある。ところでセマンティックWeb自体は明確に定義されているわけではない。未来のWebに期待される知的な処理といった概念から、メタデータ・オントロジを適用する技術、それを具体的に実装する言語など、様々な切り口が存在する。セマンティックWebを明確定義することは難しいかもしれないが、議論をするためにはその概念についてコンセンサスを得ると言うプロセスは必要であろう。

ここでは、セマンティックWebに対する個人的な考え方の紹介と、それに基づく可能性のある応用分野として、携帯電話を中心とするモバイル分野について紹介する。

2. セマンティックWebの概要とこれまでの経緯

2.1 想定されるアーキテクチャ

当初、セマンティックWebは、W3Cのディレクターであるティム・バーナーズリー氏が主導する今後のWebがその進化の過程で具備すべき機能への考察と提案であった。その概要は、2001年の春にサイエンティフィックアメリカ[1]で紹介され話題になった。この記事は翻訳され、日経サイエンス2001年8月号で「自分で推論する未来型Web」というタイトルで紹介されている。その記事によると、未来のWebは、個人を支援するネットワーク上のエージェントとして紹介されている。

セマンティックWebを実現するためのアーキテクチャは図１のように想定されている[2]。図から分かるとおり、セマンティックWebは7階層のレイヤーから構成されている。最下位の第一層は、UnicodeとURIで構成される最も基本的な層である。第2層がXML、ネームスペース、XML Schemaで構成されるXML層である。この層で、DTDやRELAXではなく、XMLSchemaを挙げているのはW3Cであるから当然であろうが、XMLSchemaが今後、安定して使用されるか否かについては議論のあるところである。第3層は、RDFとRDFSchemaで構成されるメタデータ層で、自己記述された第2層のXMLによる文書を対象とする。

第4層は、オントロジ層で、種々のカテゴリにおける用語群が定義されている。この層は第3層におけるメタデータ群（RDF）とその枠組み（RDFSchema）を対象に用意されている。第5層はロジック層で、第4層の用語間の関係やカテゴリのモデルを規定するためのルールが定義されている統合論理基盤である。第6層はプルーフ層で、第5層のロジックの正当性を管理する層である。第7層はトラスト層で利用者の意図や信念に関係する。第3層から第6層に関しては、関連する文書やデータが付随するため、それらの信頼性を保障する必要がある。そのためにこれらの層ではデジタル署名機能が要求される。図1の構成と機能はこのようなものであろう。セマンティックWebが大々的に公表されたのは、２０００年の１２月にワシントンDCで開催されたXML2000におけるティム・バーナーズリーの特別講演によるものであった。それによると、Webは、単に参照するだけの静的なコンテンツからナビゲートを支援する動的（自己記述的）なメディアに移行し、Webの意味情報の共有が新しいコミュニ

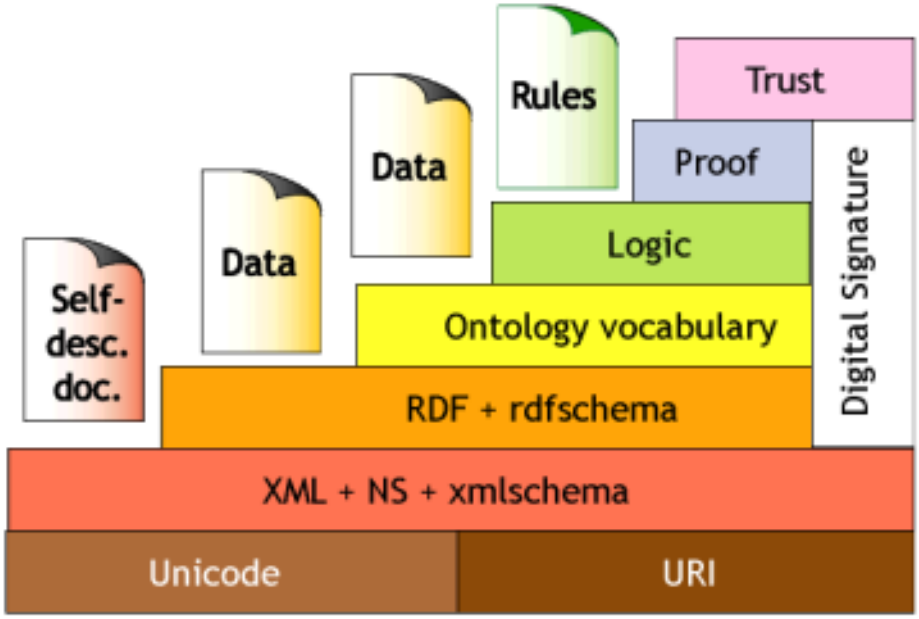


図1　セマンティックWebのアーキテクチャ階層

ケーションを生み出すと言う。そのためには、人間による自然言語の介在なしに、計算機による意味処理の実現が課題となる。

2.2 RDF層とオントロジ層

2.2.1　RDF

図1における、RDF層とオントロジ層に関して若干の説明を追加しよう。RDFはXML1.0が勧告化された1年後の1999年2月に正式な勧告となっているが、必ずしも普及しているとは言えない。RDFは、主題（Subject）、述語（Predicate）、対象（Object）からなる言明（ステートメント）と考えることができる。例えば、「源氏物語の著者は紫式部である」という言明で、主題＝「源氏物語」、述語＝「著者である」、対象＝「紫式部」、ということになる。これを具体的に記述すると以下のようになる

```
<rdf:Description about=’源氏物語’>
  <Author>紫式部</Author>
</rdf:Description>
```

RDFの実装において、主題部分はWeb上に置かれるので、通常は特定の名前空間で定義されるURIが対応し、そのプロパティ（述語）と値（対象）の3つ組で言明が記述されることになる。主題に対するプロパティと値の集合が、その主題に対するメタデータである。

2.2.2　オントロジ

オントロジはRDFの言明群によって記述される用語定義として実装することが可能である。例えば「光源氏の父親は桐壺帝である」という言明は以下のように記述される。

```
<rdf:Description about=’光源氏’>
  <Father>桐壺帝</Father>
</rdf:Description>
```

同様に、「夕霧の父親は光源氏である」という言明は以下のように記述される。

```
<rdf:Description about=’夕霧’>
  <Father>光源氏</Father>
</rdf:Description>
```

人間の常識を用いると以上の２つの言明から「夕霧の祖父は桐壺帝である」という言明が導かれる[3]。

```
<rdf:Description about=’夕霧’>
  <GrandFather>桐壺帝</GrandFather>
</rdf:Description>
```

これは、「父親の父親は祖父である」という知識があり、祖父という概念が存在して初めて可能となる。Webがこのような知識を持つためにオントロジが定義される必要がある。ここでは以上の例で用いた「家族」というカテゴリでオントロジを考えてみよう。このオントロジは、人間の性別と親子関係のみから導くことが可能である。男性の親は「父親」であり女性の「親」は母親である。共通の親を持つ人同士は「兄弟」、「姉妹」であり、親の兄弟は「叔父」であり、姉妹は「叔母」である。親の男親は祖父であり、女親は祖母である。.....

2.2.3　必要最小語彙

このように性別と親子という単純な関係から家族に関係する概念を網羅する膨大な用語が定義される。家族オントロジにおける性別と親子のような他から定義できない基本的な用語を必要最小語彙（minimum vocabulary）と呼ぶ[4]。新たな用語の定義にあたり、先のオントロジ層の説明で述べたとおり、命題論理と量化子（述語論理）が用いられ、未定義の用語が既知の用語の組み合わせに変換されて処理されることになる。オントロジとはまさにこのような手法による特定の世界の網羅的な用語記述にほかならない。

オントロジは、RDFと述語論理を使って定義することが可能であるが、RDFによる命題定義を継承関係を用いて系統的に行う枠組みをRDFスキーマが提供してくれる。そのため、RDFの世界でオントロジを定義するにはRDFスキーマを用いるのが効率的である。最近、RDFスキーマをさらにオントロジ定義の観点から具体的に特化したDAML（DARPA Agent Markup Language）＋OIL（Ontology Inference Layer）が注目されている。

2.2.4　オントロジのビジネス適用への挑戦

DAMLは文字通りエージェント技術の世界からの産物である。エージェントの世界ではエージェント間通信の意味的な世界の定義でオントロジが検討されてきた。エージェント通信言語としてはKQML（Knowledge Query Manipulation Language）[5]やFIPA（Foudation for Intelligent Physical Agents）[6]のACL（Agent Communication Language）[7]が主流の言語として検討されてきたが、どちらかと言うと学術的な領域に閉ざされてきたと言える。RDFとRDFスキーマの世界でオントロジを定義するセマンティクWebの世界は、オントロジ関連技術をアカデミックな世界からビジネスの世界に拡大する挑戦的な試みである。

2.3 DAML+OIL

2.3.1　DAML−ONT

DAMLプロジェクトは2000年の8月に米国の研究者を中心に開始された[8]。その後、DAML － ONTという言語仕様が作成され[9]、その開発と普及の努力が開始された。DAML － ONTは、クラス、特性といったRDFやRDFスキーマの用語を用い、種々の関係や集合演算などに関する用語を系統的に定義したものである。類似の概念としては、かつてのAIにおける知識ベース、特にフレームをベースとした知識表現が挙げられるであろう。具体的な処理を行うには、DAML － ONTの処理系を用い、そこで使えるプログラム言語で記述することが必要になる。

2.3.2 OIL

OIL（Ontology Inference Layer）は、主に欧州の研究者を中心に開発された記述論理（Description Logic）に基づく歴史的背景を持つ言語である。記述論理は、知識表現の一分野として位置付けられ、古典的なフレームやオブジェクト指向表現、セマンティック・データモデル、型システムなどに関係する。このような古典的なフレームベースの知識をRDFスキーマの枠組みを使ってWeb上に再構築した言語がOILである[10]。

2.3.3 DML+OILへの進展

その後、上記の米欧のオントロジ研究グループは、連合組織を（The Joint United States / EuropeanUnion ad hoc Agent Markup Language Committee）を形成した[11]。そのメンバーの検討で、ともにRDFスキーマをベースとするオントロジ言語であるDAML － ONTとOILは、統合されてDAML+OILとなった。最初のバージョンが2000年12月に仕様化されたが[12]、その後2001年3月に最新版である現行のバージョン[13]に改められた。

DAML+OILを用いて、個人としての人間（Person）のクラスを定義する例を図2に示す。上記

```
<daml:Class rdf:ID= " Person " >
  <rdfs:subClassOf rdf:resource= " #Animal " />
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource= " #hasParent " />
      <daml:toClass rdf:resource= " #Person " />
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality= " 1 " >
      <daml:onProperty rdf:resource= " #hasFather " />
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource= " #shoesize " />
      <daml:minCardinality>1</daml:minCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

図2 DAML+OILによるPersonクラスの記述例

DAML+OILのクラスとしてのPersonは、Animalのサブクラスで、両親を持ち、単一の父親を持ち、靴のサイズが定められていると記述される。この程度の記述例では、DAML+OILの真価は不明であ

る。このような記述は従来のAIの世界で、LispやPrologを用いて定義していた知識ベースと同等である。実際のアプリケーションが出てくれば、具体的に分かってくるのであろう。DAML+OILをWebサービスのカテゴリへの適用を狙った例として、DAML － Sという事例があるが、これは、Webサービスにおけるサービスをプロセスの集合として定義し、複合的なプロセスを単純なプロセスに変換して、最終的に問題を解決しようとするものである。言ってみれば、サービスをプロセスの要素還元で捉える思想であるが、現在のバージョンが0.6であることを考えれば分かるとおり、未完成であり評価ができるレベルに達してはいない。セマンティックWebの真価を問える時期は、このあたりの技術の進展を見てからであろう。

3. セマンティックWebを取り巻く環境

3.1 かつてのAIとの比較

3.1.1 かつてのAIとも対応付け

以上のように、RDFやRDFスキーマ、DAML+OILを用いて、かつてのAI分野における知識表現と同様な定義が可能となることが分かるであろう。しかし、この手法は、かつてのAIと同様なものであろうか、それとも異なるものであろうか。その議論はセマンティックWebに関心を寄せる人にとっては基本的な問いであろう。しかし、かつてのAI自体、必ずしも明確に定義されていたわけではない。

セマンティックWebの関係者からしばしば聞かされるのは、「セマンティックWebはかつてのAIとは異なる」という台詞である。DAML － ONTやDAML+OILの仕様を見る限りは、特に画期的な技術が実装されているわけではない。かつてのAIは、ミンスキーのフレームによる知識表現をベースに、Lispによるルールシステムや、PrologといったAI言語を用い、フレームのスロットの名称やファセットの値に関連する命題を定義し、述語論理で演繹したり、仮説検証を行うものであった。

DAML+OILは、かつてのAIにおけるフレームをRDF、RDFスキーマの枠組みで実現したようなものである。かつてのフレーム自体、Lispの属性リスト（Property List）の拡張と考えられていたので、その観点からすると上記の視点は妥当なものと言えるであろう。そうなると、DAML+OILレベルのセマンティックWebは、かつてのAIとほとんど同じレベルということになるであろう。

3.1.2 異機種システムとの相互運用

しかし、かつてのAIの大きな課題として、レガシーシステムのような異機種システムとの相互運用の課題があった。これは特にLispやPrologの専用マシンとしてエキスパートシステムを実装したような場合には大きな問題であり、その解決に種々の工夫を払ったものである。その対策として、専用マシンやLisp、Prologのような言語を止めて、C言語でエキスパートシステムツールを構築したりしたのであるが、限られた分野で使われたに過ぎなかった。

Webをベースにしたセマンティックwebは、その相互運用という点に関してはかつてのAIに比べると格段に進歩している。RDF、DRFスキーマ、DAML+OILといった枠組みで、プログラム言語から知識表現を独立させ、アプリケーションの相互運用と同時に、知識ベースの相互運用をも可能にした点である（とは言っても、これはセマンティックWebの功績ではなく、XMLの功績である）。

このようなアーキテクチャをかつてのAIで行おうとしたら大変なことであったことを思うと、セマンティックWebはかつてのAIとは異なると言って良いであろう。

3.2 モジュール化の視点

以上の議論から分かるとおり。従来のAIの欠点の一つは相互運用性の不十分さ、ようするにモジュール化の問題であった。そこでモジュール化の視点からセマンティックWebを考えてみたい。

3.2.1　モジュール化の歴史

コンピュータソフトウエアは、最初の蓄積プログラム方式の計算機であったEDSAC以来、モジュール化を目指してきた。サブルーチンという概念は、1949年に作成されたEDSACで既に使われていた。1950年代に使われた科学技術計算用のFORTRAN、1960年代の事務処理用のCOBL、ALGOLなどでは、この汎用的な手続き呼び出しを高級言語で行うことを可能にした。1970年代の構造化プログラミングは、複雑膨大化したプログラムをモジュールで管理するためにソフトウエアモジュール間の関係を単純で分かりやすく構成するための技術であった。

1980年代のオブジェクト指向技術は、モジュールをオブジェクトとしてカプセル化することにより管理しやすくすると共に、モジュール生成の容易化を目指してクラスというテンプレートを定義するとともに、類似のテンプレートを作りやすくするために継承というメカニズムを用意した。1990年代の分散オブジェクトは、オブジェクト指向技術をネットワークに適用し、遠隔のモジュールをローカルなモジュールのように呼び出せるクライアント・サーバ方式をプログラミングパラダイムに導入した。2000年代はWebプログラミングの時代である。Webを効率的に管理するためにXMLによるデータ･文書のモジュール化を行う必要がある。そのために、当初はDTDというテンプレートを用意したが、DTDがXML構文ではないことから最近はXMLスキーマやRELAXNGに変わりつつある。

3.2.2　XMLにおけるモジュール化の枠組み

XMLスキーマにおける型の導入は両刃の剣である。型は基本的に振る舞いから制約を与える。従って動的なメディアに対しては有効であろう。動的なメディアはプログラミング言語と融合しており、プログラミング言語において型は有効な道具だからである。一方、静的な文書において、型は不要である。静的な文書は、表示・印刷されれば良いのだから、数字のデータも文字型でかまわない。integerとかfloatは余計なお世話である。XMLの使われ方は圧倒的に整形式（well−formed）が多い。文字だけのDTDですら使われてこなかったのである。それがさらにデータ型を意識したXMLスキーマなど、処理を伴わない一般の文書を扱いたい人間から見ると余計なお世話も良いところなのである。従って、従来の文書という観点からテンプレートの良さを評価する限りにおいて、不要なものが無いだけRELAXNGの方がXMLスキーマよりも自然で素直である。

3.2.3　知的システムにおけるモジュール化

以上モジュール化の観点からプログラムと文書を概観したのであるが、以上に基づきセマンティックWebにおけるモジュール化を考えてみたい。知的という観点でWeb上のコンテンツをモジュール化しようとしていると考えることが可能であろう。セマンティックWebにおける知的とは、取りも直さず、推論、メタデータ・オントロジというキーワードになるであろうがここには大きな論理的な飛躍がある。現状の推論は、せいぜい記号論理学的な述語論理レベルである。

述語論理レベルの機能をモジュールとして導入する試みはすでに存在している。PROLOGは言語仕様に述語論理を導入した。しかしこれは完全に失敗した。その理由は、アプリケーション自体をPROLOGで書かねばならず、機能・性能が制約されると同時に、既存のアプリケーションを統合できないからである。エキスパートシェルと呼ばれた知識ベースシステム開発環境は、PROLOGよりは多少ましなシステムであった。言語としては、当初はLISPで書かれたが、後にCやC++のような手続き

型言語やそのオブジェクト指向拡張言語でも実装されるようになった。この場合には、知識ベースと推論エンジンがモジュールであった。

この手法によるエキスパートシステムや知識ベースシステムは特定の領域には使われた。C言語のような既存言語のモジュールだったので、既存のアプリケーションとの相互運用も可能であった。しかしながら、最近はそのような使われ方が宣伝されることも無くなったのであまり使用されてはいないと思われる。そのあたりの理由は、最新のプログラミング環境との相性の問題が挙げられるであろうが、知識ベースの維持管理の問題も当然のこととして考えられる。

セマンティックWebの場合も、もし実現され得るならば、基本的にはエキスパートシェルのような考え方をとるのが自然であろう。その場合、知識ベースに相当するのがオントロジ層であり、推論エンジンに相当するのがルール層ということになるであろう。

3.3 知識の構築よりは維持管理が問題

3.3.1 AIビジネスからの教訓

セマンティックWebとかつてのAIとの比較はさておき、かつてのAIからの教訓を生かすことは重要であると思われる。かつてのAIはビジネス的には確かに失敗であった。しかし技術的にも失敗だったのであろうか。また、たとえそうであってもこの失敗を通じた教訓もあったはずでありそれをセマンティックWebに生かすことは可能なはずである。

かつてのAIブームの発端は20年近く昔の話になる。知識ベース構築ツール、推論エンジンなど、GUIベースのシステムが開発され、それを適用した数多くのシステムが構築された。それらの多くは実用に至らなかった。その理由の一つは先に述べた相互運用の問題であるが、さらに大きな理由は、運用、保守などのシステムのライフサイクルに関連する要因であった。知識ベースを構築できても、それを維持管理する人間は構築者本人でなければならない場合が殆どであったのが実情である。それほどまでに知識ベースの構築・管理は高度なスキルを必要とするものであった。それ故に、「ナレッジ・エンジニア」なる用語が生まれ、知識ベースの構築・管理の専門家が今後の技術エリートとして注目を集めたこともあった。しかしそれらの期待は全て幻となって消え去った。

私もフレームとルールに基づいたシステムを検討したことはあったが、実用的なシステムとしては、ルールを駆動させるよりはルールに基づいた決定木を生成し、それの探索を行わせる方がはるかに容易であり、そのようなシステムを提案し構築したこともある。その基本的な考え方は、当時のAIの基礎となった「プロダクション・ルール・システム」において、ルール競合を避ける場合の探索は、決定木で記述可能であり、そのパラダイムでシステム構築を行う方が、システム構築、運用、維持管理が楽であるということであった。この考えは、先に述べたように、構築しても維持管理ができないと言われた当時のエキスパートシステムに対する一つの解決法であった。

3.3.2 メタデータ、オントロジ、ルールの付与と管理

セマンティックWebにおいては、メタデータ、オントロジ、ルールなどがWeb・コンテンツの付加的な情報として管理されることになるであろう。その場合、そのような付加的なデータを誰がどのように付与するかが問題である。現在、HTMLでコンテンツを記述している人々がRDFベースのメタデータ、オントロジ、さらにルールなどを記述することはあり得ないであろう。然らば、かつてのナレッジエンジニアのような付加情報を作成する専門家集団が出現するのであろうか。これもありそうもない。

結局、コンテンツ作成者が意識することなく、そのコンテンツに相応しいメタデータ、オントロジ、ルールを自動生成するようなシステムの出現を期待することになると思われる。しかしこれも現状の技術レベルでは容易ではなさそうである。コンテンツの作成、運用、廃棄といったライフサイクル的な観点からは、セマンティックWebが離陸する可能性は低いと言わざるを得ないであろう。ただし、これは、一般向けのWeb・コンテンツの観点からであって、特定の領域、特に軍事や防衛、宇宙開発などと言った分野は例外である。かつてのエキスパートシステムも特定の領域では現在も使われ続けているのである。

3.4 記号論理学からの課題

3.4.1 記号論理の適用範囲

世界を記号論理の世界で定義する試みは、例えばヴィトゲンシュタインが”論考”[14]で試みたりしているが、そのためには基本的な語彙を必要とする。そのような語彙を必要最小語彙（minimum vocabulary）と呼ぶ。これは、ある種の論理体系における公理系のようなものと考えれば良い。他の語彙は、必要最小語彙から命題論理、術語論理などを用いて演繹的に定義される。

オントロジは、ある種の領域における語彙集である。この場合も当然のことながら必要最小語彙を必要としそれが問題になる。DAML+OILやOWLでは、それらがXML、XMLスキーマ、RDF、RDFスキーマなどの用語である。ただしそれらは単なる構文的な定義にすぎないので、意味的には別のアプローチを必要とする。そのような意味では各種分野のメタデータが最小必要語彙として期待されていると考えられる。

セマンティックWeb関連の論文では、しばしば論理記号で記述されるモデルが議論の対象となっており、それらの基礎として記述論理（DL: Description Logic）[15]について言及されている。記号論理学は論理哲学や分析哲学の基本的な道具であり、フレゲー、ラッセル以来、一世紀の歴史があるが、科学技術の世界との相性が良いとは言えない。一昔前にPrologで述語論理（命題論理と量化子）をコンピュータに実装しようとしたが実用システムには結びつかなかった。このような経緯を乗り越えるような展開をセマンティックWebでは期待したいところであるがその可能性は大きいとは言えない。

ラッセル、ヴィトゲンシュタイン、チューリング、ゲーデルらが展開した論理分析哲学に基づく意味論的な世界は、記号論理をベースとする厳密な定義による世界の定義の限界を提示した。ラッセルは彼の晩年の著書「人間の知識」[4]で人間の知識は不正確で不完全であることを述べ、ゲーデルは、不完全性定理で、前提を明確に定義するほど、結論で導かれる内容は狭まらざるを得ないことを指摘した[16]。このロジックを乗り越える展開がなければ、記号論理の世界の成果をセマンティックWebの世界にダイレクトに適用することは困難であろう。

3.4.2 記号論理と自然言語との距離

結局、セマンティックWeb実現に関する基本的な問題は、集合論に基づく記号論理的な枠組みと自然言語との距離であり、それをコンピュータ的な世界で解釈すると、プログラム言語と自然言語とのギャップということになるであろう。メタデータやオントロジという用語は、記号論理的な世界が、自然言語の世界に向かって精一杯背伸びして到達したマイルストーンであろうが、それでも人間が理解し、思考し、記述し、語る自然言語の世界からは程遠い存在と言わざるを得ないであろう。そうなると、冒頭で紹介したXML2000におけるティム・バーナーズリーの特別講演で述べられたセマン

ティックWebの狙い自体が無謀であったということになる。人間の知性というものは、既存の人間の知識ではうかがい知ることができない深遠なものなのであろう。

最近は、この分野に対して脳科学的な分野からもアプローチが試みられている[17]。人間の脳は約1000億の神経細胞（ニューロン）から構成され、認知や思考はその活動の結果であるとされている。脳科学の世界では、人間の認知活動や思考活動において「クオリア（qualia、単数形はquale）」という心的要素を取り上げる。この用語は「質感」などと訳されているが、要するに意味を構成する基本的な要素と考えれば良いであろう。このクオリアとニューロンとの対応関係や脳におけるマップが明確化されつつある。クオリアは2種類に大別される。感覚的クオリア（sensory　qualia）と志向的クオリア（intentional　qualia）である。前者が、視覚、聴覚、触覚のような人間の感覚に根ざした質感であるのに対し、後者は言語的、社会的な文脈の下に置かれた質感である。

人間の言語は、異なる感覚を統合する志向性の作用の頂点に存在すると考えられる。要するに感覚的クオリアと志向的クオリアのシステマティックな協調作業における氷山の一角である。言語の意味とは、言葉が何かを指し示すことであり、種々の感覚を網羅した志向性のネットワークのもたらすものである。その結果、言葉は普遍的な力を持つ。言葉の意味が文脈に依存するのも、文脈自体が志向性のネットワークの作用で生成されるためである。

自然言語を以上のような観点で捉えると、集合論に基づく記号論理の世界との乖離を実感せずにはいられない。しかし、DAML+OILのようなアプローチに代表される「人間による自然言語の介在なしに、計算機による意味処理」としてのセマンティックWebは、これを乗り越えなければ実現は不可能であろう。従って近未来におけるビジネスへの適用や事業化といった観点からセマンティックWebを期待するのは無理があるが、逆の観点から考えると、膨大な基礎研究領域が広がっているということに他ならない。

4. 携帯電話から見たWebの可能性

4.1 個人を支援するエージェントとしての視点

セマンティックWebに関して、基礎研究はともかく、実用化や事業化といった観点からは見通しが暗いのであるが、それでもWebは進化し続けるであろう。そのような観点で捉えると、サイエンティフィックアメリカで紹介された「個人を支援するネットワーク上のエージェント」としてのセマンティックWebの方が「人間による自然言語の介在なしに、計算機による意味処理」よりは実現の可能性がある。そのようなシナリオを携帯電話の側から考察してみよう。

4.2 携帯電話上の電子秘書

4.2.1　携帯電話上のXML処理系

「個人を支援するネットワーク上のエージェント」を携帯電話利用者側から見ると、それは携帯電話上の電子秘書の実現ということになる。現在のi−アプリの世界は、10KBとか30KBのメモリ空間におけるJavaによる処理系に過ぎないが、これが100KBとかMBの世界に移行するのは時間の問題である。このような処理空間にどのようなアプリケーションを埋め込むかが今後の携帯電話の課題であろう。

それは、現在の携帯電話が保有している機能の自然な拡張や高機能化が基本であろう。その方向性の一例として「モバイル・インターネット」というキーワードを使って、今後のユビキタス・ネット

ワークの個人指向のノードについて検討したことがある[18]。そのアーキテクチャの概要を図3に示す。

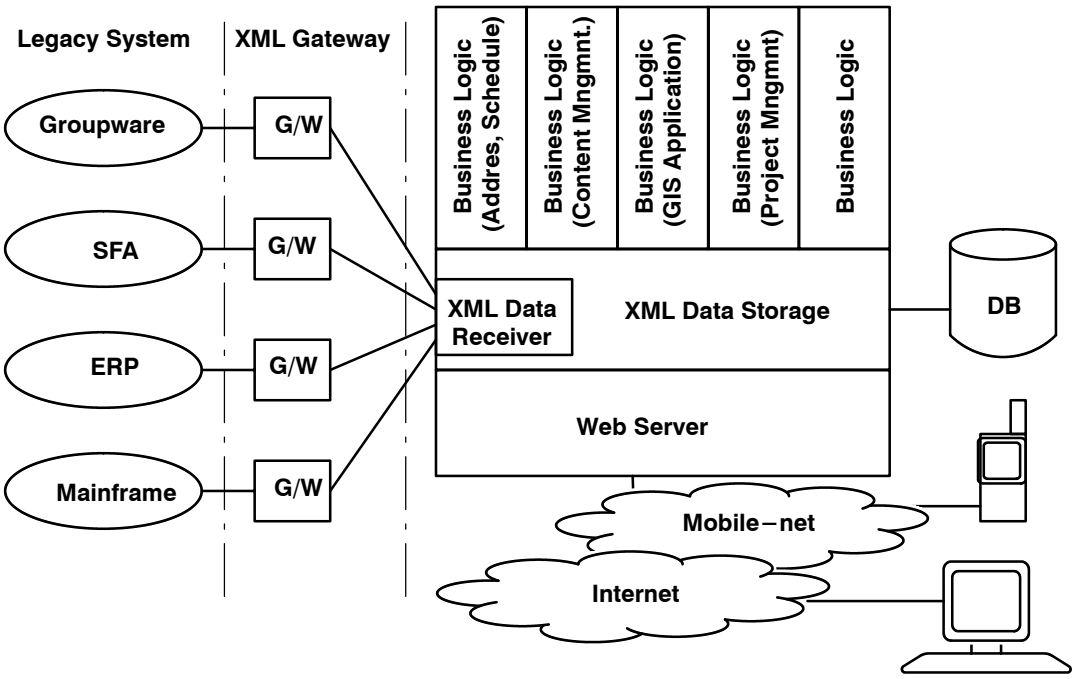


図3　モバイル・インターネットシステムの構成

4.2.2　コアアプリケーションとしてのPIM

ここで述べる携帯電話の高機能化とは、XML処理の枠組みを保有するサーバ機能を携帯電話の中に実装してしまおうとする試みである。携帯電話は個人の属性情報を効果的に管理することが可能である。例えば、ミーティングを設定するような場合に、日時、場所、参加者、議題などが必要な情報となるが、これらの情報は関係者の携帯電話上のPIM（Personal Information Mnager）に情報を送付することにより系統的、有機的、かつ効果的に関係者の業務を支援することが可能である。

4.2.3　日時と場所の情報の獲得と活用

日時に関しては、タイムスタンプを活用した情報履歴の管理、関係者相互間の知的なスケジュール調整などが今後のPIMの基本的な機能となるであろう。場所に関しては、GPSを組み込んだ携帯電話の登場などから推察されるとおり、地図情報システム（GIS）を組み込むことが一般化されるであろう。さらに、GISシステムと鉄道の運賃表や時刻表とを組み合わせることにより、目的地のところ番地や電話番号を入れると、現在地点から鉄道の最寄の駅までのルートを選定して案内し、さらに目的地の最寄の鉄道の駅を選択して最適の列車や乗り換え情報を提示し、最寄の駅から目的地までのルートを選定するようなサービスが可能となりつつある。

4.2.4　認証とセキュリティ

さらに携帯電話の特徴として、個人認証機能が挙げられる。その携帯電話を保有しているのは、まずはその正規の持ち主であり、さらに身体的な特徴と照合することが可能になればその個人認証機能は完璧に近くなる。その認証機能と組み合わせた決済機能の組み込みも検討されており、その実現は

時間の問題である。そうなると、常日頃持ち歩く手帳や財布は携帯電話で済むようになってしまう。その逆に、将来は携帯電話が無いと、生活自体が成り立たなくなる可能性すらあるということだ。このような社会の出現が、望ましいか否かは議論が分かれるところだろう。そのような問題に対するアセスメントを考えながら、今後の技術開発は進める必要がある。

4.2.5 音声認識技術の活用

以上のような携帯電話の利用状況を想定すると、携帯電話の使い勝手が極めて重要な課題となる。現在の小さなキーを複雑に操作するような状況はあり得ないであろう。そうなると音声認識と簡単なキー操作の組み合わせで必要な機能を抽出し、それを実行するようなインタフェースが必須となるであろう。その究極の姿が携帯電話中の電子秘書である。

4.3 具体的な実現性

4.3.1 ナレッジナビゲータ

このような電子秘書は、以前、アップルが「ナレッジ・ナビゲータ」というキーワードで提示したコンセプトに近いものであるが、それはノートパソコン程度のハードウエアを想定したものであった。その当時、携帯電話は出現してはいなかったからである。携帯電話上の電子秘書は「ナレッジ・ナビゲータ」に比べるとはるかに進歩したものである。ナレッジ・ナビゲータが電話の取次ぎやスケジュール調整が主たる役割であったのに対し、携帯電話上の電子秘書は、GPSを用いて現在位置をGISシステム上で確認し、道案内をすると同時に、個人の認証、決済といった信用を要する業務や経済的な活動まで支援することが可能である。

4.3.2 携帯電話上のXML処理系

なお、携帯電話上のXML処理に関しては、30KBのメモリ空間上のi−アプリにおけるJava上で、XMLパーサを使って処理する検討が行われている。さらに村田氏により、RELAXNGによるバリデータも実装されているので、今後の展開にあたり技術的なネックは殆ど無いと考えられる。

5. あとがき

セマンティックWebの可能性について考察を加えたが、RDFとDAML+OILで代表される、メタデータ、オントロジ関連の技術に関しては、発展的な見通しが得られず、どうも悲観的にならざるを得ない。逆を言えば、人間の知識や知的活動というものの奥深さ、深遠さというものを痛感したのであった。それは、主に脳科学[17]や言語学[19]からのアプローチを種々の書籍を通じて知ったからである。セマンティックWebのような分野を開拓することを志すコンピュータ科学者は、コンピュータのことだけでなく、脳科学や言語学を始めとする幅広い分野の知識を習得する必要があると感ぜざるを得ない。

それにしても、驚かされたのは、バートランドラッセルが半世紀前に「人間の知識」[4]で記号論理と科学的な知識との距離を明確に提示してそれを厳密に考察している事実である。私事で恐縮だが、数人のグループで3年余の歳月を費やして「人間の知識」を最近読み終えたところである。難解ではあったが、読み応えがあった。

ラッセルは一世紀前にPrinceples of Mathematics[20]を著して、世界というものを論理的な要素に還元して把握しようとしたが、その試みには失敗した。その失敗の理由を反省を交えて検証した著書が「人間の知識」であると言える。これを読んで、かつてのAIや現在のセマンティックWebは、ラッ

セルで言えば、Principles of Mathematicsレベルの試みに過ぎないのではないだろうか。要するに、そのレベルの課題に対し、コンピュータを用いて挑戦しているだけではないのかとの疑念を抱かざるを得ないのである。

以上のように現状のセマンティックWebの進展の可能性は見えないのであるが、それでもWebは進化し続けるであろう。その観点から最後にその可能性を携帯電話の側から考察してみた。諸兄のご批判とご意見を頂けると幸いである。

参照情報と文献

[1] T. Berners−Lee, et. al.; ”The Semantic Web”, SCIENTIFIC AMERICAN, May, 2001

[2] http://www.w3.org/2000/Talks/1206−xml2k−tbl/slide10−0.html

[3] http://www.inpaku−genji.pref.kyoto.jp/

[4] B.Russell; ”Human Knowledge”, G. Allen & Unwin, (1948), 日本語訳（鎮目訳）”人間の知識”, バートランド・ラッセル著作集, No.9, No.10, みすず書房, (1960)

[5] http://www.cs.umbc.edu/kqml/papers/

[6] http://http://www.fipa.org/

[7] [7] http://www.fipa.org/repository/aclspecs.html

[8] http://www.daml.org/

[9] http://www.daml.org/2000/10/daml−ont.html

[10] http://www.ontoknowledge.org/oil/

[11] http://www.daml.org/committee/

[12] http://www.daml.org/2000/12/daml+oil−index.html

[13] http://www.daml.org/2001/03/daml+oil−index

[14]ヴィトゲンシュタイン;“論理哲学論”,中央公論世界の名著58−ラッセル・ヴィトゲンシュタイン・ホワイトヘッド編,中央公論（1971）

[15] http://dl.kr.org/

[16] 廣瀬、横田; ”ゲーデルの世界”, 海鳴社（1854）

[17] 茂木健一郎; “心を生み出す脳のシステム”, NHKブックス, #931, 日本放送出版協会（2001）

[18] 大野, 前, 吉田, 近藤; “モバイル・インターネット環境構築支援システムの検討”, 情報処理学会研究報告, FI−66, DD−32, (2002.3.15)

[19] スティーブン・ピンカー（椋田訳）;“言語を生み出す本能（上・下）”,NHKブックス,#740・741,日本放送出版協会（1995）

[20] B. Russell; “The Principles of Mathematics”, (1902), Paperback ed., Routledge, (1992)

Gellishについて
－STEP APモデルの拡張－

委 員　森　福 瑞

本文はSTEP AP（アプリケーション・プロトコル）モデルを拡張してSmantic Webへ進化させていくGellishについてご紹介します。

オランダSHELL社のAndries van Renssen氏が、自社の設計とエンジニアリングのベスト・プラティクスを作るための仕組みを開発してきました。エンジニアリングのノウハウを共有するための有効な手段をずっと研究しています。その成果の中、Gellishは基盤となる重要な要素だと思われます。

STEPは、本来いろいろな産業界のユーザから見た業務の共通プロトコルを規定するものです。GellishはSTEPのAPモデルをベースとして、さらに自然言語やエンジニアリング以外の分野へ発展しています。STEPの実用化アプローチの１つを示しています。

Gellishとは

- 構造化された自然言語
- オブジェクトやメッセージの記述
- エンジニアリングデータやビジネスデータ交換に適用可能

STEPの拡張、自然言語バージョン

- EPISTLE Core Model (=> AP221, ISO15926)
- EPSTILE Reference Data Library (ERDL)
- STEPlib

Gellish は、構造化された自然言語から構成されます。記述対象は、オブジェクトおよびメッセージを含みます。適用分野は、エンジニアリング系だけでなく、ビジネス系にも拡張可能で、広範囲なデータの交換に利用できます。

Gellish のベースとなるものは、ヨーロッパのプロセス産業標準 EPISTLE (The European Process Industries STEP Technical Liaison Executive)のコアモデルです。このコアモデルはいわゆるメダモデル、または Generic モデルです。プロセス産業に必要となる基本要素を規定しています。このコアモデルから STEP の AP221 及び ISO15926 が開発されています。STEP の AP221 は、プロセスプラントの詳細設計の標準モデルです。ISO15926 は、石油とガス業界の標準モデルです。

このコアモデルには、抽象的な要素から構成されて、実際のエンジニア用語が入っていません。エンジニアの用語は、次の階層に当たります。ISO15926 では、Reference Data Library で規定します。STEP の AP221 では、STEPlib で規定します。両方で規定しているエンジニア用語は、今後共通な国際規格で統一されます。

SHELL 社は、国際共通のエンジニア用語以外に、さらに独自の用語を自社ののライブラリに拡張しました。

Gellishの定義

•Atomic sentence = <the name of a concept> <the name of a class of relation> <the name of another concept>.

一般形　**A is_related_to B**.

•A message = a series of atomic sentences

•Gellish xmlns='http://www.steplib.com'

language = Gellish_English

Atomic sentence-1

…

/Gellish

Gellishの例

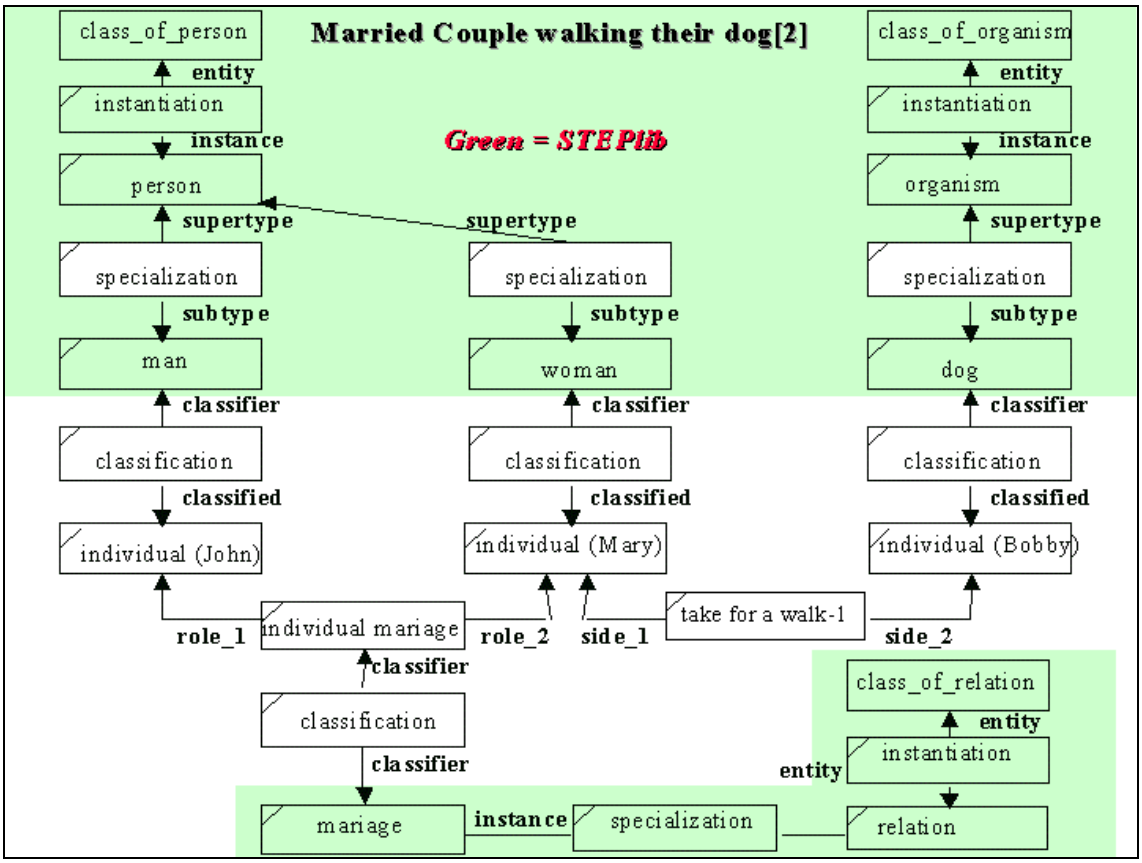
•a wheel can_be_part_of_a car.

•'motor with serial number 1234' is_part_of 'car with chassis number 2468'.

•a "married couple walking their dog".

Gellish の定義ですが、Gellish の基本形は、「A is related to B」で、コンセプト A とコンセプト B との関連を記述するものです。Gellish のメッセージは、Gellish の基本形の集合から構成されています。

Gellish の例が上に示されています。例１、a wheel は a car の部品です。この例の wheel も car も STEPlib に定義されている用語を使用しています。例２、’モータ番号 1234’が’シャーシ番号 2468 の車’の部品を表し、固有名詞が’’または””で囲んでいます。例３、「a married couple walking their dog」という自然言語のメッセージです。基本形に分解されたものを次に図示します。



この例は、以下の基本形から構成されています。

'John' is_a man

'Mary' is_a woman

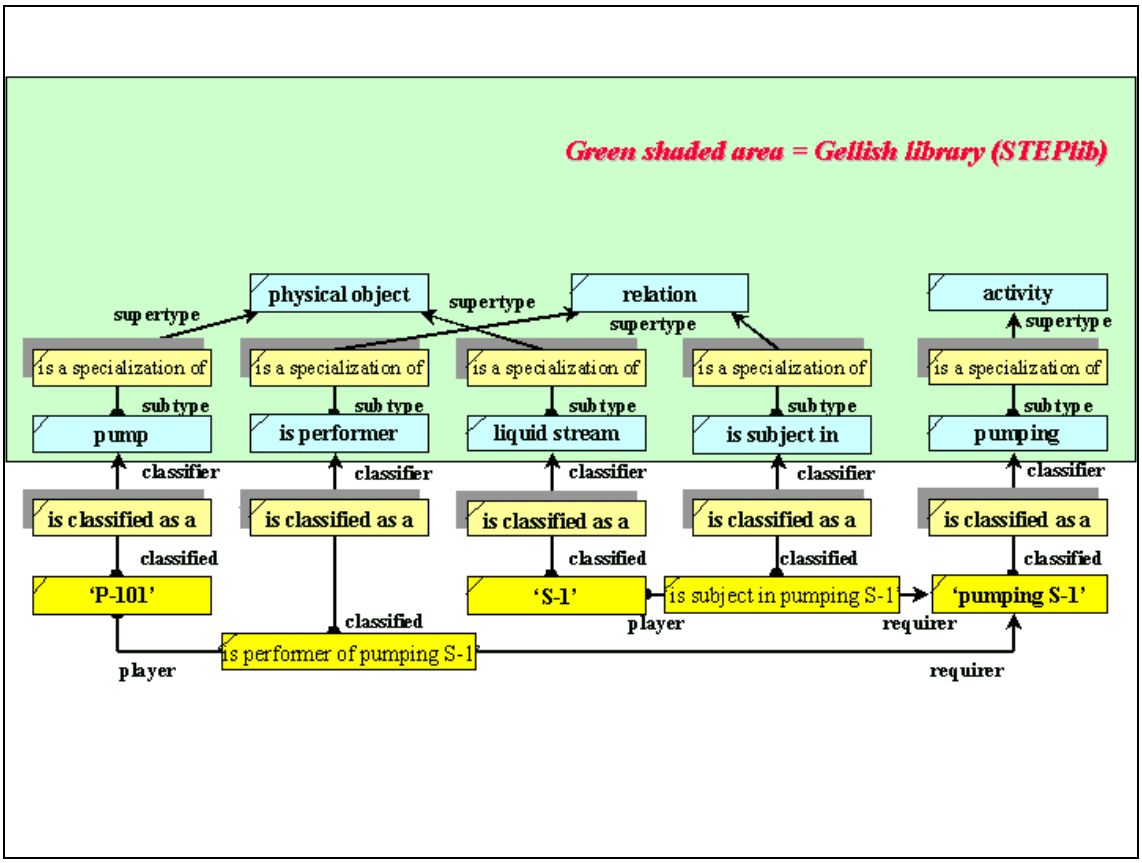
'John' is_married_with 'Mary'

'Bobby' is_a dog

'Mary' is_taking 'Bobby' for_a_walk

図の影部分（緑）は、STEPlib で定義されている用語です。この例は、STEPlib の用語 classification (is_a)を参照するだけで、再定義する必要はありません。

また、できるだけ自然言語を使用するために、John と Mary の individual marriage (is married with)には、関連そのものに名前を付けていません。



次は、エンジニアリング系の例、“P-101 is pumping S-1” 。これは以下の基本形から構成されます。

'P-101' is_a pump

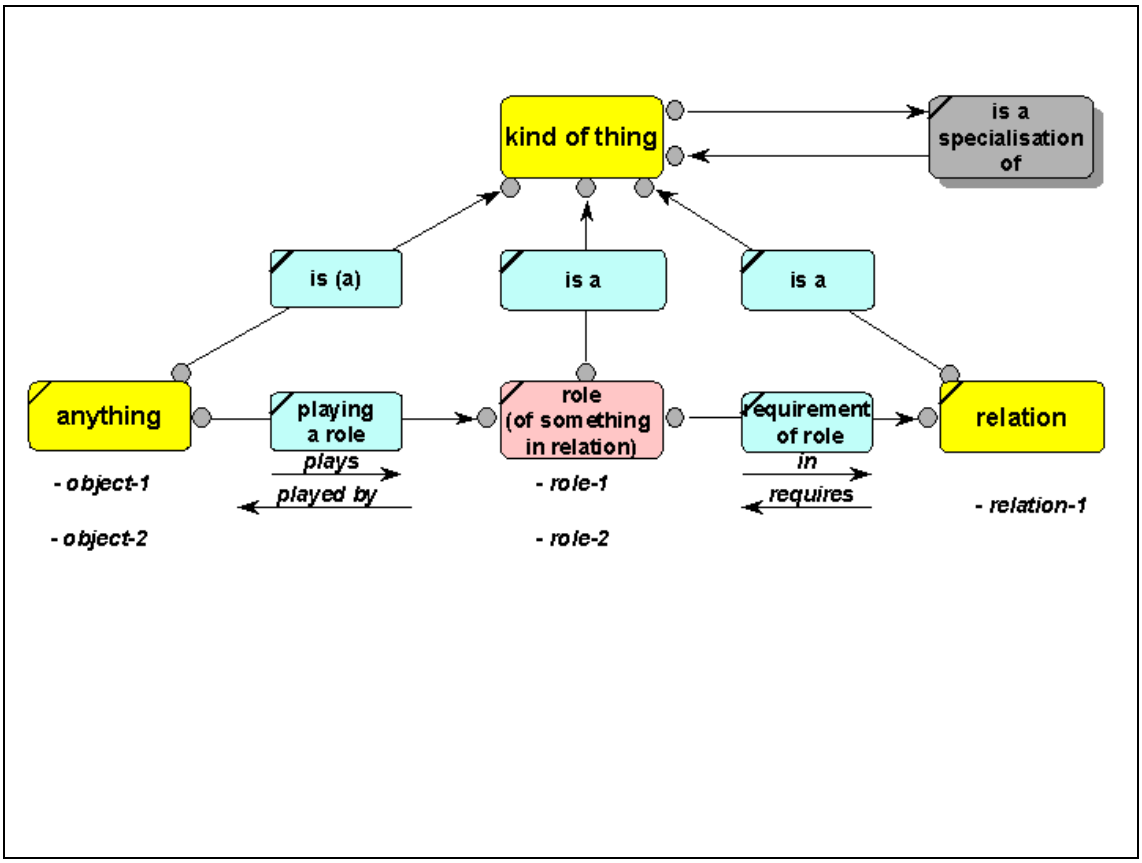
'S-1' is_a liquid stream

'pumping S-1' is_a pumping

'P-101' is_performer_of_pumping 'S-1'

'S-1' is_subject_in_pumping 'pumping S-1'

Pump、liquid stream、pumping、is performer、is subject in は STEPlib で定義済の用語です。Pump と physical object との specification(supertype/subtype)関連も STEPlib で定義されています。



Gellish の基本定理

上図には、Gellish の基本要素及び構造を示しています。

任意のメッセージは、２つ（または２つ以上）の object 間の基本関連(role の箱)及び object, role, relation の classification(is a の箱) 関連で表現できます。

例えば、「impeller 01 is part of centrifugal pump 02」は以下のように分解されます。

基本関連

- object 01 plays role R1
- role R1 is required by relation C1
- relation C1 requires role R2
- role R2 is played by object 02

classification 関連

01 is classified as an ‘impeller’

- R1 is classified as a ‘part’
- C1 is classified as a ‘part-whole relation’
- R2 is classified as a ‘whole’
- 02 is classified as a ‘centrifugal pump’

実装上は、明示的な role と classification を省くことができて、下記の３つの基本形だけで十分です。

- 01 is classified as an ‘impeller’
- 01 is part of 02
- 02 is classified as a ‘centrifugal pump’

STEP APモデル	Gellish
Instantiation	Explicit classification relations
Entities have fixed Attributes	Explicit Relations between objects Explicit roles of objects in relations
Fixed Subtyping of entities	Specialization relations between classes
Entity and attribute types are not instances (fixed)	Entity types and attribute types (classes) are instances

Gellish と STEP AP モデルの違いを上の表に示しています。

Gellish は、classification 関連を明示的に表現します。

STEP AP モデルでは、Entity は固定の Attribute を持つが、Gellish では、role 関連とほかの関連で表現します。

従来の STEP AP モデルでは、Entity 間の supertype/subtype は固定しています。Gellish では、specialization 関連で表現します。

従来の STEP AP モデルでは、利用者がモデル内に定義されていない Entity と Attribute を追加できません。Gellish は Entity も Attribute も instance として扱うので、追加が可能です。(ただしルールが必要です)

以上の観点から見ると、Gellish が柔軟性に富んだモデルということが理解できま

す。

STEPlib には、現在約２万の用語及び同じ数の関連が定義されています。STEPlib のホームページに最新の用語集が Excel ファイルの形で一般公開されています。オランダ SHELL 社の Andries van Renssen 氏ご提供の情報によると、SHELL 社がさらに自分のノウハウ（現時点は約１万の用語）を蓄積して、自社の設計やエンジニアリング業務に活用しています。

本文は今後 STEP を活用する例としてご参考になれば幸いです。

STEP と Semantic Web との融合を目指して
—制約自然言語の提案—

委 員　太田 吉美

1. はじめに

STEP（Standard for the Exchange of Product model data、ISO10303 を中心に、ISO13584、ISO14959、ISO15531、ISO15926 などの規格がある）は、膨大な知的資産であると認識している人は少ないが、間違いなく「ものつくり」の技術知識基盤として利用できる知的資産である。他を見渡してもこれ以上の知的資産はない。これまで開発を進めてきた関係各位の努力は尊敬に値する。

STEP は先取りの国際標準としてスタートし、早 20 年を経過しようとしている。この間、情報技術の進展はすざましく、従来の技術を瞬く間に陳腐化させてしまう。STEP も同様で、技術的にも STEP の大幅な見直しが必要な時期にきている。

一方、Web の進化は劇的であり、Web に意味を持たせ、Web の利便性を飛躍的に革新するセマンティック Web が提唱されている。セマンティック Web の世界では、これを普及させるためのキラーアプリケーションが不在で、おそらく、企業間の情報交換や情報共有から普及するものと考えられている。このような意味からも、STEP とセマンティック Web との融合は必然的な方向である。

2002 年度から始まった e-ENG のアドホック部会であるセマンティック Web 部会では、「ものつくり」の技術知識基盤をめざし、STEP とセマンティック Web の融合をテーマに活動している。

本報告では、STEP を「ものつくり」のための重要な知的資産と認識した上で、STEP の概要について述べ、STEP の課題を明確にする。次に、STEP や CALS の実証から発展し、情報や知識の共有化に至った技術知識基盤（テクノインフラ）構築のプロジェクトとこれに関連する Web 技術（Web サービスとセマンティック Web）の概要について説明する。さらに、STEP とテクノインフラやセマンティック Web との融合方法について考察する。

また、新たに提案する制約自然言語と STEP との融合を検討する。制約自然言語は、テクノインフラの成果を発展させたもので、基本構造は主語＋述語＋目的語で、主語や述語、目的語の取り得る内容を辞書で限定する。これにより、一般の自然言語処理の難しさを排除する。日本が世界をリードしている機械翻訳技術との連係を図ることもできる。例えば、辞書を多言語対応にすることにより、海外の協力も得やすい。

STEP の知的資産と制約自然言語を融合することにより、日本発の「ものつくり」の技術知識基盤を早急に立ち上げることができる。是非、皆様のお知恵を拝借したい。ご意見やご批判は yoshimi-ota@k3.dion.ne.jp までお願い致します。

２．STEP の概要

STEP は ISO が進めている国際規格の一つで、ISO/TC184（Technical Committee 184）/SC4（Sub-Committee 4）で 1984 年に審議が開始された。TC184 は産業分野のオートメーションに関する標準化を審議する組織で、その中の SC4 で STEP を審議している。ISO/TC184/SC4 で審議された結果は、国際規格 ISO 10303 （Industrial Automation System : Product Data Representation and Exchange）として登録されている。

２．１ STEP のニーズ

プロダクト･モデルのデータ交換標準化のニーズとしては、

（１）情報システム間のデータ交換
（２）情報システム間の連携、統合化、オープン化
（３）顧客との契約、取引の電子化
（４）国内外他社との共同事業
（５）資材、部品調達のグローバル化
（６）アウトソーシング
（７）ソフトウェアやハードウェアに依存しない製品情報の長期管理保存
（８）技術提携や技術供与
（９）設計･生産情報を知的資産として蓄積･活用

などいろいろ挙げることができる。

STEP はこのようなニーズに応えようとするもので、企業間のデータ交換という本来の目的以上に、CAD／CAE／CAM システム統合化の手段としても利用できる。図 2.1 は 1995 年に作成したものであるが、まだ、現状から脱していないように思える。残念ながら 10 年以上、あまり進歩していないことになる。ほとんどのドキュメントは電子化されているが、まだ紙や PDF（Portable Document Format）をマスターとしているケースが多い。

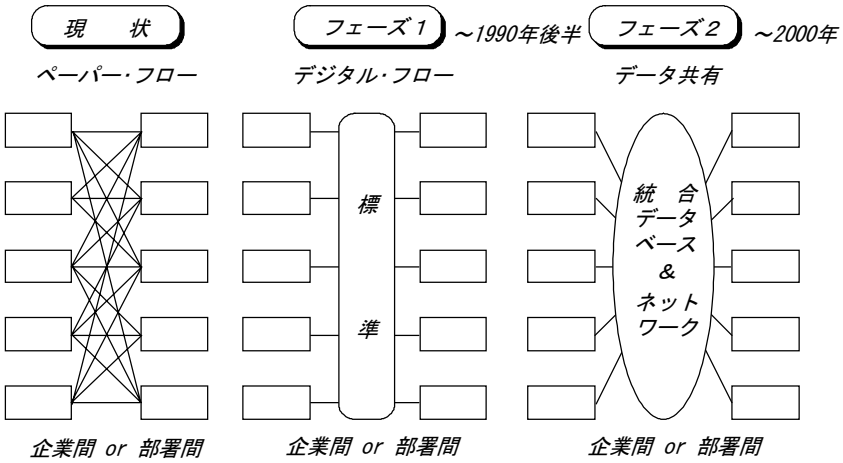


図 2.1　STEP による統合化アプローチ

２．２ STEP の開発レベル

STEP の開発レベルは、次のように考えられていた［1]。

（１）レベル 1：受動的ファイル交換

・トランスレータを介してデータ交換

（２）レベル 2：アクティブなファイル交換

・ローカルな作業ファイルを介してデータを操作、検索

・トランスレータはアクセス関数を使用してアクセス

（３）レベル 3：データベースによる共有

・製品データは、データベース管理システムに格納し、データを共有

・トランスレータは DBMS データ操作言語を使用して直接アクセス

・製品のマスター･データベースとして利用

（４）レベル 4：知識ベースによる共有

・製品データは、ナレッジ･ベース･マネージメント･システム（KBMS）に格納し、知識として共有

・トランスレータは KBMS データ操作言語を使用して直接アクセス

・製品のマスター･データベースとして利用

現状の STEP は（１）から（２）のレベルであり、（３）（４）の実現は夢になりそうである。

２．３ アプリケーション･プロトコルの開発方法

STEP では、各種産業分野における製品のライフサイクル全般の情報交換をねらいとしている。このような情報モデルの開発では、製品のライフサイクル全般にわたり、作業フローを分析し、各作業フローでの必要な情報を洗い出し、情報の構造化を行うこと必要である。こうした開発のため、STEP では各種の標準的な手法が利用されている。STEP では、次の三段階のステップで AP（Application Protocol）が開発される（図 2.2 参照）。

（１）AAM：Application Activity Model

まず、対象となる業務の内容や交換する情報について、関係者の間で共通の概念、理解を持つことが必要である。このため、STEP では、AAM を作成する。AAM では、対象とするアプリケーション分野のアクティビティを分析し、各アクティビティと情報の流れを表現し、アクティビティでの入出力情報を明確に定義する。この記述には、IDEF0（Intergrated Computer aided manufacturing DEFinition）の表記方法が使われる。

（２）ARM：Application Reference Model

ARM では、AAM で分析した結果に基づき、対象とするアプリケーションで必要な情報についてのプロダクト･モデルを作成する。この記述には、IDEF1X や EXPRESS-G などの表記方法が使われる。

（３）AIM：Application Interpreted Model

AIMでは、ARMで作成したプロダクト・モデルを、STEPの統合リソースにマッピングし、マッピングできないものは新たに追加して具体的なデータ交換の実装方法であるAIMを作成する。AIMの記述では整合性を確保し、曖昧さを排除するために、データ記述にEXPRESSという形式言語を利用する。

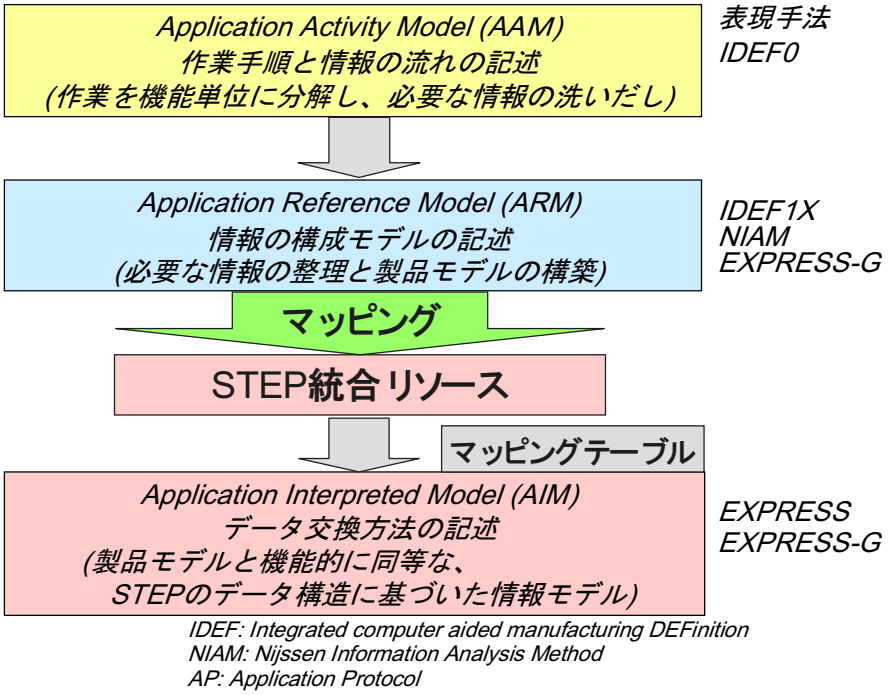


図 2.2　AP 開発方法

2．4　STEP 規格の構成

STEPでは製品のライフサイクル全般を通じて、曖昧さのない計算機処理可能な製品情報の記述と交換の形式を規定するものである。この目的を達成するため、規格の構成要素を分類し、製品情報を構造化し、概念モデルと実装形式を分離し、可能なすべての部分を形式言語で記述する方式を採用している。

STEP 規格は、広範囲にわたるプロダクトデータに関する表現や交換に利用されるため、各分冊（Part）に分けられて、7 つの文書クラスから構成されている。分冊は、必要に応じて追加できる構成になっている（図 2.3 参照）。

Part 1 は、STEP の概要と基本的な考え方を記述している。Part 11 は、STEP の記述方法を示したもので、EXPRESS 言語参照マニュアルになっている。STEP では、整合性を確保し曖昧さを排除するために、EXPRESS という形式言語を利用する。この言語は、人に理解できると同時に、計算機によるデータ交換処理が可能になっている。Part の 20 番台では、実装方式を記述している。30 番台では、適合性試験などの規約を規定している。40 番台では、プロダクト・モデルを統合するためのリソースについて規定している。100 番台は、基本的なアプリケーション・リソースについて規定している。

AP に関するものは、200 番台の番号がとられている。AP では、個々の応用分野の要求に従い、実際に交換される一群のデータ構造を規定する。STEP の利用者は、統合リソースを直接利用することはなく、自分に必要な AP を選択して用いることになる。

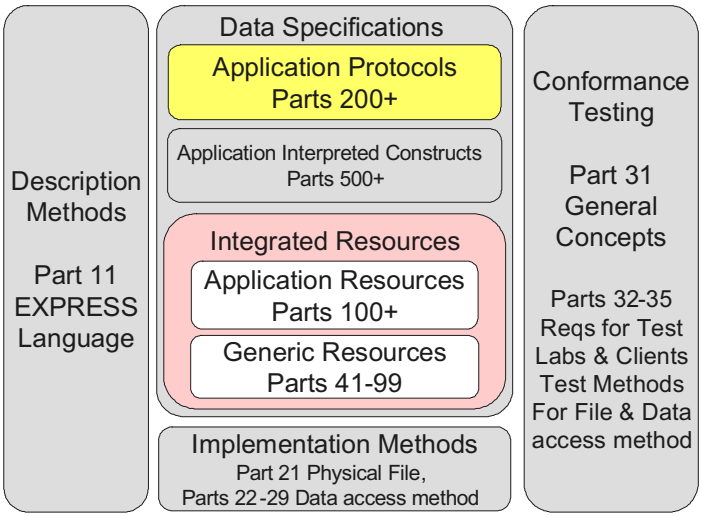


図 2.3　STEP 規格の構造

図 2.4 は統合リソースの構成を示したものであり、図 2.5 は統合リソースの構造を示したものである。また、図 2.6 は統合リソースの一部を詳細に展開（形状の Part42 や表示関連の Part46、製図の Part101 は展開していない）したものであり、これからも分かるように「ものつくり」に必要な基本的な情報について整理されている。このことは注目に値することであるが、一般にはなかなか理解されていないのは残念である。

図 2.7 は AP の ARM から統合リソースへのマッピングの過程を示したもので、これらの結果は統合リソースへのマッピングテーブル（MT s ）とその結果として、AIM にまとめられる（図 2.8 参照）[1]。

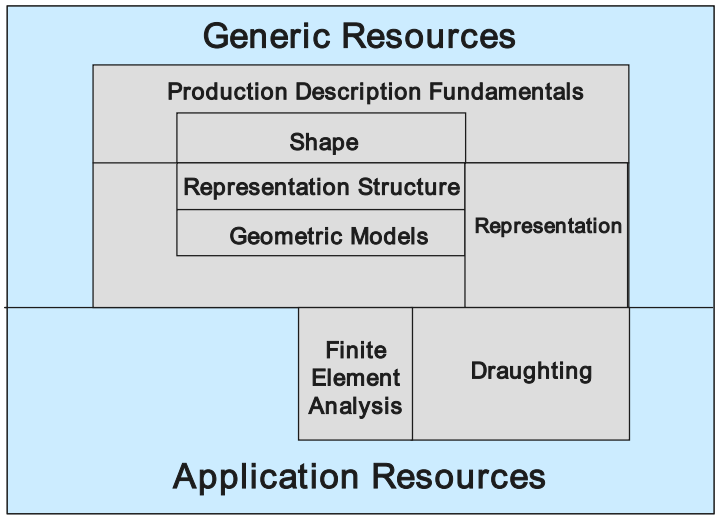


図 2.4　STEP 統合リソースの構成

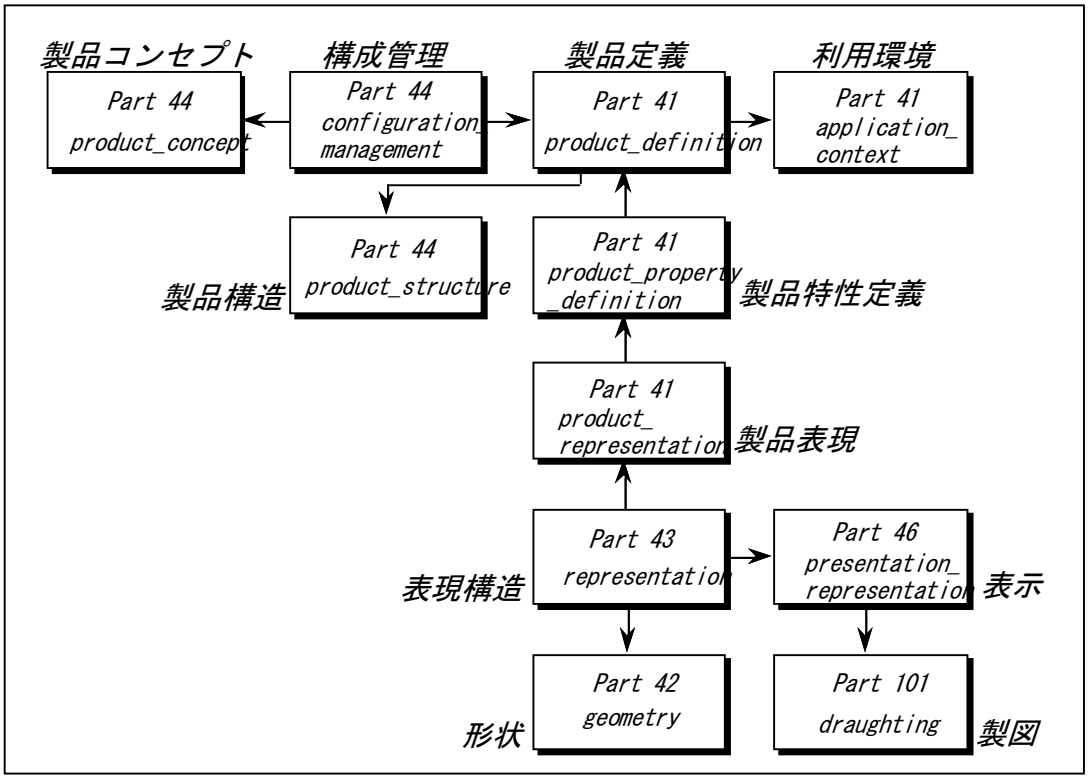


図 2.5　STEP 統合リソースの構造

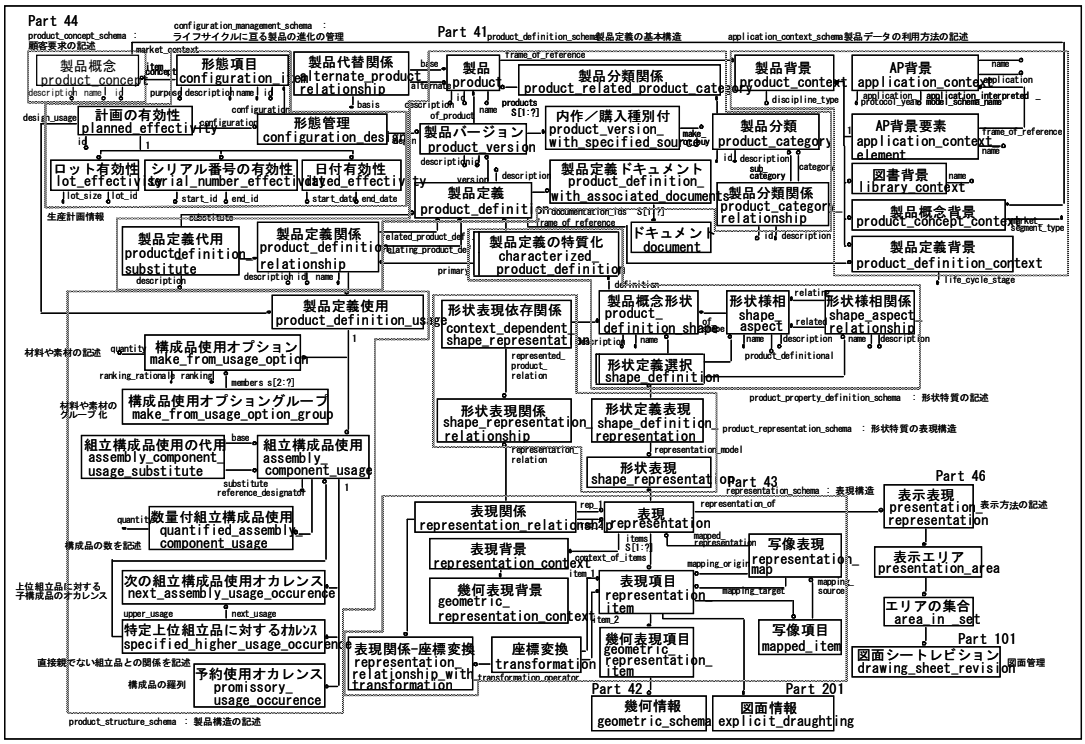


図 2.6　STEP 統合リソースの構造（一部詳細）

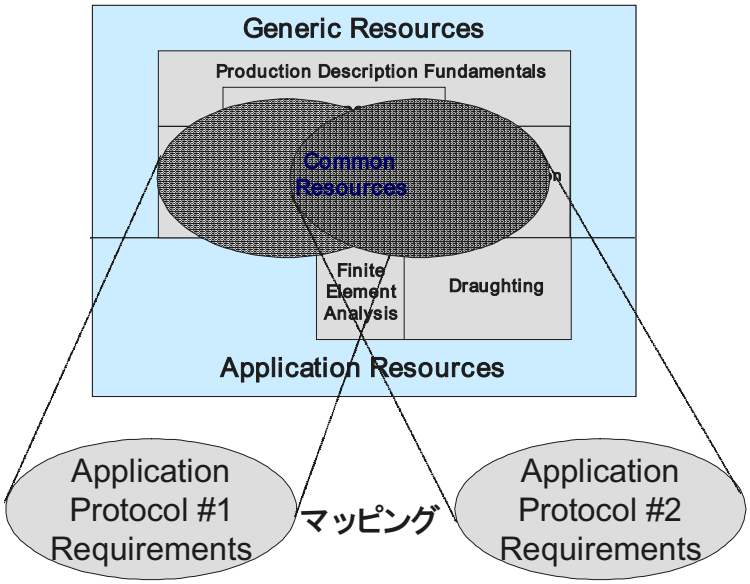


図 2.7　STEP 統合リソースへのマッピング

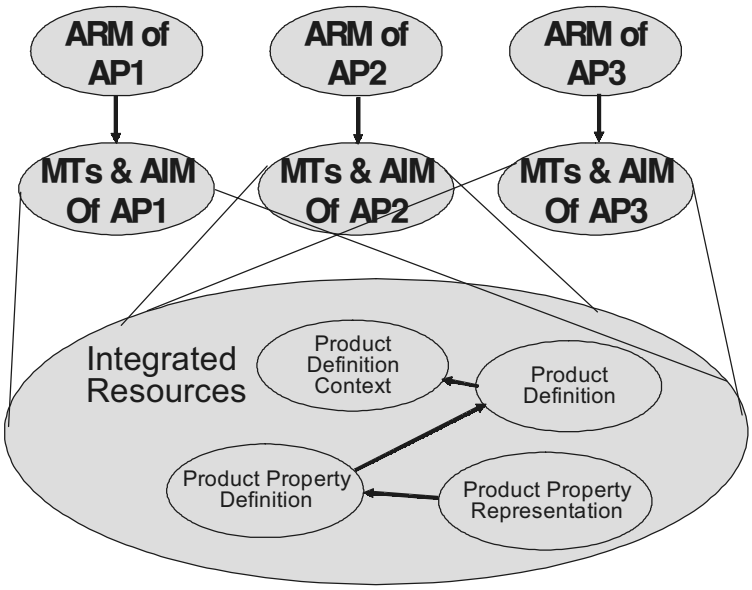


図 2.8　AP と統合リソース

図 2.9 は STEP 規格（2002.11.13 時点）を一ページのまとめたものであり、約 60 に近い分冊が国際標準となっている[2]。

ISO TC184 SC4

STEP on a Page

ISO 10303

APPLICATION PROTOCOLS AND ASSOCIATED ABSTRACT-TEST SUITES

I 201 Explicit draughting [ATS 301 = X]
I 202 Associative draughting [X]
I 203 Configuration-controlled design (c2=I,a1=I)[X]
I 204 Mechanical design using boundary rep [I]
X 205 Mechanical design using surface rep [W]
X 206 Mechanical design using wireframe [X]
I 207 Sheet metal die planning and design [I]
X 208 Life-cycle product change process [X]
I 209 Composite & metal structural anal & related design[X]
I 210 Electronic assy, interconnection & packaging design [X]
X 211 Electronic P-C assy: test, diag, & remanuf[X]
I 212 Electrotechnical design and installation [C]
X 213 Num control (NC) process plans for mach'd parts [X]
I 214 Core data for automotive mech design processes [F]
E 215 Ship arrangement [X]
E 216 Ship moulded forms [X]
X 217 Ship piping [X]
E 218 Ship structures [X]
X 219 Dimension inspection [X]
O 220 Proc. plg, mfg, assy of layered electrical products [X]
C 221 Functional data & their schem rep for process plant [X]
X 222 Design-manuf for composite structures [W]
X 223 Exch of design & mfg product info for cast parts [@]
I 224 Mech pdt def for p. plg using mach'n'g feat (e2=X,e3=A) \[X,@]
X 226 Ship mechanical systems [C]
I 227 Plant spatial configuration(e2=C) [X]
X 228 Building services: HVAC [X]
X 229 Design & mfg product info for forged parts[X]
X 230 Building structural frame: steelwork [X]
X 231 Process-engineering data [X]
I 232 Technical data packaging: core info & exch [I]
W 233 Systems engineering data repr (to be PAS 20542)[X]
X 234 Ship operational logs, records, and messages[X]
W 235 Materials info for des and verif of products [X]
W 236 Furniture product and project data[W]
W 237 Computational Fluid Dynamics
A 238 Computer numerical controllers
W 239 Product life-cycle support
W 240 Process plans for machined products

DESCRIPTION METHODS

I 1 Overview and fundamental principles
I 11 EXPRESS language ref man. (c1=I,c2=C e2=C e3=X(ISO 20303=X a1=X)
I 12 EXPRESS-I language ref man (Type 2 tech report, not a 10303 part)
X 13 Architecture and Methodology reference manual
E 14 EXPRESS X Language reference manual

COMMON RESOURCES (with 13584-20 logic. model of expr.(I) and 15531-42 Time (W))

APPLICATION MODULES (Technical specifications)

Because there are many of these planned SOAP has been forced to be SOAP.5, STEP on a page and a half. For their listing, please access the file via the SOAP home page.

Legend: TS Status
0-10 =O=prop-->apvl for ballot
10-20=A=NP blt circ-->NP apvl
20-60=D=DTS dev-->reg as TS
>60 =T=TS Published

INTEGRATED-APPLICATION RESOURCES

I 101 Draughting (c1=I)
X 102 Ship structures
X 103 E/E connectivity
I 104 Finite element analysis
I 105 Kinematics (c1=I, c2=I)
X 106 Building core model
C 107 Finite-element analysis definition relationships
C 108 Prmetizat'n&Constraints for expl geom prod mdls
W 109 Assembly model for products
W 110 Mesh-based computational fluid dynamics

INTEGRATED-GENERIC RESOURCES

I 41 Fund of prdct descr & spt (e2=I,c1=I)
I 42 Geom & top rep (c3=I,e2c1=I,e3=E)
I 43 Repres specialization (e2=I,c1=I,c2=I)
I 44 Product struct confg (e2=I,c1=I)
I 45 Materials (c1=I)
I 46 Visual presentation (c1=I, c2=I)
I 47 Tolerances (c1=I)
X 48 Form features
I 49 Process structure & properties
I 50 Mathematical constructs
E 51 Mathematical description
W 52 Mesh-based topology
W 53 Numerical Analysis

APPLICATION-INTERPRETED CONSTRUCTS

I 501 Edge-based wireframe
I 502 Shell-based wireframe
I 503 Geom-bounded 2D wireframe
I 504 Draughting annotation
I 505 Drawing structure & admin.
I 506 Draughting elements
I 507 Geom-bounded surface
I 508 Non-manifold surface
I 509 Manifold surface
I 510 Geom-bounded wireframe
I 511 Topological-bounded surface
I 512 Faceted B-representation
I 513 Elementary B-rep
I 514 Advanced B-rep
I 515 Constructive solid geometry
X 516 Mechanical-design context
I 517 Mech-design geom presentation(c1=I)
I 518 Mech-design shaded presentation
I 519 Geometric tolerances (c1=I)
I 520 Assoc draughting elements
@521 Manifold subsurfaces
E 522 Machining features

IMPLEMENTATION METHODS

I 21 Clear-text encoding exch str (c1=I,e2=I)
I 22 Standard data access interface
I 23 C++ language binding (to #22)
I 24 C language binding (to #22)
C 25 EXPRESS to OMG XMI
X 26 IDL language binding (to #22)
I 27 JAVA language binding (to #22)
@28 XML rep for EXPRESS-schemata & data
X 29 Ltwt Java binding (to #22) \(DTS)

CONFORMANCE TESTING METHODOLOGY & FRAMEWORK

I 31 General concepts
I 32 Requirements on testing labs and clients
X 33 Structure and use of abstract test suites
I 34 Abstract test methods for Part 21 implementation.
C 35 Abstract test methods for Part 22 implementation.

Legend: Part Status (E, F, I safe to implement)
0=O=Preliminary Stage (Proposal-->appr for NP ballot)
10=A =Proposal Stage (NP ballot circ-->NP approval)
20=W=Preparatory Stage (Wkg Draft devel.-->CD regis)
30=C =Committee Stage (CD circulation-->DIS regis)
40=E =Enquiry Stage (DIS circ.-->FDIS registration)
50=F =Approval Stage (FDIS circ-->Int'l Std regis)
@=At ISO, approved for publication (ISO status 40.95 or 50.99)
60=I =Publication Stage (Int'l Std published)
98=X=Project withdrawn

jgnell, 89-Oct.-23; **rev. 02-11-13**. Origin: ISO 10303 Editing Committee. On-line: http://www.nist.gov/sc5/soap/

図 2.9 STEP on a Page

２．５ EXPRESS 言語

STEP ではプロダクトモデルの記述における整合性を確保し曖昧さを排除するために、データ記述に EXPRESS という形式言語を利用する。この言語は人に理解できると同時に計算機によるデータ処理が可能になっている。

EXPRESS の特徴をまとめると次のようになる。

（１）オブジェクト指向言語

（２）人間と計算機に理解できる言語

（３）データ定義と制約の記述が可能

[例]　円は半径を持つ（データ定義）、半径は正の値をとる（制約）

（４）エンティティとインスタンスの区別

[例]　円：エンティティ、半径が 10mm の円：インスタンス

（５）分野ごとの独立性を保証（スキーマの概念）

（６）普通のプログラム言語と同様に宣言文、実行文、アルゴリズムなどの記述が可能

STEP では EXPRESS 言語を用いて、利用分野ごとのデータ交換の仕様（データの種類や構造）を記述する。EXPRESS を用いることにより、データ交換の仕様のあいまいさを排除し、独自のデータ交換仕様を作成できる。

STEP 規格は、EXPRESS を用いて AIM の定義を行うことにより、次のような特徴を備えることができる。

（ｉ）曖昧さのない規格の解釈が可能。

（ⅱ）統合リソース、AIC、AIM などの各モデルのモジュール化を行うことができる。

（ⅲ）物理的な媒体上での実体の表現（ファイル形式など）を、EXPRESS で記述されたモデルからの形式的な写像の形で定義する。従って個々のモデル毎にファイル形式などを定義する必要がない。

（ⅳ）記述されたモデルを計算機プログラムによって処理することによって、内容の自動的な検証、応用プログラム開発の効率化などへの応用が可能となる。

EXPRESS 言語の主な構成要素としては次のものがある。

(A) スキーマ（Schema）：一つのまとまった情報記述構造

(B) エンティティ（Entity）：オブジェクト指向言語のクラスに近い概念で、情報のまとまりの単位

(C) 継承関係：下位型となるエンティティにおいて SUBTYPE 宣言を用いて上位型を指定することで定義

(D) データ型：属性の型を定義

(E) インスタンス：定義されたエンティティの各属性が実際の値をもち、参照構造を実体化したもの

(F) EXPRESS-G：EXPRESS の図式表記法

データ型としては次のものがある。

（a）単純データ型

・数値データ型（NUMBER）

・実数データ型（REAL）

・整数データ型（INTEGER）

・論理データ型（LOGICAL）

・文字列データ型（STRING）

（b）集合体データ型

・配列データ型（ARRAY）：構成要素順序あり、インデックス付、要素の重複可

・リストデータ型（LIST）：構成要素順序あり、要素の重複可

・集合データ型（SET）　：構成要素順序なし、要素の重複不可

・BAG 型（BAG）　：構成要素順序なし、要素の重複不可

（c）名前付きデータ型

・エンティティデータ型（ENTITY）

・定義データ型（TYPE）

（d）構成データ型

・列挙データ型（ENUMERATION）

・選択データ型（SELECT）

２．６　知識資産としての例（プロセス･エンジニアリング）

以下では、プロセス･エンジニアリング関連の STEP の AP である ISO10303 AP227 の概要について説明するが、その知的資産について認識して頂きたい。

図 2.10 は AP227 の AAM の例である。AAM は、プロセス･エンジニア、建築技術者ならびに施工業者を含むプラント空間構成情報の利用者の視点で作成している。

AP227 のプラニング･モデルを図 2.11 に示す。プラントは、プラント･アイテムから構成され、プラント･アイテムはパイピング･コンポーネントとしての属性や形状、接続情報を持つことを示している。

また、AP227 の ARM の全体を図 2.12 に示す。ARM は、IDEF1X で記述しており、AP227 で規定しているエンティティの構造及び制約の図式表現を表している。全体のモデルは、漸く A0 サイズに収まる程度である。図 2.13 は図 2.12 の'Plant'というエンティティについての部分を拡大したものである。

さらに、AP227 の AIM の全体を図 2.14 に示す。AIM は、EXPRESS-G で記述されているが、実際の交換や共有のための情報モデルは、EXPRESS で記述されている。これも全体のモデルは、漸く A0 サイズに収まる程度である。図 2.15 は図 2.14 の'Product'というエンティティ（AIM でのメイン･エンティティ）についての部分を拡大したものである。

AP227 は、1000 ページを超える膨大なドキュメントとなっており、プラントの空間構成

モデルに関する貴重な知的資産となっている。

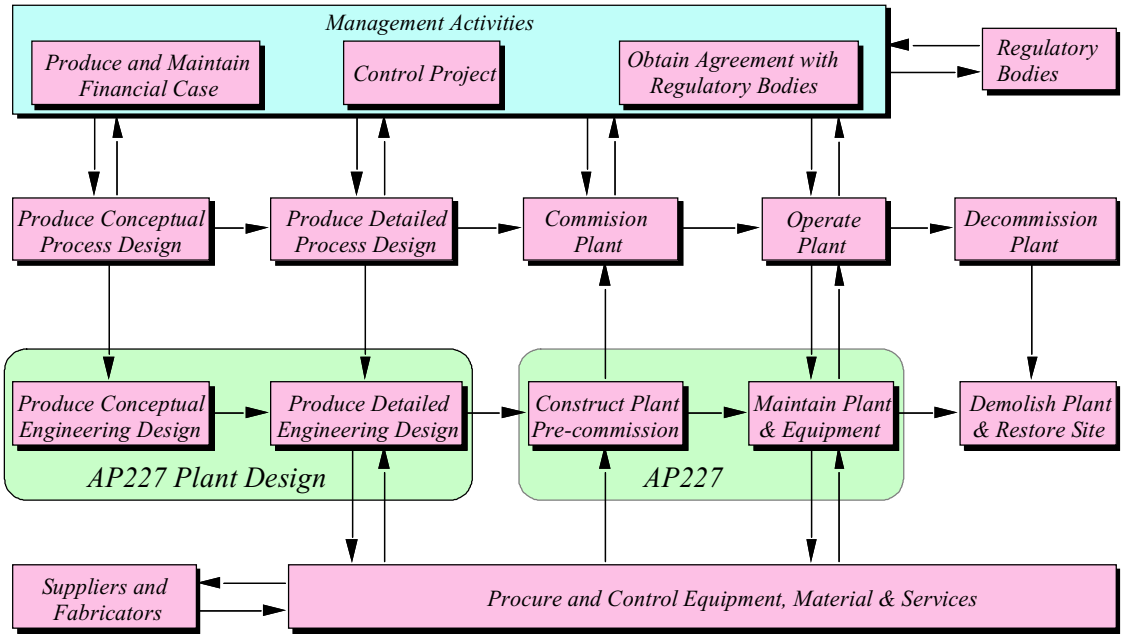


図 2.10　プラントのライフサイクルと AP227 のカバー範囲

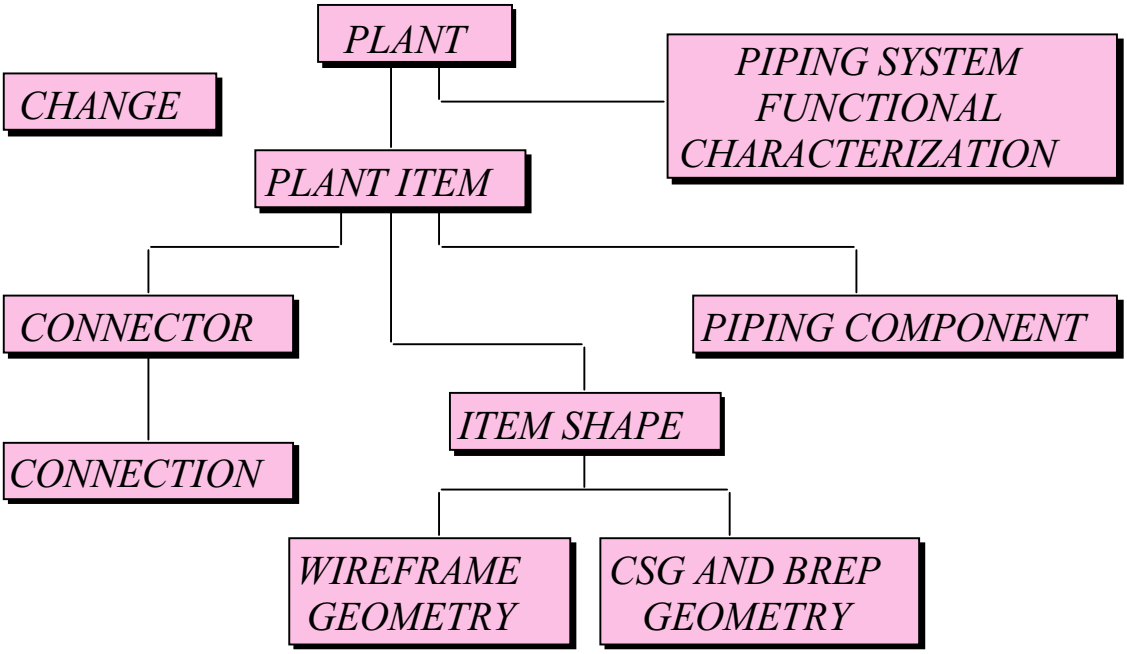


図 2.11　AP227 のプラニング･モデル

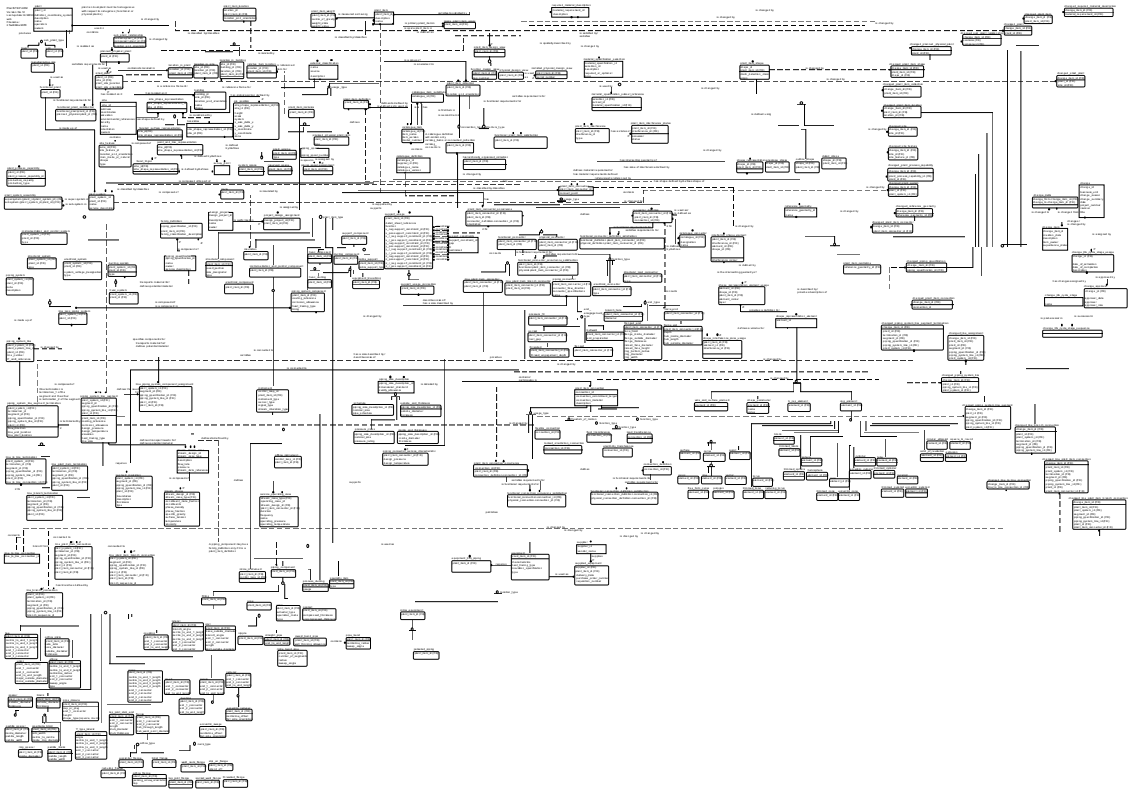


図 2.12　AP227 の ARM

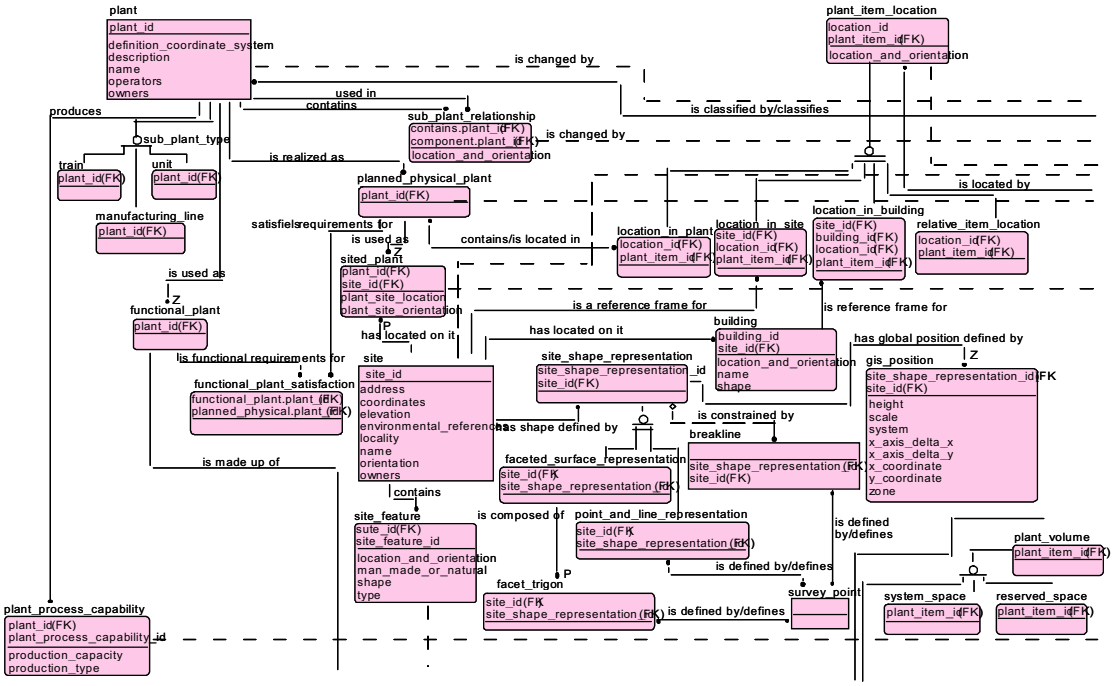


図 2.13　AP227 の ARM の部分拡大

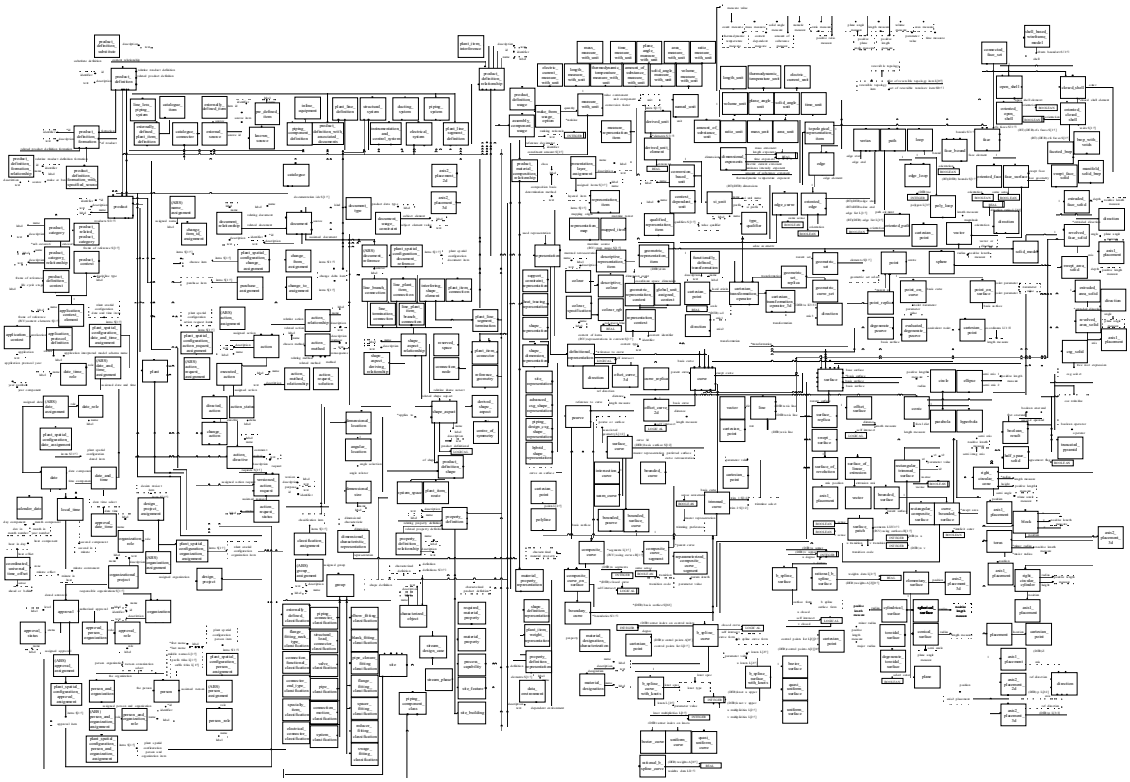


図 2.14　AP227 の AIM

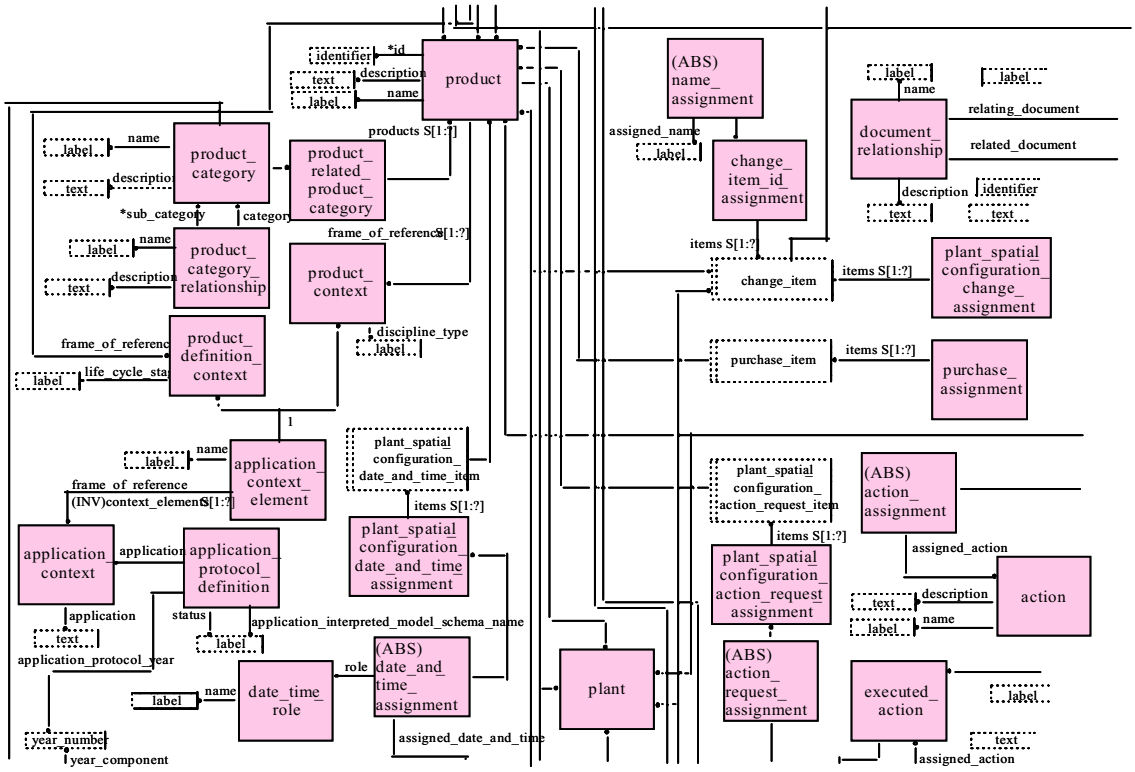


図 2.15　AP227 の AIM の部分拡大

2．7　データ交換方法

STEP での交換方法は、ファイル交換、アプリケーション・インターフェース、データベース・インターフェースなど様々な形態に対応することが考えられている。基本的には、EXPRESS で記述された規格を同様に別な形式言語に写像することにより、交換形式が曖昧さなく規定される。

図 2.16 は、ファイル形式（テキスト･ファイルや XML ファイル）による CAD システム間のデータ交換の方法を示したものである。

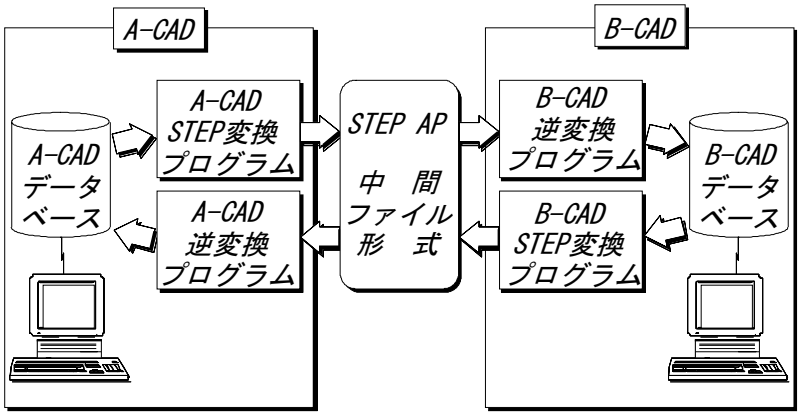


図 2.16　テキスト･ファイル形式によるデータ交換

また、EXPRESS 表現記述を C++、C、IDL（Interface Definition Language）、Java などの言語に変換する SDAI（Standard Data Access Interface）も開発されている。STEP は、プロダクト･データの交換が目的であるが、さらに CAD／CAE システムの統合データベースとしても利用できるものとなっている。

最近では、図 2.17 に示すように、XML 対応のインプリメンテーション方法（Part28、XML representation of EXPRESS schemas and data）も開発されている。

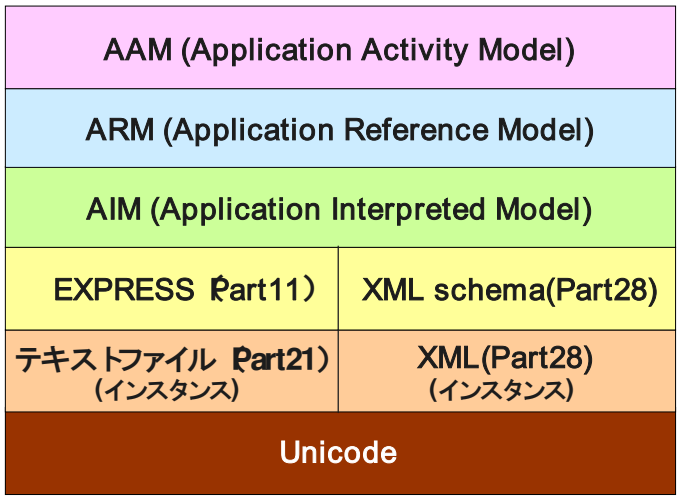


図 2.17　STEP における XML の位置づけ

ISO10303 Part28 は、EXPESS スキーマとインスタンスを XML 形式に変換する規格であるが、図 2.18 に示すようにいろいろな対応がとれるようになっている。

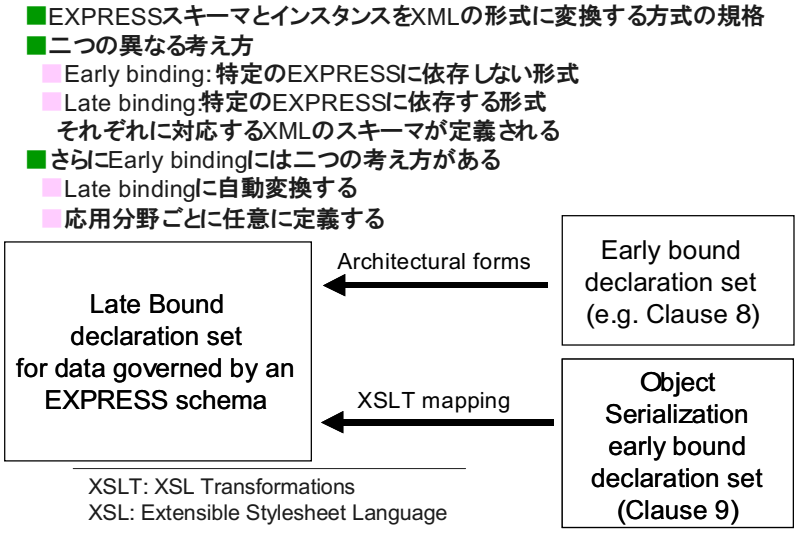


図 2.18　ISO10303 Part28

２．８　STEP の課題

先に述べたプロセス･エンジニアリング分野の例が示すように、STEP の開発方法論及びその開発してきた膨大な成果は、知的資産としての価値がある。しかしながら、現状の STEP についての課題も多い。以下に課題をまとめる。

（１）現状の STEP は、データ交換の規格であり、情報共有や蓄積の仕掛けではない。

（２）AP 間のインターオペラビリティ（相互運用性）が図られていない（プロセス･エンジニアリング分野に関連する AP としては５つ以上もあり、それぞれの AP に対応したトランスレータを用意することは大変である）。

（３）AP の拡張性に乏しい（新しい項目の追加や変更が容易でなく、再度 ARM と AIM を変更し、各国の承認が必要になる）。

（４）ARM から AIM へのマッピングが非科学的である（ARM の本来の意味が AIM に反映されない可能性がある。また逆マッピングが 1 対 1 でないことが起こりえる）。

（５）ARM はあくまで参考図書であり、正式な情報表現モデルは AIM である（ドメインのユーザーは ARM で理解するし、ARM のレベルで最終的にも記述したい）。

（６）AP の開発スピードが遅い（AP の開発には最低でも５年はかかる。AP 開発のリソースの確保や各国間の調整などある程度はしかたがない面もあるが）。

以上のような課題を現状の STEP の枠組みの中で解決するのか、さらには発想を転換して新たな開発手法を導入するかの分かれ道にきている。

プラント CALS/STEP 実証事業プロジェクト（1996～1997 年）では、「STEP を含めたプラント関連国際標準は、データ交換には有効であるが、プラント･ライフ･サイクルの各フ

ェーズにおけるデータ交換が主体であり、プラント･ライフ･サイクル全般に亘る情報共有のための標準ではない」という結論であった。現状の STEP は、2.3 節で述べたようにデータ交換のレベルであり、情報や知識の共有のレベルに進化する必要がある。

STEP は技術情報の標準化という基本的に重要で難しい事業を行ってきた点では高く評価されるべきであるが、従来の方法論を見直し、上記課題を解決する新たな方法論を導入すべき時点にきている。

3. Web の進化と技術知識基盤構築

2 章では STEP が知的資産としての価値があることを評価し、STEP の現状の課題について述べた。そして、新たな開発方法論を導入すべきと指摘した。3 章ではどのような新たな方法論があるかについて検討する。

新たな方法論として、IMS（Intelligent Manufacturing Systems）の VIPNET（仮想企業体ネットワーク）プロジェクトの中で進めてきた、情報や知識の共有を狙いとする技術知識基盤（テクノインフラ）の構築についてその概要を説明する[3-10]。

また、Web の進化に伴い、世界的にも情報や知識の共有化の重要性が認識され、テクノインフラと同じような考えのプロジェクトが始まっている。次世代の Web 技術として、注目されつつある Web サービス[11]やセマンティック Web[12]がそれであり、以下ではこれらの概要についても説明する。

3．1　テクノインフラの構築

企業はグローバル化や大競争の進展に伴い、産業構造の変革や業務の革新などの本質的な問題に直面している。今後、企業が生き残るためには、自らのコア･コンピタンス（事業の中核となる製品や技術･サービス）を明確にし、自社にない製品や技術･サービスなどは、他の企業や機関のコア･コンピタンスを活用することが重要となっている。

一方、インターネットの普及により、情報や知識は国境を越え、瞬時に世界を駆け巡ることができる。今や研究開発から設計・生産、運用・保守まで、国境や地域を越えて最適な組み合わせを追求することが可能となっている。

製品や技術･サービスなどのライフサイクルに亘り、情報や知識を蓄積･共有･活用･創出できるテクノインフラを構築し、コア･コンピタンスの最適な組合せを可能にする仮想企業体を実現できる環境の整備があらゆる意味で重要な課題となっている。

このようなテクノインフラを構築するためには、製品や技術･サービスなどのライフサイクルに亘る、情報や知識を蓄積･共有･活用･創出するための理論的な体系化が必要不可欠である。また、ライフサイクルの過程では、情報や知識の変更や新規追加などができる機構が必須である。

IMS-VIPNET プロジェクトでは、このような要求を満たすモデルとして GPM（Generic

Product Model）を開発してきている。GPM は、あらゆる製品や技術･サービスなどのライフサイクルに亘る、情報や知識を蓄積･共有･活用･創出することを支援できるモデルで、従来のデータ構造やデータベースなどの考え方を革新する技術でもある。

GPM は、いろいろなデータ構造の基本要素（オブジェクトとオブジェクトの関連（アソシェーション）という要素）を抽出したもので、この基本要素を組み合わせることにより、複雑な現実世界を記述しょうとするものである(図 3. 1 参照)。また、GPM は述語（動詞）を中心にした、意味ネットワークモデルであり、自然言語の構文（ステートメント）も記述できる重要な特徴がある。

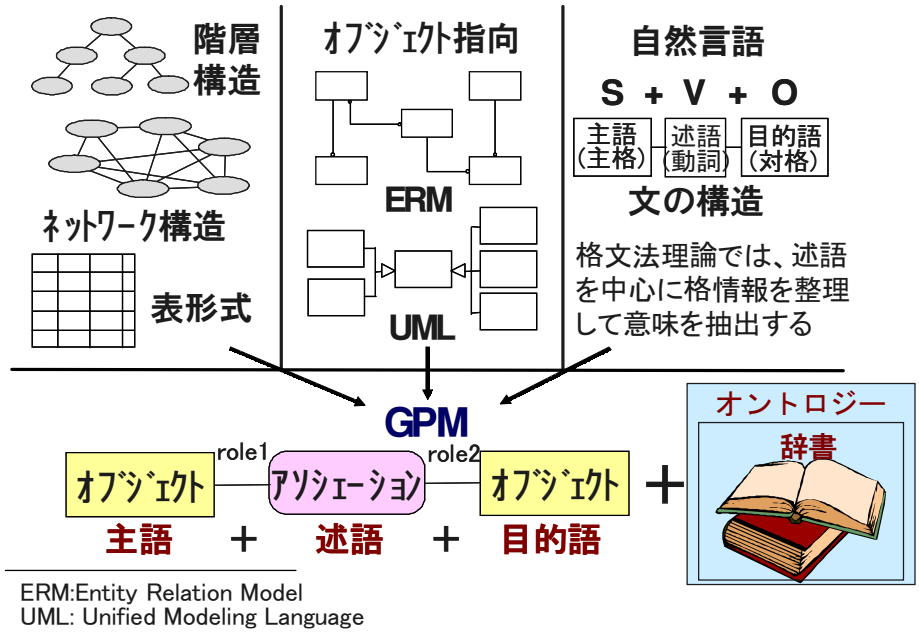


図 3.1　GPM の原理

GPM は概念モデルで、オブジェクトやアソシエーションの内容は、辞書で規定する。また、実際の情報や知識の内容（インスタンス）は、GPML（Generic Product Model Language）や GPM-XML（eXtensible Markup Language）と呼ぶテキスト形式で記述する。

辞書では、記述対象となる“もの（オブジェクト）”の構成要素やその分類についての用語（クラス）や、そのプロパティ（形状、材質、物理単位など)、アソシェーション（述語）を定義する。構成や分類、所有などの関係は、アソシェーションで記述する。現状では、STEP の用語を中心に、形状や物理単位などの共通化できる用語を約 2,000 語、プロセス･エンジニアリング分野に関する約 2,000 語の整備が進んでいる。他の分野に適用するには、その分野固有の辞書を追加することになる。この辞書を豊富にすることにより、知的資産の相互運用性（Interoperability）が可能となる。

テクノインフラでは、既存システム・アプリケーション層、情報や知識を蓄積する内部構造層（GPM、GPML、GPM-XML、GPM データウェアハウス)、情報や知識を共有する Web サービス層（エージェント技術)、情報や知識を活用するテクノインフラ適用層（オン

トロジー技術、データ・マイニング技術、安全評価技術、環境への影響評価技術）の４階層からなるアーキテクチャーを考えている（図 3.2 参照）。これにより、内部構造を GPM で共有化でき、Web サービスを共通化することができる。この共有化が情報管理のプロセスをドラマチックに簡素化することになる。

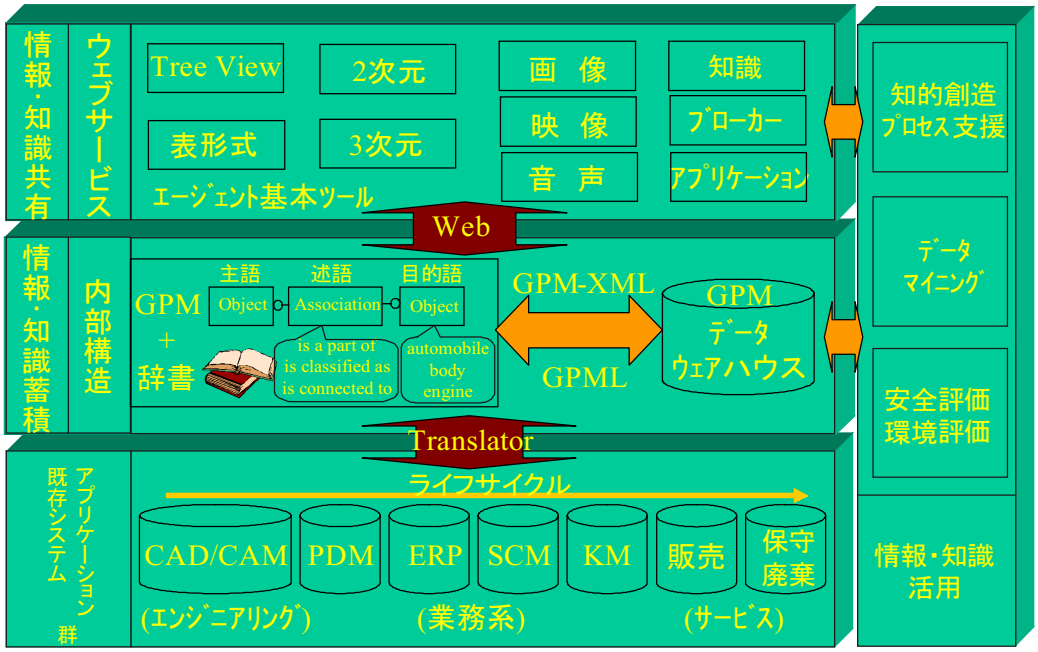


図 3.2　テクノインフラのアーキテクチャー

このアーキテクチャーに基づき、基本機能を開発し、プロセス･エンジニアリングの一部に適用し､情報や知識を蓄積･共有する実験を行い､アーキテクチャーや GPM､GPM-XML､データウェアハウス、エージェントの有用性を確認している。図 3.3 は既存の CAD システム等で設計された、P&ID（Piping & Instrument Diagram）や３次元 CAD 情報、施工図情報などを GPM-XML 形式に変換するトランスレータを用いて､データウェアハウス内に蓄積し、インターネットを介してデータウェアハウス内の情報をエージェント基本ツールである Web サービスで検索･閲覧したものである。辞書で規定した用語（製品を構成する用語）を表示①し、これをベースとして、ライフサイクルに亘る情報や知識を検索･閲覧できる。例えば、①から系統名をピックして、P&ID を表示②し、次にその系統の３次元情報を表示③し、さらに、③のある配管ラインをピックして、その配管ラインに関する施工図を表示④し、また、別の配管ラインの保守情報を表示④することが容易にできる。勿論、機器や部品の仕様も簡単に検索･閲覧できる。

テクノインフラのベースとなる、GPM やその実装手段は、あらゆる産業や技術分野に適用でき、将来への変更や拡張が容易で、柔軟性があると言える。これは、一般の製品のライフサイクル支援システムが抱える多くの問題を解決できる。

本来、テクノインフラはあらゆるシステムに非依存であるべきであり、このことは、IT

（情報技術）の急劇な進展に伴う陳腐化にも対応できる。また、学術などの研究分野においても、研究成果などを蓄積･共有･活用するテクノインフラとしても期待でき、研究開発の効率的な推進が図れ、人類共通の知的資産として活用できる。

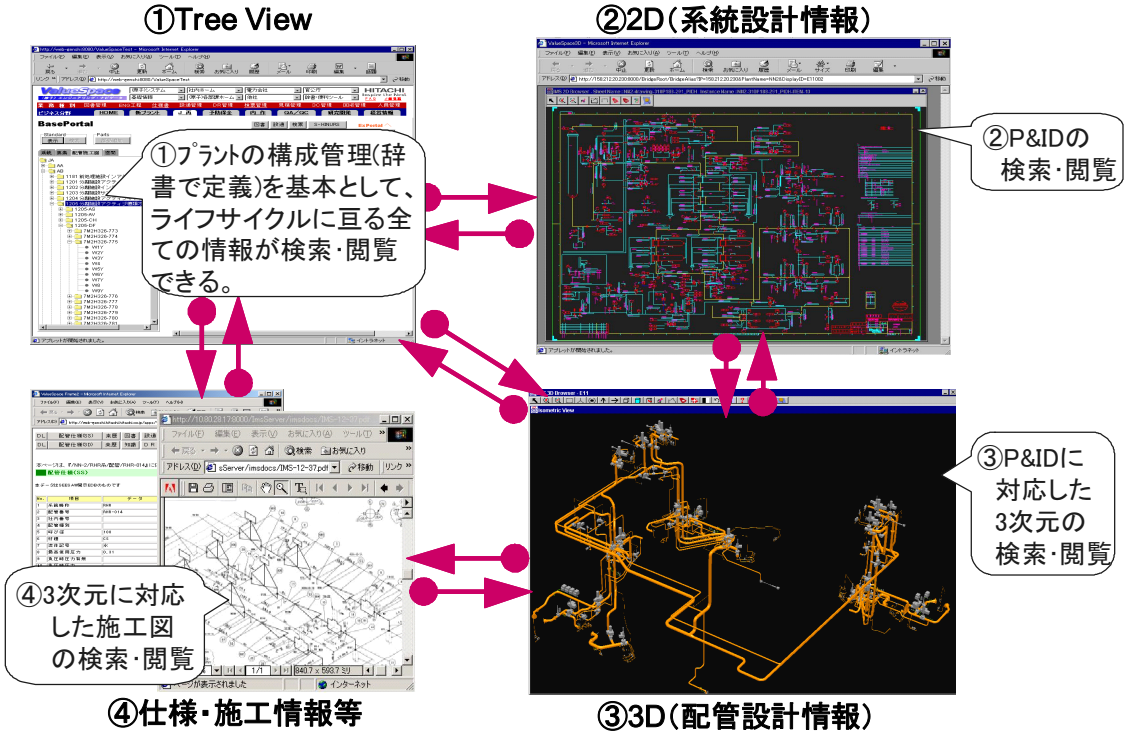


図 3.3　Web サービスによる情報や知識の検索･閲覧

3．2　Web サービス

従来の資産（ハードやソフト）を所有する情報システムの運営、保守費用は年々増大している。例えば、多くの企業では、データサイロ（データが孤立した状態）をなくすため、既存アプリケーションの連携や Web 化に大きな投資をしてきた。また、膨大な投資をして、ERP（Enterprise Resource Management）システムを導入したが、柔軟性のない組織（企業サイロ、蛸壺ともいう）になってしまったという報告もある[11]。

これに対して、Web サービスは、インターネット上でサービス･プロバイダーが情報技術を提供し、ユーザーは、ただそれを利用するというもので、資産の所有を前提にしないものである。これによって、今後予想される、従来型の情報システムの運用･保守費用の増大に対して、低価格なサービスを提供しょうとするものである。

Web サービスは、従来の IDC（Internet Data Center）や ASP（Application Service Provider）などの概念をさらに進化させたものである。マイクロソフトは“ドットネット”、オラクルは“ネットワーク･サービス”、IBM は“ウェブサービス”、サンマイクロシステムズは“オープンネットワーク環境”とそれぞれ呼び方は違っているが、Web 上でのサービス提供のことを言っている。

多くの人は、またかと懐疑的である。ただし、従来と重要な違いは、これらの企業が新しいITアプローチ環境のために必要なインフラストラクチャーを作るのに、大量の投資を行っていることである。１～２年に亘り、これらの努力を続け、新たなインターネット･ベースのサービスの安定した流れができ、伝統的な内部システムに対して、大幅なコスト削減を提供することができるようになり、企業間の協業についての新たな機会を提供することになる。ゆっくりだが、確実に、ITについての古い見方を修正することが求められる。

Web サービスは、インターネットで構成されたオープンな世界である。ユニークな内部システムを構築し運営する代わりに、企業はそれらに必要な機能、例えば、データ記憶装置、処理パワー、または具体的なアプリケーションなどを外部のサービス･プロバイダーから借りることになる。

Webサービスの３階層のアーキテクチャーを図3.4に示す。Webサービスでは、種々のアプリケーションの間で情報が容易に交換されることを可能にする XML や SOAP（Simple Object Access Protocol）、WSDL（Web Service Definition Language）[13]、UDDI（Universal Description, Discovery, and Integration）[14]などの通信プロトコルやソフトウェア標準がベースになる。これらのツールは、アプリケーションが他のアプリケーションと自由に接続し、それらから電子メッセージを読むことを可能にする Web サービスの共通語を提供する。テクノインフラと同様に、この標準が情報管理のプロセスをドラマチックに簡素化する。

サービス･レイヤーは、プロトコルと標準をベースにし、共有できるユーティリティの集合を提供する。例えば、インターネットの上のトランザクションやビジネス機能によって、請求や支払を第三者に委ねることなどを可能にする。さらに、サービス･レイヤーは、ユーティリティのセットから成り、通常サード・パーティーが提供、管理する。それは、メッセージの送信、入手可能なサービスの識別、および信頼性と整合性の保証などを容易にする。要するに、サービス･レイヤーは、（１）Web がユーザーとプロバイダーを見付けることやお互いを接続させるなどの Web サービスのユーザーを手助けする役割、（２）重要なビジネス業務を実行するのに必要な信頼された環境の提供、などの主要な役割を果たす。頑強なサービス･レイヤーは、Web サービスの潜在的なインパクトを加速、拡大する。

アプリケーション･レイヤーは、アプリケーション･サービスの多様な配列から成り、CAX（Computer Aided Design, or Engineering, etc.））、 ERP、SCM（Supply Chain Management）、SFM:（Sales Force Management）、CRM（Customer Relation Management）、KM（Knowledge Management）などのビジネスを支援する機能を提供する。このアプリケーション･レイヤーは日々、業務担当者、顧客、パートナーに見えるものとなる。いくつかのアプリケーション･サービスは、所有権を主張する特定の企業またはグループ企業のために、また他のアプリケーション･サービスは、すべての企業間で利用されることになる。あるケースでは、企業は自分自身のアプリケーション･サービスを開発し、新たな潜在的富をもたらす資源を創出する他の企業へ売ることも選択できることになる。

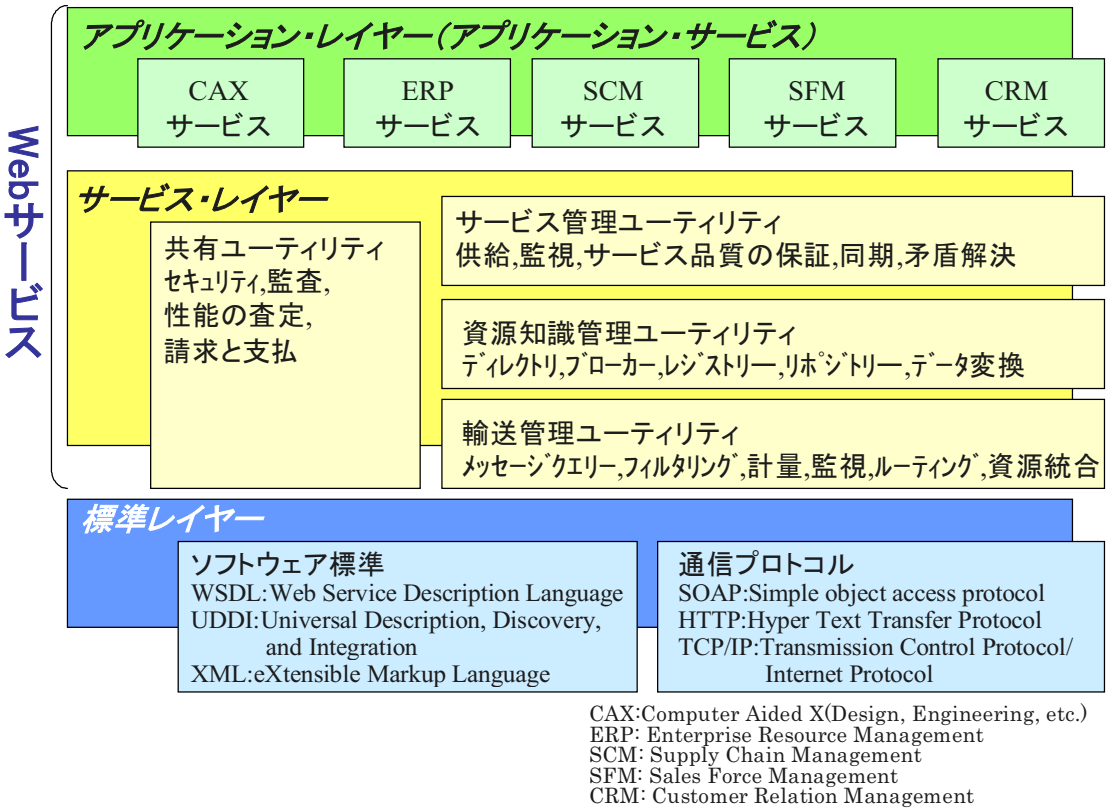


図 3.4　Web サービスのアーキテクチャー[11]

3．3　セマンテッック Web

セマンティック Web は、Web サイト上の情報にセマンティックス（意味）情報を付加して、人の代わりにソフトウェアで自動処理する新しい Web の世界を拓くと期待されている技術で、1999 年に Web の創始者である Tim Berners-Lee 氏により提唱されて以来、第二の IT 革命を起こす次世代の Web 技術の有力な候補として注目を浴びている[15-19]。

セマンティック Web は、「ただの表示目的ではなく、コンピュータにより、様々なアプリケーションを横断して、データの再利用や統合、自動化できるようにするため、Web 上でデータを見つけたり、データ間を関連付ける」というビジョンである。

セマンティック Web では、Web コンテンツの意味やその関連を定義するため、XML 上に、RDF（Resource Description Framework）[20]やオントロジー語彙集（Ontology vocabulary）などからなるアーキテクチャーを提案している（図 3.5 参照）。

XML は、タグ付き言語と呼ばれるもので、データに自由なタグを付けることができ、そのタグによってデータに意味を持たすことができる。XML をローカルに使うなら勝手にタグを決めればよいが、利用者で共有しょうとすると、新たなルールが必要になる。利用者が勝手にこのタグを決めたら大変な混乱になる。

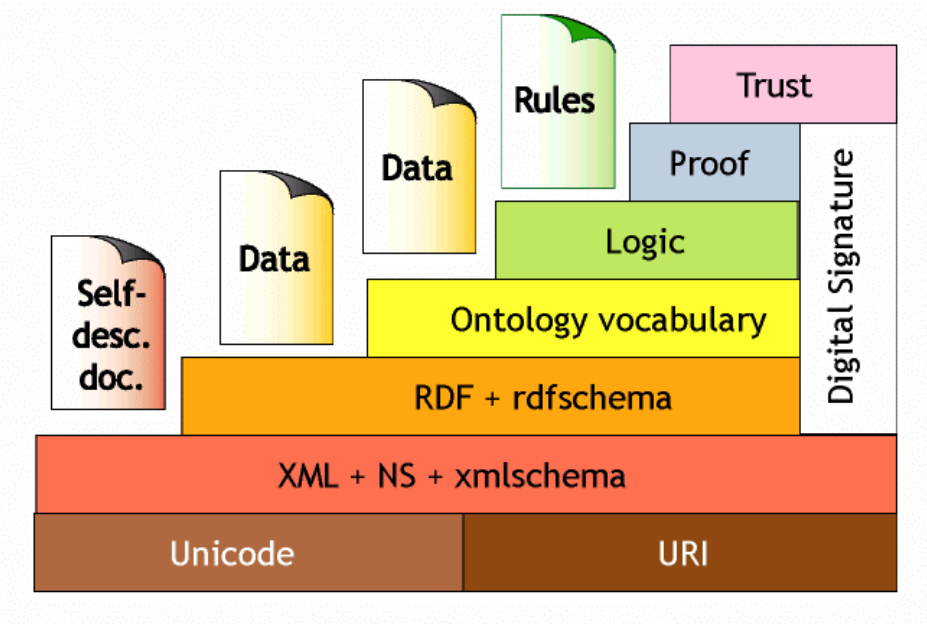


図 3.5　セマンティック Web のアーキテクチャー[12]

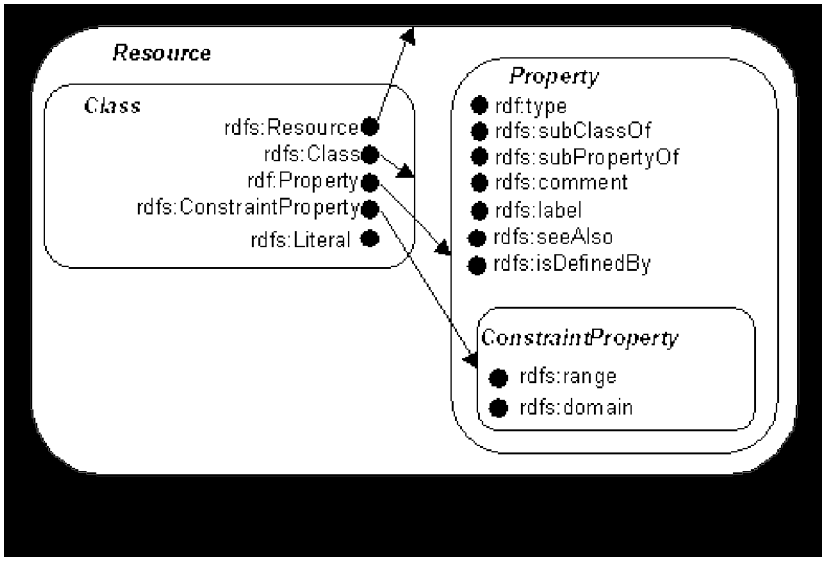
セマンティック Web では、この問題を解決するために、タグをオントロジー語彙集として標準化することを狙っている。セマンティック Web では RDF という、メタデータ（metadata）を記述するための汎用的な枠組みを利用する。RDF は Web のリソースの内容を表わすプロパティとその値（プロパティ値）を表現するための XML をベースにしたモデルで、W3C の標準である。元々は図書の分類を記述するために開発されたもので、ダブリン･コア･メタデータ（Dublin Core Metadata Initiative が開発した図書分類のためのメタデータ）[21]と組合せて利用することを進めている。RDF スキーマを図 3.6 に示す。

メタ（meta）とは、ギリシャ語の「（あるものに）添えるもの、付随するもの」を意味する言葉に由来するもので、メタデータとは、情報検索・収集に必要なキーワードを指し、データのためのデータということになるが、タグであることに変わりはない。要するに、RDF はタグの使い方に制限を加えたということになる。

セマンティック Web では、RDF より上位のオントロジー語彙集は、OWL（Web Ontology Language）[22]で規定する言語で記述する。オントロジーとは、元来は哲学の用語で、「ある」ということの意味を問う学問分野だったが、「知識を積み上げる」ための「概念及び概念間の関係の明確な定義の集まり」というような意味で使われている。

OWL のベースである DAML＋OIL[23,24]は、DARPA（Defense Advanced Research Projects Agency）からの資金援助を受けた DAML（DARPA Agent Markup Language）プロジェクトで開発されていたオントロジー記述言語 DAML と、欧州の IST（Information Society Technologies）プログラムからの資金援助を受けた On-to-Knowledge プロジェクトが開発した OIL（Ontology Inference Layer）を統合した言語である。DAML＋OIL は、RDF をベースとして、クラス（概念）やプロパティに対する制約の記述能力をさらに向上させた言語で

あり、述語論理で意味を定義することを特徴としている。



RDF や OWL を使えば、より複雑な情報を Web 上に掲載でき、それをコンピュータが的確に解釈し、自動処理できることになる。いわゆるエージェントに命令を与え、そのエージェントがインターネット上のリソースにアクセスし、自動処理するということが現実に近づく。Web サービスが実現するアプリケーションも、さらに高度なものになると考えられる[14]。

セマンティック Web は、2001 年２月に、Web 関連の標準化機関である W3C（WWW コンソーシアム）の活動として承認され、現在、最も注力しているテーマである。

セマンティック Web については、80 年代の人工知能の知識表現システムに似た高度で複雑なモデルであるとして、セマンティック Web の実現性を問題視する意見も少なくないが、ゴールが示す世界は素晴らしく、セマンティック Web の実現に向けて、Web は進化を続けるとものと考えられる。

4. STEP と Web 技術の融合

ここでは、STEP の知的資産としての価値を継承し、新たな発展をめざす方法論を検討する。

4．1　STEP とテクノインフラとの融合

3.1 節のテクノインフラのところで述べたように、既に、STEP の形状（Part42）に関する用語を中心に約 2,000 の用語を辞書として整備している。また、アプリケーション分野としては、プロセス･エンジニアリングの分野では、AP221（Functional Data and Schematic Representation for Process Plants、系統設計モデル）や AP227（Plant Spatial Configuration、プ

ラント空間構成モデル）、ISO15926（Oil & Gas Facilities）を中心に約 2,000 の用語を辞書として整備している。

他の産業分野に適用するためには、対象となる分野のライフサイクルに亘る用語辞書を整備することが必要となる。既に AP として開発できていれば、その用語をまとめて用語辞書を作ることで対応できる。GPM-XML では、XML の名前空間を定義できるので、アプリケーション分野ことの名前空間を用意することにより、用語間の衝突は回避できる。

テクノインフラでは、ドメインの用語をそのまま使うものであり、STEP の課題にあった、ARM から統合リソースへのマッピングや相互運用性の問題は起こらない。また、辞書を豊富にすることにより、知的資産を将来的に拡張できるメリットがある。

4．2 STEP とセマンティック Web との融合

STEP とセマンティック Web との融合方法としては、以下のように二つの方法が考えられる。

（1）現状の STEP の枠組みをそのまま継承し、セマンティック Web に融合する方法

この方法では、以下について検討することが必要である。

・EXPRESS スキーマを RDF スキーマに変換

・インスタンスを RDF スキーマに基づき RDF（Property Graphs）に変換

・自動変換する方法

この方法は、2.8 節で述べた、ISO10303 Part28（XML Binding）の XML スキーマを RDF スキーマに置き換えることにより、比較的容易に実現できる可能性がある。RDF Binding として、STEP の規格の一つの分冊としては成立する。従来の STEP の枠組みをはずれるものではなく、従来の STEP グループの理解は得られやすい。STEP の EXPRESS と RDF スキーマの対応関係を表 4.1 にまとめる。

表 4.1　STEP の EXPRESS と RDF の対応関係

STEP (EXPRESS)	セマンティックWeb, RDF
ENTITY, TYPE	Class
SUBTYPE	subClassOf
NUMBER, REAL, INTEGER, LOGICAL, STRING	Datatypeとして定義
ARRAY LIST SET BAG	Datatypeとして定義
ENUMERATION	one of
SELECT	disjoinUnionOf

しかしながら、この方法では STEP の情報をセマンティック Web の世界にあげることはできるが、アプリケーション･システムから見ると、STEP の枠組みが残ることになり、アプリケーション･システムから直接、セマンティック Web の世界へあげる方法に比べれば、無駄が多く、非効率であるといえる。STEP の本質的な課題を解決するものではない。

（2）STEP のよいところとセマンティック Web のよいところを融合させる方法

従来の STEP の成果を生かし、アプリケーション･システムとセマンティック Web の両方向のやりとりができる方法で、以下について検討することが必要である。

- RDF スキーマで記述（EXPRESS に依存しない）
- ARM のエンティティやプロパティを RDF のクラスで記述
- 統合リソースのエンティティやプロパティ（不要なものは捨て、必要なものを）を RDF のクラスとして記述
- インスタンスを RDF スキーマに基づき RDF（Property Graphs）で記述（アプリケーション･システムとセマンティック Web との両方向のトランスレータを用意する）

この方法は、4.1 節の方法と似ており、STEP の従来の枠組みをすべて捨て、セマンティック Web の RDF に統合化する方法（図 4.1 参照）であり、STEP グループの理解を得ることは難しいように思える。

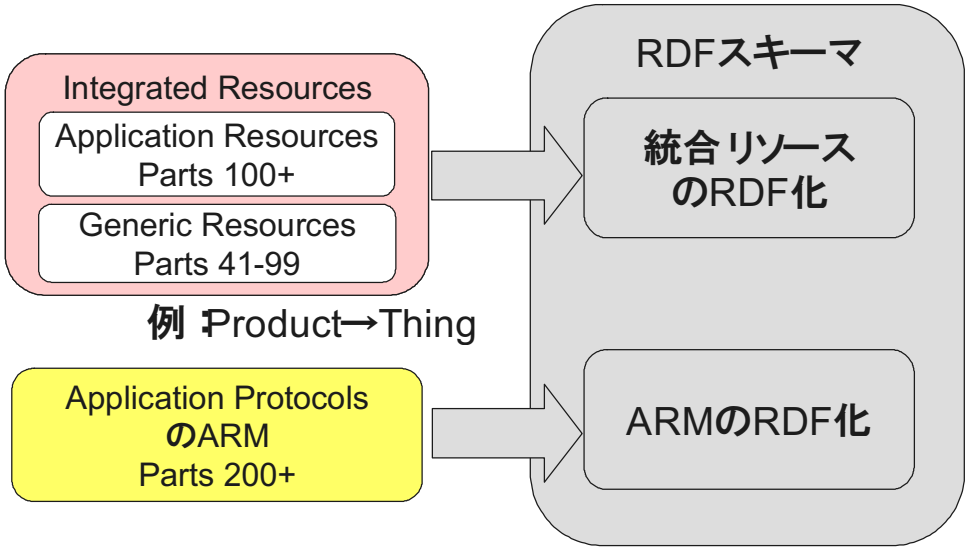


図 4.1　RDF スキーマをベースとする方法

しかしながら、この方法は STEP の枠組みを介さず、アプリケーション･システムとセマンティック Web との両方向のやりとりができる効率的な方法である。また、クラスの追加は容易であり、将来の拡張に向けた柔軟なシステムが構築でき、インターオペラビリティも解決できる可能性が高い。

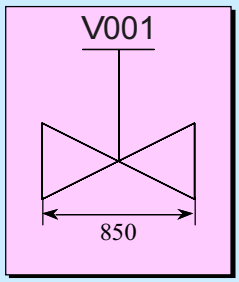
4．3 制約自然言語の提案

STEP も先取りの標準と言われてきたが、いつの間にか色があせてきた。XML についてもいつまでブームは続くのであろうか。セマンティック Web は究極のモデルだろうかという疑問が湧いてくる。STEP のねらいは、人と機械や機械と機械のスムーズなコミュニケーションを実現することであった。人と人のコミュニケーションは言葉（音声）や文章、絵や図面、ジェスチャー、造形などの媒体を介して行われている。絵や図面、造形などの媒体によるコミュニケーションを機械に理解させることは相当に難しい。言葉（音声）や文章などの自然言語によるコミュニケーションも相当に難しいが、機械翻訳の研究が長年行われてきており、自然言語を理解するための辞書などの整備もかなり進んでいる[25,26]。人と機械や機械と機械が自然言語でコミュニケーションできればすばらしい。日進月歩で変化している自然言語そのものの理解は相当に難しいが、言語の構造や語彙を制約することにより、かなりのことが実現でる。そこで、「究極のモデル」として、自然言語の使い方を制約した「制約自然言語」（Formal Natural Language）を提案する。これと STEP の知的資産を融合する方法を検討する。

制約自然言語の例を図 4.2 に示す。図 4.3 は図 4.2 の日本語の内容を英語にしたものである。これはバルブの記述例で、カタログなどに書いてあるのは自然言語形式での表現である。このような自然言語形式は、人間には理解し易いものであり、コンピュータでもこのような表現形式を理解できると便利である。

図 4.2、図 4.3 で、V001（固有名詞）や 350mm（特性値）は、ユーザにより決められるもので、実体（インスタンス）である。バルブ、調整弁、面間寸法などは、プラントを構成する要素の用語で、概念（クラス）である。また、「～です（is a）」「～に分類される（is classified as）」「～を所有する（possess）」などが概念間、概念と実体、実体間の関連を表わす述語である。

私（V001）は*バルブ*です。（以下の情報を持っています。）
私（V001）は*調整弁*です。
私（V001）は*呼び径*200Aの特性を持っています。
私（V001）は*呼び圧*20Kの特性を持っています。
私（V001）は*面間寸法*350mmの特性を持っています。
私（V001）は*フランジ形式*の*接続端*を持っています。
私（V001）は右図のような*形状*をしています。
私（V001）はSCPH11で作られています。
私（V001）は230万円です。
私（V001）の*製造番号*は6789です。
私（V001）は日本バルブで作られました。



形　状

ｱﾝﾀﾞｰﾗｲﾝ文字:ﾕｰｻﾞｰ定義(インスタンス)、太文字:用語(述語)、ｲﾀﾘｯｸ文字:用語(概念(クラス))

図 4.2　制約自然言語の例（日本語）

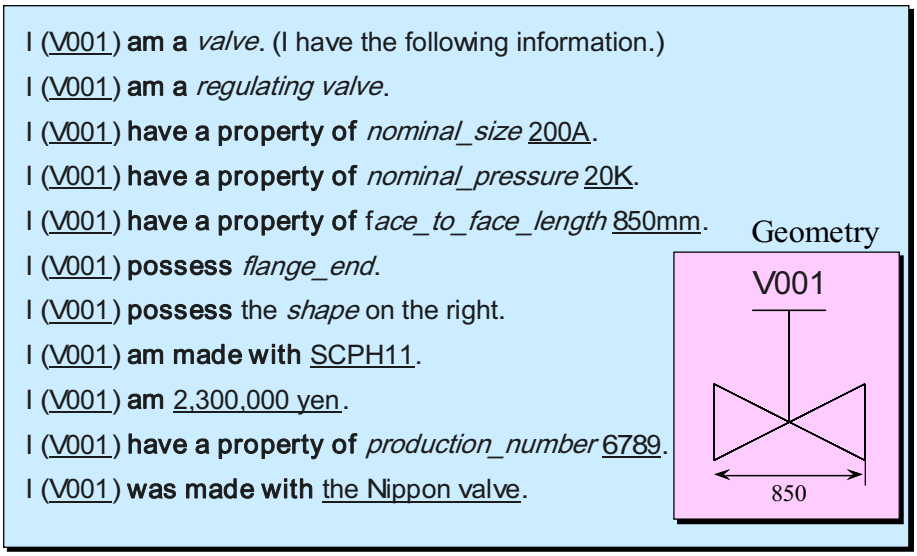


図 4.3　制約自然言語の例（英語）

制約自然言語の構造を図 4.4 に示す。制約自然言語の基本構造は、主語＋述語＋目的語で、主語や述語、目的語の内容を辞書で規定する。辞書は大きく概念（クラス）の辞書と述語の辞書に分けられる。辞書は日本語とし、英語、中国語、韓国語などの多言語に対応できる構成にする。

この提案は、セマンティック Web と同様、新たな標準開発に対する合意を得ることは難しい。しかしながら、日本発の技術知識基盤として、e-Japan の基幹に成り得る技術であり、世界に発信する価値は大いにある。制約自然言語については、次章でさらに検討する。

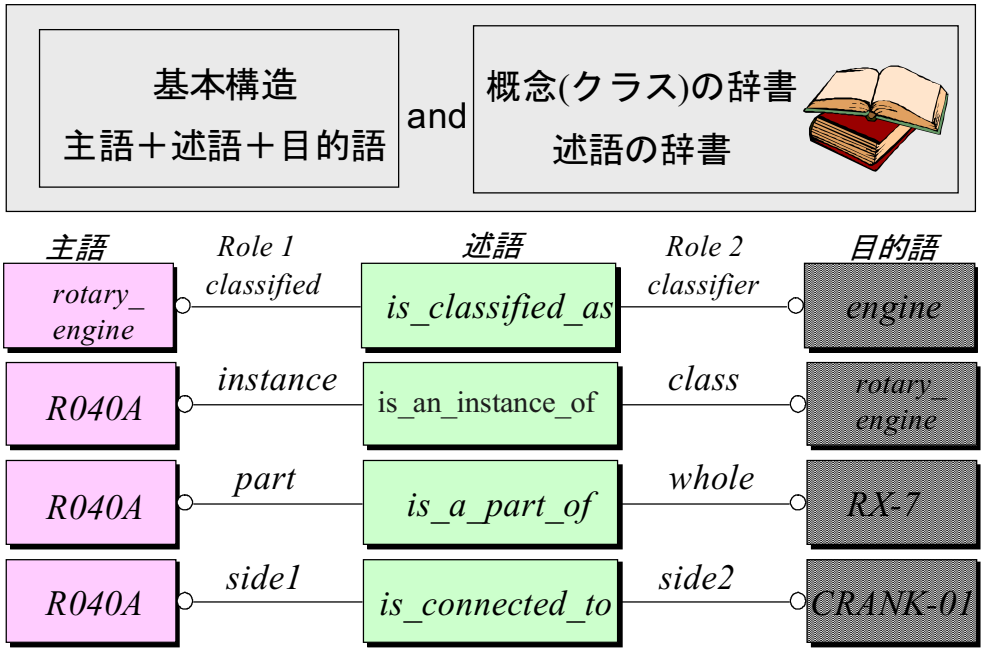


図 4.4　制約自然言語

5. オントロジーとしての制約自然言語

本章では、制約自然言語について、オントロジーの面から詳細に検討する。

5．1　オントロジーとは

従来、ファイルやデータベースが知的資産の管理対象であったが、インターネットの爆発的な進展に伴い、Web が知的資産の管理対象として重要性を持ってきた。現状のインターネットは、広告の手段であるが、インターネットの特徴の一つはオープン性であり、人類の共通の情報や知識基盤としての価値を持つものとなって行く。このような知識基盤では、情報や知識の共有が重要であり、Web 間で共通の表現構造や語彙が必要となる。このような統合化のためのオントロジーを定義する必要がある。

オントロジーとは、言葉の意味を明示的に記述し、言葉の間の関係を表現したものである。暗黙の内に共有している概念を明示的に定義することで、言葉が持つ多義性を排除することができる。概念の明示的な定義では、矛盾がないように論理を組み立てる必要があるため大変な労力を要する。また、専門家を総動員し、知識を出し合い、合意を得ながら進める作業には大変な時間が必要となる。このような意味からも STEP は、オントロジーの一つであり、同様に大変なマンパワーを要している。

一方、言葉の意味は、その言葉が存在している文章や文書の集合体である。知識は言葉によって形式知化される。言葉の意味に関するオントロジーがあれば文章の言葉を切り出すことにより、グループ化や階層化を図ることができる。

オントロジーは哲学や人工知能、自然言語処理といった様々な分野で研究が進められており、オントロジーの定義はいまだに研究者によって議論が続いており合意された定義はない。しかしながら、情報科学においては、「対象とする領域における概念の属性と包含関係について首尾一貫した定義を与え、それを人間と計算機の両方が理解できる形式で記述したもの」と定義されている[27]。哲学的視点は、言葉をより一般的に取り扱おうとしているだけで、本質的なねらいは同じである。

以下では、オントロジーがどのように構成され、どのような特徴を持ち、どのような機能を持つかについて検討する。

5．1．1　オントロジーの構成

オントロジーは以下のような手順で記述される[28]。

（1）概念の切り出し

（2）概念の分類、階層的表示

（3）概念間の関係記述

（4）形式的定義

まず、対象となる世界に存在する概念を集める。集められた概念は様々な関係に基づい

て組織化される。多くのオントロジーでは、このような概念と概念の関係が述語論理などを用いて、公理の形で形式的に記述されている。
例として、図 5.1 の「積み木の世界」に関するオントロジーを作成してみる[29]。この世界はテーブルの上にいくつかの積み木が載っている世界で、記述するために必要な概念を切り出す。この例では、積み木の A、B、C と TABLE の４つで、図のように分類でき、階層的に表現できる。積み木やテーブル以外にもそれらをまとめる“もの（thing）”という概念を導入している。

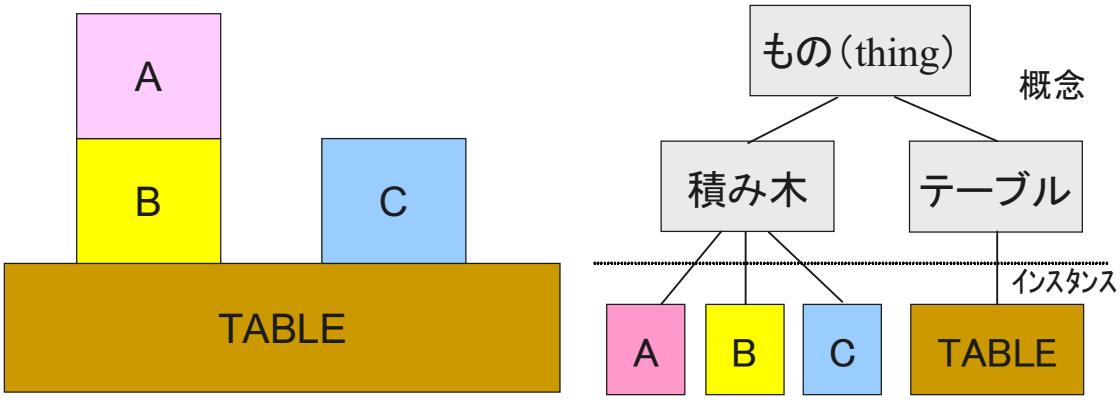


図 5.1　積み木の世界とその概念の分類

次に概念間の関係を記述する。上下関係に注目した場合、“A は B の上にある”、また、“B は TABLE の上にある”、“C は TABLE の上にある”という関係が記述できる。また、“A の上には何も無い”、“C の上には何も無い”という記述もできる。“上にある”や“何も無い”も「積み木の世界」を記述するため概念である。そして、ものｘがものｙの上にある状態を ON(x, y)、ものｘの上に何も無い状態を CLEAR(x)と形式的に表現した場合、概念の間の関係は ON(A, B), ON(B, TABLE), ON(C, TABLE), CLEAR(A), CLEAR(B)と記述できる。このように記述すると「積み木の世界」は以下のように形式的に表現することができる。

$$ON(A, B) \wedge ON(B, TABLE) \wedge ON(C, TABLE) \wedge CLEAR(A) \wedge CLEAR(C)$$

上記の２つの項から、三段論法で ON(A, TABLE)という命題を導ける。

５．１．２　オントロジーが持つ性質

オントロジーが持つ性質としては以下のようなものがある [28]。

（１）目的依存性
（２）普遍性（一般性）
（３）完全性（定義はなるべく必要十分条件がよい）
（４）共通性（合意）
（５）全体性（対象の包括的記述）
（６）網羅性（対象領域のカバー）

（７）体系性・一貫性（無矛盾な推論ができること）
（８）最小性（モデル対象をできるだけ少ない公理で表現すること）
（９）安定性
（１０）形式性（計算機理解可能性）
（１１）推論可能性
（１２）部分性
（１３）明瞭性･客観性（オントロジーは客観的な定義と言語による正確な意味をもった語彙を提供すべき）
（１４）部品性
（１５）拡張性（既定義の用語を変更せずに新たな一般語や特殊語が導入できること）
など。

オントロジーをどのような目的で利用するかによって、出来上がるオントロジーは変わる。例えば、自然言語処理に携わるものにとっては“語彙”や“辞書”といった観点が重要視され、知識ベースの研究者にとっては“実行可能性”や知識ベース構築の“部品”という考え方に基づいたオントロジーが作成される。さらに、対象とする世界が変われば、オントロジーを構成する概念も変わってくる。こういったことから、オントロジーは目的依存性を持つと言われている。

また、オントロジーは一般性を持っていなければならない。万人が興味をもつものしかオントロジー構築の対象にしてはいけないというのはつよすぎる制限であるが、一人の人しか興味を持たない領域などに関するオントロジーは意味が無い。そして、そのオントロジーは多くの人の合意を得ているべきである（標準化）。

５．１．３ オントロジーの機能

オントロジーはその利用目的に合致した様々な機能を持っている。具体的には以下の機能があると言われている[2]。
（１）人にも計算機にも理解可能
（２）概念定義（語彙の提供）
（３）メタモデルの提供
（４）暗黙情報・知識の明示化
（５）設計意図の明示化
（６）知識の体系化
（７）標準化
（８）内容の論理化

自然言語処理を目的とする人は、オントロジーに語彙の提供という機能を期待するだろう。オントロジーは対象とする世界を記述する際には厳密に定義された標準的な語彙を提

供する。

また、計算機上でのモデル化を行う目的でオントロジーを扱いたい人は、オントロジーが持つメタモデル的機能に期待する。オントロジーは概念とそれらの間に成立する関係を明示的に規定したものであり、これらの概念と制約の元で構築されたモデルによって様々な問題解決を試みることが可能となる。

５．１．４ オントロジーの記述レベル

オントロジーの記述レベルには、次の３つに分けられる [30]。

レベル１：概念の切り出しと階層的記述

・例：Yahoo オントロジー，Metadata タグ

レベル２：意味定義の明示化･形式化

・豊富な関係定義公理の利用

・推論可能性（自分自身）

・例：多くのオントロジー

レベル３：構築されたモデル実行に関わる計算論

・モデルの推論可能性

・例：タスクオントロジー

５．２ 制約自然言語

５．２．１ 制約自然言語のねらい

制約自然言語のねらいは、あるドメイン（世界）の情報や知識を体系的に積み上げ、蓄積･共有･活用･創出できるようにする（機構化）技術を確立することであり、まさしくオントロジー技術である。

ドメインを記述するためには、

（１）ドメインについての共通の認識・理解（概念の共有）

（２）ドメインを明示的に記述する方法（モデル）

が必要である。

制約自然言語は図 5.2 に示すように、ドメインの構成要素をオブジェクトとし、それらの関連を示す述語からなる集合体として明示的に記述しょうとするものである。

ドメインをさらに分析すると、オブジェクトは図 5.3 のように概念（クラス）と実体（インスタンス）に分けることができる。概念と実体は連係しており、is_an_instance_of（=is_a）なる述語で記述する。実体（インスタンス）の記述では、分類関係を表す述語 is_classified_as は現れない。

制約自然言語では、概念（クラス）や述語などを用語辞書として定義し、必要に応じて拡張できる仕掛けになっている。

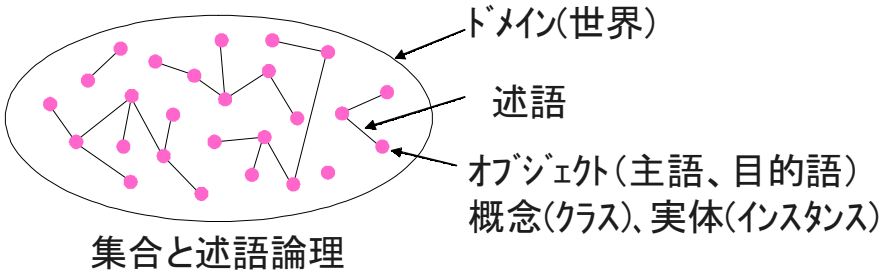


図 5.2　ドメインの明示的な記述

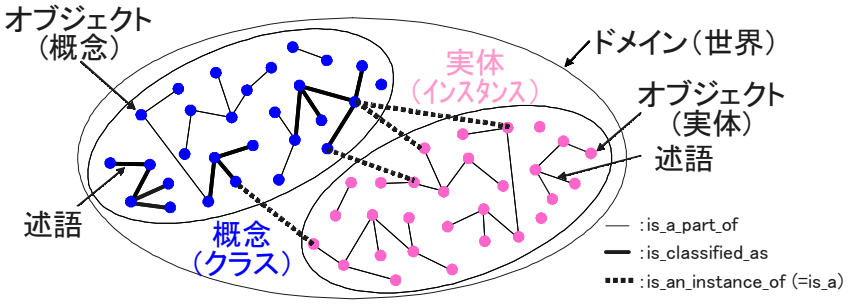


図 5.3　概念（クラス）と実体（インスタンス）

製品やサービスなどのライフサイクルにわたる情報や知識を体系的に積み上げるためには、対象とする製品やサービスなどについてのオントロジーが必要不可欠である。図 5.4 に示す例は、制約自然言語による人工衛星の構成管理を記述したもので、黄色い四角形がクラスで、水色の四角形が実体（インスタンス）で、六角形が述語で、is_a（=is_an_instance_of）はインスタンスとクラスを関連付ける述語である。また、構成は is_assemble_from なる述語で記述する。クラス間の分類には、 is_classified_as という述語を用いる。クラスには宇宙機の構成要素以外に、物理量や形状などを表す属性やアクティビティ、組織、述語などの用語を記述している。クラスは拡張でき、新たな概念や知識の追加が可能である。

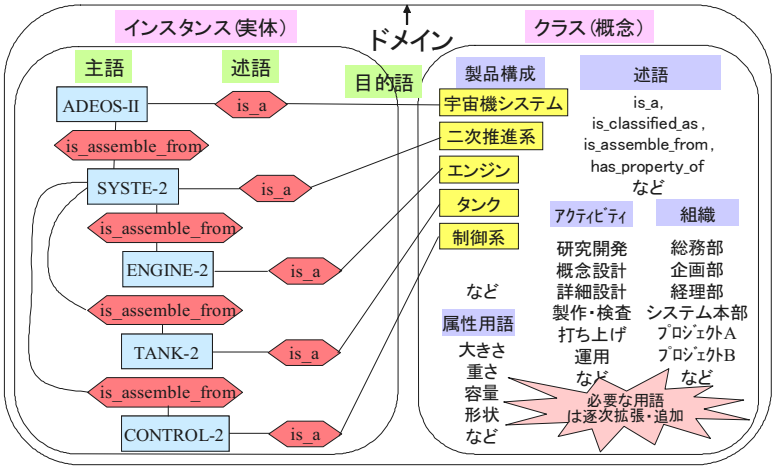


図 5.4　ドメインの記述例

５．２．２ 制約自然言語と述語論理

制約自然言語は上記の概念（クラス）と実体（インスタンス）および概念間、実体間、概念と実体間を関連付けるオントロジー･モデルである。制約自然言語は図 5.2.6 に示すように、オブジェクト間の多項関係を記述するモデルで、オブジェクト及びオブジェクト間の関連を示す述語で表現する非常にシンプルな情報･知識記述モデルである。制約自然言語は述語を中心とした、意味ネットワークモデルである。

制約自然言語をオントロジーとしてみると、述語を中心としたモデルであり、先に述べた「積み木の世界」と同様に、述語論理（Predicate Logic、表 5.1 参照）として取り扱うことができる。

数学だけではなくもっと広いく、日常生活の会話から科学の専門的議論にも適用できる証明の方法論というものを体系化する学問が論理学である[31]。証明とは、ある文章（命題）が真であることから別の文章が真であることを導き出す作業である（「証明」のかわりに「推論」「論証」ともいう）。一番良く知られた「証明」は、次のような「三段論法」である。

前提１：恐竜は絶滅した。
前提２：ティラノサウルスは恐竜である。
結論　：したがって、ティラノサウルスは絶滅した。

上記の例で、「恐竜」を「昆虫」と変え、「絶滅した」を「火を吐く」と変え、「ティラノサウルス」を「猫」に変えると、全体の証明は次のようになる。

前提１：昆虫は火を吐く。
前提２：猫は昆虫である。
結論　：したがって、猫は火を吐く。

この証明は正しいだろうか？下段の例は、前提と結論ともすべてが偽であるが、記号的な証明としては正しいと言わざるをえない。論理的証明は、内容ではなく単なる形式によって正しさが保証される。上記の例からも分かるように、真なる前提が手に入ったときに、そこから間違いなく真なる結論を導き出す手続き、それが「証明」である。

また、諸言語に現れた人間共通の思考のあり方を、どの特定の言語の語彙や文法にも縛られない諸言語共通の最小限のルールで記号化したものが論理学である。言語がどのようなものであろうと、そして世界で何が起こっていようと、確実に成り立つ命題の真偽の組み合わせのパターン化を体系化するのが論理学である[31]。

制約自然言語において、記述論理を適用する場合には、真なる命題の蓄積が重要となる。真なる命題から原理・原則を導くことは大いに可能性がある。これ以上の検討は今後の課題である。

表 5.1　述語論理

	用　　語	記号	意味
演算	積、連言（conjunction）	∩、∧	and
	和、選言（disjunction）	∪、∨	or
	含意（implication）	⊃、⇒	If-then
	同値（equivalence）	≡、⇔	equal
	否定（negation）	¬、～	not
量化子	全称（universal quantifier）	∀	all
	存在（existential quantifier）	∃	some exist

５．２．３ ドメイン定義の方法

対象とするドメイン（世界）の情報や知識を体系的に積み上げ、蓄積･共有･活用･創出できるようにするため、ドメインにおける構成要素（オブジェクト）を明示的に定義する必要がある。定義する方法としては、以下の方法がある。

（１）トップダウン：最初は、最も一般概念を定義し、それからそれらを展開する。

（２）ボトムアップ：最も具体的な概念を定義し、それからより一般クラスでそれらを構成する。

（３）組合せ：最初は、より顕著な概念を定義し、それからそれらを一般化させ展開する。

定義する上で、スコープを制限する必要があり、以下の事項について十分検討する必要がある。

(a) 始めからすべてを定義できない

・ローマは一日にしてならず

・マジックはない（No Magic）

・小さく始めて大きく育てる（Step by Step）

(b) 必要性の高い範囲から検討する

(c) Simple is Best

(d) 他のオントロジーを活用する

・相互運用性の検討が必要

制約自然言語は図 5.5 に示すように柔軟で、将来の拡張に対応できるモデルで、必要なところから記述し、情報や知識を逐次拡張することができる。

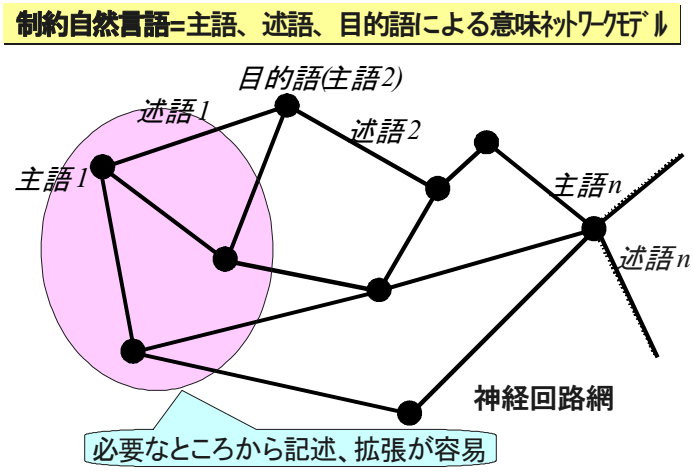
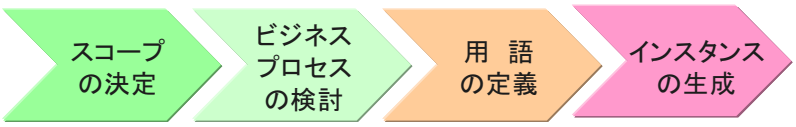


図 5.5　意味ネットワークモデル

５．２．４ ドメイン定義の手順

以下ではドメインにおける要素（オブジェクト）を具体的に定義する方法について検討する。ドメイン定義の手順を図 5.6 に示す。

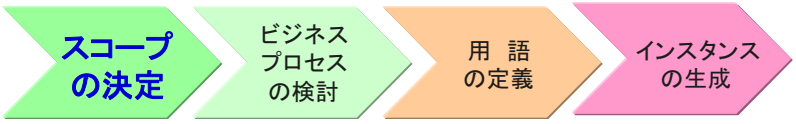


- ■スコープの決定
- ■ビジネスプロセスの検討
- ■用語（概念･述語･プロパティ･修飾語）の定義
- ■述語やプロパティ･修飾語の制約定義
- ■インスタンスの生成
- ■評価･検討（実際には繰り返しのプロセス）

図 5.6　ドメイン定義の手順

（１）スコープの決定

スコープの決定では、図 5.7 の項目について検討することが必要である。



- ■カバーするドメインは何か？
- ■何のために制約自然言語を利用するのか？
- ■制約自然言語における情報や知識が提供する解決策は？

図 5.7　スコープの決定

これらの質問への回答は、ライフサイクルの間に変わる可能性もある。

（2） ビジネス･プロセスの検討

ビジネス･プロセスでは、図 5.8 の項目について検討することが必要である。

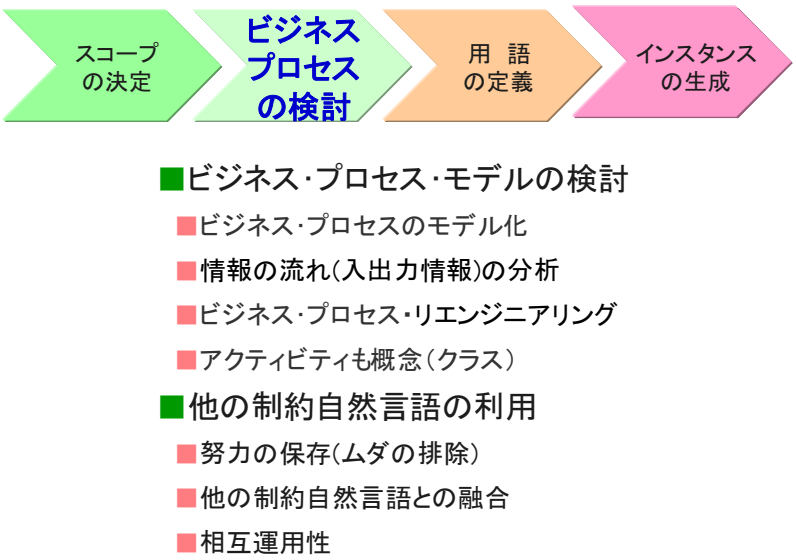


図 5.8　ビジネス･プロセスの検討

（3） 用語の定義

用語の定義では、図 5.9 の項目について検討することが必要である。

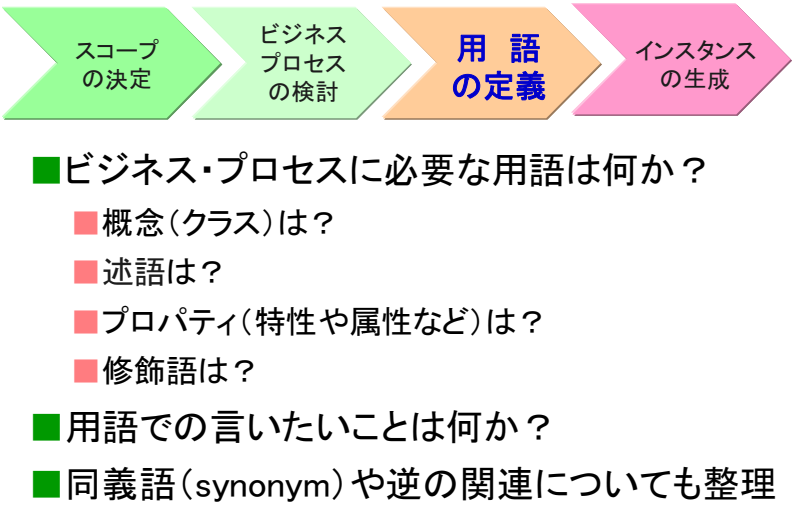


図 5.9　用語の定義

用語の辞書化では、以下の項目について検討することも必要である。

・用語を文書化することは重要

・概念（クラス）の定義と共有

・用語の辞書化

・用語定義の自然言語による記述

・同義語リスト、多言語対応

・用語間関連の図式化：リレーショナル・ダイヤグラム（図 5.10 参照）

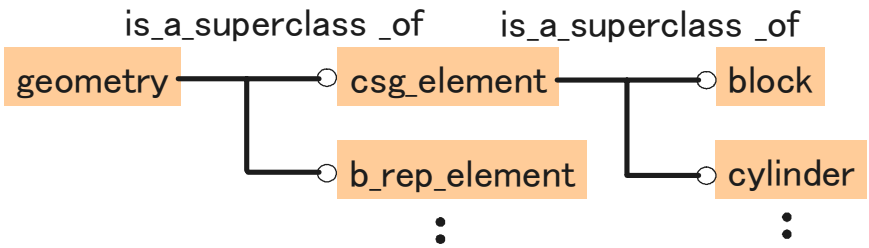


図 5.10　リレーショナル・ダイヤグラム

(a) 概念（クラス）の定義

概念（クラス）定義の例を図 5.11 に示す。

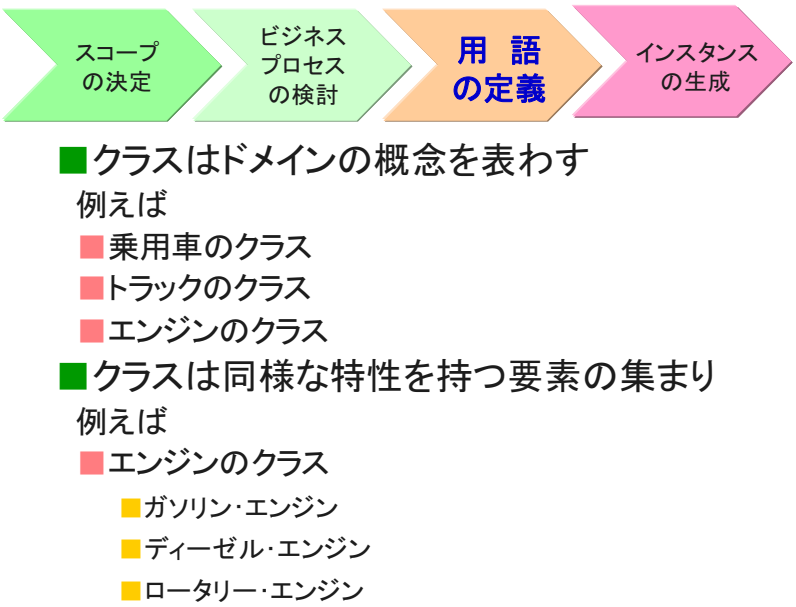


図 5.11　概念（クラス）の定義

(b) 述語の定義

述語定義の例を図 5.12 に示す。

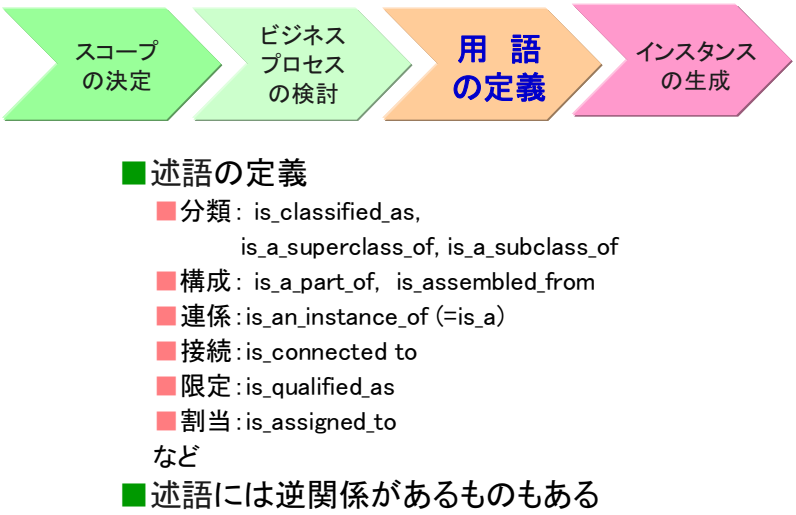


図 5.12　述語の定義

(c) プロパティ（特性、属性など）定義

プロパティ（特性、属性など）定義の例を図 5.13 に示す。

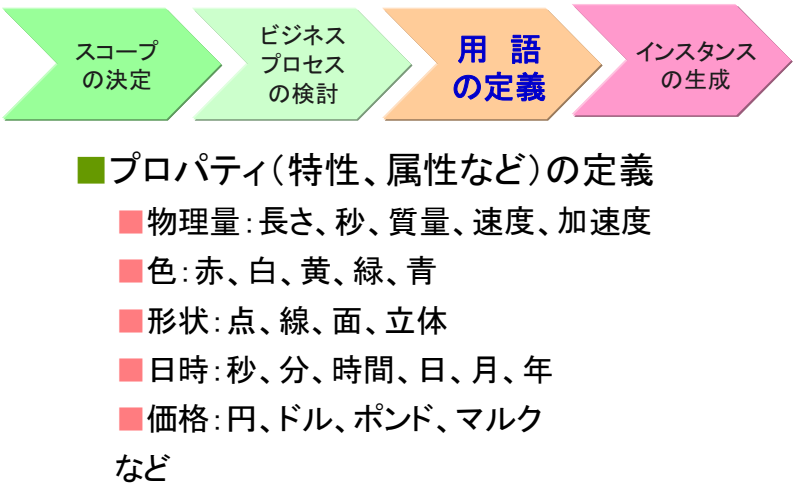


図 5.13　プロパティ（特性、属性など）定義

(d) 修飾語の定義

修飾語（プロパティの限定子）定義の例を図 5.14 に示す。

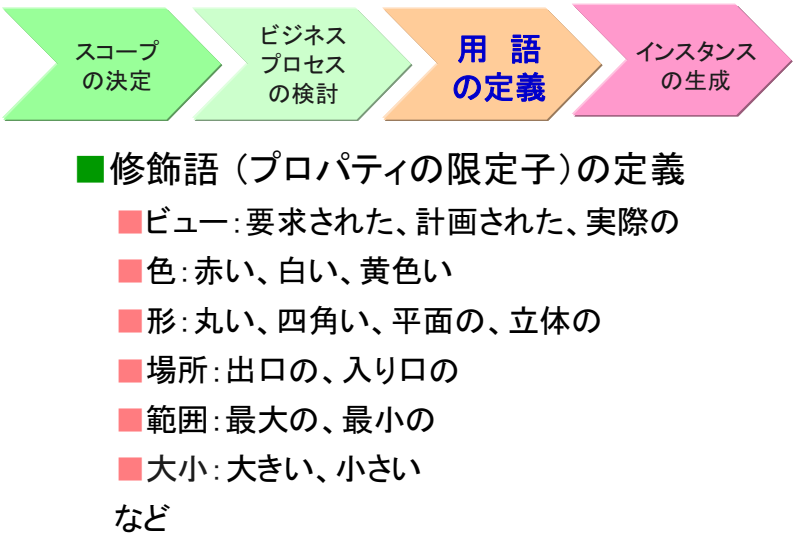


図 5.14　修飾語の定義

（4）インスタンスの生成

インスタンスの生成では、図 5.15 の項目について検討することが必要である。

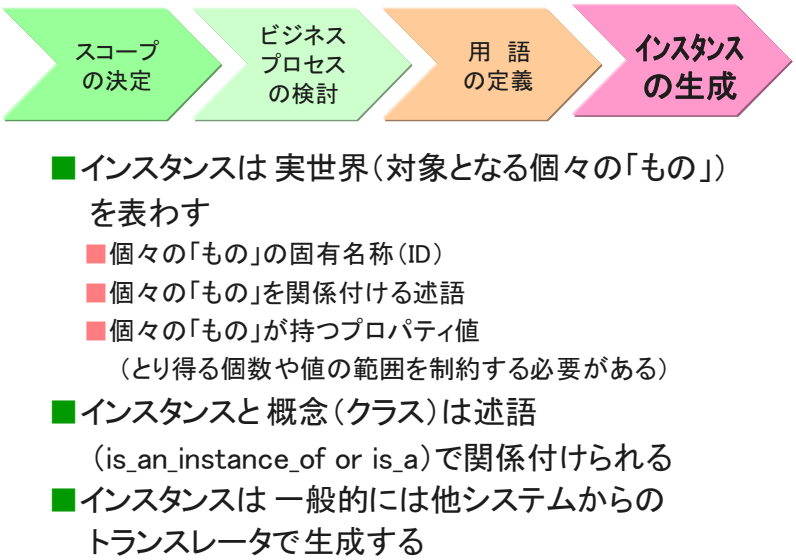


図 5.15　インスタンスの生成

以上のドメインの定義をまとめると、図 5.16 のようになる。オブジェクトは、インスタンス（実体、実際のフィールドにおける設備名称や数値を表す）とクラス（概念）に分類する。述語は、クラス間、インスタンス間、インスタンスとクラス間の関連付けの役割を

果たす。

クラスと述語は、辞書で規定する。すなわち、分野ごとの辞書を用意することになる。述語では、データ構造の主要な概念である分類や構成、接続関係などを用語として規定する。

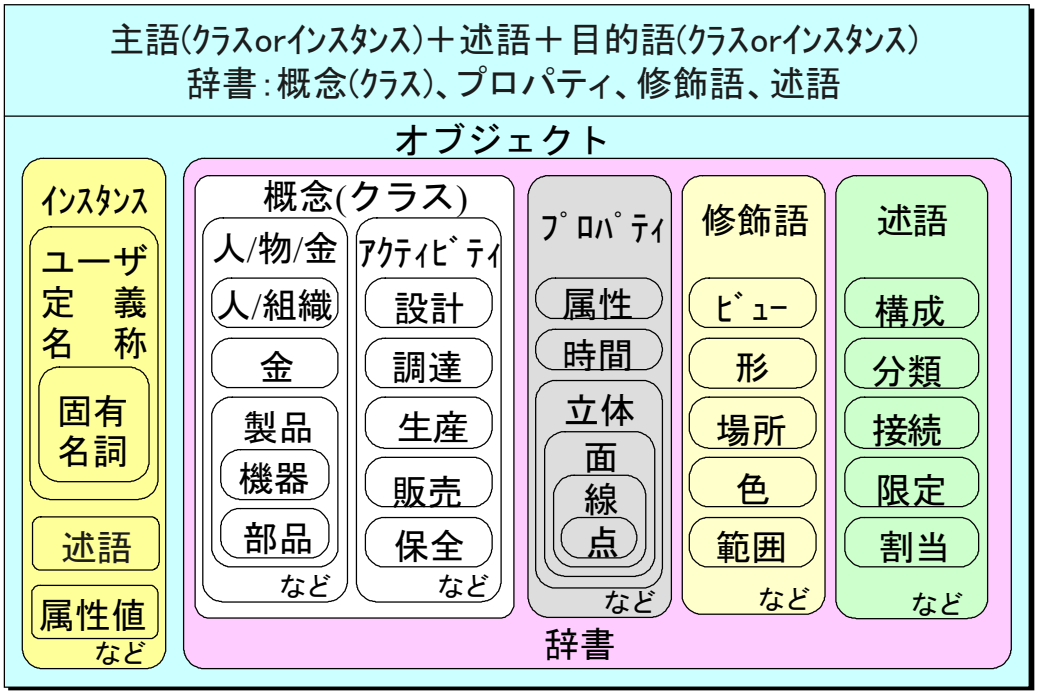


図 5.16　制約自然言語における辞書とインスタンス

5．3　STEP と制約自然言語

STEP と制約自然言語との融合は、基本的には、4.1 節で述べた方法と同様になる。STEP の ARM と統合リソースは、概念（クラス）、プロパティ、修飾語、述語に整理し、辞書を作成する。統合リソースから作成した辞書を共通辞書とし、アプリケーション分野は分野毎の辞書を作成する。用語間の衝突がなければ一つの辞書としてまとめる。

今後、開発する分野については、初めから概念（クラス）、プロパティ、修飾語、述語を定義し、辞書に追加する。

ユーザーは、ユーザー自身の情報システムの情報内容を辞書の用語を用いて制約自然言語の形式に変換する。他のシステムで作成された制約自然言語形式の情報内容を辞書に従い解釈し、自分のシステムに取り込む。

ユーザーがパートナーとの間で独自の辞書を作り、情報共有することも可能となる。ただし、この場合、第３者は独自の辞書の部分の内容は理解できないが、共通辞書の部分については理解可能であり、理解できる部分を自分のシステムに取り込める。

標準化という視点で考えると、辞書の維持や追加･変更などの管理組織や制約自然言語を使い易くするための環境作りなどを今後検討しなければならない。

6. おわりに

「ものつくり」の技術知識基盤として、従来の STEP の知的資産を生かし、さらに発展させるため、人にも機械にも理解できる制約自然言語と STEP の融合を提案した。

制約自然言語の基本的な発想は、IMS-VIPNET プロジェクトで開発された GPM であるが、GPM を自然言語に近づけたいとの思いからである。一方、欧米発のセマンティック Web というといろいろな人が興味を示すが、それよりも前から、日本の中で、GPM と唱えてもなかなか理解して頂けないもどかしさを感じてきた。日本発の技術をみんなで育てるという活動が今、求められている。流行を追いかけるのではなく、本質的な問題を解決すべき時点にきている。

従来の STEP/CALS の流れと、Web 技術の進化であるセマンティック Web、さらには、従来の基幹システムを変革する Web サービスの流れが融合して、新たな大きな流れを生み出そうとしている（図 6.1 参照）。これはまさしく、第二の IT 革命の始まりである（図 6.2 参照）。今後は、安全性の高い多様なネットワークが形成され、さまざまな活動シーンにおいて、時間や場所の制約を超えて、必要とする情報や知識を誰もが意識しないで簡単に活用できる「ものつくり」の技術知識基盤が実現できる。その先はユビキタスという、いつでも、どこでも、だれでもが高度情報化社会（情報技術により、「人を幸せにする社会」）の恩恵を享受できる環境が実現のものとなる。

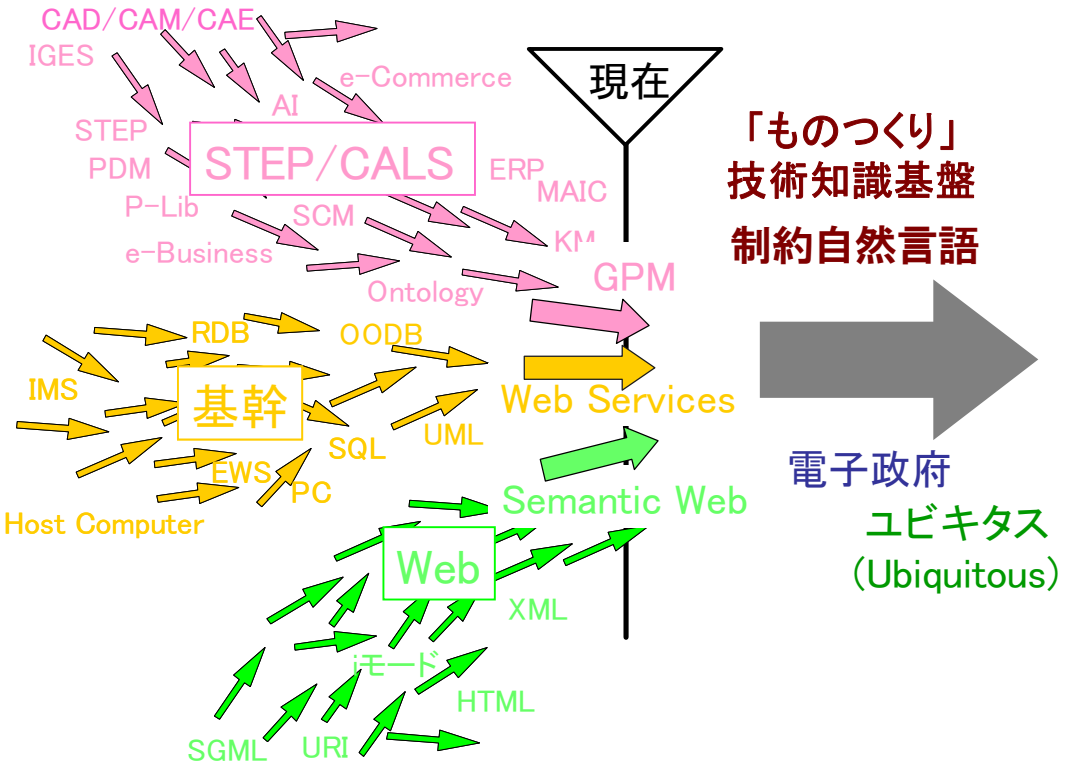


図 6.1　情報･知識共有の大きな流れ

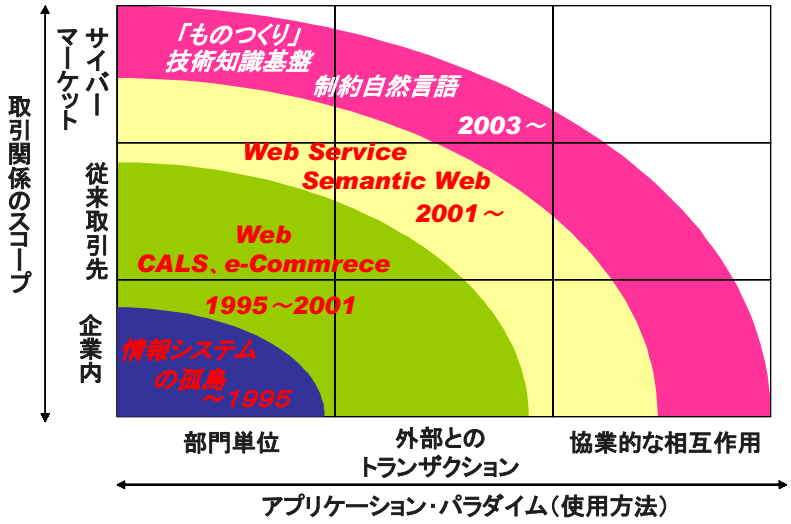


図 6.2　第二の IT 革命

参考文献

[1]NIST SP939 STEP The Grand Experience, 1999.07

[2] http://www.nist.gov/sc5/soap

[3]仲: 技術情報基盤構築へ向けて, 化学工学会, 62 巻, 4 号(1998)

[4]仲: 情報技術と化学技術, 化学工学会, 64 巻, 5 号(2000)

[5]化学工学会 Process Engineering 特別研究会・Process STEP 専門委員会偏:活動報告 1996-1998(1999)

[6]太田,仲ほか:プラントデータウェアハウスモデル：GPM(Generic Product Model)の開発,CALS/EC Japan 1999

[7]太田,好永: IT による情報・知識の共有と蓄積-技術情報基盤(テクノインフラ)の構築-, 日本機械学会誌, 103 巻, 977 号(2000)

[8]太田,好永:エンジニアリング情報化の潮流, 配管技術, 42 巻,11 号, 日本工業出版(2000)

[9]好永,太田,仲ほか:仮想企業体ネットワーク, IMS 研究成果報告会論文集(2001)

[10]太田,仲ほか:技術情報基盤のための情報・知識記述モデル, IMS 研究成果報告会論文集(2001)

[11]John Hagel and John Seely Brown: Your Next IT Strategy, Harvard Business Review October 2001

[12]http://www.w3.org/2000/Talks/1206-xml2k-tbl/

[13]http://www.w3.org/TR/wsdl

[14]http://www.uddi.org/

[15]Tim Berners-Lee,他: The Semantic Web, SCIENTFIC AMERICAN May 2001(日本語訳:自分で推論する未来型ウェブ,日経サイエンス 2001 年 8 月号)

[16]日経 IT プロ, 記者の眼, 2001.09.19 号

[17]次世代インターネット Semantic Web と STEP の将来, CEO-ECOM-IIMMC 合同フォーラム(2001)
[18]セマンティック Web コンファレンス予稿集, 慶応義塾大学 SFC 研究所(2001)
[19]ライフサイクル支援のための XML 技術, 精密工学会(2001)
[20]http://www.w3.org/RDF/
[21]http://dublincore.org/
[22]http://www.w3.org/TR/owl-ref
[23]http://daml.semanticweb.org/
[24]http://www.ontoknowledge.org/oil/W3C/
[25]http://www.jsa.co.jp/EDR
[26]ttp://www.ipa.go.jp/STC/NIHONGO/IPAL/ipal.html
[27] T. R. Gruber: A translation approach to portable ontology, Knowledge Acquisition, Vol.5, No.2, pp.199-220, 1993
[28] 日本情報処理開発協会:大規模知識ベースに関する調査研究—オントロジー工学に関する調査研究—報告書、1999
[29] www.jaist.ac.jp/library/thesis/ks-master-2002/paper/t-yagyuu/paper.pdf
[30]溝口理一郎: オントロジー研究の基礎と応用, 人工知能学会誌 Vol.14, No.6 (1999)
[31]三浦俊彦: 論理学入門, NHK ブックス (2000)

仮説：STEP はセマンティック Web のキラーアプリになる

委 員　綾 日天彦

1．プロローグ

E-Japan第１弾の成果として、2005年に日本は世界で最初にブロードバンドと携帯電話とユビキュタスコンピュータが融合した情報環境が整備された国になると予想されている。ユーザの多様性に合わせ高齢者･非熟練者であっても誰もが多形態化するコンピュータを、何時でも何処でも自由に使いこなせる社会環境が整ってきた。次世代情報環境では、コンピュータ利用のパラダイムが変る。

Webの基盤技術XMLと製品モデルのISO標準STEPは、ネットワークとコンピュータとソフトウェアとコンテンツを明確に分離するアーキテクチャを決める標準として捉えることが必要である。コンピュータはネットワークで結ばれ、ソフトウェアで処理が行われる。コンテンツは、この３層の上に独立して存在することになる。各層が独立性を保つことは、Webが進化を続けて行くための必要条件である。

XMLの進化したセマンティックWebとISO標準STEPの統合化は、日本の強さを活かせる課題である。「情報伝達の媒体が図面から情報モデルに変る日STEPはセマンティックWebのキラーアプリになる」という仮説について、以下に論点を整理する。

2. CAD の進化と STEP

2．1．CAD の進化

１９７０年代、対話方式の図形処理技術が生まれ、１９８０年代、CAD を利用した製図コストが手作業による製図コストより低くなって CAD が商品として新たな市場が誕生した。2003 年現在、市販の３次元 CAD は、詳細設計・生産準備段階の業務合理化に必要不可欠なツールとなっている。

上流の概念設計段階と下流の生産管理・運用保守から廃棄・再利用にいたる段階の業務は、依然として専用アプリケーションを開発して対応することが必要な段階にある。２１世紀の製造業の新たな命題は、地球環境との調和である。このため省資源・省エネルギー・リサイクルが重要な課題になっている。製品開発の最初の段階で、シミュレーションによって製品の機能予測をすることが必要となっている。次世代 CAD 製品は、この問題を解決する機能を持つことが課題である。

2．2．STEP の基本問題

STEPは1984年に、情報伝達の媒体を図面から情報モデルに変えるというビジョンをもって先取型標準としてスタートした。何もないところに論理的な枠組みを作って、次世代IT技術開発を先導するという新しいかたちの標準化として始まった。1994年に第1版がISO標準規格になったが、市場標準にはなっていない。

STEPの伸び悩みは、基盤となる理論と実装技術の両方が研究開発段階にあることを成熟技術と誤認していたことにあると考えられる。3次元CADおよびPDMの分野では、標準が確立し公開されている。しかし、実用化の視点から見ると、試行段階に止まっている。ビジョンとソフトウェア実装技術の乖離が大きく、具体的メリットをエンドユーザが実感できないことに原因がある。

現在、市販のCAD/CAM、PDM、CAE製品は、それぞれ固有の領域をもっている。当面の課題が、これらの統合化であり、次の課題が3次元デジタル模型として情報モデルを共有することであり、将来の課題が、データと情報を形式知に変え知識を共有し、創造的な知的生産性を向上させることである。図1は、経営的視点からCADの進化を予測したものである。

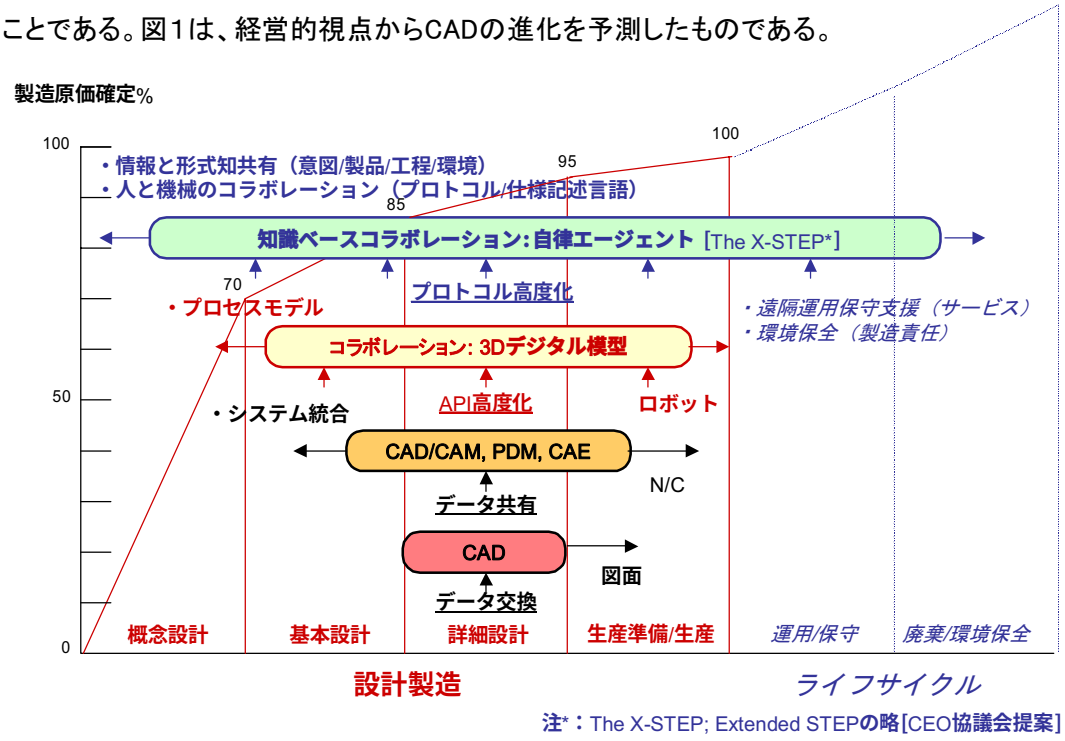


図 1 CAD の進化

オブジェクト指向技術は、システム構築技術として工学的規範性に基づく強力な枠組みを提供する。しかし、モデルを記述するための数学的論理性についての規定はない。モデル内部の属性情報・因果関係情報の記述が利用者に任されている。オブジェクト指向技術ですべてが解決できると考えて開発が行われたことが、90年代多くの技術開発プロジェクトが成功しなかった主要な要因である。

モデル内部の記述を数学的に許される変換の集合にすると、情報の生成は自動化できる。米国および欧州では基礎となる数学理論の重要性に気づいている。21世紀に入って、Formal Method Program（数学と論理学を基盤としたソフトウェア開発方法論）として、産学協同の戦略プ

ロジェクトが始まっている[1)]。

3. Web の進化とセマンティック Web

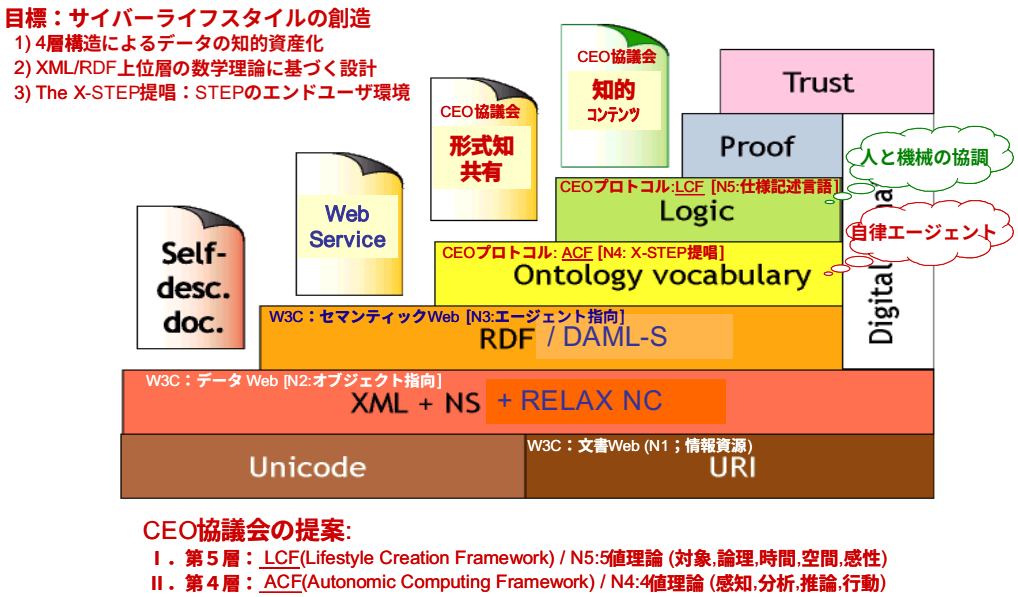
3. 1. Web の進化

インターネットは、創造と進化の最中にある。Webには、網の目状に何ものにも制約されない方法で、様々なアイディアを統合できるパワーがある。「これまで無関係と思われていた情報を相互に連携することができる機能を開発すれば、コンピュータは人間の知的創造性を支援することができる」というのが、Webを主導しているTim Berners-Leeの基本理念である[2)]。

いろいろな種類のコンピュータが協調して機能できる作業空間ができると、日々変化し更新されつづけている最新の情報を、いつでも誰でも共有できる情報環境がでる。Webの進化によって、データ自体がどのような形で管理されていても、それらをXMLに変換し、インターネット上で取り出し可能とする限り、問い合わせ言語によって統一的な検索が可能な段階に入っている[3)]。

3. 2. セマンティック Web のビジョン

インターネット環境でコンピュータは、計算する機械から人と人とのコミュニケーションを媒介するメディアとしての機能をもつ機械に進化すると予測される。Webは、単なる情報流通の基盤から、Webサービスとしてサービス流通の基盤に発展している。2001年7月、第一回セマンティックWeb国際シンポシウムが、スタンフォード大学で開催された。巨大な分散された情報を意味構造で連携させることにより、人と機械の協調が知的コンテンツを媒体として実現できる。Webは言語メディアに進化する。インターネット環境は、産業社会のインフラになり、人々のライフスタイルを変えて技術とビジネスの環境を変える[4)]。



CEO協議会の提案:
Ⅰ. 第５層： LCF(Lifestyle Creation Framework) / N5:5値理論 (対象,論理,時間,空間,感性)
Ⅱ. 第４層： ACF(Autonomic Computing Framework) / N4:4値理論 (感知,分析,推論,行動)

図 2 CEO 協議会の提案

図2は、Tim Berners-Leeの基本理念を拡張して、STEPのエンドユーザ環境を実現するためのCEO協議会の提案を示したものである。STEPはセマンティックWebの環境で、情報モデルを共有するための中核になると予想される。

4. STEP はセマンティック Web のキラーアプリになる

4. 1. XML と STEP の統合

XMLの本質は、Web上の多様なデータの内容と形式を定義するための言語である。どんな内容でも、人間とコンピュータの両方可読にし、文書処理とデータ処理を統合化する基本仕様についての合意である。現在、既存システムのデータ統合と既存出力系のフォーマット変換に威力を発揮している。Webサービスとして異種アプリケーション連携の枠組に進化し、既存システム統合化プラットフォームの地位を確立している。次の技術課題は、論理･時間･空間･感性の数学を基盤とした意味構造の標準表記法の開発と、文書処理･データ処理の統合化である。

STEPの本質は、モデルの情報完備性を保証することである。どんな角度でとりだしても整合性をもつモデルの標準表記法を決めることが規格開発の目標である。産業構造改革の視点から見ると、情報伝達の媒体を図面から情報モデルに変えることに意義がある[5)]。現在、主要な3次元CADベンダは、データ交換標準としてSTEPインターフェースを提供し始めている。PDMのデータ交換標準として効用を実証し、STEP-NCプロジェクトではXMLインターフェースを開発して情報伝達の媒体を図面から情報モデルに変えることに成功している。次の技術課題は、セマンティックWebとの相互運用性の確立である。

4. 2. セマンティック Web のキラーアプリ

セマンティックWebは、Webをコンピュータによる意味処理に適用しようとする長期戦略的な提案である。誰がどのように高度に複雑なモデルの開発を行うかが問題で、成果を直ちにビジネスに応用することは、困難であらうと予想されている。幸いなことに、STEPでは既に複雑な情報モデルが開発されている。また、次世代CADの重要課題が一意性を必要条件とする仕様記述言語の開発である。STEPは、セマンティックWebのキラーアプリになると予測する根拠である。

システムが巨大化すると要求が複雑化するので、一意性を持って要求仕様を曖昧さなく記述することは、困難度の高い仕事になっている。システム開発の実態は、要求事項を自然言語で書いた仕様書を、プログラマーがプログラム言語で、コンピュータに理解できる命令のかたちに書き直すことである。仕様記述、コーディング、コンパイル、リンクを行って実行オブジェクトをターゲットにロードし、実機デバッグを行うというのが一般的なシステム開発の手順である。

要求を仕様書にする段階、仕様書をプログラム言語に書き直す段階のいずれにも間違いが生じる可能性が存在する。現在のシステム開発環境には、連続性がなく、それぞれの工程間で連携が取られていないため、要因を上流工程に自動的にフィドバックすることはできない。要求定義、仕様書、その他の詳細な記述法を統一することによって、保守も関連する文書も最新の状況を維持することが可能になる。

仕様記述言語の第一の要件は、利用者、要求提示者、仕様書記述者およびシステム開発者とコンピュータの相互の円滑な意思疎通を実現することである。第二の要件は、一意性を持つ曖昧さのない記述を可能とすることである。数学理論に基づいた形式言語を使用すれば、正確さが保証されることになり、検証作業は自動化できる。[6] 難易度の高い数学理論は、コンピュータ内部で自動的に処理することになる。

4．3．IT技術革新のロードマップ

通信とコンピュータの技術革新が進展し、価格性能比が10^2倍以上になると、コンピュータ利用のパラダイムは変る。2005年、e-Japan計画が完成すれば、10^3倍技術による通信回線が全国的に整備され、低コストで高品質の動画像情報の双方向通信が可能な情報通信基盤が整う。コンピュータのハードウェアは、半導体製造技術に90nmルールを採用する高機能マイクロプロセッサの製品化のよって、10^2倍技術が実用化すると予想されている。文書処理とデータ処理の統合化を実現するための必要条件は整うと予想される。

次世代CADの基盤技術である仕様記述言語の開発には、強いユーザニーズがある。ユビキュタス時代へ対応するためのソフトウェアの生産性向上は、国家戦略の課題である。STEPのエンドユーザ環境の開発によって、セマンティックWebのキラーアプリが生まれると考えられる。

5．X-STEPの提案

5．1．製品ライフサイクル管理への対応

STEPが目標としている製品の全ライフサイクルを通しての情報モデルを共有は、次世代CADの必要条件である。これを実現するためには、セマンティックWebとSTEPのアーキテクチャの相互運用性を確立して、概念モデルと論理モデルの連携を実現することが必要である。

製品の全ライフサイクルにおいて、製品やシステムは技術革新や環境変化のため、大きく変化する。多種多様な情報モデルを統合的に管理し、誰もが利用できる技術の開発が必要である。この問題を解決するためには、数学を基盤とした仕様記述言語とモデル記述言語の標準開発が必要条件になる。

1997 年の電子商取引共通基盤整備事業の一環として、「XML と STEP の統合化による組織知と技術知の蓄積と流通」のテーマで、CEO 協議会はこの理念を実証するための実験を行った。実

証実験で、当時の実装技術では、XML ベースで時間軸を扱う汎用のワークフローエンジンの開発ができなかった。セマンティック Web のアーキテクチャは、この問題を解決する基盤になる。

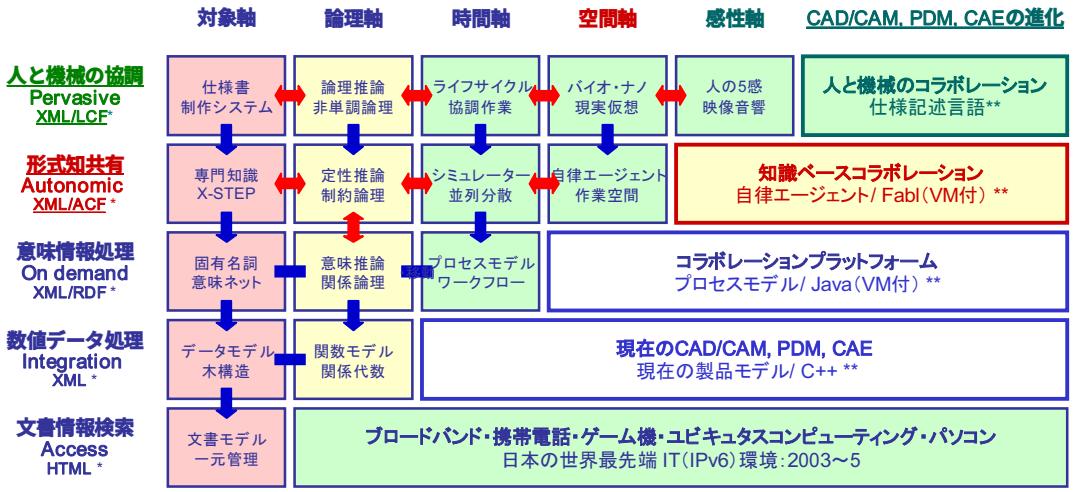


図 3 4 層構造によるデータの 知的資産化

図3は、RDFの上位層をX-STEP（Extended STEPの略）とするCEO協議会のアーキテクチャ提案である。モデル内部の記述を数学的に許される変換の集合にすることで、4層構造によるデータの知的資産化を提案している。現在のCAD/CAM、PDM、CAE製品は、第2層の技術である。第3層は、現在、Webサービス（またはオンデマンド）として、実装技術が実用段階にきている。第4層と第5層は、仮説の提案である。

利用者の視点からみると、第2層で論理軸（関数）、第3層で時間軸（ワークフロー）、第4層で空間軸（ルール）、第5層で感性軸（パターン）が、高次構造化されたデータ資産（コンテンツ）になる。データ構造を高次構造化すると、エンドユーザが自分たちの言葉で、プログラマーなしでコンピュータを利用することが可能になる。

第2層の理論基盤は、写像と関数をベースとする2値理論である。プログラミングの世界では、関数を計算規則として扱われる。関数は処理の単位である。関数型プログラミングは、数値計算処理の並列化の基盤として重要な役割を占める。[7)]

第3層の理論基盤は、関係論理（一階述語論理）をベースとする3値理論である。数学的には有向ラベル付グラフと呼ばれる。意味をベースとする言語メディアの基本仕様である。ワークフローの部品化によって、エージェント指向技術の基盤としての役割を担う。W3Cは、第3層RDFの基盤理論は、有向ラベル付グラフであると明記している。[8)]

コンピュータ科学の分野では、代数系言語で記述されるデータ操作は、数学的に許される変換規則を用いて論理系言語でも記述でき、その逆も真であることが示されている。2値の関係データベースを基本として、3値の知識ベースを構築できることは証明されている。[9)]

第4層の理論基盤は、制約論理をベースとした4値理論である。数学的には、ペトリネット理論が基盤理論となる。作業空間を場として定義することで、自律エージェントの定義ができる。また、ソリューションの記述ができるので、第5世代コンピュータ開発研究から生まれた定性推論の組込みが可能になり、シミュレータ構築の基盤としての役割を担う。定性推論は系の構造を微分方程式で記述し、系の動的変化をとらえるもので、定量的な情報が十分得られない系においても挙動が予測できる。[10)]

ペトリネットは、多くの事象が同時進行する並列システムのモデル化と解析に利用するための数学理論である。ユニークな特徴は、グラフィック視覚化・シミュレーション・数学的方法論の3機能を統合化するモデルが記述できることである。乗物に例えると、陸を走り水に潜り空を飛ぶ3機能をもった夢の自動車の機能をもった数学理論である。ペトリネット理論を用いて、有限空間では4値理論と3値理論は、数学的に許される変換を用いて等しいことが証明されている。[11)]

第5層は、矛盾した論理にも対応できる論理推論が理論的基盤である。注釈述語論理および非単調論理などを実装することによって、人とコンピュータの協調できる環境が構築できる。コンピュータ科学の分野では、ペトリネットをベースとした言語は、文脈依存であることが知られている。仕様記述言語の理論的基盤が高レベルペトリネットである。

UML言語は、第3層の技術標準である。データ･機能･状態の3側面の表記法を標準化したものであり、統合化はヒトの仕事である。これが現在のシステム開発業務の実態である。高レベルペトリネットを実装すれば、UMLの3モデル統合化業務は自動化できることが知られている。[12)]

5．2．ユビキュタスコンピュータの時代

至るところにコンピュータが存在するユビキュタス時代になると、知識管理の手法が変ると予想される。コンピュータが扱う情報は、数値・文書・図形・画像から発展して、音声・映像から各種のセンサー情報を加えて、コンピュータはメディアに進化する。ユーザインターフェースは、知的コンテンツに変る。

第5層の高レベルペトリネットを実装したプラットフォームができると、発見的方法と厳密な論理的推論との相互作用が可能になる。知識管理は、知識を形式知として保存するかたちから、現場の生データをセンサー情報で集めて暗黙知のままデータベースに保存し、分析することから新たなノウハウや法則を発見するかたちに変る。

6．教育システムの刷新

6．1．「動け日本」への対応

現在、日本では、経済成長の原動力となってきた製造業において、国際競争力を失いつつある産業が見受けられる。非製造業の生産性は、平均的にはその製造業を下回っている。単なる欧

米追従ではなく、世界が評価する日本の強みを自覚して、経済の活性化を図ることは緊急の課題である。

新しいモノづくり･サービスづくりによる需要の創生と知的生産性の高度化によって、日本経済は再活性化できる。地球全体のエネルギー消費、CO_2 の発生の制約などを考えれば、今後の経済成長は、資源的環境的な制約から従来の延長線上には、大きな期待はかけられない。情報通信技術をインフラとした新しいモノづくり、サービスづくりに依存するしか方法はないと考えられる。

コンピュータ利用形態は、難しいパソコンの操作方法を覚える必要がないコンテンツ・メディア方式に進化する。システムの開発方式は、小さく生んで大きく育てる方式に変る。莫大な時間とコストを必要とする現在のシステム開発方法論が前世紀の遺物となる日は、遠い将来の夢では無い。

コンピュータ利用の常識は変る。新しい情報環境によって、殆どのソフトはインターネット上の部品とサービスになる。セマンティック上に STEP のエンドユーザ環境が実装されると、設計・生産へのコンピュータの利用に関するこれまでの常識は変る。

６．２．知的 CAI プラットフォーム

すべての製造業は「設計の生産性を10倍にしたい」「納期を短縮したい」「コストを抑えたい」「設計製造の拠点を海外に展開したい」といった目標をもっている。方法論や開発環境や意志決定のプロセスを変えなければ、これらの目標は達成できない。

現在のCADは、類似の製品を繰り返して設計する場合、設計時間を大幅に短縮できる。しかし、概念設計が必要な新規設計にはほとんど効果がない。概念設計は新しいアイディアを生み出す過程である。概念設計にCADを使うには、コンピュータが知識を蓄え、蓄えた知識を使って推論する機能が必要になる。

ブロードバンド・ユビキュタス・セマンティックWebの3技術をベースとした情報環境は、コンピュータ利用のパラダイムを変え、21世紀型ライフスタイルを創り出す社会基盤になると予想される。新しいモノづくり・サービスづくりを実現するためには、新しい設計方法と最適なツールとよく訓練された人材が必要になる。緊急を要する日本の課題は、大学を基点とした知識共有技術の開発と社会人の生涯教育の高度化である。

このためには、産学官が共同して知的CAIプラットフォームを開発することが必要であると考えている。知識ベースCAD開発の先進導坑が、知的CAIプラットフォームの開発である。情報化社会における日本の未来は、製造業におられる人々のチャレンジ精神の復活と、新しい時代に対応するための新技術の習得にかかっている。

参考文献

1） http://www.itrd.gov/pubs/blue01/transmittal.html : 21 世紀の IT 革命 (2000)

2） TimBerners-Lee:Weaving the Web: the original design and ultimate density of the World Wide Web by its inventor、Harper Business (2000)

3） 吉川正俊:Web データベースの基盤技術としての XML、 情報処理 42 巻 7 号 (2001)

4） 横井俊夫:言語メディアを物語る・情報処理学会編、共立出版 (1998)

5） 綾日天彦:インターネット時代の協創設計・情報処理学会編、共立出版 (1996)

6） Christopher A. Goad・平野 哲雄訳:RDF に準拠した記述式計算、International Semantic Web Workshop (2001); CEO 協議会ワークショップ (2002)

7） 市川忠雄他:かわりゆくプログラミング・情報処理学会編、共立出版 (1994)

8） http://www.w3c.org/2000/10/swap/Primer.html:鈴木亮平訳:3 値理論を用いた RDF とセマンティックWeb 入門; CEO 協議会ワークショップ (2002)

9） 高濱忠彦:文書データベースの構築技術の特徴と効用、行政と ADP (1998.1)

10） 溝口文雄:人間に近づくコンピュータ・情報処理学会編、共立出版 (1996)

11） 村田忠夫:ペトリネットの解析と応用、近代科学社(1992)

12） 宮下洋一他:NEC におけるオブジェクト指向の取り組み、NEC 技報 47 巻 6 号 (1994)

総括と今後の方針

主 査　井 越 昌 紀

分散化された情報の島を結合する能力を持つXML技術系の新たな流れであるセマンティックWebと、製品モデル記述の先取り標準を目指すSTEPを融合することにより、機械の助けを借りて人の能力の柔軟性をさらに向上させ、製品のライフサイクルに関わるあらゆる問題解決のための情報処理を統合的に可能とする日本から発信できる新しい情報記述モデルの検討を行うこととする。

本年度は、製品のライフサイクルにおける視点の多様性を委員の専門の多様性に求め意見を交換した。とくに、次の点に留意して意見を述べることとしたが、充分に意見交換する時間が無かった。

1）「セマンティックＷＥＢとＳＴＥＰを融合」することによって何がうれしいのか、何が解けるのか？
2）それぞれの技術はどのようなものか？
3）それぞれの技術に関連した技術にはどのようなものがあるか？
4）これをまとめる枠組みはどのようなものか？

討論の結果、製品のライフサイクルに関わる情報は多岐多様であり、融合することにより解決できる問題のビジョンの展開するためには特徴あるコンセプトを持つモデルが必要であるとし、討論の中から次のような方針を持つこととした。

1）　情報記述の枠組みとして階層性を持たせたモデルを考える。その参照モデルとして W3C で検討されている Tim Berners-Lee の７層モデルを参考にする。
2）　各層は相当の数学的基盤を持つべきである。
3）　最上位層は「人と機械の協調のための対象の仕様記述」とし、対象は委員の専門や興味の分野とする。

次年度の方針案として、

1）ベースとしての綾委員の階層モデルについて次年度以降さらに検討する。
2）ライフサイクルに関わる情報の種類についてリストアップする。
3）セマンティック Web の最新技術動向を調査する。
4）人と対話する上位層の高次記述言語について調査する。
5）児西委員の知識地図について勉強し、セマンティック Web と STEP（エンジニアリング）との融合によってもたらされる利益について知識地図により考察する。
6）利益をもたらす技術分野について提言をまとめる。

などの中から、実現可能な部分について実行する。

第２部：
「プロダクト・データ活用企業事例」

ハーレーダビッドソン社（USA）
－新製品開発での企業間コラボレーション事例－

１．調査概要

時：平成１５年１月
訪問先：Mr. Patrick D.Keller(Director of Enterprise Risk Management)
　　Mr. Shingo Takeda(Curriculum Developer
　　:International Technical Instructor)
場所：3700 W.Juneau Avenue,Milwaukee,WI USA
内容：Harley-Davidson 社の最新製品 VRSCA V-Rod の開発における、ドイツ P 社との共同開発のあらましとその体験から得られた協業を生かすための知恵を当時の統括責任者 Keller 氏にお尋ねした。

2. Harley-Davidson 社概要

1903年に William S. Harley とDavidson兄弟によって創設された世界的オートバイメーカー。本社はミルウォーキー（米国ウィスコンシン州）。米国の生産設備はミルウォーキーエリア、イーストトロイ、トマホーク（以上ウィスコンシン州）、ヨーク（ペンシルベニア州）及びカンザスシティー（ミズーリ州）
社員は全世界で約8000人。年間売り上げ３４億ドル（2001年）。
2001 年に 2002 年モデルとして液冷エンジンを持つ VRSCA V-Rod を投入した。

3. Keller 氏インタビュー要約

Q.このプロジェクトの概要について

→ビジネス部分については、技術サイドとの協力の他にも、財務部門との間でプロジェクト管理のための提携を行った。また P 社との間で、エンジン構築を行うためのジョイントベンチャーをカンザスシティーに立ち上げ、そのための P 社との間の交渉、関係構築を Keller 氏が行っている。現在の職を得る以前には Harley-Davidson 社（以下 HARLEY 社）のマネジメント・ディレクターを勤めていた。

Keller 氏は、このジョイントベンチャーにおいて開発されるバイクがどのようなものになるべきか、という最初のコンセプト構築からビジネスリレーション、開発、そして開発のための資金確保などの全ての業務に関わった。同社ではプロジェクトの全てをこれだけ一人で全部やる、という例は無かったため、プロジェクトの名前をとって「P4 Guy」というあだ名がついた。

このプロジェクト以前は、バイク部門におけるチーフエンジニアを 7 年間勤めていた。このプロジェクトは 1995 年 5 月に開始され、約 8 年が経っている。プロジェクト開始にあたっては、どのようなコンセプトのバイクを作ろう、という構想はある程度あったものの、では実際にどのようなエンジンをいれるか、などに関しては全くアイデアが無かった。しかしとにかくやろう、ということになりプロジェクトが始められた。実は当初、別のプロジェクトのヘッドになるために引き抜かれたのだが、実際にマーケットリサーチを行った結果、別プロジェクトに関してはあまり良いアイデアではない、ということがわかり、とりやめとなってしまった。そのミーティングを行っている最中に、この新しいプロジェクトのアイデアが生まれた。

デザインには、1年半かかった。最初の半年で、まずデザインで「するべきではないこと」が何であるか、ということがわかった。残りの半年でなんとなくアイデアが浮かび、最後の半年でやっときちんとしたデザインが生まれた、という感じである。

エンジンベースは、レースバイクで使われるVR1000であるが、実際にロードレースをするようなお客さんはそれほどいない、ということでストリートバイクベースに作り変えた。このため、実際に利用されているパーツはVR1000とは全て異なっており、全て新たに開発されたものとなっている。VR1000エンジンをP社に持って行き、これをベースに、新しいコンセプトにあう製品デザインの作成を要請した。同社からはパフォーマンススペックを全てP社に渡し、メカニカルデザインはP社が全て行った。またチューニング、パフォーマンス開発、排気テストなどは同社が行った。このように分業が行われたわけだが、エンジンはバイクにとって非常に重要な部分であるため、その開発段階でのコミュニケーションは非常に密接で、開発プロセスは非常にコラボレーティブなものであった。

Q. 何名、いくつのビジネスユニットが参加しましたか

→参加者、参加部門に関しては、それぞれの組織がかなり違うので一概にはいえない。同社からは最初あまり参加しておらず、6名ほどであった。P社からはフルタイムの開発チームとして20名が参加した。しかしP社では、専門によって部門をわけており、専門家がそのつど支援のために別途参加したりもしていたので、開発チームのみで考えると10人から12人ほどだったと思う。HARLEY社からは、ドイツのP社本社に4人駐在させている。内訳は、調達が2名、エンジニアリングマネジャー1名、プロジェクトマネジャー1名である。プロジェクトマネジャーがKeller氏の直接の部下となっている。Keller氏自身は、少なくとも月に1回、2-3週間のドイツ出張を繰り返した。最初はあまりドイツ出張をしていなかったのだが、それでは全く事が運ばないということがわかったので。

彼はエンジニアリング部門の出身ではあるが、このプロジェクトにおいてはエンジニアリングマネジャーではない。ミルウォーキーにはモーターサイクルデザイングループがある。それに加えそれぞれの専門家が必要に応じて支援を行うという形で、10-20名のプラットフォームグループというものを立ち上げた。ここではフレーム、車輪、電気系統などのサポートを行った。当初モーターサイクルデザイングループが、パワートレインに関する活動を一部分行ない、またP社のパワートレイングループにおいても業務が行われた。

コンポネントのいくつかは、Harley社のものとは全く異なるものを利用することになるため、デザインを行う以前の段階から、サプライヤーの選択を行った。ここでは、P社、Harley社がお互いのサプライヤーの情報を交換しあい、その中からベストのサプライヤーを選ぶ努力が行われた。デザインを行う以前からサプライヤー選択を行った理由としては、デザインのコラボレーションの段階から、サプライヤーにも参加して欲しかったからである。選択にあたっては、エンジンはカンザスシティーで製造するため、なるべく地理的に近いサプライヤーが望ましかったが、もちろんそれが主要なファクターというわけでもなかった。

特に今回のように、サプライヤー側からのサポートを期待している場合、コラボレーションを行うプレイヤーを選択することは非常に重要であった。このため、Harley社、P社それぞれの調達･購買担当者は、パーツの購買や調達、サプライヤーとの関係構築に関して、非常に多くのディスカッションを重ねていった。

Q．全てのスタッフが一箇所に集まって話し合いをしましたか

→全てのスタッフが集まったわけではなく、またミーティングも一度ではなく、様々な場所で何度にもわたって話し合いが繰り返された。特にミーティングではお互いにどのような質問が飛び出すかは予測不可能であり、質問が出るたびにそれを持ち帰り、内部で検討し、そしてまた同じテーブルについて話し合いを繰り返すという積み重ねで、そのプロセスには非常に時間がかかった。ここでは大規模なミーティングではなく、小規模なミーティングを繰り返し、その結果をそれぞれが社内に持ち帰り全体に伝える、という方法を取った。このほうが効率的であったと思う。またここで学んだ重要なルールとしては、このような話し合いは少なくとも一部分はフェース・ツー・フェースでやらないといけない、ということであった。最初ドイツとの間の連絡は電話や電子メールを通じて行っていたが、もともと関係の基礎が構築されていない間柄では、この方法でやりとりを続けるのは非常に難しかった。まずは一度会って、お互いの顔を見、お互いを知り、メッセージの裏側にある背景や相手の状況を理解してから初めて電子メールや電話などが使えると思う。結果、電子メールはコミュニケーションの手段としては非常に重宝した。また電話やビデオ会議なども利用したが、それはKeller氏がドイツに赴き、そこで長い時間を過ごし、関係を構築したからこそ活用することができたものだと考えている。実際に現地に赴き、相手先が抱える問題を理解したり受け入れたり、またお互いが抱える問題を共に解決しようという姿勢を持ったからこそコミュニケーションがうまくいった。

とにかくこの開発事業は非常に大変であった。契約書とは、非常にスタンダードな状況において、お互いがどのような対応をとるか、という基本合意でしかなく、今回のエンジン開発のように、状況が難しくなればなるほど、元々の契約書の内容からは想像を絶するような努力が必要になる。このような状況を克服する上でもお互いの信頼というものは非常に大事であり、またその信頼はフェース・ツー・フェースによって生み出されたと思う。今回の開発では、パーツの設計など様々な業務があったが、その中でも技術的にも、また2社の関係を維持するといった上でも、エンジン開発が一番難しかった。

P社に駐在するHarley社職員は、当初P社が提供する顧客専用のオフィスで業務を行ない、またP社のデザイナーは、それぞれの部門で業務を行っていたため、同じ施設内にいながらも、顔をあわせるということが非常に難しかった。P社のデザイナーが、顧客用オフィスに出向いてHarley社職員と話すことはできたが、Harley社の職員が、P社内の様々な部門を自由に行き来することは禁止されていたからである。これではコラボレーションもうまくいかない。後にようやく、フルタイムでこちらの設計に関わってくれるP社のデザイナーが選ばれ、同じオフィスに入れてくれた。これでやっと、両社のエンジニアリングデザインのマネージャーや、それぞれの購買・調達部門の担当者など、Harley社の製品を開発するという同じ業務についている人達が、フェース・ツー・フェースで、ずっと机をつき合わせて、同じオフィスで一緒に仕事をできるようになった。とにかく同じ部屋にいないと、何も始まらない。P社とHarley社ではお互いの企業文化も違うし、それぞれが問題にぶつかったとき、どのように対処していくか、違うやり方を学びあう上でもフェース・ツー・フェースは非常に良いことであったし、オープンマインドになるという点からも重要であった。

また言葉の壁の問題もあり、それぞれがお互いの言葉を学びあった。最初は英語でミーティングをやることになっていたが、それでは非効率だと思ったので、ドイツ語でやってもらうことにした。P社側が自分達のことをどう言っているのかも知りたいし、文化的にも意見があわないとはどのような状況のことを指すのか、きちんと理解したかったので、Harley社側でもドイツ語の勉強をした。

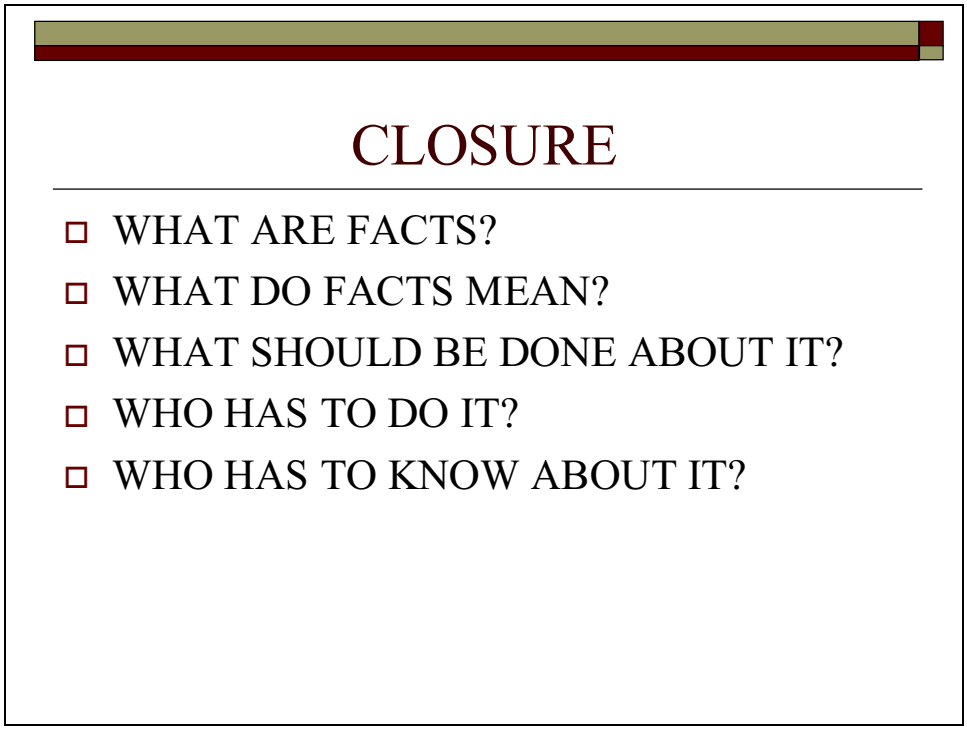
このような努力は、チームとして一つのことを達成する上で必要なものであった。言葉や社会、国籍が違っても、それぞれが同じ「人」だということがわかり、またそれでもみんなが一つのチームだということを理解するに至り、チームは上手く動くようになった。Keller 氏の仕事は、全員がきちんと働くことができる環境を整え、解決するべき問題に向かって全員が一丸となって進んでいけるようにする、ということであった。特にチームのメンバーが多種多様な場合、ある個人のために仕事がうまく進まない、という問題はよくあることである。特にそれが文化的な問題である場合、お互いを非難しがちである。そこで、もしも何か問題が起きた場合、それはチームのヘッドである Keller 氏の責任であるというルールを作った。しかしどのような問題であっても、それによってプロジェクトが成功するか失敗するかがかかってしまう。こういった意味でも、チームワークは非常に難しいものであった。

またこのプロジェクトは、スタッフ全員に、通常の能力以上のものを求める非常に厳しいものであった。しかし一度チームが動き出すと、プロジェクトが難しい分だけ、そのモーティベーションは非常に高いものであった。このプロジェクトに関わった P 社のスタッフは、2-3 年に渡り、1 週間に 6 日間出勤、勤務時間は 1 日に 12-13 時間にのぼった。日曜に出社しなかったのは、それが単にドイツでは違法であるからで、それが無ければ毎日でも出社してきそうな勢いであった。実は P 社がこのプロジェクトのために最初に開発したエンジンは失敗に終わっている。というのは、P 社では当初、自動車のエンジンを作るよりはバイクのエンジン開発は簡単だろう、と高をくくっていたのである。しかしこの失敗がバネとなり、P 社側のプロジェクトへのコミットメントがさらに深まり、人員を増やしたり、トップレベルの専門家が投入されるようになった。最初は簡単だろうと思っていたバイクエンジンの開発であるが、最終的にはそのリクワイヤメントは非常に複雑で難しいものであり、P 社の担当者も、今までで開発したエンジンの中でも難しいものの一つであったと言っていた。

コラボレーションにおいては、フェース・ツー・フェースが非常に大事だと思う。電子的ツールも使う必要があるが、それは補完的な役割で利用される。このプロジェクトにおいては、ドイツ、ミルウォーキー、カンザスシティー、そして世界中のサプライヤーなど、地理的に離れた相手とのコミュニケーションツールとして、電子メールが非常に有効であった。時差もあまり気にすることもなく、またボイスメールと違い、内容が記録として残る点も良かった。電子メールを利用すると、内容が残り、それをデータベースとして利用することができるので、後にどこでいつ、何が起こったかなどの追跡を行なったり、責任の所在確認にも役立った。また時差を活用することもできた。ミルウォーキーとドイツは 7 時間差があるため、何か質問がある場合、米国側が退社時間直前に質問を送って置けば、翌日出社するころには返事が来ている。また電話も、複数のスタッフが参加することができるため多用された。この場合は、時差の関係から午前中に行われる場合が多かった。ビデオ会議ツールも利用したが、広帯域の問題や画像が鮮明でないなどの点から、あまり良いとは思わなかった。しかし一方で、Harley 社内の他のサイトとの間でビデオ会議を行った際には、相手のジェスチャーや表情がわかるので、電話以上に相手の意図がわかりやすい、という利点があることにも気付いた。

コラボレーションのためには、自分に与えられた業務だけに責任を持つのではなく、より広範囲の責任をそれぞれが自覚しなければならない。個人のパフォーマンスは、プロセスの一部なのだという意識が必要で、そのような意識のもとコラボレーションの必要性が生まれてくる。仕事を細かく分業して、個人がその与えられた仕事だけに責任を持つというのではいけない。ここで自分は「Interdependence（相互依存）」という言葉を使いたい。

お互いが何をしているのかを理解し、それを考慮した上で、自分が行った業務の結果を、次のプロセス相手に渡していく、ということが重要である。チャートを見ると、このチャートの中の誰も、一人で仕事を達成するだけの知識は持ち合わせていない。よりよい意思決定するには、全員のよい知識を活用しなければならない。



このプロセスの目的は変化を作るということである。変化を作るということは、「Closure」がコンスタントに失われていくということである。何か変更が行われるごとに、デザインや製造が変わっていく。このため、自分がするべきことだけをやっていると、周囲の事実が変わっていることに気がつかず、「Closure」が無くなっていることに気付かなくなる。今現在変化が起こっているということについてコミュニケーションが無ければ、人々は何が実際に起こっているのかがわからず、フラストレーションをためる一方である。このため、どのような変化がおきているかを知るための方法をプロセスに組み込まなければならない。どれだけ多くの変更があるか、またその規模によって方法は変わるが、コミュニケーションを通じて、それぞれの「Closure」を取り戻す方法を構築しなければならない。これにより、人々は必ず周囲で起こっている変化やそれぞれの「Closure」についての情報をきちんと知ることができるようになる。例えばデザイナーが週ごとにミーティングを開催すれば、その間に他のデザイナーがどのようなことを考えているかなどを理解することができる。そして「Closure」を得る時間を短縮すればするほど、その間に起こる問題の発生を抑えることができる。そのための方法として、同じ部屋でミーティングをする、ということが挙げられる。しかしいつもミーティングを行えるわけではないため、現在の状況について、絶えず何らかの形でコミュニケーションしなければならない。

（報告者注）
「Closure」そのものはコラボレーションを行うための手段やプロセスそのものではなく、コラボレーションにおいて、参加者の間で認識あわせが必要となる「事実」（今何が起こっているか、その事実に対して何が行われるべきか）や「するべきこと」（誰が何をするべきか、誰がこの事実を知っておくべきか）、ということに成ります。何かしら複数の人や組織が集まってプロジェクトを行っていると、シチュエーションに合わせて「Closure」は非常に流動的になります。このため、ミーティングなどを通じてコミュニケーションをとることで、「Closure」に関する共通認識をいつも合わせておくべきである、ということを Keller 氏は話されている。

Q. このような情報をデータベースに載せたりしましたか

→私の意見としてはデータベースを成功裏に利用しているとは思えない。というのはプロジェクト管理システムなどのデータベースは維持が大変だからである。プロジェクト管理システムではプロジェクトのステータスや情報を共有したりできるが、問題はこのようなデータベースをアップデートしなければならないという点である。またスタッフにいつもデータベースに情報をアップデートさせるのも大変である。もしもグループが大きくなりすぎて、コミュニケーションが難しくなった場合には、このようなツールの利用も良いとは思う。しかし問題としては、全ての人々に情報を入力させることが難しいという点が挙げられる。このようなシステムに情報を入れるにあたっては、通常他人に何をしてほしいか、という情報をインプットはするものの、他人が自分に何をしてほしいか、という情報については受け入れたがらない場合もあるからである。

CAD データに関しては、Harley 社にもデータベースはある。しかし問題は技術ではなく、ヒトの問題である。技術はヒト無しでは活用できない。例えばデータベースに関しては、誰がこのような情報を変更する必要があると考えるか、そしてなぜ変更が行われるべきと考えるか、そしてそれを利用して何をするか、という点が難しい点である。またデータベースを作ったとしても、そこに保存されるデータは果たして良いデータ（業務に活用できるという意味？）なのかを見極めなければならない。テストデータなのか、開発段階のデータなのか、それともこの通りに構築すれば、すぐに顧客に販売できるようなデザインデータなのか？このようなことは、全てデータの周辺の問題として、別に取り組まなければいけない問題である。

私が考えるデータベースは、既に関係が構築されている人たちの間では利用することができるかもしれないが、データベースそのものが関係構築を行うとは思えない。

Q. 様々な地域に分散している人々にどうやってこの原則を広げて行くのですか

→沢山の場所にいる人たちに情報を広げるという点については、まず CAD については、同じフォーマットのものが、デザイナー、製造エンジニアなどの間で共有できるようにするべきであるが、現在そのようなことは実現されていない。しかしパーツの現在のステータスを見たり、誰が開発に取り組んでいるかなどの情報に、誰でもアクセスして見られるようにしなければならない。またこのような情報共有を自動化したい場合、特定のパーツ開発に関わっている人々は誰であるか、という定義づけを行い、何かその部分に問題や変更が発生した場合、その情報が自動的に送信されるようにするべきである。これにより、パ

ーツ開発に関わるデザイン、テスティング、購買担当者やサプライヤーとの間で、新たなコネクションが生み出される。

データベース構築においては、製品に関する情報や組織に関する情報の双方がデータベースと関連づけられて初めて、データベースを活用したコミュニケーションを行うことができると思う。これらの情報があって初めてコミュニケーションの自動化を図ることができる。しかし組織やパーツなどの製品構造はコンスタントに変わっていくものなので、それにあわせてデータベースを維持していくのは非常に大変な作業である。このような業務はプロジェクト管理ソフトウェアを利用して行うことができるが、しかしこれらのソフトウェアでは CAD データを管理することができない。

Q. コラボレーションのルール、スタンダードについて

→ルール作りは、業務のどこまでを個人でやるべきか、またはチームでやるべきかという境界線を引くためにも必要である。例えばプロジェクトの中で何か変更があった場合、誰が誰に報告義務があるかなどを決めておかなければならない。例えば P 社がエンジンデザインで何か変更を行った場合、それを Harley 社に報告せずに進めてほしくなかったため、チェンジコントロールシステムを利用していた。これはあまり良いものではなかったが。しかしチームで知識を共有するにあたっては、そのためのルールを作らなければならないことは非常に基本的なことである。

Q. P 社とのコラボレーションは初めてだったのですか

→P 社との間では、30 年にわたってコラボレーションを行っている。しかし今回のプロジェクトが最大で、最も深いものであった。

Q. P 社と Harley 社の間で新しいルールがありましたか

→ルールは何百種類とあるため、一つ一つあげることは難しいが、例えば何らかの変更があった場合、どのようにコミュニケーションするかという点については、週ごとのレビューミーティングが行われた。これは時に週 2 回行われることもあった。またサプライヤー選択に関するルールもあった。これは技術的なものから、工場にどれだけ近いか、などがポイントとなった。またデータに関するルールとして、P 社は当初 Harley 社が利用している Pro/ENGINEER でのフォーマットを利用することに合意した。しかし実際には、P 社内において Pro/ENGINEER を利用することのできるエンジニアが非常に少なかったため、P 社が通常利用している CATIA フォーマットでデザインを提出してもらい、それを Harley 社側が Pro/ENGINEER フォーマットに変換するという作業を行った。これは P 社側に Pro/ENGINEER を使いこなせる人があまりいなかったため仕方の無いことであったが、本来であれば、スタンダードな CAD システムを設定しておき、全員が利用することができることが望ましい。また CAD システムを利用してモデルを作成するにあたっても、そのためのスタンダードを作っておくことが重要であろう。P 社と Harley 社との間では、今だにこれらに関する違いは残っている。

エンジニアリングスタイルについては、P 社はデザインに関して、エンジニアの力が強い、という感じがあった。一方 Harley 社では、エンジニアは機能に関することについては影響力をもっているが、デザインに関しては周囲の意見もかなり取り入れなければならないという違いがあった。例えば Harley 社では、スタイルが一番重要な部分となっているが、自動車のエンジンデザインを行ってきた P 社にとっては、そのようなことは理解できない

ことであった。このため、Harley 社のエンジニアリングスタイルに関し、最初から学んでもらう必要もあった。これは非常に時間のかかるプロセスであった。

Q. コラボレーションを通じた仕事のスタイルの変化は

→エンジニアリングスタイルに関しては、P 社と Harley 社との間のベストなスタイルを組み合わせようとしたが、これを説明することは非常に難しい。2 社の間でコラボレーティブスタイルを使ったとしかいいようがない。

P 社は主に技術面の貢献を行い、それ以外の業務に関しては Harley 社の伝統に基づく方法を取った。しかし難しかった点は、今まで自分ひとりの責任で行ってきたことを、チームとの間で共有しなければならないという「責任感の共有」という部分であった。これは今までに個人プレイヤーとして仕事をしてきた人たちにとっては大変なことであった。「責任感の共有」を実現するためのプロセスとしては、他のチームメートの話をきくということ、そしてチームメートの意見を自分の中で消化すること、そして、これらの意見をもとに、何がプロジェクトにとってのメリットになるのか、という議論を進めていく、ということが行われた。このプロセスを進めていくことで、チーム参加者は、周囲の人たちが行っている業務に感謝し、何がチームにとって良い結果を導くことになるのかを考えるなど、オープンマインドになっていった。

Keller 氏自身がプロジェクトを始めて驚いたことは、P 社には P 社のやり方があり、P 社のスタッフはそれに満足していたという点である。このため、Harley 社のやり方にわざわざ変えなければならないことには当初抵抗があったようである。このようなことは、やはり 2 社がコラボレーションを行って初めてわかったことである。

このような障害をどのように解決していくかということについて腹を割って仕事を進めていくうちに、お互いへの尊敬が生まれた。当初はプロセスを管理するため、責任者がコンスタントにチェックを行わなければならなかったが、そのプロセスがひとりでに動くようになった。このように、プロセスが自ら動くようになると、そのプロセスの実現が双方の社員にとっての喜びに代わっていった。

Q. ではこの障害はあってよかったものでしたか

→Harley 社、P 社双方からのチームがコラボレーションをすることで、Harley 社でもない、P 社でもない新しい視点を持つチームが生まれた。逆にこの新たな視点を持つチームが、親会社である Harley 社や P 社に良い影響をもたらしたといえる。しかし当初は親会社のほうが、それに対して抵抗を示した。親会社にしてみれば、このような新たな「ハイブリッド・チーム」は、単に相手企業の影響を受けてしまったチームとしか見えず、逆にチーム孤立してしまった時期もあった。後に、チームにいた社員が親会社に戻り、親会社に良い影響を及ぼした場合もあったが、P 社においては、逆に古い体質になじめず、結局会社を辞めてしまった人もいた。

Q. P 社と Harley 社との間の関係は、一時的なジョイントベンチャーなのか、それともこれからも合弁は続いていくのか

→P 社との関係には 2 種類ある。一つは、今回のようなエンジン開発などワンポイントの共同開発を行なう関係。そしてもう一つは、生産現場においての関係である。カンザスシティーの工場は Harley 社のものであるが、双方が共同出資したジョイントベンチャーユ

ニットが生産を行い、Harley 社がそこからエンジンを買い取る形となっている。エンジンを作っても、それが最終的には製品としてお客様に届かなければ意味がないので、P 社側も、生産のための設備投資や人的投資など、Harley 社と同じ位のコミットメントが必要であった。P 社はエンジン生産において、品質、コスト、納期の管理を行っている。またエンジン価格についても、生産前に価格が設定されており、そのコスト内で生産を行えるようにしている。

Q. なぜ P 社と協業したのですか

→他の候補企業もあったが、P 社とは技術的な提携を 30 年にわたって続けており、このプロジェクト以前にも、ノイズ低減に関するプロジェクトを P 社と行っていたため、経営陣が今回もやれるだろうと好感を持っていた。ここで提携しているのは、他企業とのコラボレーションを主に行っている P 社・エンジニアリングという企業である。プロジェクト開始当初は、P 社はあまりこのプロジェクトを重要視していなかったようだが、その後、精鋭スタッフが送り込まれるようになった。

以上

デーナ社(USA)社

ーPDM Implementation Initiativeー

1. 調査概要

時：平成１５年２月

訪問先：

Mr. John Hornbostel (Executive Director)
Mr. Brad Knauf (Director,Information Technology Automotive Systems Group)
Mr. Hiroshi Olin (Director of Business Development Automotive Systems Group)
Mr. Bob Keister (Director of Global Core Engineering Traction Technologies Group Spicer Technology,Inc)
Mr. Ivan Stretten(Manager,Computer Aided Engineering Services,RapidPrototyping,Engineering Graphics Solutions Advanced Technology Resource Group)
Mr. Richard L. Dafforn (Mechanical Testing Manager Advanced Technology Resource Group)

場所：8000 Yankee Road,Ottawa Lake,Michigan USA
内容：米国自動車部品大手サプライヤにおけるPDM Implementation Initiativeのお話を伺った。

2. Dana 社概要

Dana 社は２００４年で１００周年を迎える自動車部品メーカーである。同社は 1904 年、ユニバーサルジョイントを開発したエンジニア、クラレンス・スペンサー氏と、ビジネスマンであるチャールズ・デーナ氏などにより創設された。2001 年の売上げは 103 億ドルで、現在北米、欧州、南米、アジア太平洋を中心に世界中に約 6 万人の従業員を抱えている。同社は主に、アクスルやプロペラシャフトなど、自動車の構造そのものに関するパーツを製造する部門（Under the Vehicle）、そしてエンジンやエンジン関連のパーツを製造する部門（Under the Hood）、サービスパーツ・アフターマーケット製品製造・販売部門、またシャシーなどモジュールの組み立てを行うシステムインテグレーション部門などに分かれている。日本においては 1973 年から製造を行っており、現在世界市場における日系の大口顧客はトヨタとなっている。2001 年に於いて、日本企業との取引は、全世界で 6 億ドルの売上げを計上している。

3. PDM (Product Data Management) Implementation Initiative について

【PDM 導入背景】

Dana 社では、現在 PDM（製品データ管理）の導入イニシアチブを実施している。現在 PDM の導入を通じ、エンジニアリングにおける以下の 3 つの分野についての業務効率化を目指している。

－CAD データなどのデザインデータの管理
－デザイン分析やレポートなど、エンジニアリングに関するドキュメント管理
－ワークフロー管理

エンジニアリングの世界は1970年から現在までで大きく様変わりしている。1970年代、CADは単に設計を行うなど、一つの「タスク」に注目して利用されるシステムであった。それが1980年後半から1990年頃より、よりエンジニアリング部門全体に注目したツールとして利用されるようになった。ここではソリッドモデルなど、デザインのために作成された情報を、資材や構造解析などにも活用されるようになった。そして現在、これらのCAD情報は、エンジニアリング部門だけのものではなく、さらにビジネス全体での利用に注目したものへと進化している。PDM導入の背景には、このように進化するCADシステムを、企業全体で活用できるようにしたいという目的があるといえる。

【Dana社におけるエンジニアリングの課題】
Dana社では、エンジニアリングに関して以下の点を改善するべきと考えている。
－社内、また顧客企業との間でのコミュニケーションの改善。
－生産性の向上。
－さらなる効率化。
－顧客の満足度の向上。

さらに以下の点を課題として捉えている。
－将来のビジネスをさらに発展したものとするため、R&Dを積極的に行わなければならない。
－VE（Value Engineering） のための様々な価値分析方法を活用し、既存の製品に対する顧客満足度を向上させる。
－顧客企業が製品開発サイクルを短縮する傾向にある中、同社もそれに素早く対応できるような体制を整えるため、業務の効率化を進めていく。
－効率化のため、製品開発フェーズの段階から、製造や調達、販売などといった全ての業務ラインにおけるコミュニケーションを改善していかねばならない。

また顧客企業がDana社に求めているものとしては、以下の点が挙げられる。
－自社製品に役立つような新しい、イノベーティブなパーツの開発。
－より良いプログラム管理。
－スムーズなプログラム開始。
－製造開始前の調整など、エラーを最小限に抑える。
－顧客に対するエンジニアの素早いリスポンス。
－エンジニアリングサポートの改善。
－品質に関する問題を最小限に抑える。

このように様々なエンジニアリングの課題や顧客からの要望があるが、実際にDana社で行われているエンジニアリング活動を見てみると、現在リソースの8割が「メンテナンスエンジニアリング」に利用されているのが現実で、新製品の開発などに投入されるリソースは2割程度に過ぎない。Dana社としては、新製品の開発に6-7割のリソースを割きたいと考えているが、では「メンテナンスエンジニアリング」のリソースをどのように抑えるかが問題となる。

【メンテナンスエンジニアリングとは】
Dana社では、「メンテナンスエンジニアリング」を以下の業務と捉えている。
－既存の製品製造コストを削減するための努力。
－プラントサポート。
－欠陥製品削減のための努力。

－既存製品のアプリケーションエンジニアリング業務。

Dana 社では、メンテナンスエンジニアリングを削減する方法として、以下の点を改善するべきだと考えている。
－APQP（Advanced Product Quality Planning：製品品質計画）プロセスの改善や、DOE（Design of Experiment：実験設計）、また SixSigma を利用するなど、さらにロバストなデザイン実験方法を採用する。これにより、正しいデザインを行い、後々のデザインのやり直しなどを避けることができる。
－情報検索方法の改善。現在、エンジニアは業務時間の 40%を、意思決定のために必要な情報を検索に費やしている。これはいわば付加価値の低い業務であり、結果生産性を下げることにつながっている。
－既存のデザインデータを再利用する。これは製品開発における投資を削減する上で有効である。
－ECO（Engineering Change Order：技術変更指示）をなるべく減らす。特に製品の製造が始まってからの変更は、最初のデザインが正しく行われていなかった、という証拠である。

【パーツライブラリの導入】
上記に挙げられた改善点のうち３番目に関連し、Dana 社では PDM の一環として、既存のパーツ情報を全て納めた「パーツライブラリ」の構築を行っている。例えばエンジニアがワークステーションを利用し、新たなパーツのデザインを行う際、デザインに再利用することができるような既存製品・コンポネントの情報がないかを、「パーツライブラリ」を検索して確認することができる。「パーツライブラリ」を利用することの利点としては、以下の点を挙げることができる。
－新たなデザインを一から作り直す必要がないため、付加価値の低いタスクを減らし、デザインプロセスをスピードアップすることができる。
－既存のパーツをデザインの中に組み込むことで、新たに工場で生産しなければならないパーツを減らすことができる。
－生産にあたっては、既存のツールやゲージを再利用することができるため、新製品のための投資を抑えることができる。
－工場フロアにおいても新たな設備の導入やセットアップを行う必要がなく、工場における効率化にも貢献する。

「パーツライブラリ」はエンジニアリンググループや部門に関わらず、どこからでも情報にアクセスできるようになっている。ここでは主に製品の３D モデルや２D での設計図、また製品に関するテキスト情報などが保管されている。
製品開発を行なうにあたっては、異なる部門間のコミュニケーションを改善しなければならない、という観点から、「パーツライブラリ」をはじめとする PDM の構築が行われている。エンジニアリングに関する情報は、最終的にはマーケティング、調達、製造、プロセス計画やツーリング部門などあらゆる部門間で検索、共有が可能なものでなければならない。またより良いエンジニアリングを行うにあたっても、これらの異なる部門が持つ情報を利用することは非常に重要である。

【PDM を通じたプロセス統一】
また、PDM 導入に際しては、社内のビジネスプロセスの変更を行うことも重要であると考えている。製品開発改善のためには、まず達成するべきビジネス目標が設置されるが、その目標を達成するためには、今までのプロセスを変更しなければならない。そしてそのプロセスを変更するためのツールとして、PDM のような技術が活用されると考える。従来製

品開発業務を行うにあたっては、製造などの異なる業務分野の間に大きな情報の壁があった。今まで製品開発に関しては、それぞれの部門が業務に必要な情報を作成してきたが、部門間で情報が共有されてこなかったため、重複して作成される無駄な情報も沢山あり、非常に非効率的であった。Dana 社では、PDM 導入を通じ、エンジニアリング、製造、組み立て、マーケティング、調達、販売などにおける製品開発プロセスを全て統一することで、情報の重複などによるオーバーヘッドを削減しようとしている。

【PDM を利用して実現できること】

ドキュメント管理システムに関しては、Dana 社ではエンジニアリングスペックや技術要件などに関する情報などのドキュメントを管理するシステムを既に活用している。これらのドキュメントは本社に一括管理されており、米国や世界中の生産施設が同じ情報にアクセスできるようになっている。しかし現在、エンジニアリングで利用される情報はこれだけではなく、プロセス計画情報、部品票（BOM：Bills of Material）、ツール情報、モデル情報、テスト情報、品質情報など非常に多岐に渡っており、またこれらの情報は異なる部門の管轄下に置かれている。これらの情報は、エンジニアリングにおける正確な判断のための材料として、また製品の改善を行うために非常に重要な情報となる。データ管理システムを利用し、これらの情報を全て統合することで、エンジニアは異なる部門が持つ情報に一括してアクセスすることができるため、今まで情報を探すだけに宛てていた業務時間を、さらに付加価値の高い業務を行う時間に代え、生産性を大幅にアップさせることが可能になると考えている。

PDM を利用し、新たなプロジェクトを開始するにあたっては、まず新たなプロジェクトファイルが作成される。このファイルは、PDM を通じ、顧客が提示してきたエンジニアリングに関するリクワイヤメント情報、ラボデータ、ベンチマーキング情報など、製品開発を行なうための意思決定をサポートする情報にリンクされる。エンジニアはまた、PDM を利用し、デスクトップから直接パーツデザインの立体的なモデルなどを引き出して閲覧することも可能となる。過去には、このような情報を引き出すためには、エンジニアがわざわざデザイン部門まで赴き、これらの情報を受け取らなければならなかったが、PDM を利用することで、誰の手も煩わすことなく、またデザインエンジニアリング業務を中断させることなく、情報を得ることが可能となる。
PDM システムは現在エンジニアリング関連の情報統合を進めているところであり、実際の利用は部分的なものに止まっている。エンジニアリング情報の統合が終わり次第、次のフェーズにおいて、品質などに関する情報の統合も進めていく予定である。

またこれらのエンジニアリング情報は、特にアクセスを制限することなく、どの部門のエンジニアであっても情報を引き出すことが可能となっている。例えばモジュールやシャシーを作るシステムインテグレーショングループでは、アクセルやドライブシフトの情報、自動車の構造に関するコンポネント情報など、他部門がデザインしたパーツの情報が必要となる。これらの情報が、PDM を通じて簡単に引き出せるようになっている。

【PDM 導入効果への期待】

PDM を利用することで、「メンテナンスエンジニアリング」のための時間を削減し、その分新製品開発の時間増加が実現されることを期待している。このようなシステムを利用し、「メンテナンスエンジニアリング」時間を削減することについては、米国において様々なケーススタディが行われている。これによると、エンジニアが PDM のようなシステムを利用し削減できる業務時間は、プログラムごとに平均して 220 時間にもなると考えられている。つまり、削減された 220 時間を新製品開発に充てられるわけである。Dana 社においては、PDM 導入によりどれだけ効果があがったかについては、これから計測していかねばな

らない。しかしASG部門の100名のエンジニアに対して行った調査では、エンジニアリングに関する意思決定に必要な情報を探すという業務に、1週間のうち平均2日間が費やされているということが判明している。PDM導入により、このような大きな無駄を大部分削減できるのではないかと考えている。またワークフローにおいては、今までエンジニアリング変更を行う際のプロセスとして、マニュアルで130以上のプロセスがあった。これを90%削減できると考えている。

【コラボレーションツールの利用】
世界中にデザイン施設を展開するDana社においては、時差の違いを活用し、デザインを分業して行えるようなコラボレーションを進めている。これは南米やアジア、インドにあるデザインセンターを活用し、デザインプロセスを共同で行うというものであるが、ここではそれぞれのデザインセンターが自動的に情報を引き出すというよりは、こちらが一方的に情報を渡すという形になっており、電子的というよりはどちらかというとマニュアルプロセスでの作業が行われている。

一方、現在Dana社が利用しているエンジニア間のコミュニケーションツールとしては、「E-Vis」と呼ばれるシステムが利用されている。これは、デザインを行うにあたり、異なるロケーションにいるエンジニア同士がコミュニケーションを図ることができるというもので、主に CAD ワークステーションを利用して行われている。このツールを利用すると、例えばパーツを組み立てたり分解してみたり、また組み立てプロセスを作成するなどの作業を行うことができる。このツールはDana社のイントラネットを利用し、社内エンジニア間のコミュニケーションツールとして利用されているほか、外部のエンジニアとの間でも、セキュアなウェブサイトを通じてやり取りを行うことができる。

現在このツールは、社内製造グループにおけるデザインレビューに、また欧州の顧客企業1社との間で、アクセルのデザインレビューにも利用している。このようなツールを利用することの大きな利点としては、電話やファックスなど他のコミュニケーション手段よりも数倍効果的であること、また出張のための経費や時間を抑えることができることが挙げられる。また異なる施設で働くエンジニアがインタラクションを行うことができるため、従来のオペレーション方法を根底から変えることも可能となる。例えば地理的にデザイン業務を分散するなど、ワークフローを大幅に変更することも可能であるし、パートナーや様々な部門との間の関係が変化していくと考えられる。

しかしDana社では、コラボレーションツールを利用して100%全てのコラボレーションを実現することはできないと考えている。現在利用されている「E-Vis」は、CADツールそのもの、というよりは、CD (Collision Detection) や Interference Detection など、デザインレビューのためだけに利用されているのが現状である。顧客企業はこのツールを利用し、Dana社が作成したデザインをチェックし、そのフィードバックをもとにDana社では新たにデザインをしなおす、という作業を行っている。つまり、このツールを利用し、実際にエンジニアがその場でデザインを行っているというわけではない。

デザインに関するリクワイヤメントや、その他の初期的なコミュニケーションに関しては、ほとんどが対面式、またはメールなどの電子的な方法で行われる。これらのやりとりを通じ、まずは顧客のニーズをつかみ、それをもとにDana社がプロポーザルを提出するという段取りとなる。しかし、顧客から出される要件は、最初から全てが決定されているわけではなく、コンスタントに変化していくものであるため、開発は非常にインタラクティブでダイナミックなプロセスとなる。例えば顧客企業側で、総車両重量、ギアレシオ、エンジンのトルク出力などの要件が急に変わることもある。またパッケージングの問題や、組

み立てに関する問題が生じてきたり、顧客企業のマーケティング部門が、どのようなコンセプトの製品を作りたいかを変更してきたりもする。このため、最初にもらった要件が、開発が終了するまでに何度も変わることは稀ではない。

結局、コラボレーションツールは、このような要件の変更にあわせ、きちんとデザインが出来ているかをチェックするのに役立っているといえる。要件にあわせてデザインされたパーツ情報は、３Ｄの CAD モデルとして電子的に顧客に送信される。顧客はその情報を自らが持つデザイン情報などに組み込んでみて、組み立てに問題がないかどうか、変更が必要かどうか、パッケージの点から何か問題がないかどうかなどをチェックすることができる。

顧客企業は、それぞれ異なる CAD システムを利用しているという現状があるため、Dana 社は全ての顧客の要求に応えられるような体制を整えている。ヨーロッパの顧客とコラボレーションを行った時は、たまたま同じ CAD システムを利用していた。しかし、「E-Vis」を利用してやり取りされる CAD データは、全て中立的なフォーマットに変更されるため、どのような CAD システムとの間でも互換性のあるデータのやり取りが可能となっている。

Dana 社は Harley Davidson 社を訪問したこともある。やはり同社の指摘するとおり、全てのフェイス・ツー・フェイスのやりとりを電子的なものに変えることができるとは思わないが、一部分を電子的なものに代えることは可能だと思う。HD 社と話して後で気がついたことは、例えば最近ではビデオ会議ソリューションにおいてもリアルタイムで見ることのできる非常に性能の良いものが出ており、このようなものを補完的に利用することができると考えている。

コラボレーションツールを利用し、顧客企業との間でその場でデザインを行ったことは無いが、社内では製造プラントとの間でやったことがある。ここでは、製造プラントにいるプロセスエンジニアと、Dana 社技術センターにいるデザインエンジニア、アプリケーションエンジニアが、組み立てプロセスのデザイン変更をインタラクティブに行った。

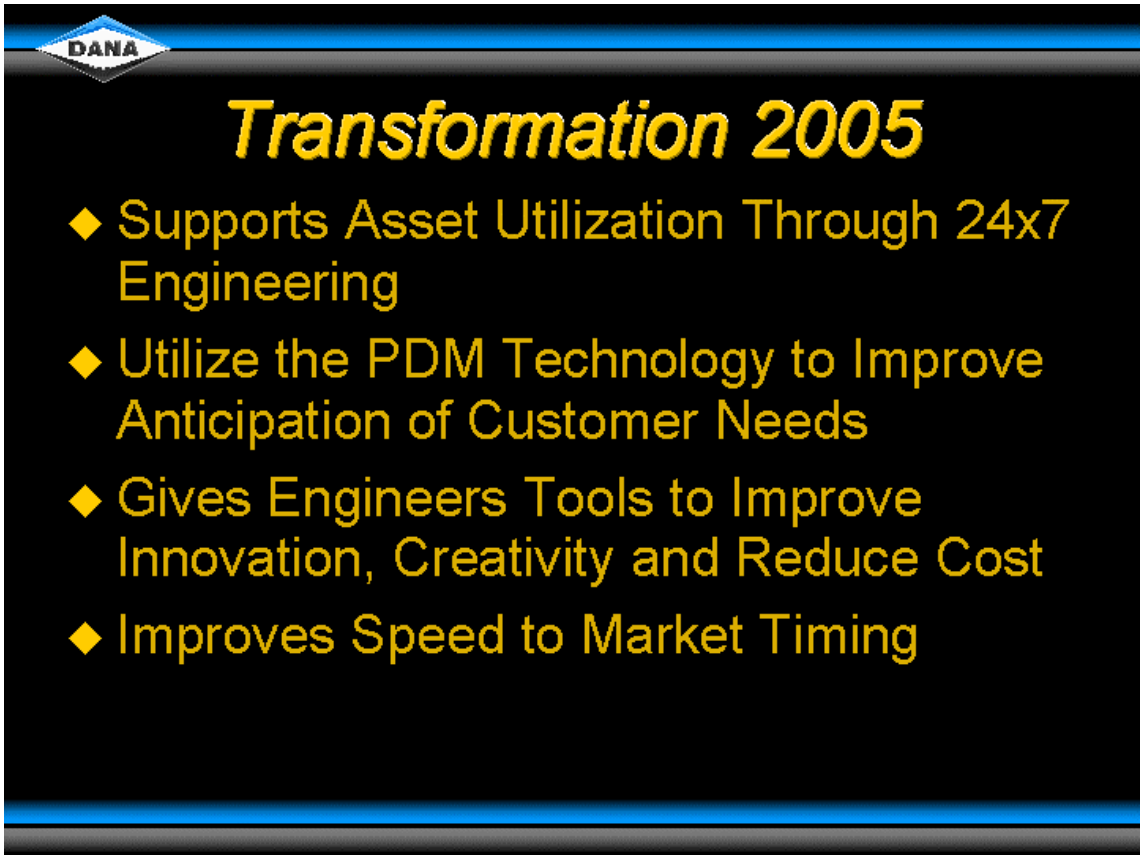
【PDM の導入状況と今後について】

PDM システム導入の第 2 フェーズにおいては、主に製造、品質、販売、マーケティング、調達などの分野に注目し、それぞれの部門の間にあった情報の壁を取り除き、情報統合を進めることにより、最終的には製品開発方法の統合を実現していきたい。例えば品質管理グループが品質チェックを行う際、エンジニアリンググループが作成した立体データを活用することなどが可能となる。また将来的には、MRP・ERP などの基幹システムとの間での統合も行っていきたいと考えている。これらの情報統合により、サプライヤーとの間のコラボレーションをさらに進めて行ったり、またマーケティングなど他の業務においてエンジニアリング情報を活用することが可能になっていくだろう。

現在の PDM 導入状況であるが、現在エンジニアリングプロセスに注目した情報統合を行っている。特にワークフローの部分でも、デザインワークオーダー、デザイン分析、ギア・スプライン分析、ノイズ、振動、プロトタイプ、資材分析、テスト、エンジニアリング変更などに注目している。さらに、これらのプロセスを行うことによって発生する技術スペック、顧客プログラム、デザイン検証、計画、レポートなど様々なドキュメントの管理にも注目している。現在はこのようなドキュメントは、異なるプロセスごとに別々に保存されているが、最終的にはこれらのドキュメントが、いかなるプロセスにおいても活用できるようになる。

【TeamCenter デモンストレーション】
PDM の一環として、Dana 社では EDS 社が開発したデータ管理システム「TeamCenter」を利用している。デモンストレーションでは、エンジニアがある部品の耐久度を変えるよう、デザイン変更の指示を行う手順が紹介された。まずユーザは、ユーザ名、パスワードを利用してログインを行う。ユーザのレベルによって、利用される機能の制限などを行うことが可能となっている。ここでは、ユーザはまずデザイン変更の指示をテキストベースで作成、またこの特定の業務がいつまでに行われるべきかなどの情報を入力、保存する。またデザイン変更の指示は、文章だけでなく、実際に変更が行われるべきパーツのソリッドモデルを利用しても行われる。ユーザは製品番号などの情報をもとに、パーツの図面情報の検索、取り出しを行うことができる。ユーザはこのソリッドモデル上に、ビジュアライゼーションツールを利用し、どの部分の変更が行われるべきなのかを、矢印や文章などを利用して書き込み、デザイン担当者にこれらの指示情報を送付する。
このように、デザイン変更に関する指示を、文章で示すだけでなく、実際の図を示して説明することで、コミュニケーションがよりスムーズになる。また現在のライフサイクルにどのようなオーダーが出されているか、それぞれのタスクは誰が担当しているのか、またタスク終了までにどれだけかかるのかといった情報をトラッキングする機能もついているとのことである。

現在 Dana 社では、同社会長を中心に、Dana 社が将来どのような企業に成長したいかの指針を示す「Transformation 2005」と呼ばれるイニシアチブを行っているところである。PDM システムはこの「Transformation 2005」の目標達成をサポートするためのツールとして捉えられており、Dana 社は E コマース利用を積極的に行っていることは間違いない。



4. バーチャルファクトリーについて

Advanced Technology Research Group は、Dana 社技術センターにおけるエンジニアリンググループの一つである。Dana 社にはそれぞれの製品開発をサポートする技術センターが複数あるが、このグループは、特定の製品やシステムのみに焦点を置くのではなく、Dana 社全体における技術サポートを行っている。

エンジニアリングの世界では、３D の CAD モデルを利用することは長年のスタンダードとなっているが、製造部門においては、工場のフロアデザインなどは未だに二次元の図面が利用されている。そこで同グループでは、AutoCAD ソフトウェアを利用し、工場フロアの図面も２D、３D 両方で作成できるようにし、またこの「バーチャルファクトリー」を作成するための一貫したプロセスを作成した。通常、製造施設において新たなフロアプランを設計した場合、機材を搬入して設置した後で初めて何らかの問題があることがわかり、また機材の移動を行わなければならない、などの問題があった。しかし３D モデルを利用することで、実際に工場で業務を行う人々が、機材の設置を行う以前に問題がないかどうかを確認することができる。ここで利用されている閲覧ソフトウェアは、PDM で利用されているものと同じであるため、最終的にはこのような情報をも PDM に組み込み、製造や開発などに役立てていきたい。

このイニシアチブそのものは 1997 年に開始されたが、それ以前には 25 万ドルのグラフィックソフトウェア、4 万ドルのバーチャル・リアリティソフトウェアなどが利用されていた。しかし現在利用しているツールはこれらのソフトウェアよりも格段に安い値段で、また PC 上で利用することができるため、技術的にもコスト的にも利用しやすいものとなっている。このように工場フロアの設計をビジュアライズ化することにより、メンテナンスや実際に工場でのオペレーションを行う従業員全てが、共通の理解のもと業務を進めていくことができる。工場フロアのほかにも、パーツの組み立て方法や、組み立てプロセス、ビルの概観なども、全て同じソフトウェアを利用し、アニメーションとして提示することができる。

5. B2B E コマースについて

同社では特に MRO の調達を行うための B2B イニシアチブを 2000 年から行っている。当初は、世界中に展開する Dana 社施設全ての調達をサポートしてくれるサプライヤーとの取引を行うことを希望していたが、世界レベルでオンラインサポートで行ってくれるようなサプライヤーは存在せず、結果国ごとに違うサプライヤーと契約を結ぶことになってしまった。間接材調達をオンラインで行う目的としては、カタログから必要な商品を検索する時間や注文処理などの時間の短縮、また購買力を高め、ディスカウントを得ることができるなどを期待している。

米国では Grainger 社を始めとする MRO ベンダーとの契約を結んでおり、コスト削減においてある程度の効果を見出している。しかし海外に目を向けてみると、それほど大きな効果をあげているわけではない。特に、最初はグローバルレベルでのサプライヤーからのサポートを期待していたわけだが、それが実現しなかったため、海外ではそれぞれが違うサプライヤーとの取引を行うことになってしまい、これがシステムを複雑にしてしまう結果となっている。このため、コスト削減も期待したほどは実現しなかった。

B2C に関しては、アフターマーケット製品などに関して顧客にオンライン販売を提供するようなイニシアチブを行っている。

また一方、オークションも 2000 年から 2001 年にかけて試してみたが、あまりうまく行かなかった。これは、オークションを通じて安価な製品を納入してくれるサプライヤーを見つけることができたとしても、その新たなサプライヤーの背景調査や品質調査、信用度の調査などを行うためのコストが逆にかさんでしまったためである。

6. CAD システムについて

Dana 社では、顧客である自動車製造企業にあわせて CAD システムを使い分けている。これは顧客企業のリクエストによるものである。Dana 社内でも、異なる部門においては異なる CAD システムを利用している場合が多い。このため、これらの情報を、顧客企業が指定する CAD フォーマットに翻訳する、という作業が行われてきた。しかし最近では、顧客企業が製品開発を行う最初の段階から、顧客企業が指定した CAD フォーマットを利用することを要求してきている。サプライヤーとの間で共有される CAD 情報に関しても、どの顧客企業のためのパーツを製造するかによって、特定の CAD フォーマットを一貫して利用するようにしている。CAD データの翻訳については、Dana 社がインドの企業と共に立ち上げたジョイントベンチャーが全て請け負っている。

7. 顧客企業との間の PDM 接続について

フォードや GM 社などの顧客企業との間での PDM 接続を行っている。しかし全てが自動化されているわけではない。全ての情報を顧客やサプライヤーとの間でオープンにすることは、セキュリティ認証の問題のほかにも、知的所有権の観点からまだ問題が多い。このため PDM の相互接続を通じ、必要な情報の送付などの業務の一部分を自動化しているにすぎない。Dana 社にとっての利点は、これらの企業との間での連絡コストを削減できる点である。

顧客企業、サプライヤーなどを含めたバーチャル企業となる取り組みについては、例えば顧客企業が新たにネットワークなどを構築する際、Dana 社からも構想の段階からアイデアを求めてくるなど、協力しあう関係になっていると思う。最終的には、大枠は顧客企業が決めるが、さらに我々がサプライヤーからの要望なども吸い上げてトップダウン、ボトムアップ両方の方法でコラボレーションのための枠組みを作れるようになれば一番理想的だと思う。

以　上

禁　無　断　転　載

設計・製造データの活用と流通に関する調査研究
製品表現のための先進技術動向研究報告
平成　１５年　３月　発行

発行所　財団法人　日本情報処理開発協会
電子商取引推進センター
東京都港区芝公園３丁目５番８号
機械振興会館　３階
TEL：０３（３４３６）７５００
印刷所　株式会社三菱電機ドキュメンテクス
東京都中央区湊３丁目５番 10 号
TEL：03-5566-0681

（本報告書は再生紙を使用しています。）　14-E009