資　料

わが国が行う情報技術研究開発のあり方に関する調査研究（その７）

平成15年3月

財団法人　日本情報処理開発協会
先 端 情 報 技 術 研 究 所

KEIRIN この事業は、競輪の補助金を受けて実施したものです。

わが国が行う情報技術研究開発のあり方に関する調査研究（その７）

目　次

まえがき

先端情報技術研究所が平成7年10月に設立され、翌年の平成8年度から、「わが国が行う情報技術研究開発のあり方」に関する調査研究が開始された。それから、早や8年が経過した。

この8年間は、IT革命が始まり、そして、その第一段階が終結するという期間であったと考えられる。そして、今、IT革命の第二段階が始まろうとしているように思われる。

この8年間に、インターネットが急速に普及し、それによって地球はいろいろな意味で小さくなった。グローバルマーケットの成立は企業活動に大きな影響を与えた。生産者と消費者間の距離が縮まり、ユーザニーズを素早く捉え製品に反映することが企業生き残りの条件となった。時間と距離の短縮は、製造業の部品調達の世界市場への拡大と下請け構造の崩壊を招くなど、産業構造の変革をもたらした。デジタルエコノミーの世界の実現である。また、その影響はわれわれの身辺にも及び、行政サービスのインターネット化、電子政府の構築などが世界各地で進行した。

そして、今、インターネットはさらなる進歩を遂げようとしている。その一つがグリッド・コンピューティングである。電力網において、電力を余裕のある個所から不足する個所へ送電し、余剰電力を有効利用するのと同様に、インターネットに接続されたコンピュータ群を計算資源と見て、その余剰計算能力を融通しあう仕組みが実現しようとしている。

また、ソフトウェアの開発方法として、ソース・プログラムを公開し、多くのソフトウェア研究者、技術者が、半ばボランティア的に集まり、ソフトウェアを開発するというオープンソースソフトウェアプロセス（OSSP)という手法が広く行われるようになった。Linuxやグリッド・コンピューティングのためのミドルウェアもそのような方法で作られている。

このようなインターネットの普及やソフトウェアの開発方法の進歩の中で、わが国は情報先進国の地位から脱落してしまった。世界の最先端からの遅れは、縮まるどころか拡大方向にある。その上、従来は情報技術の後進国であったアジア諸国、特に、中国の台頭が著しい。わが国もe-Japan計画などを実施し、国を挙げて、ソフトウェアやIT技術の研究開発を加速しようとしているが、その効果はあがっていない。

「わが国の情報技術の研究開発のあり方」の調査を進めるに当たっては、米国を中心に、その国のIT関連技術開発の仕組み、法制度を調査した。まず、米国のIT関連研究開発を、その開始から、発展段階、そして、その成果の商品化、さらにそれに基づく市場創成までの段階を追跡した。

また、米国の産業の新陳代謝を進める仕組み、法制度も調査し、わが国の仕組み、法制度と比較することで、わが国の抱える問題点を明確化し、その改革提言を行った。

昨年までの調査により、米国では、優れたアイデアを捉え、研究開発をスタートさせ、発展

させて、得られた成果を商品化し、市場を創出するという、一連の流れを省庁連携によって、連続的に支援する構造が存在することを明らかにした。

われわれは、この構造をフロントランナー構造と呼ぶこととし、この構造こそが米国をして、ITやバイオテクノロジーなどの先端技術領域を一早く開拓し、先端産業の覇者たらしめるものであると結論づけた。

わが国の研究開発は、このような省庁間連携はもとより、産学官の連携も効果的に行われておらず、また、研究の管理方法や予算使用上の制約も多く、その上、人件費に間接費を算入できないという、米国の合理的な仕組み、法制度とは、比べることができないほど時代遅れのものであることが判明した。また、成果管理規則についても、箱物作り時代の制度が生きており、ソフトウェアや IPR などの無形物が大きな付加価値を持つ時代には全くそぐわないものであることもわかった。

このような研究開発環境では、いくら研究者が努力しても勝敗は明らかで、このような調査結果を得たことで、本調査研究の目指した、わが国の情報技術の研究開発投資に対する成果の少ないこと、効率が上がらないことの主要な原因を究明し得たものと考えた。そして、これにより、当初の調査活動も一段落したものと考え、本調査事業を本年度で終了することとした。また、同時に研究所もクローズすることとなった。

今後は、このような調査結果を基とする改革提言の実現のための努力が残されている。できるだけ多くの関係者に上記の調査結果と改革すべき点を知ってもらうことがその第一歩である。

本年度は、本調査の最後として、このようなフロントランナー構造の実現上、最も実現が難しいと考えられる省庁間の協調と連携について、米国の仕組み、法制度をさらに調べることとした。また、わが国のソフトウェア産業に将来大きな影響を与える可能性のある中国のソフトウェア産業や市場、およびソフトウェア技術についての調査を行うこととした。

先端情報技術開発の動向については、例年どおり、高速コンピューティングと通信についての調査 WG（HECC-WG）と、人間主体の知的システムについての調査 WG（HCIS-WG）により、調査を実施した。特に、上で述べたインターネットの概念拡張を伴う発展形であるグリッド・コンピューティングについての調査、および大規模化、複雑化が進むソフトウェアの生産性向上を基礎理論から追求し、新しいソフトウェア工学を構築することを意図した NITRD 計画の新テーマ、Software Design and Productivity (SDP)に注目し、その現状を調査した。

本調査報告書は、当研究所の作成する最後の報告書となる。そこで、簡単に本調査事業を開始した動機と実施体制、そして調査結果の概要を最初の章にまとめておくこととした。

IT に限らず先端技術開発に関する先進国の地位をとり戻し、技術立国を可能とするためには、常に先進諸国の研究開発動向やその仕組み、法制度について継続的な調査を実施し、研究投資

の増額や人材育成と同時に、仕組み、法制度も進化させることが不可欠である。

当研究所が実施してきたような調査や改革提言を行う組織が再び作られることを願うものである。

第１章

国が支援する情報技術研究開発のあり方

第１章　国が支援する情報技術開発のあり方

1.1　本調査開始の動機と従来経緯

1.1.1　調査目的と方針

当初の調査目的は、平易に言えば、次のようなものであった。すなわち、「わが国のソフトウェア技術の研究開発やソフトウェア産業育成のための投資は、通産省関連だけで毎年200億円近くある。それにも拘わらず、十分な成果が挙がっていないように見える。これは研究開発や産業育成の仕組みに何か問題があるのではないか。その問題の究明と改革提言を実施せよ」ということであった。また、「調査は、国や通産省に遠慮せず、悪い点ははっきり悪いと書くタブー無しの方針で実施せよ」との条件もついていた。

このような要請に基づき、当研究所は次のような方針を立てて調査を実施することとした。

1）ソフトウェアやIT分野において先頭を走っているのは米国である。従って、まず、米国の連邦政府の実施するIT研究開発計画（R&D Initiative）について、その実施の仕組み、法制度を調査し、わが国のそれと比較する。この日米比較により、わが国の仕組み、法制度の問題点、改革すべき点を明確化する。

2）わが国の国のプロジェクト（ナショプロ）を実施した経験や、国の支援を受けて研究開発を実施した経験のあるメーカやソフトウェア企業の担当者を集め、彼らの意見を求め、国の仕組み、法制度の問題点、および改善策を調査する。また、ナショプロや国の支援の仕組みなどの領域を越え、わが国の大学、国研のあり方、省庁縦割りの弊害などについても、同様に調査することとした。要するに、Customer Satisfaction (CS)の調査を行うこととした。

その後、「国が投資すべきIT関連の重点分野」についての調査が追加された。ソフトウェアに関して、将来予測を伴う重点分野の調査は難しい。半導体のようなハードウェアやいわゆる箱物と呼ばれるものは、アイデアのみでは製品は出来ず、周辺技術や製造技術も合わせ持たなければならない。これらをすべてそろえるには、一定以上の時間や資金を必要とする。このため、LSIの集積密度の進歩の類は、従来の進歩のスピードを延長したロードマップ上に乗ってくる。

これに対して、ソフトウェアや特許のようなアイデアがすぐ製品となるものは、いつどのようなものが現れるかの予想が困難である。また、今後、発展方向の予測も難しい。将来に向け

ての重点分野の予測は、広く薄く投資し、どのような芽がでるかを観察し、大きく成長しそうな芽がでたら、それに投資を集中するのが次善の策である。これは、米国において、NSF や NIH が多数の研究者にグラントという小額の投資を行い論文を発表させる手法と同様のものである。

わが国の IT 研究開発の仕組み、法制度は、大きなソフトウェア市場に発展するような新概念や新応用を産み出すのに適したものには進化しておらず、いわゆる箱物（固いハードウェア）開発を前提としたものとなっている。従って、世界の誰かの網に魚がかかったかどうかをいち早く知ってこれを追いかけて投資をするキャッチアップ型がよいと考え、重点分野の調査を実施した。

従って、この調査に関しては、世界の先端 IT 分野の動向を第一線の研究者を集めて随時フォローしてゆくほか、米国の NITRD 計画（当初の HPCC 計画）の進捗をフォローし、これをもとに重点分野を選択した。

1.1.2 調査体制

当研究所のマンパワーは、所長、事務管理者３名、経理担当者１名、研究員６名の体制であった。所内の研究員のみでは、上記の調査範囲をカバーするのは困難であった。このため、メーカやソフトウェアハウス、大学、国研などの研究者の協力を仰ぎ、次のような委員会や WG を設けた。集まってくれた研究者の持つ知識や彼らの議論により調査データの収集、分析を行った。

１）技術政策委員会

委員長：故水野幸雄氏（NEC 顧問）、メンバー：メーカ６社の技術担当役員、または
それに準ずる技術企画担当者、通産省担当者等

委員会の目的：

－各年度の研究所調査テーマ、計画などの審議、

－前年度の調査結果のヒヤリングや提言についての審議

－国の IT 政策などの紹介、質疑応答

２）研究開発体制のあり方検討作業委員会

委員長：後藤滋樹氏（早大教授）、メンバー：メーカ＋ソフトウェアハウスの
研究所長、IT 関連事業部の企画担当者

委員会の目的：

－ナショプロの実施体制、予算や成果の管理体制などについての問題点や

改善すべき点などの議論

－プロジェクトの実行体制や予算制度の日米比較の議論

－わが国の IT 研究開発政策や投資についての問題点の議論

－わが国の国研や大学の体制や連携に関する問題点の議論

3）先端 IT 技術分野の動向調査 WG

a)High-End Computing and Communication WG (HECC-WG)

委員長：山口喜教氏（筑波大教授）、メンバー：大学の若手教授、電総研の中堅研究員、メーカ研究所の課長クラスなど、第一線の研究に従事している者

b)Human　Centered Intelligent Systems WG (HCIS-WG)

委員長：奥乃　博氏（京大教授）、メンバーは上記 WG と同様

これら WG の目的：

－各委員の専門分野周辺における内外の技術動向調査

－国のプロジェクトとすべき研究開発テーマや開発目標の議論

このほか、米国の研究開発の仕組み、法制度に関して、米国の調査会社や NSF などを退職したプログラムマネージャなどに調査を依頼し、研究の実施方法や研究予算、成果の管理方法などを詳細に調査した。

このような３つの階層から成る委員会／WG を構成し、日米比較に基づく議論をもとにわが国の研究開発の仕組み、法制度の問題点や改革提言を明確化し、各年度の報告書を作成した。

特に、米国の連邦政府支援の研究開発計画（R&D　Initiative）の仕組み、法制度は、わが国のそれと比較し、はるかに進歩しており、わが国のメーカ、国研、大学の研究者は、米国の研究者に比べ、研究管理や予算の使途に関し制約が多く、事務処理負担も重い不利な環境におかれていることが明らかとなった。

1.1.3　調査結果、および改革提言の広報と実現に向けた活動

本調査活動においては、単に、報告書を発注者（スポンサー）である通産省やメーカ６社に提出するのみならず、調査結果や改革提言の広報と実現に向けた活動が求められた。このため、当初から実施していた、当研究所のホームページへの報告書やスライド資料などの掲載に加えて、平成 11 年度より、次のような広報活動を実施した。

1）調査報告書の配布（約 300 部）

－通産省、文部省、内閣府などの IT 研究開発計画や予算を扱う部局

－NEDO や IPA などのファンディングを実施している政府機関
－大学において情報やソフトウェアなどを専門とする研究者（教授）
－技術マスコミ（新聞記者など）
－IT 関連メーカ、ソフトウェアハウスなどの代表者、技術開発担当者

2）口頭発表、説明会の実施
－AITEC セミナーの開催（毎年実施、通常 150 名程度が参加）
－通産省の関係部局での説明会
－学術会議（50 周年記念講演会や関連小委員会）
－情報処理学会全国大会　e-Japan 講演会など
－規制改革会議ヒヤリング
－そのほか、各種の招待講演

3）学会誌、新聞、雑誌などへの発表
－情報処理学会本誌解説論文
「国の資金による IT 研究開発における仕組みや法制度に起因する
研究環境の日米格差について」2000 年 10 月号
－新聞記事掲載（日経新聞、経済産業新報など）
－情報関係雑誌など

このような広報活動により、米国の連邦政府が実施する研究開発計画の仕組み、法制度と比べ、わが国のプロジェクトの仕組み、法制度がいかに制約が多く、かつ研究開発の現場への権限委譲が不充分であることが、関係者にかなり広く知られるようになった。

特に、米国ではファンディングする省庁側にプログラムマネジャー（PM）と呼ばれる研究者がおり、この PM に、研究目的、実施方法や予算の使途変更、人の雇用、研究開発成果の処分方法、プロジェクトの評価などの権限が委譲されており、ファンドを得て研究する研究者は、この PM と電子メイルや電話で交渉し了解を得ることで、分厚い書類などを作成することなく各種の変更ができることや、費目の流用や予算の繰越しが容易に行えることが、大学や国研の研究者に知られるようになった。

また、会計制度に関しても、複数年度会計であり、繰越などが、PM の了解で容易にできること、間接費が国の費用に算入できること、ソフトウェア開発において企業が所有していたソフトウェアを組み込んだ時、その費用が算入できることなども知られるようになった。

このような米国の制度の合理性は、独立行政法人化を控えていた国研や大学、文部省関係者、さらには、内閣府の総合科学技術会議事務局の関係者などには、タイムリーな情報として使ってもらえたと思われる。実際、一部機関でのPM制度の採用、科研費への間接費（最大30%）の導入、競争的資金による人の雇用の自由化、日本版バイドール法の適用などがかなりの大学や研究機関で実現した。

しかし、残念ながら、複数年度会計、国の研究費についての間接費の導入、民間会計の導入など、財務省の権限に属するものは全く手付かずである。わが国の公会計は特殊であり、人件費について間接費を認めていない。国のプロジェクトが箱物作り主体で人件費の割合が少ない時代にはあまり大きな問題ではなかったかもしれないが、現在のようにソフトウェア開発がプロジェクトの大半を占め、費用のほとんどが人件費となっている状況では、間接費持ち出しの「裸の人件費」は、中小、ベンチャー企業の国のプロジェクトへの参加を阻害する大きな要因となっている。

このようなわが国の時代遅れの仕組み、法制度のために生じる研究環境の格差は、わが国の研究投資の効率を悪化させ、研究者のオーバヘッドを増やす結果となっている。

以上のようなわが国の仕組み、法制度の問題点が、多くの関係者の知るところとなり、改善への動きが活発化することとなった。

1.2 本年度の調査方針と調査テーマ

本年度は、本調査の最終年度であることを意識した調査テーマを選択した。以下、その調査の背景と目的を紹介する。調査結果の概要は本報告書の第2章以下にまとめた。また、その詳細は、別冊の報告書を参照されたい。

1.2.1 米国の連邦政府のR&D計画における省庁間の役割分担と連携の仕組み

昨年度の調査では、米国の連邦政府の研究開発支援は、単に研究開発段階に留まらず、研究成果の商品化や起業の支援、さらに、市場創成の支援と途切れなく続く「支援の連鎖」を作り出していることを述べた。そして、それが米国産業の新陳代謝と常に高付加価値の製品を生み出す企業群の創成を可能としていることを示した。この「支援の連鎖」を昨年度調査では、「フロントランナー構造」と呼んだ。

この構造の先端は、アイデアを産み出す若手研究者を抱える大学と、それら研究者のアイデアを基礎研究としてスタートさせる「グラント」と呼ばれる小額だが使途自由の資金を提供するNSFやNIHである。グラントの成果は論文であり、論文発表により、アイデアが多くの研究者に知られ共有される。

その後、優れたアイデアは多くの研究者により、さらに発展させられる。この段階での支援は、DOEやNASA、DARPAが行い、さらに、実用化、商品化にむけて研究開発が継続される。もちろん、その途中段階では激しい競争や淘汰がある。商品化、または起業段階では、SBIRやSTTRという支援の仕組みがある。この支援は、SBIRについては10の省庁が、STTRについては5つの省庁が競い合いながら実施している。

米国ではこのような省庁間連携が連邦政府の研究開発計画では広く行われている。米国最大のIT研究開発計画であるNITRD計画も11の省庁が参加し連携している。このような研究開発の発展段階の一部、もしくは全部について、複数の省庁が連携して支援し、「支援の連鎖」が作られ、それが小さなアイデアから、一大市場を創成するという流れを形成している。

わが国の常識に照らして考えれば、各省庁は、常に予算のぶんどり合戦や、新しい分野については縄張り争いを演じるとの認識がある。研究開発や調達に関しても、省庁間には壁があり、同じようなテーマの研究開発や物の調達が、ばらばらに行われ、ほとんどの場合、その間に協調や競争の仕組みはない。

米国には、大統領府にNSTC（国家科学技術会議）、OSTP（科学技術政策局）、OMB（管理予算局）などの、省庁の上に立つ調整組織がある。これら組織は、各省庁の計画する研究開発テーマを監視し、類似テーマは一まとめにして代表するPM（プログラムマネジャー）も一人に絞り、協調や競争を起こさせる。こうすることで、研究開発の効率とスピードアップを図っ

ているといわれている。また、このような調整により、各省庁にもメリットがあるとも言われている。NITRD においても、NCO（国家調整局）がこの計画全体を管理し、研究の無駄な重複を防ぎ、効率向上に大きな貢献をしているといわれている。

本年度は、米国の研究開発における「支援の連鎖」について、なぜ各省庁にもメリットがあるのかなど、さらに深い調査を実施することとした。

この調査報告では、我々が昨年度の調査で、「フロントランナー構造」と呼んだ「支援の連鎖」は、「イノベーション・パイプライン」と呼ばれている。この調査結果によれば、米国においても利害の対立する省庁間の調整は容易ではなく、失敗例も多いとのことである。

しかしながら、未踏領域へのチャレンジでは、複数の省庁が協調し、多くの予算や人を投入することで、より多くの可能性を追求でき、国としても、参加する省庁としても、単独で行うより、多くのメリットを享受できるとのことである。

「わが国の情報技術研究開発のあり方の調査」の最後の提言は、現在、ばらばらのわが国の省庁間の研究開発を連携させることであろう。それによって、わが国も国全体を挙げての「イノベーション・パイプライン」を実現し、研究開発の効率改善やスピードアップを図りたいものである。

本調査については、本報告書の第４章に概要が、また、調査資料：「米国の連邦政府 R&D 計画における省庁間の役割分担と連携の仕組み」に詳細が報告されている。

1.2.2　先端情報技術の動向調査

先端情報技術の研究開発動向については、例年通り、次の２つの WG により調査を行った。

a)高速コンピュータと通信（High-End Computing and Communication）WG (HECC-WG)

b)人間中心の知的システム（Human Centered Intelligent Systems）　WG (HCIS-WG)

これら WG では、各委員の専門分野を中心として、新しい動きを追っているが、本年度は、中心的話題として、HECC-WG では、グリッド・コンピューティング、HCIS-WG では、Software Design and Productivity(SDP)を取り上げている。

1）グリッド・コンピューティング

グリッド・コンピューティングは、高速ネットワークに接続された、スーパーコンピュータのような強力な計算パワーをもつコンピュータ群を一まとめにして、さらに強力な仮想的コンピュータを実現しようというアイデアに端を発している。

これは、電力網（Electric Power Grid)が、余剰電力のある地域から、多くの電力を必要とする地域へ送電し、融通しあうことで多量の電力を供給する事例から、発想したと言われている。

このようなコンピュータの余剰計算パワーを有効利用するためには、使用者が実行したい大

規模計算を分割し、多くのコンピュータへ分散させ、計算を並列に実行できなければならない。この実現のためには、使用されるコンピュータが同じ顔（インタフェース）を有しなければならない。

グリッド・コンピューティングの研究開発は、この同じ顔にみせるためのミドルウェア開発がその中心となる。また、異なるハードウェアやOSを持つコンピュータ群を同じ顔にみせるための共通仕様の決定が主な仕事となる。

2000年11月に共通仕様を決めるための世界的標準化団体、Global Grid Forum(GGF)が設立された。グリッド・ミドルウェアはオープンソース方式で開発が行われており、多くの研究者が参加するようになった。

また、インターネットに接続されたPCを数百万台用い、地球外生命探索のための計算を行う実験が行われたり、その他、多くの科学技術計算にグリッドが利用されるようになった。

2002年には、IBMをはじめとする大手メーカがグリッド・コンピューティングへの本格的参加と実用化を目指すことを発表したことから、急速に注目を集めるようになった。わが国においても、平成15年度から3年間の計画でサイエンス・グリッド、およびビジネス・グリッドのプロジェクトが実施される計画である。

グリッド・コンピューティングは、これまで別々の要素と考えられてきたネットワークとそれに接続されている多くの異なるコンピュータのインタフェースを共通化し、コンピュータの一部分を一まとめにして仮想的な一台のコンピュータとして利用可能とできる点が革新的である。このような仮想的構成をVirtual Organizationの機能と呼んでいる。

異機種や異種OS、異種データベースが全て同じインタフェースを有することで、従来のメイルにより交信したり、ホームページを見たり、個々のデータベースをアクセスする段階から進んで、多数のコンピュータやデータベースを一まとめにして使うことができる。これにより、あたかも強力な一台の仮想コンピュータや、いくつものデータベースを統合して多くの知識を加え合わせた仮想的な大規模データベースが構成できることになる。ネットワーク上にある計算資源やデータなどの資源を活用する自由度が大幅に増加することになる。

もちろん、インターネットのようなものの上で、このような仮想的環境を作るためには、機密保持や信頼性、安全性の確保などの問題が解決されねばならない。しかし、世界中に支店網を展開している会社内部のイントラネット上において、サーバー間の負荷の集中時に余剰計算パワーを融通したり、故障時のバックアップを行うことなどが容易となる。「2005年には、ビジネス分野の売上げが科学技術分野の売上げを追い越す」ともいわれている。

このような状況の中で、2002年度より、NITRD計画は、グリッドのミドルウェアやツールのサービスを目的とする調整部会、Middleware and Grid Infrastructure Coordination(MAGIC)を設けた。その役割は、NITRD計画に参加している各省庁のグリッド・コンピュー

ティング開発計画を調整し、さらに米国内外のグリッド研究の協調を図ること、それにより、現在も米国中心で進んでいるグリッド・コンピューティング研究のバックアップをさらに強化することを意図しており、米国の戦略のしたたかさを感じる。

このようなグリッド・コンピューティングに関する動向については、その概要が本報告書の第２章にまとめられている。また、HECC-WG の報告書を参照願いたい。

２）Software Design and Productivity (SDP)

SDP は、1999 年２月の PITAC 勧告によって、NITRD 計画の Program Component Area (PCA)として追加されたテーマである。その目的は、現在のソフトウェアの抱える問題は深刻であり、信頼性は低く、安全性、可用性の面でも、きわめて不充分であるとの認識の上に立ち、今後、ますます、大規模、複雑化するソフトウェアの生産性を向上するための技術を理論レベルから再構築しようとするものである。

このため、多くの研究者の知恵を集める目的で、ワークショップなどを開催し、その研究開発計画の立案を公開で行っている。このワークショップでの現状認識や将来に向けての提言は、きわめて多岐にわたる興味深いものである。

また、このような議論を通して得られた研究開発方針が Blue Book　2003 にまとめられている。ここでは、現世代のソフトウェアは、技術的にきわめて脆弱であり、われわれは、その開発とメンテナンスに巨大な資金を投入している。しかし、相互運用性が無く、拡張性も無く、また、コスト・パーフォマンスもよくない。これまで技術者は、形式的枠組みを持たず、詳細な設計図、関連する基本原理も持たず、訓練で培われた知識、品質保証された構成部品、コスト・パーフォマンスのよい開発プロセスなども持たなかった。つり橋を架けるような方法でソフトウェアを作ってきた。

NITRD の研究では、ソフトウェア開発のための新しい科学的モデル作りや、試験や評価方法に重点をおき、特異的な少数の人しか理解できないコードから脱却し、合理的でモジュラリティがよく、再使用可能な工学的設計への移行を目指す。

このためには、自動化された技術を使い、ソフトウェア製品の設計、構築、試験、検査を事前に可能とすることが必要である。何百万ステップもの大規模ソフトウェアをその利用以前に検査し、そこに潜む弱点を正確に指摘することが可能となるような技術を目指す。

というわけで、科学的基盤の構築から、理論や新知識、技術を積み上げて行こうという基礎研究を中心とする大構想が述べられている。しかし、なんといっても米国のプロジェクトであり、このような大構想だけでなく、すでに、一部で実用化されている将来性のある技術や、オープンソースソフトウェアプロセス（OSSP)などの利用についても研究を進めている。

このような将来に向けてのソフトウェアの基礎研究プロジェクトは、わが国では全くと言っ

てよいほど行われていない。

SDP の現状調査に関しては、第 3 章でかなり詳しい報告を行っている。

1.2.3 わが国 IT 開発拠点の中国移転に関する調査

最近、わが国企業の中国進出が大きな動きとなっている。IT 産業も例外ではない。特に、ハードウェア製品の製造拠点の中国移転は、安価な労働力、関連産業の集積効果、近い将来に作り出されるであろう大市場への期待などが動機となっていると考えられる。

これまで、本調査では、IT 研究開発の先頭を走る米国を中心として、研究開発投資、仕組み、法制度、技術開発動向などを調査してきた。そして、その結果を基に、日米比較の観点から、わが国の抱える情報技術研究開発の仕組み、法制度、重点投資分野などについての問題点を分析し、キャッチアップのための改革提言を行ってきた。

ところが、わが国の IT 機器ベンダーや大手ソフトウェアユーザ企業が、コスト低減を主目的に、中国のソフトウェア企業への業務委託などを行うようになってきた。現状では、中国のソフトウェア技術は、平均的に見れば、未だ、わが国のソフトウェア企業に及ばないといわれている。しかし、中国のソフトウェア企業が、わが国ソフトウェア市場で一定の競争力を持つようになっており、既に、中小の日本のソフトウェア企業とは競争関係に達しているとも言われている。

この傾向が進み、中国ソフトウェア企業の技術力が向上すると、研究開発分野など高付加価値領域にも遠からず影響が及ぶ可能性がある。その結果、ソフトウェア産業においても、産業の空洞化がおこることを懸念する声もでるようになった。

このような背景から、前門の米国だけではなく、後門の中国についても、そのソフトウェア産業の技術力や市場について、状況把握することが必要になったと考えるに至った。以上の理由から、中国の IT やソフトウェア産業の現状、ソフトウェア市場、日本のソフトウェア企業の中国進出とその実態、中国ソフトウェア企業と日本ソフトウェア企業の連携などについて調査を行った。調査は文献調査のほか、日本と中国のソフトウェア企業関係者へのインタビューにより行った。

この調査結果は、本報告書の第 5 章に概要がまとめられている。詳細は、調査資料：「わが国 IT 開発拠点の中国移転に関する調査」を参照されたい。

第２章

米国ハイエンドコンピューティング技術の研究開発動向

第２章　米国ハイエンドコンピューティング技術の研究開発動向

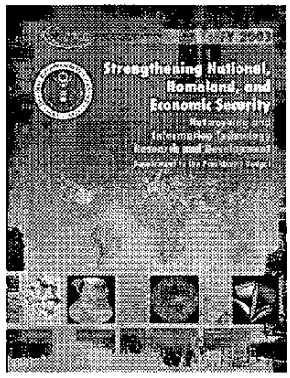
2.1　FY2003 Blue Book にみる政府支援の IT 研究開発

2.1.1　NITRD 計画の推進体制

米国政府による情報技術(IT)研究開発において重要な役割を果たしているのが、1992 年に開始された High Performance Computing and Communications (HPCC) 計画を起源とする Networking and Information Technology Research and Development (NITRD) 計画である。NITRD 計画の特徴は、大統領直属の組織体制により立案、管理される省庁横断的な IT 研究開発プログラムという点にあり、このことが、国家としての戦略的な IT 研究開発の実施に大いに貢献している。

通称 Blue Book と呼ばれている「IT 研究開発 (NITRD) に関する大統領予算教書補足資料」は、毎年発行され、NITRD 計画の活動状況の概要をまとめている。この Blue Book の内容を調査することにより、現在の米国政府の IT 研究開発がどのような技術分野に着目し、どのような方向に進んでいるかを知ることができる。

2002 年 8 月に公開された FY2003 (2003 年度版) Blue Book:"Strengthening National, Homeland, and Economic Security"は、同時多発テロ以降の米国の立場を色濃く反映したものになっている。ここでは、政府支援による IT 研究開発の最重要目的は「米国の国家、国土および経済の安全保障の強化」のためであり、巻頭の 10 ページにわたって、NITRD 計画により開発された先進技術が国家、国民および経済の安全の確保にいかに貢献しているかが強調されている。ただし、内容を読む限り、その研究内容は従来からそれほど変化があったようには見えない。同時多発テロ以降、軍事、安全保障のための予算が増加しているが、従来からの IT 研究開発に対する影響は少ないようである。



FY2003 Blue Book

NITRD 計画の推進体制に関しても、2002 年度から大きな変更はない (図 2.1 参照)。参加省庁に関して若干の変更があるが、本質的な変更ではなく、他に LSN Coordinating Group 内のチームが再編成され、Middleware and Grid Infrastructure Coordination(MAGIC)が追加されたのが目につく程度である。

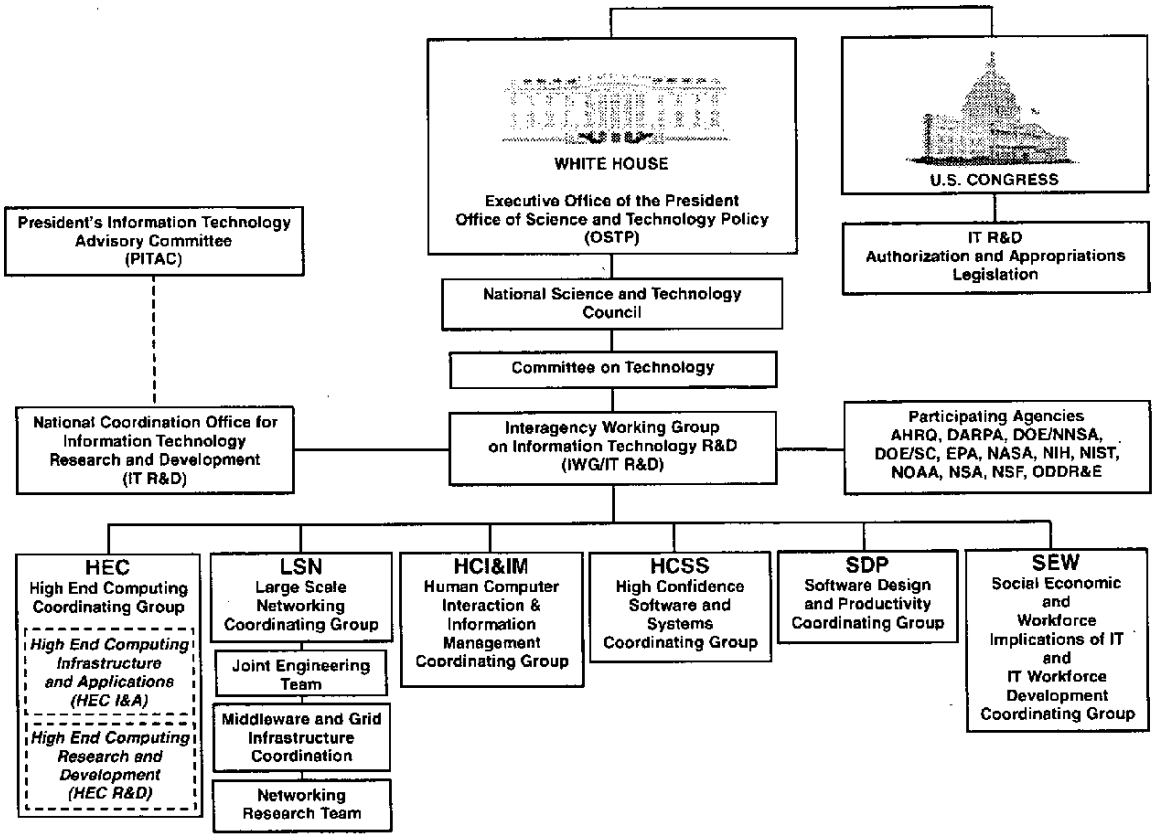


図 2.1 NITRD 推進体制

2.1.2 NITRD 計画の予算規模

FY2003 Blue Book に掲載されている、NITRD 関連の省庁および PCA ごとの 2003 年度予算要求額を表 2.1 に示す。ハイエンドコンピューティング関係(HEC)予算は全体の半分近く(44.8%)を占めており、当初より NITRD 計画の中心的存在となっている。

また、FY2004（2003 年 10 月～2004 年 9 月）大統領予算教書における NITRD 関連予算要求額が 2003 年 2 月 3 日に OSTP のサイトに掲載された。これを表 2.2 に示す[1]。
(http://www.ostp.gov/html/budget/2004/OSTP%20NITRD%201-pager%20(OMB).pdf)

[1] この要求額は大統領から議会に提出された予算案にもとづくものであり、議会での審議により大きく変わる可能性を持っている。例えば表 2.2 における FY2003 の予算額は表 2.1 の Blue Book 記載額からかなり増加していることがわかる。

表 2.1　NITRD 計画　FY2003 予算要求額（単位：100 万ドル）

省庁	HECI&A	HECR&D	HCI&IM	LSN	SDP	HCSS	SEW	合計		比率
NSF	**215.2**	**68.3**	132.2	102.4	45.8	50.1	64.8	679		35.9%
NIH	**88.2**	**8.9**	90.8	117.5	5.8	3.7	11.8	327		17.3%
DARPA	**16.8**	**81.9**	35.5	29.2	60			223		11.8%
NASA	**68.4**	**26**	23.8	4.5	40	43.3	7	213		11.3%
DOE/SC	**98.5**	**39.3**	16.4	28.1			3.5	186		9.8%
NSA		**31.9**		1.3		28.1		61		3.2%
NIST	**3.5**		3.2	6.2	7.5	2		22		1.2%
NOAA	**13.3**	**1.8**	0.5	2.8	1.5			20		1.1%
ODDR&E		**1.8**	1.8	6.3	1.9	1		13		
AHRQ			5	4				9		0.5%
EPA	1.8							2		0.1%
小計	**505.7**	**259.9**	309.2	302.3	162.5	128.2	87.1	1,755		92.9%
DOE/NNSA	**41.4**	**39.5**		14.7	34.2		4.3	134		7.1%
合計	**547.1**	**299.4**	309.2	317	196.7	128.2	91.4	1,889		100.0%
比率	**29.0%**	**15.8%**	16.4%	16.8%	10.4%	6.8%	4.8%	100.0%		

表 2.2　NITRD 計画　FY2004 予算概要（単位：100 万ドル）

省庁	2002 実績	2003 要求	2004 要求	増加率 '03 to '04
National Science Foundation	662	678	724	7%
Defense	439	442	461	4%
Health and Human Services	347	374	441	18%
Energy	306	310	317	2%
NASA	181	213	195	-8%
Commerce	36	38	39	3%
Environmental Protection Agency	2	2	2	0%
合計	**1,973**	**2,057**	**2,179**	**6%**

表 2.2 を見ると、NITRD 計画の予算は FY2004 予算案においても全体的には順調な伸びを示しており、あいかわらず Health and Human Services（バイオ、医療関係）の大きな伸びが目立っている。その一方で NASA 関係の予算が 8%削減されているのが目に付く（ただし、NASA の FY2004 予算は、2003 年 2 月 1 日のスペースシャトル、コロンビア号事故の影響で大きく変わる可能性あり）。

本資料では 2004 年度における NITRD 計画の重点項目として以下を挙げている。

- ネットワークの「信用」（セキュリティ、信頼性、およびプライバシー）
- 高信頼(high-assurance)ソフトウェアおよびシステム
- マイクロ、組み込み型センサー技術

- ハイエンドコンピューティング・プラットフォームのコスト、サイズ、および消費電力を低減するための革命的アーキテクチャ
- 情報技術による社会的、経済的インパクト

2.1.3 ハイエンドコンピューティング研究開発の内容

以下に、NITRD 計画における主要なハイエンドコンピューティングおよび高速ネットワーキング関係の研究開発の内容を、FY2003 Blue Book にもとづいて記載する。これらの内容は NITRD 計画の 7 つの領域(Program Component Area：PCA)のうち、HEC I&A、HEC R&D および LSN に相当している。

(1) ハイエンドコンピューティング

ー複雑さのフロンティアを探求する新技術ー

主要な研究課題

- 高性能コンピュータ・アーキテクチャー科学的発見と国家安全保障のための最大の能力をもたらすために、ハードウェア設計とアーキテクチャ、システム・ソフトウェア、科学的応用などの間の複雑さのトレードオフを理解し、それを克服する
- 次世代スーパーコンピュータ・プラットフォームを実現するための革命的手法ー新しいアーキテクチャ、量子及び生物分子コンポーネント、ハイブリッド技術、チップ上の再構成可能なシステム、processor-in-memory（PIM）技術を含む革新的なプロセッシング及びストレージ概念
- システムソフトウェアー先進的なプログラミング環境、コンパイラ、ライブラリ、ミドルウェア、及びパフォーマンス・エンジニアリング技術
- 科学計算とシミュレーションのための新しいハイエンド・アルゴリズムとコード；分散型テラスケール・コンピューティング環境のための、統合化・最適化されたソフトウェア基盤

2003 年度における各機関の代表的活動

政府機関	活動の内容
NSF	テラスケール基盤；異機種分散型ハイエンドシステムのためのシステム・ソフトウェア、ミドルウェア、ソフトウェア環境、ライブラリ、可視化、データ管理、及びアルゴリズム；グリッド資源管理；量子及びバイオコンピュータの概念
DARPA	多様なアーキテクチャ；ハイエンドかつ生産的で、堅牢なインテリジェントコンピューティング・システム；プロセッサ-イン-メモリアレイを含むチップ内及びチップ間通信のための光学材料の超高密度集積技術

DARPA/NSF	テラスケール・コンピューティングとストレージに用いる生物分子構造
NASA	ハイエンドソフトウェア及びシステムのチューニングと管理技術；インフォメーション・パワー・グリッドの技術とツール；情報物理学（センシング、処理及び記憶システムの特性）；量子及びナノスケール技術
NIH	ハイエンド生物医学コンピューティング；立体分子構造を決定するためのツール；測定データのイメージを表示及び分析するための方法
DOE/SC	テラスケールのモデル化とシミュレーション・アプリケーションのためのスケーラブルな数学的アルゴリズムとソフトウェア基盤（オペレーティングシステム、コンポーネント技術、最適な数学的ソルバ）；テラスケール科学のためのパートナーシップ
DOE/NNSA	米国の核貯蔵量管理のためのスーパーコンピュータ用モデル化とシミュレーションを可能にする高速計算、テラバイトデータ記憶と検索、及び可視化における科学及び工学上の革新
NSA	ハイエンドシステム製造企業との共同研究；オペレーティングシステムとプログラミング言語の改良；特殊目的のデバイスのための基盤技術（電力制御、冷却、相互接続、スイッチ、および設計ツール）；コンピュータメモリ性能；量子情報システムの基礎物理
NOAA	強化されたモジュラー・オーシャン・モデル（MOM）、フレキシブル・モデリング・システム（FMS）、スケーラブル・モデリング・システム（SMS）による改良された気候及び気象モデル
NIST	量子コンピューティング、安全な量子通信、最適化と計算幾何学、フォトニクス、ナノテクノロジー、オプトエレクトロニクス、及び新しいチップの設計と製造方法に関する研究
ODDR&E	量子通信とメモリに関する大学を中心とした研究
EPA	空気、水、及び土壌の相互作用のような複雑な環境現象をモデル化するためのパラダイム、技法及びツール

(2)　ハイエンドコンピューティング能力

―発見のためのテラスケール・インフラストラクチャー―

1)　NSF の分散型テラスケール施設(Distributed Terascale Facility：DTF)

NSF から$53M の資金提供を受けて 4 ヶ所の研究所（NCSA、SDSC、ANL、Caltech）にある Linux クラスタを 40 ギガビットの高速ネットワークで接続してグリッド環境（TeraGrid）を構築する。2003 年から運用を開始する予定。

（注：Blue Book には書かれていないが、2002 年 10 月に$35M の追加資金援助と、上記の 4 サイトに加えて Pittsburgh Supercomputing Center(PSC)の参加が決定された。これで合計 5 サイトとなり、ピーク性能の合計は 20TFlops、ストレージ容量の合計は 1 ペタバイト近くになるという。）

2)　DOE の先進的コンピューティングによる科学的発見（SciDAC）

2001 年度から始まった Scientific Discovery through Advanced Computing

(SciDAC)プログラムは、DOE のミッションに関連する科学分野(気象、核エネルギー、化学等）の基礎研究を、テラスケール・コンピュータを利用して推進するための、ソフトウェアおよびハードウェア基盤を開発することを目的としている。DOE による $57M の資金提供により、合計 51 のプロジェクト（3～5 年計画）が DOE の 13 ヶ所の研究所と 50 以上の大学で進行中である。

3) 革新的なアーキテクチャにおける長期的研究

現在のハイエンド・プラットフォームは性能のスケーラビリティ、製造コスト、運用コスト等において限界に達しようとしている。DARPA の高生産性コンピューティング・システム（High Productivity Computing Systems：HPCS）プログラムは、民間企業や大学と協力して 2010 年までに性能、コスト、ソフトウェアの生産性／移植性、堅牢性、信頼性等を飛躍的に向上させる新しいアーキテクチャやコンポーネント技術を開発することを目標としたものである。

(3) 大規模ネットワーキング

ー未来のインターネット：ダイナミックな柔軟性、広帯域幅、及びセキュリティー

主要な研究課題

- 信用：セキュリティ、プライバシー、および信頼度
- 適応性のある、ダイナミックでスマートなネットワーク
- ネットワーク性能の測定とモデル化
- 何十億個ものワイヤレス機器とセンサーを含む、異種間ネットワークトラフィックの大幅な増加に対応する拡張性のある技術
- ミドルウェアのような垂直統合とサポーティングのツールやサービスを含む、ネットワーキング・アプリケーション
- 革命的な研究：複雑さの理論、一般化された制御理論、接続性が指数関数的に拡大する状況で、ネットワーク機能の進展に取り組むための異なったモデル

2003 年度における各機関の代表的活動

政府機関	活動の内容
NSF	ネットワーク化されたアプリケーションの性能を最適化するミドルウェアを中心とした研究；大学間における高性能な接続；ネットワーク・モニタリング、問題検出と解決、アクティブ／インテリジェントネットワークを支援する先進的自動化ツール、協調的アプリケーション、及び革新的アクセス方法のような戦略的インターネット技術

NIH	スケーラブルでネットワークを意識したワイヤレスの地理情報システム(GIS)、及びネットワーク化された医療関連環境におけるセキュリティ技術のアプリケーションの実証実験
DARPA	何百ものノードから成るネットワーク上における、ミリ秒単位から時間単位までのレンジにわたる振る舞いを予測できる、拡張性のあるネットワークのモデリング及びシミュレーションツール；ハイブリッドの光/RF 自己回復型ネットワークの実証実験
NASA	先進のコンピューティング、ネットワーキング及び協調技術を開発、実証するための、高速テストベッドネットワークの導入；グリッド環境のためのネットワークサービスの統合（QoS、受動的モニタリング、リソース確保）；ハイブリッドな衛星/モバイル、無線/アドホック・ネットワークアプリケーションと「未来のオフィス」の労働環境の実証
DOE/SC	分散型ハイエンド科学アプリケーションに対して、テラバイト/秒のスループットで TCP による信頼性の高い転送を可能にする高性能転送プロトコルの研究；大規模な科学的共同研究のためのエンド・ツー・エンド性能モニタリング、ネットワーク診断、及び拡張性のあるサイバー・セキュリティ・サービスの開発
NSA	バーストスイッチ技術、供給(Provisioning)、メッセージ受け渡し、低出力無線ネット、高速システムのファイヤーウオール、セキュリティと相互運用性の問題などを含む、先進的ネットワークトポロジーとプロトコル、ネットワークコンバージェンス、全光ネットワーク、及びネットワーク管理に関する研究
NIST	産業用プロトコルのモデル、新しい仕様の検証、適応制御メカニズムを含む、パーベイシブ・コンピューティング・デバイスのネットワーク通信のための標準規格；アドホック無線ネットワークの基準とプロトコル；応答性に優れたスイッチング・インフラストラクチャーの開発手法；インターネット・インフラストラクチャー・セキュリティのためのプロトコルと標準規格
NOAA	過酷な気象現象の予測と警報及び危険物への対応を支援する、拡張性のあるネットワーク能力とアプリケーションの早期導入
ODDR&E	リアルタイム耐故障ネットワーク・プロトコルに関する大学を拠点とした研究

1) グリッドのためのネットワーク向けミドルウェア"MAGIC"

ネットワーク環境でのアプリケーションの効率的実行には、ミドルウェアが重要な役割を果たす。最近注目を集めるグリッド（Grid）技術においては、NITRD 計画の下で開発された Globus と呼ばれるミドルウェアパッケージが標準的に利用されるなど、政府支援によるミドルウェアの研究が継続して行われている。

- NSF ミドルウェアイニシアチブ(NMI)

NMI は NSF からの資金提供により 2001 年 9 月から開始されたイニシアチブであり、ネットワーク環境での再使用可能で拡張性に富んだミドルウェアを開発、配備、および

サポートすることを目的としている。

- MAGIC(Middleware And Grid Infrastructure Coordination)

MAGICは、NITRD組織の中の大規模ネットワーキング調整部会の内部に新たに設けられたチームである。本チームは、従来からあるJETおよびNRTを主としてグリッドという観点から結びつけるもので、以下の役割を担っている。

- 省庁間のミドルウェアとグリッドに関する活動の調整
- 相互運用可能なグリッド技術の研究とその配備の促進・奨励
- 使用に適した、広く利用可能なミドルウェアツールとサービスの開発推進
- 上記技術における有効な国際協調に関するフォーラムの開催

(4) 先進のネットワーキング・アプリケーション

―人材とIT資源を結びつけ米国の科学面でのリーダーシップを目指す―

2003年度における各機関の代表的活動

政府機関	活動の内容
NSF	地震工学シミュレーションのためのネットワーク (Network for Earthquake Engineering Simulation：NEES) 国立バーチャル天文台(National Virtual Observatory：NVO) グリッド物理学ネットワーク(Grid Physics Network：GriPhyN)
DARPA	ネットワークを基本とした総合監視装置 (Network-Based Total Surveillance System：NBTS)
NASA	航空安全シミュレーション
DOE	研究者を科学とコンピューティング機能に接続
NIST	製造のためのソフトウェア相互運用性テストベッド 国際iGridデモンストレーション

2.2 ハイエンドコンピューティング分野の新しい動向

2.2.1 SC2002に見る動向

2002年11月16日から11月22日の間、米国メリーランド州ボルチモアにおいて第15回International Conference on High Performance Networking and Computing（通称SC2002）が開催された。この会議は毎年11月に開催されるスーパーコンピューティング関連の国際会議（開催地は米国）であり、会議の性格から、展示等は企業よりむしろ大学や国の研究機関からの参加が目立つのが特徴である。同時多発テロの影響により低調であ

った前回とは違い、今回は展示参加が 221 団体、参加者が 7000 人を超えて史上最大規模となった。

今回のテーマは"From Terabytes to Insights"（テラバイトから洞察へ）であり、ハイエンドのコンピューティング能力がもたらす科学研究の飛躍的進歩といったところに主眼が置かれていたようである。基調講演や招待講演でも科学研究に対するハイエンドコンピューティングの貢献の大きさと、さらなる高性能化への期待が述べられていた。

SC2002 の主要な話題は、やはり地球シミュレータをはじめとするハイエンドプラットフォームとグリッドおよび高速ネットワーク関連であった。展示関係では CRAY X1、IBM p655、Origin 3900 などのサーバ新製品の展示と科学技術アプリケーションのデモによる高性能のアピールが目立った。

なお、SC2002 での講演、発表および展示内容については、東京大学の小柳教授のレポートが詳しい。(http://olab.is.s.u-tokyo.ac.jp/~oyanagi/reports/SC2002)

(1) 地球シミュレータの与えたインパクト

SC2002 における最大の話題は、2002 年 6 月にその圧倒的な性能により Top 500 リストの第 1 位に躍り出た地球シミュレータであった。会期中には地球シミュレータセンター長の佐藤哲也氏の招待講演があったほか、米国政府の NSF や DOE のディレクターの講演でもしばしば言及されていた。また、投稿された論文の中から選ばれて、Gordon Bell Awards を受賞した 5 件の論文うちの 3 件が地球シミュレータ関連であった。さらに、パネルディスカッションではそのものずばりの"The 40Tflop/s Earth Simulator System: Its Impact on the Future Development of Supercomputing"というタイトルのセッションまで設けられ、活発な議論が行われるなど、その影響の大きさがうかがえた。

上記のように地球シミュレータが大きく取り上げられた背景には、「危機感を煽って米国政府からの R&D 予算獲得の口実とする」という理由も存在したであろうが、以下のような議論を呼び起こすきっかけを作ったということもあったと思われる。

①　実効性能に関する議論

現在の主流は、汎用スカラープロセッサと汎用ネットワーク、すなわち Commodity Off-The-Shelf(COTS)製品を使用した超並列スカラー型システム（PC クラスタを含む）である。しかしこの方式における実効性能（実アプリケーション全体を通しての平均 Flops）は、一般的にピーク性能の 10 パーセント程度と言われており、ベクトル型システムより劣っている。米国が捨てたベクトル方式を採用した地球シミュレータが、実アプリケーションの「全球大気大循環シミュレーション」でピーク性能の 66 パーセント（26.58TFlops）

という驚異的な実効性能を実現したことが、システムアーキテクチャと実効性能についての議論をあらためて呼び起こしているように見える。

② ハイエンド領域も COTS でいいのか？

米国のハイエンドコンピュータの主流は、上記の COTS 製品を使用したものである。安価な点において、この方式の一般市場での優位は、もはや動かしがたいところまで来ている。しかし、国家の安全保障や科学研究などの最高性能を必要とする領域で、COTS をベースとしたシステムしか解がないことへの問題意識も高まりつつある。

地球シミュレータは、わが国ではさほど話題になっていないが、それとは対照的に、米国の関係者には非常に大きなショックを与えたようである。科学技術における米国のリーダーシップの危機といった論調まであり、DOE などが中心となって 2 年以内に地球シミュレータの性能を凌駕するシステムを開発しようと躍起になっている。

一方、わが国は地球シミュレータ以後、国が主導する高性能プラットフォーム開発の具体的計画を持っていない。このままだと 2～3 年後には米国が再び世界一の座を取り戻し、ハイエンドコンピューティング分野における日本発のセンセーションも、一時的な現象に終わるのではなかろうか。

(2) Top 500 にみる性能動向

毎年 6 月にドイツで開催される ISC(International Supercomputer Conference)と 11 月に開催される SC において、"Top 500 Supercomputer Sites"（LINPACK ベンチマークによる性能測定結果の上位 500 サイト）が発表されている。

SC2002 で発表された Top 500 には、地球シミュレータに続き、新たに 2 位と 3 位にロスアラモス国立研究所(LANL)の ASCI Q がランクされた。また、5 位にローレンス・リバモア国立研究所(LLNL)の NetworX 社製 PC クラスタ（Xeon 2.4GHz x 2,304 プロセッサ）、8 位に NOAA の HPTi/Aspen Systems 社製 PC クラスタ（Xeon 2.2GHz x 1,536 プロセッサ）と 2 台のシステムが PC クラスタとしては初めて Top 10 内にランクインするなど、全般的に PC クラスタの増加が目立っている。

図 2.2 は 1993 年から 2002 年まで 10 年間の Top 500 の 1 位と 500 位の性能推移をグラフにしたものである。過去 10 年間は年率 1.9～2.0 倍の割合で性能が向上しており、今後もこの傾向が続くとすれば、2005 年に 1 位が 100TFlops、500 位が 1TFlops に達し、さらに 2009 年には 1 位が 1PFlops（ペタフロップス）に達すると予想される（あくまでも LINPACK ベンチマークの性能であり、理論ピーク性能ではないことに注意）。

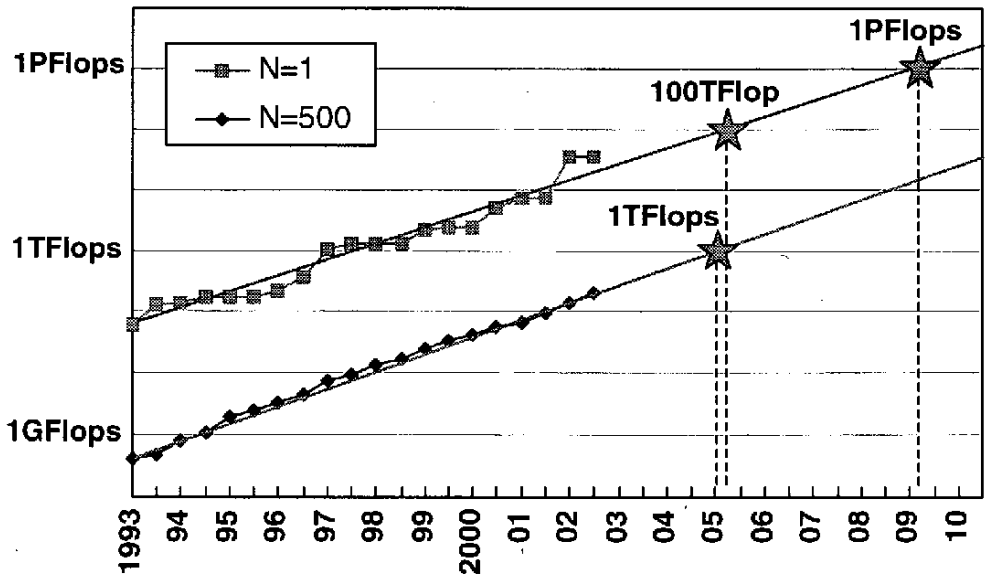


図 2.2　Top 500 の性能トレンド

2.2.2　超並列コンピューティング

(1)　ASCI 計画

DOE の国家核安全保障管理局(National Nuclear Security Administration：NNSA)が実施している ASCI 計画（Advanced Simulation and Computing Initiative[2]）は、1995 年の発足以来、米国のハイエンドコンピュータ開発の中心的な役割を担っており、当ワーキンググループでは過去の報告書において ASCI 計画の状況を毎年報告してきた。最近になっても"Red Storm"、"Purple"、"Blue Gene/L"などの計画が次々と発表されるなど、相変わらず積極的な投資が目立っている。以下では本計画の下で開発されているシステムの最新状況について述べる。

1)　ASCI Q

ロスアラモス国立研究所（LANL）に設置される ASCI Q は、HP（旧コンパック）の Alpha サーバ ES45 をベースにしたクラスタ・システムで、当初 2002 年にピーク性能 30TFlops のシステムが完成する予定であった。現在 1/3 規模のシステム（4,096 プロセッサ／1,024 ノード）が 2 セット（QA、QB）完成した段階にあり、これらの 2 セットがそれぞれ LINPACK ベンチマークで 7.73TFlops を記録して Top 500 リストの 2 位と 3 位に登場している。しかし、残りの 1/3 である QC が設置され、さらにシステム全体が統合さ

[2] 以前は Accelerated Strategic Computing Initiative と呼ばれていた。

れるのがいつになるかは不明である。一足先に完成した地球シミュレータや後述するASCI Purple と Blue Gene/L の発表の影響で影が薄くなった感は否めない。

2) ASCI Red Storm

Red Storm は CRAY, Inc.により開発、製造される予定のスカラー型システムである。サンディア国立研究所に納入され、2004 年 8 月から運用を開始する。AMD の Opteron プロセッサ（SledgeHammer）を 10,368 個使用し、ピーク性能は 41.5TFlops（LINPACK 性能目標は 14TFlops 以上、ASCI Red の 7～8 倍）と発表されている。
（SC2002 での発表資料による。）

3) ASCI Purple と Blue Gene/L

2002 年 11 月 19 日に IBM により開発、製造される ASCI Purple および Blue Gene/L がローレンス・リバモア国立研究所(LLNL)に納入されることが発表された。

ASCI Purple は IBM の POWER5 プロセッサを 12,544 個使用し（64 CPU x 196 ノード）、ピーク性能は 100TFlops、主記憶容量 50 テラバイト、ディスク容量 2 ペタバイト（約 2,000 兆バイト）という巨大なシステムである。ASCI Purple は 2004 年末までに運用を開始する予定である。

Blue Gene/L は IBM が最終的にペタフロップスを目指している Blue Gene 計画の最初のシステムであり、2005 年までに完成する予定である。Blue Gene/L は 1 チップ上に 2 プロセッサ（700MHz）を実装し、これと 256MB のメモリで 1 ノードを構成する。さらに 65,536 ノード（131,072 プロセッサ）を 3 次元トーラス接続して、ピーク性能 360TFlops を実現するとしている[3]。過去このように多数のノード（65,536 ノード）から成るシステムは例が無く、さらにノードあたりのメモリが 256MB と小さいため、高い実効性能を出すためには、これまで以上に高度なタスク割り当て制御が必要と思われる。

また、Blue Gene/L の後にもきわめて意欲的な計画が続いている。Blue Gene/P は 2006 年～2007 年にピーク性能 1Pflops（実効性能 300TFlops）、Blue Gene/Q は 2007 年～2008 年にピーク性能 3PFlops（実効性能 1PFlops）を実現するとしている。にわかには信じがたい計画だが、もしこれが予定通り実現されるとすれば、わが国のメーカはとても太刀打ちできないに違いない。

---

[3] 1 ノード（2 プロセッサ）中の片方のプロセッサは主にノード間メッセージ転送用に使用されるため、通常演算に使用されるのは 65,536 プロセッサである。したがってノード間メッセージ転送がきわめて少ない特殊なアプリケーション以外は、ピーク性能 180TFlops とするのが妥当と思われる。

(2)　その他

ASCI 計画とは別に、地球シミュレータ対抗を強く打ち出しているものに、ローレンス・バークレイ国立研究所(LBNL)、アルゴンヌ国立研究所(ANL)および IBM が共同で提案している Blue Planet 計画というものがある。その目標は、2005 年末までに地球シミュレータの 2 倍の実効性能のシステムを、半分のハードウェアコストで実現するというもので、ピーク性能は 150TFlops、少なくとも数種類の実科学計算コードで 40～50TFlops の実効性能を実現することを目指している。

IBM の POWER5 を科学技術計算向けに改良したものを 16,384 プロセッサ使用し、また Virtual Vector Architecture(ViVA)と呼ぶ疑似ベクトル処理（CRAY X1 等でも使用している、ノード内の複数 CPU を連動させてベクトル処理を行う機能）を採用する。プロセッサを POWER6、7 とアップグレードすることにより 2009 年にはペタフロップスを目指すとしている。この計画が実行に移されるかどうかは、今のところ不明であるが、もし実施されるとすれば、将来は Blue Gene 計画と統合されるものと思われる。

2.2.3　グリッド・コンピューティング

2002 年は、グリッド・コンピューティングが研究段階から実用段階へと移行する節目の年であったと言えるかも知れない。IBM などの大手メーカがグリッド・コンピューティングビジネスへの本格的投資を発表して以来、「グリッド・コンピューティング」は技術系／ビジネス系のメディアに頻繁に登場するようになった。また、わが国も含めて、科学研究コミュニティでのグリッド構築が一段と加速している。以下にグリッド・コンピューティングに関する最近の動向を、①基盤技術の標準化の促進、②科学研究コミュニティのグリッド形成の加速、③ビジネス化へ向けた動きの本格化、という 3 つの視点から述べる。

(1)　基盤技術の標準化

2000 年 11 月にグリッド基盤技術の世界的標準化団体である Global Grid Forum(GGF)が結成され[4]、インターネットにおける IETF と同様の手続きで標準化の作業が進められている。 GGF では 2001 年より毎年 3 回づつワークショップを開催しているが、2002 年 7 月の第 5 回ワークショップ(GGF5)参加者は約 900 人とそれ以前（500 人以下）の約 2 倍であり、グリッド・コンピューティングに対する関心が急速に高まっていることを示して

[4] それ以前は米国の Grid Forum、ヨーロッパの European Grid Forum(eGrid)、アジア太平洋の Asia-Pacific Grid Forum(ApGrid)が、それぞれの地区のグリッド開発を推進していた。

いる。

わが国では2002年6月にグリッド協議会が設立され、GGFに対する標準化の提案のほか、グリッド・コンピューティングの啓蒙活動、情報共有などが実施されている。

(2) 科学研究コミュニティのグリッド形成

グリッド技術は、現在の一般的なうたい文句である大規模分散コンピューティング（計算）への応用面のほかに（それ以上に？）、遠隔地にある実験装置、計算資源へのアクセスやデータベースの共有、遠隔地間でのコラボレーションなどにも応用される。最近になって、様々な科学研究コミュニティの間で、主として後者の目的によるグリッドの構築が活発化してきている。以前はグリッドそのものの実証実験的意味合いが強かったが、今は科学研究への実質的貢献を目指した応用面主体のものに変わりつつある。以下に米政府の支援により構築されている（されつつある）主要な科学研究グリッドを示す。

プロジェクト名	支援機関	分野／参照サイト
Earth System Grid (ESG)	DOE	気候変動研究 http://www.earthsystemgrid.org/
Grid Physics Network (GriPhyN)	NSF	天文学、高エネルギー素粒子物理学 http://www.griphyn.org/
National Virtual Observatory (NVO)	NSF	天文学（仮想天文台） http://www.us-vo.org/
Network for Earthquake Engineering Simulation (NEES)	NSF	地震工学 http://www.nees.org/
Biomedical Informatics Research Network (BIRN)	NIH	バイオ医学研究 http://birn.ncrr.nih.gov/

(3) ビジネス化へ向けた動き

グリッド・コンピューティング技術は、従来から政府支援のもとで大学と国研により研究開発が行われてきた。その後、昨年の報告書にあるように、AVAKI、Platform Computing、Entropiaなどのベンチャー企業を中心に企業化が始まった。2002年2月にはPlatform Computing社がオープン・ソースであるGlobus Toolkitの業界初の商業版である"Platform Globus"を発表している。また、最近になって、米国の大手メーカのグリッドビジネスへの参入が本格化しており、現在、グリッド・コンピューティングのビジネス化を目指している主要な大手メーカとして、IBM、Sun、HP、マイクロソフトなどが挙げられる。特にIBMはGlobus Projectと共同で、グリッドとWebサービスとの統合化を目指すOpen Grid Services Architecture (OGSA)の仕様を提案しており、GGFは2003年に

これを標準として認定する見込みである。IBM のグリッド事業責任者である Tom Hawk 氏は「2005 年にはグリッド関係のビジネス分野向けの売り上げが科学技術分野を追い越すことになるだろう」との見解を明らかにしている(日経 IT Pro 2002 年 5 月 13 日記事)。

以上のように、グリッド・コンピューティングのビジネス化の動きが活発になっているが、実際はグリッドというキーワードが先行し、後からビジネスの為の応用を考えているというのが正確な状況であろう。これをビッグビジネスに成長させるためには、セキュリティや信頼性などの技術的課題を克服することと並んで、（どのような技術にも言えることであるが）いかに技術の特性を生かしたビジネスモデルを作り、キラーアプリケーションを生み出せるかが鍵になると思われる。

2.2.4 将来のコンピューティングに関する研究

(1) DARPA の"High Productivity Computing System"プログラム

DARPA は 2002 年より"High Productivity Computing System"（高生産性コンピューティング・システム：HPCS）プログラムを開始した。

本プログラムは 2010 年までの 9 年間にわたる長期計画であり、現在のコンピュータシステムと、製品化がまだ当分先と言われている量子コンピュータとの間のギャップを埋めるものと位置づけられている。計画はフェーズ 1（概念研究：1 年）、フェーズ 2（R&D：3 年）、フェーズ 3（大規模開発：5 年）に分かれている（図 2.3 参照）。フェーズ 1 については BAA(Broad Agency Announcement)により企業や大学からの提案が公募され、CRAY、IBM、SGI、Sun 等の企業が提案を行っている。以下に HPCS プログラムの生まれた背景とその目的について記す。

1) 背景

- 現在の商用高性能コンピュータ技術のトレンドでは国家の安全保障に必要とされる能力とのギャップが大きくなるばかりである。
- 政府主導による研究開発が無ければ、ハイエンドコンピューティングは主要な関心が一般大衆やビジネス市場に向いた企業の製品からしか得られず、これでは重要な国家安全保障に関するアプリケーションが効果的に実行できない。
- 以下のような重要な分野において、米国の国防総省（DoD）や産業界の優位性を確保する必要がある。
  - －軍事目的の気象、海洋予測
  - －大気中の汚染物質の分布解析に関する計画演習

－暗号解読
－軍用プラットフォームの分析
－サバイバビリティー／ステルス設計
－諜報／監視／偵察システム
－巨大な航空機、船舶および構造の仮想生産／故障解析
－先端的バイオテクノロジー

2) 目的

- ハイエンドコンピューティングの新しいビジョンである「高生産性コンピューティング(HPCS)」を実現するための次世代のハイエンド・プログラミング環境、ソフトウェアツール、アーキテクチャ、およびハードウェアコンポーネントを生み出すことに焦点を当てた研究開発プログラムを設け、実効性能、スケーラビリティ、ソフトウェアツールや環境、および増加する物理的制約などの課題に取り組む。
- 今日の 80 年代後半の技術を元にした高性能コンピューティング（HPC）と将来の量子コンピューティングの間を埋める。
- 国家安全保障と産業界のユーザコミュニティのために、経済的な高生産性コンピューティング・システムを以下のような設計条件でこの 10 年の後半（2007 年～2010 年）に実現する。
  －性能：計算効率と重要な国家安全保障のアプリケーション性能を 10～40 倍向上
  －生産性：アプリケーションの開発、運用、および保守コストを低減
  －移植性：研究および業務用 HPCS アプリケーションソフトウェアがシステム仕様に影響されないようにする
  －堅牢性：HPCS ユーザにより高い信頼性を提供し、犯罪行為によるリスクを低減

3) プログラム計画

図 2.3 に本プログラムの計画を示す。

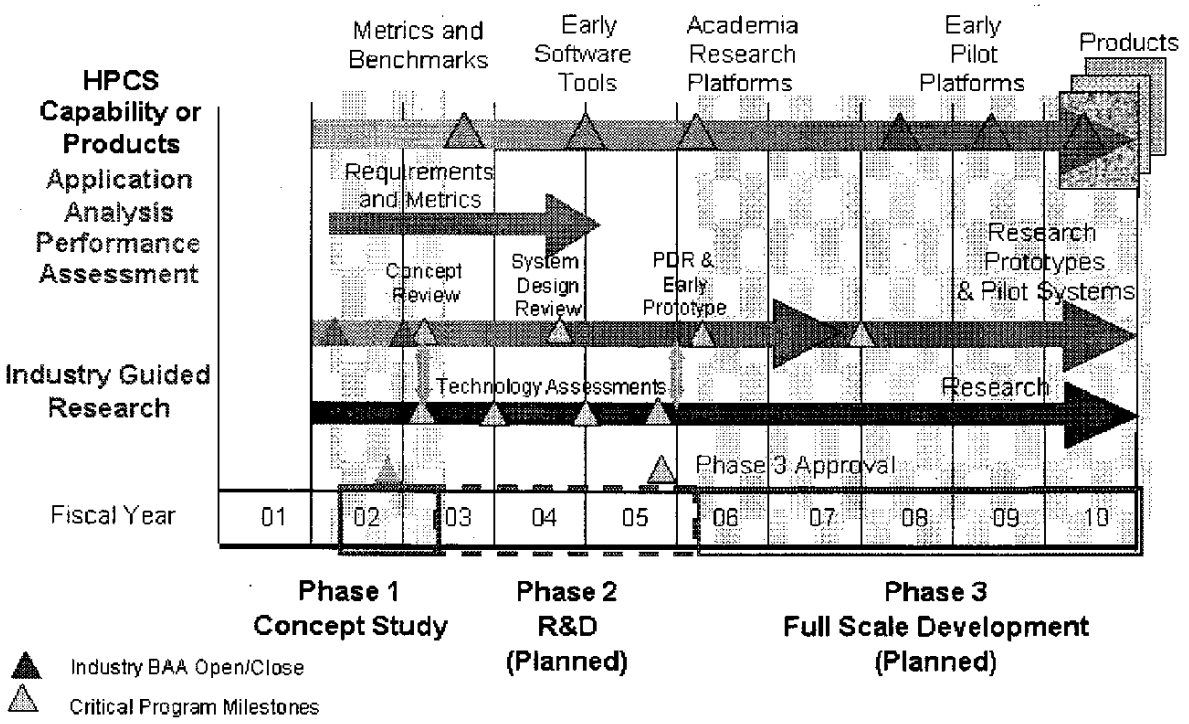


図 2.3　プログラム計画－DARPA のサイトより
（http://www.darpa.mil/ipto/research/hpcs/plan.html）

(2)　量子コンピュータの研究動向

量子コンピュータの実用化は数十年先（20 年以内から 50 年以上まで種々の意見がある）と言われている。ドッグイヤーと呼ばれる近年の IT の進展速度からすれば、はるか遠い未来の話と言ってもいい（このことを考えれば、前述の HPCS プログラムのようなものが生まれてくるのもうなずけることであろう）。しかし、米国政府はこの遠い将来の実用化に向けて、金額的にはまだそれほど大きくないにしても、この 10 年間継続して着実に研究支援を行ってきている。

以下に米国連邦政府の資金による主要な量子コンピュータ研究プログラム（公募）および政府機関における研究の一部を挙げておく。

- NSA　ARDA[5](Advanced Research and Development Activity)
  (http://www.ic-arda.org/Quantum/)
- NSF　Quantum and Biologically Inspired Computing(QuBIC) Program
  (http://www.nsf.gov/pubs/2002/nsf02017/nsf02017.html)

[5] ARDA は NSA などの情報機関が実施する先端的研究に対するファンディング組織である。

- NASA　JPL Quantum Computing Technology Group
  (http://cs.jpl.nasa.gov/qct/qat.html)
- NIST　Physics Laboratory, Quantum Information Program
  (http://qubit.nist.gov/)
- DARPA　Quantum Information Science and Technology(QuIST) Program
  (http://www.darpa.mil/ipto/research/quist/)
- DOE　LANL
  (http://qso.lanl.gov/qc/)

その他、Caltech、Stanford、MIT、UCB など多くの大学でも研究が行われている他、民間企業では IBM の活発な研究が目立っている。

また、日本においても政府系研究機関や富士通、NEC などの民間企業でも研究が行われ、米国と対等な成果を出しつつある。

第 3 章

米国政府支援
SDP(Software Design and Productivity)
研究開発の動向

第３章　米国政府支援 SDP(Software Design and Productivity)研究開発の動向

1999年2月に、米国のPITAC（ITに関する大統領諮問委員会）は、ITに関する米国政府支援の研究開発に対し、長期的基礎研究とソフトウェア研究の重要性を訴える提言を行った。この提言を受けて、米国の情報技術研究開発政策の柱であるネットワーキング及び情報技術研究開発（NITRD：Networking and Information Technology Research and Productivity）に「ソフトウェアの設計と生産性」(SDP：Software Design and Productivity)という研究領域が追加された。本章では、このSDPが何を目指しているかについて、2001年4月アーリントンで開かれたワークショップの内容から紹介し、2003年度のSDPの活動計画として、米国はどのような研究に取り組んでいるかを紹介する。また、SDPが目指すものに関連した注目すべき最近の技術、システムについて紹介する。

3.1 SDP発足の経緯とPITACの提言

1999年2月に、PITAC（ITに関する大統領諮問委員会）は、ITに関する政府支援の研究開発に対し、次の示すように、長期的基礎研究とソフトウェア研究の重要性を訴える提言[1]を行った。

- 1992年以来、IT関連の生産高はGDPの1/3を占め、数百万ドルの雇用機会を提供してきた。
- しかし、現状の政府の投資は、長期的なハイリスクな研究開発よりも、各省庁のミッションを追求する短期的な課題に偏りすぎている。
- このままであると、過去20-30年に渡って続いてきたIT革命へのアイデア供給の流れは途切れてしまう。
- 従って、2000-2004年の5年間に下記の研究開発テーマに4,743Mドルを長期的基礎研究開発に追加投資すべきである。
  - ソフトウェア
  - 大規模な情報化のためのインフラストラクチャ
  - 最高レベルのコンピューティング
  - 社会経済的影響
  - 連邦政府の情報技術研究の管理と実現

このPITACの提言を受け、政府は、2000年には「21世紀に向けての情報技術」IT^{2}計画として366Mドルの研究開発投資を追加し、2001年にはこれまでのCIC計画（2000年の予算は1,089Mドル）と統合して、ITR&D計画として、1,928Mドルと大幅に増額している。そして、現在は、ITR&D計画はNITR&D計画と名前を変更し、予算規模は1,830Mドル(2002年)となっている

PITACは、追加投資すべきとしたソフトウェアについては、以下のような現状、課題、であるとの調査結果を発表し、

- ソフトウェアに対する要求が、国家的なソフトウェアの生産能力を上回っている。
- 脆弱なソフトウェアに国家が依存している。
- 信頼性と安全性に優れたソフトウェアを構築するためのテクノロジが不十分である。
- ソフトウェア・システムは、急速な勢いで多様化と洗練化が進んでいる。
- 一般的な市民にとって、ソフトウェアへの依存度はしだいに高まりつつある。
- ソフトウェアの基礎研究に向けた国家の投資が不足している。

その対策として、次のことを提案している。

- ソフトウェアの基礎研究を最優先課題とする。

- ソフトウェアの開発手法、およびコンポーネント・テクノロジに関する基礎研究に向けた投資を強化する。
- ヒューマン/コンピュータ・インタフェースと相互作用に関する基礎研究を支援する。
- 情報管理の方法に関する基礎研究を強化する。
- 主な情報テクノロジ関連の研究構想の中に、ソフトウェアの研究を主要な課題として位置付ける。

3.2 NSF SDP ワークショップ

PITAC の提案により、SDP の研究開発予算はついたものの、PITAC の提案は、具体的にどのような研究を行えばよいか必ずしも明らかではない。そこで、NSF の Frank Anger が座長となって、後述する米国の著名な研究者（他に省庁の研究機関担当者も出席）が、2001 年 4 月 18~19 日にアーリントンに集まって開催されたのが、本ワークショップ[2]である。

3.2.1 SDP ワークショップの狙い

本ワークショップでは、Frank Anger がイントロとして、
ソフトウェアの現状は、以下のようであり、

- 全米でソフトウェア技術者 200 万人
- サポートスタッフ 35 万人
- まだまだ空きポスト多数

従って、以下のことを達成しなければならない。

- ソフトウェア生産性の劇的向上
- 高いソフトウェア品質

と説明し、前述した PITAC の提言を説明した後、本ワークショップでは、以下の点をあきらかにするのがねらいであるとした。

- 解決すべき基礎的な問題は何か
- 真の障壁と挑戦とは何か
- どの方向がもっとも有効か
- 何が理想的な成果か
- どのように研究を進めるべきか

この「どのようなソフトウェア研究をすればよいか」の議論を基にして、SDP (Software Design and Productivity) の研究計画を作ろうということで、以下の 4 つのグループに分かれて検討がなされた。

- ソフトウェア研究の将来
- 新ソフトウェア開発パラダイム
- 実世界ソフトウェア
- ネットワーク中心のシステム

これら四つのグループの参加メンバを表 3.1 に示す。

表 3.1　SDP ワークショップ参加メンバ

ソフトウェア研究の将来	新ソフトウェア開発パラダイム
Barry Boehm, USC Benjamin Pierce, Univ. of Pennsylvania Doris Carver, Louisiana State Univ. Shankar Sastry, UC Berkeley Bonnie John, Carnegie Mellon Univ. Kevin Sullivan, Univ. of Virginia William Mark, SRI International	Grady Booch, Rational Software Ralph Johnson, Univ. of Illinois Gregor Kiczales, Univ. of British Columbia Charles Simonyi, Microsoft John Vlissides, IBM
実世界ソフトウェア	ネットワーク中心のシステム
Don Winter, Boeing Phantom Works Martin Feather, Jet Propulsion lab. Gabor Karsai, Vanderbilt Univ./ISIS Patrick Lardieri, Lockheed Martin Cleve Moler, MathWorks, Inc. Edward Lee, UC Berkeley	Ian Foster, Argonne National lab. Doug Lea, State Univ. of New York at Oswego Adam Porter, Univ. of Maryland Rick Schantz, BBN Technologies Jim Waldo, Sun Microsystems, Inc

3.2.2　各検討グループからの提言

（１）ソフトウェア研究の将来

現状は、

- 現在のインフラは、低い信頼性、安全性、可用性で、まったく不適切である。
- 大規模ソフトウェアプロジェクトの半数は失敗している。
- 応用ソフトウェア同士が情報をやりとりできない。
- 過去のソフトウェア資産が１兆ドルもある。

との仮説に立って、以下の目標を立てている。

- 各領域の専門家がソフトウェアを使いこなせるようにしよう。
- 既製品ソフト業界の共通基盤となるアーキテクチャを開発しよう。
- ソフトウェア業界に対し、鍵となる概念･技術を提供しよう。
- プログラマの意図を機械が理解できる形で記録する方法を開発しよう。
- ソフトウェア開発の現状をしっかり認識しよう。

この目標実現に向けて、特に注目すべき領域は、以下であり、

- ネットワーク埋め込み型
- 対人インタラクション

- グループインタラクションの計算機支援
- 領域専門家のための環境：教員、エンジニア、金融、科学者

その領域で基礎となる科学は、以下であり、

- ソフトウェア物理：並行処理のモデルなど
- ソフトウェア経済学：柔軟性の価値を定量化など
- ソフトウェア行動科学：人間の認知・記憶に関する知見をソフトウェア設計に応用したり、ソフトウェアと人間の役割分担を明らかにしたりすることなど

そのキーポイントは、以下であると提言している。

- ソフトウェア開発のプレイグラウンドを変えよう。
  - 「プログラム」の意味の変更:

    Ptolemy（同期データフロー、状態遷移マシン、ランデブー方式同期メッセージング、連続的な時間、離散イベントといったコンピュテーションモデルのビジュアル化イメージでシステムを作成するアーキテクチャデザイン言語）

    Simulink（ブロックダイアブラムを作成することによって、システムのシミュレーションや性能の評価、デザインの改善が行える Mathworks 社のモデリングツール）
  - 「プログラマ」の意味の変更:practitioners (開業医などの専門家)
- 以下のキーとなる技術を進展させよう。
  - 軽い形式的枠組み
  - 並行性・耐故障性・安全性などのモデル
  - 「価値」に基礎を置いたデザイン
  - 分散オープンソーステスト方式
- ソフトウェア経済学を変えよう
- ソフトウェアでハードウェア を定義しよう。
  （ハードウェアがソフトウェアをではなく）

（2）新ソフトウェア開発パラダイム

新ソフトウェア開発パラダイムでは、現状は、以下のような要因によって、複雑さが指数関数的に増加している。

- 組込型、モバイル、浸透性、大域性
- 長寿命、複数バージョン混在
- 共同開発

- 利用者ごとのカスタマイズ

この難問題を以下のような革新的概念によって、一挙に解決することが必要である。

- 国立ソフトウェア考古学センター構想
- 多面的ソフトウェア
- 協調開発環境

[国立ソフトウェア考古学センター構想]

ソフトウェアの古典を蒐集および保存し、それをリバースエンジニアリングによって、その古典の芸術的本質を抽出し、Web で公開する。これにより、参考となるソフトウェアアーキテクチャの宝庫として、将来の研究の基礎として、特許取得のための事前芸術研究として、役立て、芸術的表現を奨励することをねらう。

[多面的ソフトウェア]

高級言語、オブジェクト指向プログラミング、デザイン・パターン、特定領域用言語といったこれまでの歩みは、ある側面をより理解しやすくした、あるいは、ある側面をうまく表現したというものにすぎない。本来、ソフトウェアは、様々な異なる意図を表現するものであり、新たなシステムは、共存する異なる側面が互いに影響しあって生まれるものである。このような多面的ソフトウェアについて、従来は認識不測であったが、デザイン・パターン、協調開発環境、アスペクト指向プログラミング、特定領域用言語といった研究分野では、一部、多面的ソフトウェアの考えが取り入れられ始めている。今後、多面的ソフトウェアとしてすべきことは以下である。

- どの側面が肝要か

  性能、価格、使い勝手、過去からの連続性...
- 各側面をどう表現するか
- 諸側面をどう変形・統合しシステムとするか
- 特定領域用言語のためのツール整備の加速

[協調開発環境]

協調開発環境とは、ソフトウェア開発のプラットフォームとして Web を仮想的な議論の場にするものであり、研究すべきことは以下である。

- ソフトウェア開発の社会学
- ソフトウェアプロセスの利用
- 考古学センターや多面的ソフトウェアを利用するプラットフォームの創造

これらの革新的概念の研究計画例には、以下のようなものがある。

- 側面中間形式（特定領域用言語のための汎用ツール、Weavers?）
- 対応代数 (Correspondence Calculus)

- デモプロジェクト（Demonstration Projects）
- 領域回復（Domain Recovery）
- ソフトウェア保存（Preservation）
- ソフトウェア開発民俗学（Ethnography）

（3）実世界ソフトウェア

今日のソフトウェアは、偶然的複雑性、予測不能性、合成不能性、脆弱性、実世界との相互作用に向かないといった問題点を抱えている。

また、ソフトウェアは「コードの集まり」という考え方にハマッテおり、大規模にするには人力を注ぎ込む、システム全体像の理解が喪失している、仕様とその実現が分離している、といった事態を招いている。

10年後には、どこでもコンピュータだらけになって、これまでの世界とは全く変わった世界になるというのに、今日のようなソフトウェアの作り方をしていたら、世界は危険極まわりない場所となってしまう。それを避けたければ、技術の実施を遅らせねばならず、その場合には、自動車は、コンピュータ制御とはならず、交通渋滞にはまり込んでしまうだろう。

これを解決するには、モデリング言語、システムの合成、変換、過去のソフトウェア遺産の扱い、といった課題に取り組まねばならない。

[モデリング言語]

ソフトウェアの問題点は、「プログラムコード」、あるいは「仕様」にあるのではなく、「モデル」あるいは「設計」にある。すなわち、複雑な機能の実現、設計の理解、質問の定式化、振舞いの予測、といったことを行う際に、それを表現するモデリング言語を我々が持っていないことである。

これを解決するため、以下のような研究を行うべきである。

- システムのモデリング「言語」
- 有用な抽象化を見つけること
- 計算的システム理論
- 組合わせて使える抽象化
- 時間・並行・能力などの表現

[システムの合成]

システムを組合せて作る組織的方法、例えば、要素の枠組、組合せの意味論、その場での（on-the-fly）合成、既成のソフトウェア要素との統合、といったことが欠如している。

これを解決するため、以下のような研究を行うべきである。

- 意味の枠組みと理論
- 方法論（methods）とツール
- 実験台と挑戦的問題
- 参照実装（reference implementation）
- アーキテクチャ枠組みの定義
- 分散・分割の戦略
- 粒度とモジュラー性の制御戦略

[変換]

異なる抽象間の変換についての理論，例えば、異なる抽象の間の関係、生成器（変換器）、複数の視点を許す抽象化(multi view abstraction)、物理的環境の抽象化、高信頼性ソフトウェアとシステム（HCSS：High Confidence Software and Systems）との関係: 証明つきの変換、といったことが欠如している。

これを解決するため、以下のような研究を行うべきである。

- オープンな生成器インフラ（方法論、ライブラリ）
- 生成器の理論
- Methods for correct by construction transformers (?意味不明?)
- 変換器のパラメタ化

[過去のソフトウェア遺産の扱い]

過去の遺産を扱う方法論，例えば、システムを徐々に現代化する方法、新しいものを古いものと統合する方法論、といったことが欠如している。

これを解決するため、以下のような研究を行うべきである。

- 遺産コードの部品化
- 遺産からの抽象化の抽出
- 漸次現代化、リバースエンジニアリング
- 再設計を安価に ⇔ ソフトウェア遺産からの進化

（4）ネットワーク中心のシステム

今後のシステムは、以下のような状況に進展すると予想される。

- すべてがコンピュータ
- すべてがネットワークコンピュータ
- すべてが相互依存する可能性
- 実世界と接続
- ますますヘテロに

- 接続はますます多様で不安定に

その状況下で、以下のような応用システムが登場してくるであろう。

- 交通制御: 自動車数千台からセンサーデータ
- 戦域管理: 敵対環境で多様粒度の調整·任務
- サプライチェーン管理
- 科学データの共同解析

  数千ユーザ、数千資源に対し、ソフトリアルタイムで問合せ最適化の要請
- 家庭内電力管理

このような状況、応用システムにおいては、以下のような技術的課題が生じる。

- 資源（QoS, 時間·相互依存·エネルギー消費·大きさ）を意識したプログラミング
  - 工学的トレードオフを考慮する計算モデルの欠如
  - 複雑なシステムでは端点間の特性が不明
- 動的な資源の振舞い: 故障·不均一性 等
  - 計算モデルや端点間特性の欠如
- ソフトウェア遺産
  - 設計時に想定しなかった利用形態への対応
- 危機・信用・制御の管理
  - ポリシー·保障·管理のドメイン
  - 極度に動的なシステムの安全性確保と確認手法
  - プライバシ
- スケールの問題
  - 構成要素数、構成要素の大きさ
  - 単一の計算に関係する要素数
  - ネットワーク中心システムは長時間無停止動作

これらの問題を解決するには、以下のような研究方向とアプローチが必要である。

- 「問題」を扱うプログラミングモデル
  - 工学的トレードオフ
  - 大規模システムの運用
  - 動的な資源の振舞い
  - 危機・信頼・安全の管理
- 「問題」の基本機構
  - 資源のトレードオフ: QoS、実時間性、等
  - 適応的振舞い

- さまざまな次元への拡大
- 分散管理

- 「問題」の全体を扱える、多レベルの資源管理
- ネットワーク中心の（小規模な）要素を、動的に大規模システムに合成する技術、相互運用性

この他、次のような研究も必要である。

- 新たな計算モデルを使うためのツール
- 高度に複雑化したシステムの検証・シミュレーション・理解
- 大規模分散応用システムの（自動）構成･管理

これらの研究によって、以下のような効果が期待される。

- 社会的には
  - 現状では作れないものを作れるように
  - ネットワーク化システムの信頼性･制御性向上
    工学的設計によって実現 ⇔ デバッギング
  - ソフトウェア開発コストの抑制
- 教育
  - 新たなソフトウェアの作成・システムの設計にあたれる人材の供給

3.2.3　SDP ワークショップ提案のまとめ

4 つのグループからのそれぞれの提案は、別々の方向を向いているのではなく、ほぼ同一方向を向いており、概ね、以下の提案にまとめられる。

- 有用な抽象化を含んだモデリング
  - モデリングは、対象とする特定領域のモデルを記述できるものとし、その特定領域の専門家が使いこなせる。
  - モデリングは、資源のトレードオフ・動的管理、危機・信頼・安全の管理といった、設計の意図を表現できる。
  - ある特定の側面からのモデリングを組み合わせ・合成・変換・統合して、様々な側面からのモデリングができ、システムの複雑さを軽減できる。
  - モデリングは、科学的、工学的設計に基づいた軽い形式的枠組みを持ち、これからコードの自動生成ができると共に適合性の検証も行える。
- オープンな協調開発環境
  - Web をベースとしたオープンで協調して開発できるプラットフォーム。
  - この環境には、芸術的ソフトウェアが蒐集され、ソフトウェアの宝庫とな

る。

- 柔軟性の価値なども定量化され、ソフトウェアの経済的社会的評価ができる。

- 既存ソフトウェア資産を生かす仕組み
  - システムを徐々に現代化する仕組み
  - 新しいものと古いものとを統合する仕組

3.3　2003 年度 SDP の活動計画

ワークショップからの提案を受けて、米国政府は、通称Blue Book 2003と呼ばれる「ネットワーキング及び情報技術研究開発—2003 年度大統領予算教書補足資料」[3]において、SDPに対する2003年度の活動計画を以下のように示している。

3.3.1　取り組み方針

2002 年に、半導体業界は地球上にいる人間の数より多いコンピュータ用チップを生産すると予測されている。これらチップの約 1%だけが認識可能なコンピュータに使われ、ほとんどは他のタイプの機器に組み込まれる。すなわち、小さなものでは携帯電話、ポケットベル、パーソナルデジタルアシスタント（PDA）やリモコンであり、大きなものとしては車、飛行機、製造機械、兵器システム、緊急警報ネットワークのような複雑な物理的システムなどである。組み込み型コンピューティング技術にとって最も有望な新しい方向は、ほとんどリアルタイムで重要なデータを収集し、転送することのできるネットワーク化されたマイクロセンサシステムである。このようなセンサ網は、環境と危険モニタリング、科学的農業、輸送管理、戦闘場面における偵察と早期警戒システムを含む、広範囲のアプリケーションに使われるようになっている。

IT 研究者が、国家のコンピューティング及び情報通信インフラストラクチャ上で動いている現世代のソフトウェアの重大な技術的脆弱さに取り組もうとしたとき、IT 研究者は、組み込み型技術における初期の変革において、独特の新しい難問に直面した。今日のソフトウェアは、その開発とメンテナンスに巨大な金額が費やされているもかかわらず、必要とされているほど相互運用性がなく、拡張性もなく、また、コスト効率に優れているわけでもない。

この問題を解くかぎは、これまではプログラミング“ウイザード”という寂しい技術によって行われてきたソフトウェア設計と開発を、質の高い結果を得るために広く受け入れられる原理、方法、最良の経験で支配された形式的枠組みの科学と工学の学問分野に変換することである。技術者は、技術的要求を満たした詳細な設計図、関連する基本原理や関連材料に対する最先端の訓練で養われた知識、長持ちする品質保証された構成部品を作るためのコスト効率の良い開発プロセスを持たずに、つり橋を架けるような夢を見ることはない。わが国のインフラストラクチャに根を下ろす重要な基盤として、ソフトウェアは少なくとも科学的に設計されたものであるべきである。

NITRD の研究のひとつは、ソフトウェア開発の新しい科学のために、モデルを作り出し、試験し、評価することに重点を置いている。目標は、今日の特異的で少数の人のみが理解できるコードから脱却し、合理的でモジュラリティがよく、実証的で再使用できる、

安定した工学的設計に移行することである。科学的な原理と手法は、開発者やユーザが、自動化された技術を使って、ソフトウェア製品を設計し、構築し、試験し、厳しい使用テストをすることを可能にする。ここで自動化された技術は、ソフトウェア製品が利用される前に、それらの動作を確認し、大規模ソフトウェアの何百万のもの手作りのコード中に潜む、今すぐ簡単に見つけられない弱点を正確に指摘することができるようにする。設計と実装の局面を自動化し、実践的なテストベッドを使うことは、プログラミング段階を合理化するだけでなく、高価なデバッグ工程の効率を上げ、改良することによって、現在の極めて高い開発コストを抑制することになろう。

最終的には、より高品質のより多くの製品を作ることを可能にするだけではなく、実質上開発費を下げ、維持を容易にすることにより、ソフトウェアの「生産性」を劇的に向上させる開発方法論を得ることが目的である。

ソフトウェアのための科学的基盤は、組み込み型システムにとって極めて重大なものであり、大規模な物理的システムの内側奥深くにある小さなコンピュータ・コンポーネントは、極めて多くのコンピューティング及びノン・コンピューティングタスクをリアルタイムにサポートするための機能を持っている。ちょうど、自分がどこにいるか、そして、または何をしていようとも、我々の心臓と肺は命令されなくとも機能するように、組み込み型プロセッサはその仕事を自動的に絶え間なく行うことができなければいけない。ちょうど我々の命は、新しい細胞が生まれて古い細胞が死ぬことにほとんど影響されないように、組み込み型システムは、いくつかの個々の組み込み型プロセッサが損傷を受け、破壊され、付け加えられ、あるいはアップグレードされても、全体としてのシステムが自動的に機能し続けている間、動作し続けることを期待されている。

2003 年度に、NITRD 参加機関は、ソフトウェアの品質を向上させ、その費用を全面的に削減するための基礎研究を支援する。この先進的技術は不可欠の駆動力となり、我々は、これにますます依存度を深めることになって、大規模及び小規模アプリケーションの両方において、社会により、大きな利益をもたらすことになる。

3.3.2 2003 年度の SDP の主要な研究課題

言語とコンパイラを含むソフトウェアとシステム科学；

構築方法論；

分散型で拡張性のあるシステムとネットワークの設計基盤；

効率的で、信頼性の高いソフトウェア・コンポーネントとその統合プロセスを含む自動化されたエンジニアリング；

統合ソフトウェアとシステム開発のための(仕様、分析、検証などの) 方法論；

並列実行するネットワーク化されたアプリケーションの相互運用性；
ドメイン特有の開発環境の速やかな構築とカスタマイゼーションが可能なツール環境；
組み込み型アプリケーションとシステム開発プロジェクトのための技術に関するパイロットアプリケーションと実践的研究；

3.3.3　2003 年度における各機関の代表的活動

NSF：

・実践的なソフトウェア工学研究
・コンポーネントベースのソフトウェアに於ける絶えず生じる変化に対する管理
・ソフトウェアを発展させるプロファイルとパターン
・リスクドリブンからバリュードリブンの開発モデルへと変更するための戦略的ソフトウェア設計

DARPA：

・組み込み型システムのためのプログラム作成(Program Composition for Embedded Systems：PCES)

PCES は、組み込み型システムのプログラミングのための新しい戦略を開発しており、技術的労力を抑えてより堅牢なコードを作ることを目的としている。この手法にはコアソフトウェアの設計が絡んでおり、並行操作、センサ、時間的制約のあるアクチュエータの同期、安全で高効率のキャッシュ、レジスタ、メモリ管理などの組み込み型システム全体に必要とされる横断的機能を含んでいる。目標は、ダイナミックな構成とテストのための自動化されたツールを用いた再利用可能なソフトウェアスイートである。

・組み込み型ソフトウェアのモデルベースの統合
・ソフトウェアによる制御、センサ網あるいはシステム網で構築される高度に複雑なシステムでのダイナミックな自己構成と保証

NIH：

・バイオメディカル・コンピューティング・アプリケーションを支援するソフトウェア研究

NSF/NASA：

・高信頼コンピューティングとコミュニケーションシステムの研究に関する共同研究プログラム

これは、現実世界のハードウェアとソフトウェア人工物に関する研究資金援助を評価するための NASA の新しいテストベッド施設を使って、信頼性があり、コスト効率

の良いソフトウェアをベースにしたシステムを設計し、試験し、実装し、発展させ、保証することをねらいとするプロジェクトである。

NASA：

・組み込み型あるいはロボットデバイスのための技術とツールを含む自動化ソフトウェア工学手法

・ベイジアン法を用いた仕様、ソフトウェアの実践評価

DOE/NNSA：

・全ての高度シミュレーションとコンピューティング（Advanced Simulation and Computing：ASCI）ハイエンドプラットフォームのための、共通のソフトウェア開発/実行環境の構築

これは、I/O、ストレージ、視覚化に対するニーズだけではなく、堅牢さと拡張性のためのエンドツーエンドASCIアプリケーションのニーズを支援するものである。

NIST：

・消費者団体と使い勝手に関するデータを共有するための共通の報告書形式の開発

・自動化された、知識ベースの手法を用いたソフトウェア品質の決定

・産業界のパートナーと製造B2B の相互運用性テストベッドを共有する計画

NOAA：

・コンポーネントを基本にした地球物理学環境のモジュラー研究モデルの開発

ODDR&E：

・組み込み型システムをチェックするソフトウェアモデルに関する大学における基礎研究

・リアルタイムの無停止ネットワークプロトコル

3.3.4 2003年度の予算

表3.2に示すように、SDPに対する予算は1.821億ドルであり、NITRD予算の約1割を占めている。

表 3.2　NITRD 関係機関の個別研究分野(Program Component Area)別の予算

2002年度予算（単位：100万ドル）

Agency	HECI&A	HECR&D	HCI&IM	LSN	SDP	HCSS	SEW	合計
NSF	225.1	66.4	132.6	102.8	44.4	45.8	58.8	676
NIH	79.5	7.7	82.0	105.6	5.6	3.6	11.4	295
DARPA	17.4	63.9	35.8	35.0	65.6			218
NASA	36.1	26.0	27.8	14.4	22.4	47.1	7.0	181
DOE	97.2	29.5	16.4	27.8			3.5	174
NSA		41.6		1.3		32.7		76
NIST	3.5		3.2	6.2	7.5	2.0		22
NOAA	13.8	1.8	0.5	2.8	1.8			21
ODDR&E		2.0	1.8	5.7	1.6	1.0		12
AHRQ			8.0	6.0				14
EPA	1.8							2
小計*	474.4	238.9	308.1	307.6	148.9	132.2	80.7	1,691
DOE/NNSA	42.1	33.5		25.9	33.2		4.2	139
合計*	516.5	272.4	308.1	333.5	182.1	132.2	84.9	1,830

HECI&A: ハイエンド・コンピューティング基盤とアプリケーション

HECR&D: ハイエンド・コンピューティング研究開発

HCI&IM: ヒューマン・コンピュータ・インターフェースと情報管理

LSN: 大規模ネットワーク

SDP: ソフトウェアの設計と生産性

HCSS: 高信頼ソフトとシステム

SEW: 社会、経済、および労働力の面から見た IT 労働力開発

*小計と合計に 2003 年度大統領予算教書と若干の相違があるのは、集計における個別研究分野の多少の変更と端数処理の違いによる

3.4 SDPの目指すものに関連した技術、システムの動向

SDPの将来の方向性として、前述のように、モデリング、形式的枠組み、オープンな開発環境、既存資産の活用を提言し、キーワードを並べているが、まだ、具体的なイメージがつかめない読者も多いことであろう。SDPに参加したメンバも将来構想であるから、具体的イメージを持っているわけでもなく、各メンバの将来イメージもそれぞれ異なっているので無理からぬ話である。しかし、参加メンバは、予算獲得の思惑があり、各メンバが現在研究していること、あるいは関心があることの延長線上に将来イメージがあるであろう。そこで、SDPが目指す方向に関する、以下に述べる現在の研究動向を把握し、それから、その延長線上の将来イメージがどんなものであるかを把握してもらいたい。

3.4.1 モデリング、特定領域記述言語(DSL: Domain Specific Language)

IT革命が進行し、ITと他分野との融合が加速していけば、融合分野の専門知識が希薄なITプログラマは対応が難しくなってくる。従って、ITプログラマがプログラミングするのではなく、各特定領域の専門家がプログラムできるものでなくてはならない。特定領域には、多くの場合、建築設計図、電気回路図、損益対照表といった特定領域固有のモデリングドキュメントが既に存在する。このようなモデリングドキュメントをベースにITシステムを記述しようとするアプローチが特定領域記述言語である。

これは、従来、プログラミングとは、インプリメントすることを設計してきたが、モデリングを設計することによって行おうとするアプローチであり、このモデリングが特定領域固有のモデリングドキュメントに他ならないのである。しかし、ただ単にモデリングドキュメントで終わらずに、実行できるものではなくてはならない。従って、モデリングドキュメントを「軽い形式的枠組み」にしたり、「メタモデリング言語」でモデリングドキュメントの意味を規定したりして、あるいは、暗黙的な知識を使ってシステムが自動的に、実行コードの生成をサポートにする。

特定領域記述言語の代表的なのは、スプレッドシートで表に関する処理はほぼ何でもこなすことができる。表以外で注目されるシステムに、Simulink、 Ptolemy、OMGで検討が始まったMDA、Webサービスをベースにビジネスフローを記述するBPEL4WSがある。

（1）Simulink

Simulink[4]は、MathWorks社の製品で、DSPの設計、コミュニケーションシステムの設計、その他シミュレーションシステム向けのダイアグラムブロックの開発ツールである。その特徴は、150以上のブロックを含むライブラリやモデル（出来合いブロックダイアグ

ラム）が準備されており、これらを組み合わせてブロックダイアグラムを作成、編集できる。

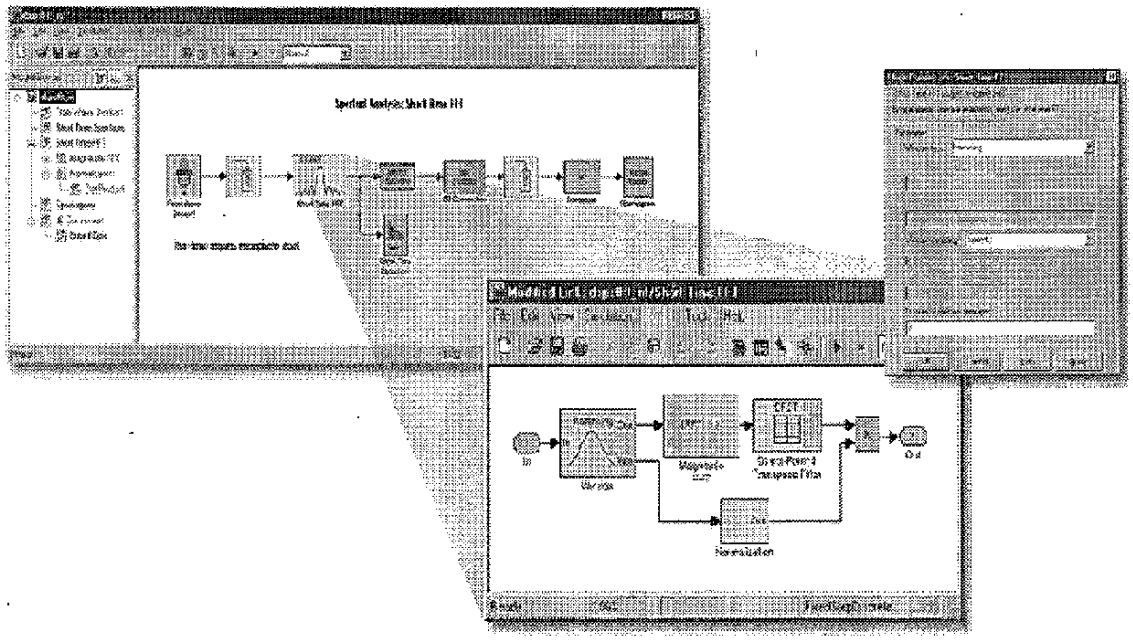


図 3.1　.Simulink FFT の例

図 3.1（http://www.mathworks.com/products/simulink/description1.jsp）は、短時間の高速フーリエ変換ブロック（FFT）の例であり、これは DSP ブロックを用いて作成されている。FFT ブロックに対するパラメタは右のダイアログボックスで入力される。ユーザがサブシステムの詳細を見たければ、右下に示すその詳細ブロックダイアグラムを表示することができる。

Simulink では、ユーザ作成のカスタムブロックをシステムに組み込む、あるいは C 言語などのコードをブロックダイアグラムに組み込むことによって、システムを拡張することができる。また、作成したブロックダイアグラムを、インタラクティブに実行することができ、結果を即座に表示したり、グラフィカルデバッガでデバックしたりすることができる。

（2）Ptolemy

Ptolemy[5]は、カルフォルニア大バークレー校のプロジェクトであり、組み込みシステムのような複雑なシステムのモデリングと設計を支援するシステムの構築を目指している。その狙いは、以下の実現にある。

①　組み込みシステムの開発におけるモデリング、設計の方法論の確立

② 適切なコンピュテーションモデルの選択によるシステム設計品質の向上

③ 多数のコンピュテーションモデルを実行するモデルの生成と相互運用

④ コンポーネントベースの設計

Ptolemy プロジェクトでは、ビジュアルエディタである Vergil を用いてモデリング、設計を行う。この Vergil の特徴は、単一のコンピュテーションモデルではなく、複数のコンピュテーションモデルの中から選択できることである。Vergil では、継承構造、手続きインターフェースを主体とするオブジェクト指向ベースのモデリングではなく、並列性やコミュニケーションモデルを主体としたアクタベースとした、以下に示す 9 つのコンピュテーションモデルが用意されている。

Communicating Sequential Processes - CSP

Continuous Time - CT

Discrete-Events - DE

Distributed Discrete Events - DDE

Discrete Time - DT

Finite-State Machines - FSM

Process Networks - PN

Synchronous Dataflow – SDF

Synchronous/Reactive - SR

2 つ目の特徴は、上記のモデルがグラフィカルに表現されることである。図 3.2 (http://ptolemy.eecs.berkeley.edu/publications/papers/01/overview/overview.pdf) は、Discrete Event モデルの例で、バス停に乗客がポアソン過程で到着し、バスが一定間隔で到着したときと、ポアソン過程でランダムに到着したときのシミュレーションの比較を示す。

図 3.3 (http://ptolemy.eecs.berkeley.edu/publications/papers/01/overview/overview.pdf) は、Finite-State Machine の例を示す。

3 つ目の特徴は、上記のグラフィカルなモデルは、コンピュテーションモデルに基づいてため、意味の形式化がなされ、実行可能で、モデルが意図するものであれば、信頼性のおけるプログラムとなる。

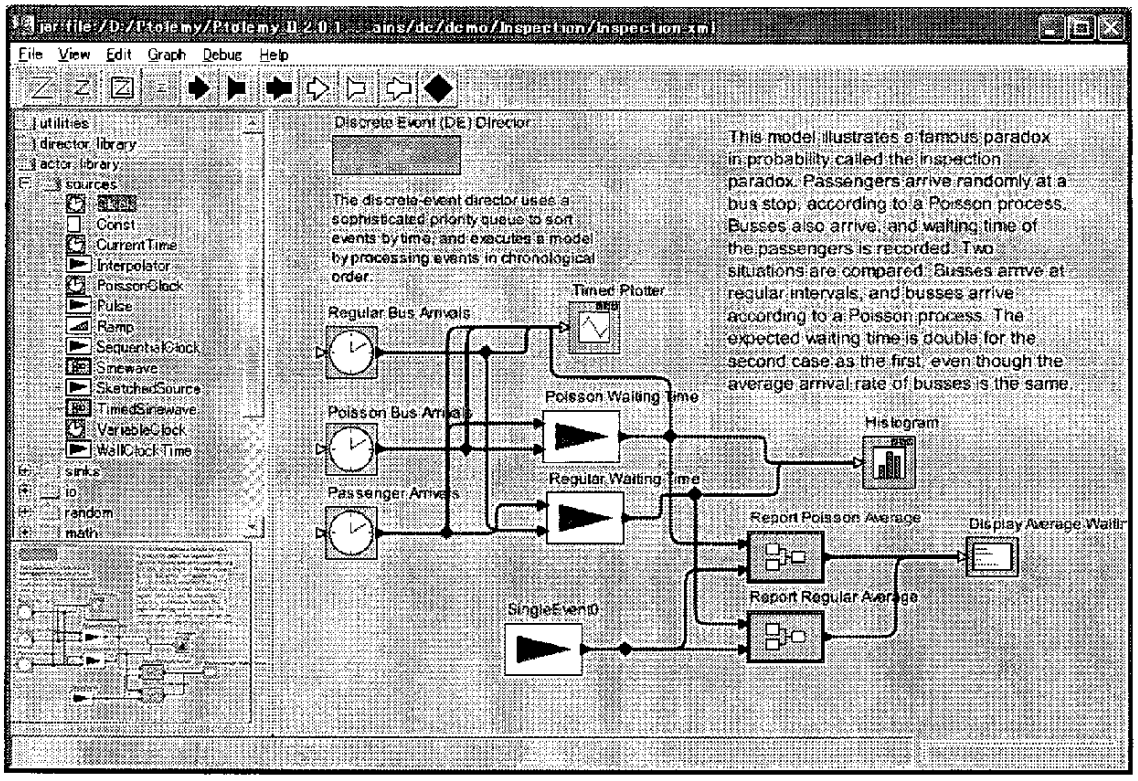


図 3.2　Vergil のバス停における待ち行列のシミュレーション例

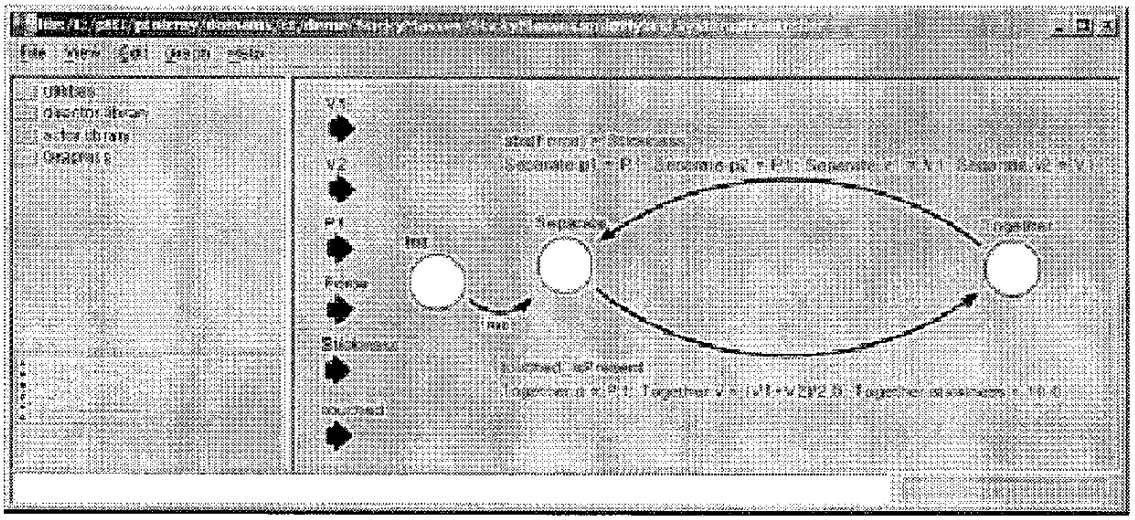


図 3.3　Vergil の状態遷移図の例

（３）UML,MDA

UML(Unified Modeling Language)は、オブジェクト指向に基づく開発方法論で使われるドキュメント(図)を統一したものであって、以下の９種類の図を規定している。

ユースケース図……システム要件となる機能を表す図

クラス図……クラスの階層構造、振る舞いを表す図

オブジェクト図……個々のオブジェクトの振る舞いを表す図
シーケンス図……オブジェクト間の協調動作（メッセージングの順序）を表す図
コラボレーション図…オブジェクト間の協調動作（メッセージングとコンテキスト）を表す図
状態遷移図……オブジェクトの状態の変化の様子を表す図
アクティビティ図……操作の中で実行されるアクティビティの流れを表す図
配置図……システムのハードウェアとソフトウェアの物理的なアーキテクチャを表す図
コンポーネント図……コードのコンポーネントの物理構造を表す図

しかし、このようなUMLには大きな二つ課題を抱えている。一つ目はUMLの図は、モデリング言語ではあるが、オブジェクト指向言語によるインプリメントを前提としたものであって、ＩＴシステム設計者、プログラマ向けのものである。業務で使うモデリングではなく、業務の専門家が使えるものではない。二つ目は、実行コードの生成までを可能とする情報は含んでいないため、設計からインプリメント、テストへとシームレスな開発が行えず、また、ツールを使えば、或る程度は補えるが、実行コードとＵＭＬとの不一致が生じたりする。

この課題を解決する動きが、MDA(Model Driven Architecture)[6]、UML2である。

MDAはOMG(Object management Group)が提案しているものであり、業務に視点を置いたモデリング(PIM: Platform Independent Model)とインプリメントに視点を置いたモデリング(PSM: Platform Specific Model)とに分けてモデリングを行い、この二つのモデリングをマッピングルールで関係付けを行おうとするものである。

PIMの一つとして、企業でのビジネスプロセスを対象としたエンタープライズ分散オブジェクトコンピューティング(EDOC: Enterprise Distributed Object Computing)向けにUML に対して、ビジネスプロセス、イベントドリブン、コンポーネント、パターンなど様々な拡張が検討されている。PSM も分散オブジェクトをサポートする CORBA、EJB、.NET、XMLが検討されている。図3.4（http://cgi.omg.org/docs/ptc/02-02-05.pdf）にEDOCのCCA(Component Collaboration Architecture)による販売の例を示す。

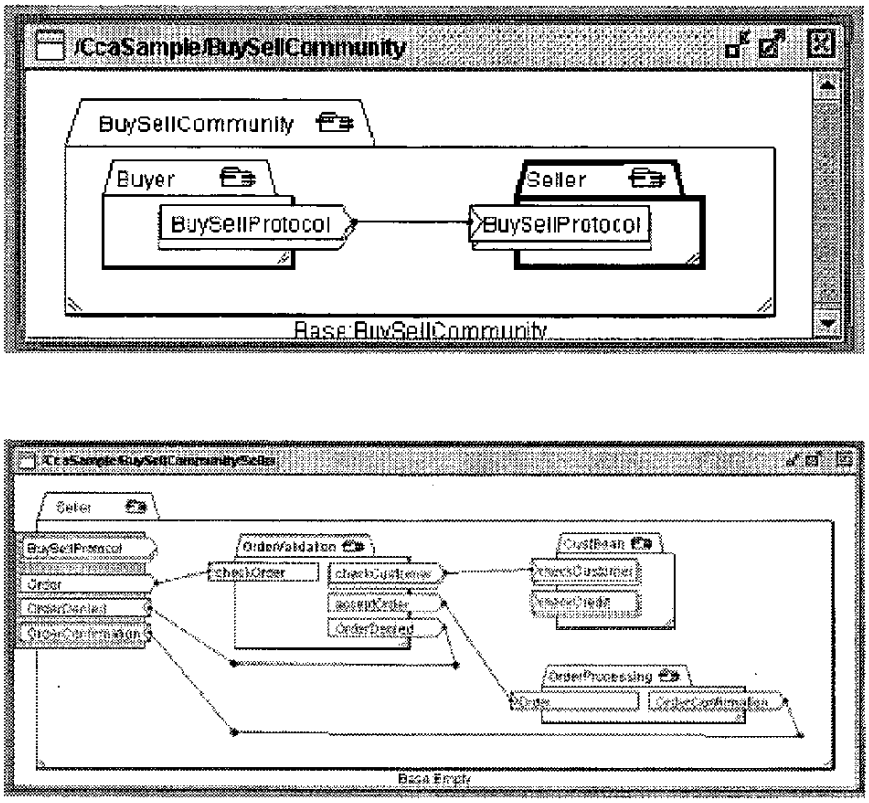


図 3.4　EDOC の CCA(Component Collaboration Architecture)による販売の例

（4）BPEL 4 WS

BPEL4WS(Business Process Execution Language for Web Services)[7]は、ウェブサービスをベースとしたワークフローによるビジネスプロセスの振る舞いを記述する XML ベースのドキュメントであり、IBM、Microsoft、BEA が共同で提案しているものである。BPEL4WS は、XLANG と WSFL の概念を融合しており、それらに取って変わるものである。BPEL4WS は、ビジネスプロセスの二つの側面、ビジネス活動における参加者の振る舞いのモデルを進行させる実行プロセスと各業者間のメッセージ交換の振る舞いを記述するビジネスプロトコルとを記述する。

BPEL4WS では、ルートの Process 要素の下位要素として、Partners 要素、Conntainers 要素、CorrelationSets 要素、FaltHandlers 要素、CompensationHandler 要素、及びアクティビティを記述する。

Partners 要素では、ビジネスプロセスの過程で登場する各業者について、Partner 要素でサービスリンクタイプと役割を記述する。Containers 要素では、ビジネスプロセスで使われる各データ内容について、Container 要素で記述する。CorrelationSets 要素では、各相互関係について、CorrelationSet 要素で記述する。エラー処理については、FaltHandlers

要素、CompensationHandler 要素で記述する。

アクティビティは、ビジネスプロセスの活動内容を以下の要素で記述する。

- Receive 要素　メッセージの受け取り
- Reply 要素　メッセージに対する返答
- Invoke 要素　パートナーの PortType で提供されるサービスの起動
- Assign 要素　新しいデータで Container の内容の変更
- Throw 要素　例外処理の生成
- Terminate 要素　ビジネスプロセスの終了
- Wait 要素　或る時間待つ
- Empty 要素　無操作
- Sequence 要素　順次に各操作を行う
- Switch 要素　分岐操作を行う
- While 要素　繰り返し操作を行う
- Pick 要素　メッセージの受け取りまたはタイムアウトでアクティビティを実行
- Flow 要素　並列操作を行う
- Scope 要素　エラー処理の範囲を定義
- Compensate 要素　補償処理の呼び出し

3.4.2　多面的ソフトウェア

(1) アスペクト指向

オブジェクト指向は、システムを「もの」、すなわちオブジェクトという側面から分割し、対象システム中の「もの」をモデル化し、その性質（手続きとデータ）を記述する。しかし、対象モデルの性質は、一つの側面ではなく、いくつかの側面を含んでいる場合が多い。このことに着目し、「もの」という側面からではなく、他の側面からもシステムを分割しようとするのがアスペクト指向[8]である。先ず、オブジェクトの側面からシステムを分割し、分割したオブジェクトを横断するような関心事があれば（複数のクラスに同じ関心事があれば）、その関心事に対する機能を一まとめにして分割の単位とする。このように、アスペクト指向とオブジェクト指向とを組み合わせてシステムを構築する。

オブジェクトに跨る関心事には、すべてのオブジェクトに共通な関心事の例として、セキュリティチェック、課金、トレース、ロギング等があり、幾つかのクラス間の取り決めの例として、Observer — Subject のようなデザイン・パターンに見られるクラス、図形描画における図形要素と画面の再描画の関係といった構成要素と全体の管理の処理、クラス間の役割を明確化するメソッド呼び出しにおけるプレコンディション・ポストコンディ

ション等がある。また、システム統合を行う場合にもアスペクト指向は有効である。例えば、販売管理システムと配送システムとを統合する場合には、商品の重量、大きさ、配送スケジュール等といった配送の側面から機能を一まとめとにして、販売管理システムに後付けで（販売管理システムに大きな変更を加えずに）追加することもできる。

このようなアスペクト指向をサポートするシステムには、Hyper/J、Aspect/J 等がある。Aspect/J を用いた図形描画における図形要素と画面の再描画の例で、具体的にどのようにコーディングするかを図 3.5 に示す。

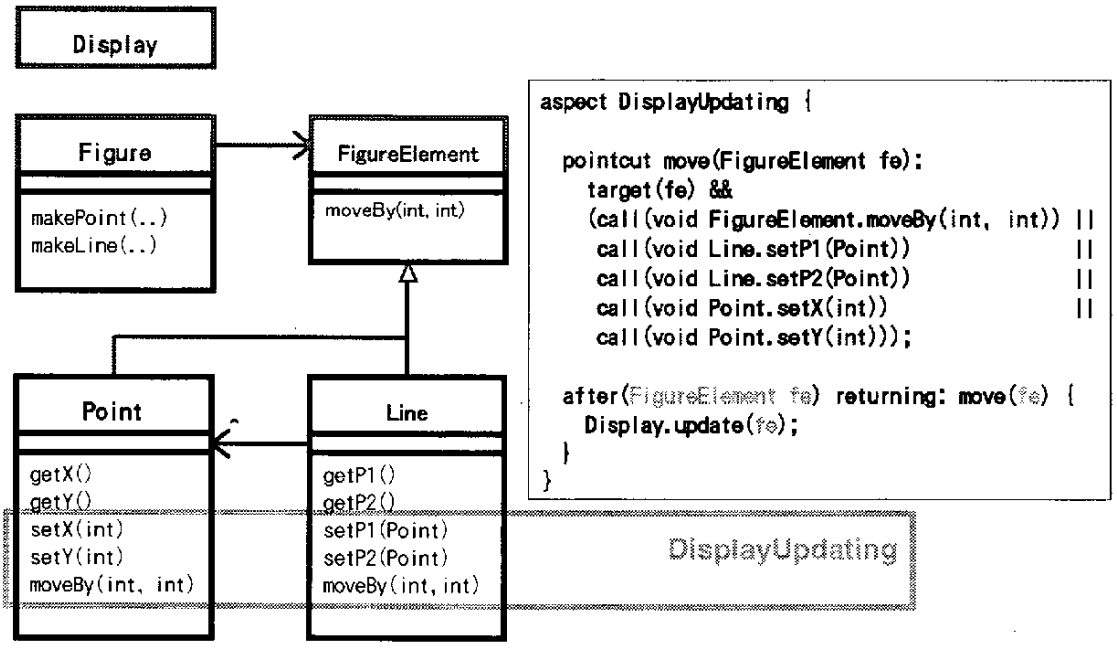


図 3.5　Aspect/J による図形要素と画面の再描画の例

図 3.5 の左に示したのが、オブジェクト指向による図形描画のクラスの階層構造である。ここで、図形要素の Point、Line が変更されたときには、必ず画面の再描画をしなければならないが、その処理は、各図形要素すべてで記述しなければならなく、モジュラリティがよくない。アスペクト指向では、その部分を別モジュール aspect DisplayUpdating で記述する。そこでは、pointcut 文で Point、Line が変更されたという関心事 move を（call(FigureElement.mobeBy||Point.setX||Point.setY||Line.setP1||Line.setP2)）で指定し、そのときのオブジェクトを fe と指定する。そして、関心事 move が呼び出された後処理として行うべき処理（画面の再描画）を after 文で指定している。

3.4.3 科学、工学的な設計

現在のソフトウェアが抱える低い信頼性、生産性、相互運用性、柔軟性は、技術的な脆弱性に起因している。それを解決するには、ソフトウェアは、科学的原理、方法、経験に基づいて設計すべきであり、このような科学、工学的な設計によって、実行コードを自動生成したり、事前にエラー、矛盾性をチェックしたり、テストを自動化したりしようとする試みである。このような試みに、軽い形式的枠組み、Semantic Web がある。

（1）軽い形式的枠組み

軽い形式的枠組みとは何かを説明する前に、先ず、形式的枠組み、擬似形式的枠組み、及び、非形式的枠組みについて説明しよう。

形式的枠組みは、Hoare のロジックやリレーショナルデータベースの計算のように、数学的理論で裏づけされた概念で記述しようとする枠組みである。これは、類似、プリ／ポストコンデション、信頼／保証、連続性といった強力な概念の宝庫ではあるが、多くのアプリケーションにおいて、実用的ではない。

一方、非形式的枠組みは、デザイン・パターン、エクトリーム・プログラミングがその例で、優秀なプログラマ、マネジャーの経験を発掘して、その経験を実践しようとするもので、非常に効果的な方法である。しかし、人の経験的能力に依存しており、経験的能力を身に付けるまでの時間、教育コストが課題である。

また、擬似形式的枠組みは、UML がその代表例であり、意外に有効ではあるが、その有効性は、ソフトウェア設計を形式化、標準化することによって得られるものであり、次善の策と言えよう。

軽い形式的枠組み[9]は、型付けシステム、モデルチェッキング、ランタイムモニタリングといった実用的で強力な概念を採り入れる試みである。　これらの概念を拡張してより強力にしたり、新たなこのような概念を見つけていけば、現在のソフトウェアが抱える問題を飛躍的に解決する可能性がある方法である。

例えば、型付けシステムは、コンポーネントとその動作環境との間の約束事を決めて置き、コンパイル時に、その約束事に従っているかを自動的にチェックするものであるが、実行時に動作環境が動的に変わるような場合においても、この型づけの概念、機構を適用されるようにすれば、証明付きコードを実現できる。また、モデルチェッキングは、ハードウェア設計において、流れを示すのに非常に有効であるが、これをソフトウェアに対しても適用できるようにする。ランタイムモデリングは、不正なふるまいを事前に推測するといった労力のいる仕事をしなくても、実行時に実際に不正な振る舞いするコンポーネントを容易に検出できるようにする。

（2）メタデータ、Semantic Web

メタデータ、Semantic Web[10]は、W3C などの団体が中心となって、その標準化、推進活動を行っている。メタデータは、データについてのデータであり、データを或る目的のために扱いやすくするデータのことである。例えば、以下のようなデータがある。

・データを構造化したデータ

・データリソースについてのデータ

・カタログデータ

・データを共通認識させる意味データ

Semantic Web は、データについての意味情報であり、W3C のディレクターであるティム・バーナーズ・リーらによって提唱された。その考え方は、サイエンティフィック・アメリカン誌で、以下のように紹介され、計算機が情報の意味を理解できるようにし、各意味情報を用いて、自動的な処理、知的な推論処理を行うことによって、人々が知識共有をグローバルに効率よく効果的に行うことを目指している。

「The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.」

Semantic Web は、メタデータの拡張として記述される。

図 3.6 に示すように、Semantic Web は、XML Schema、RDF、RDF Schema、DAML+OIL などのメタデータの記述を階層的に定義することよって記述される。

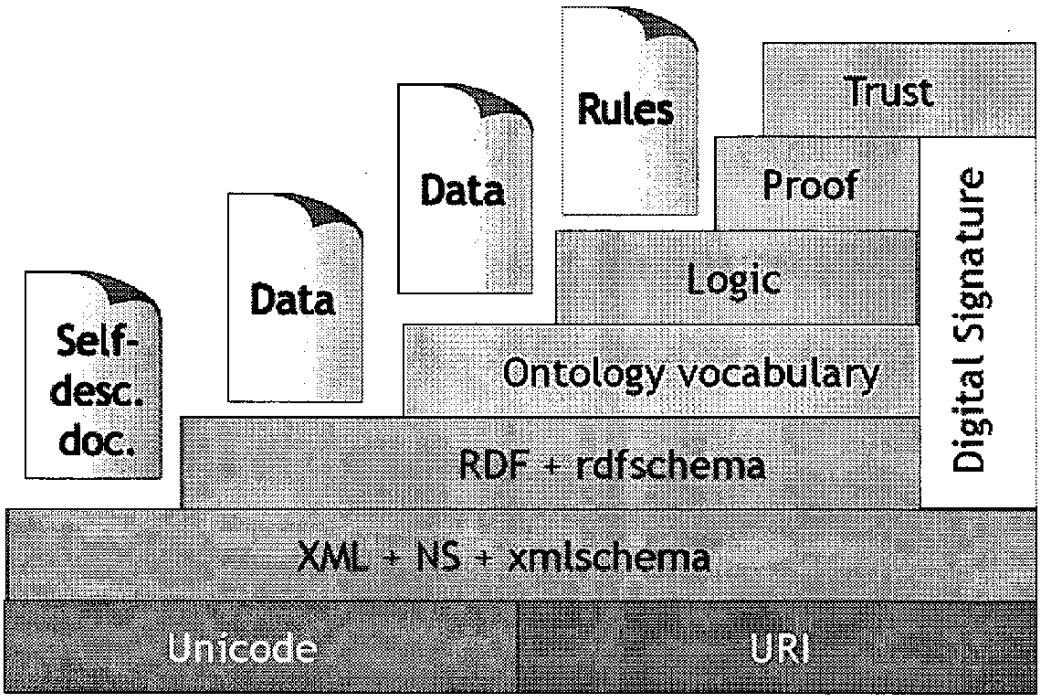


図 3.6　Semantic Web のアーキテクチャ

Semantic Web プロジェクトの特徴は、セマンテックスの標準化に向けて、非集中型のアプローチを採用していることである。データに対し意味を付与し、計算機が理解可能とするには、XML のタグ、プロパティの用語を標準化し、その意味付けを明確化するメタデータの定義が必要となるが、そのメタデータの標準化にあたっては、一つに統一するのではなく、複数の標準が存在することを許容している。それぞれの目的のために、数多くのメタデータが定義され、標準化されてくると、それらの複数のメタデータには、同義語、同音異義語などが生じ、用語の相違が生まれる。このような場合には、同義語辞書（オントロジー）を作ることによって、その用語の相違を解消していこうとしている。この同義語辞書もメタデータで、図 6 の Ontology vocabulary レイヤで定義される。同義語辞書を作成してもセマンティクスギャップ（用語の相違）は完全には解消されないが、現実的なアプローチであると言える。データベースがデータの整合性、一貫性を死守するあまり、分散化、共有化が進まなかったのに対し、Web が、データの整合性、一貫性を放棄して分散化を進め、ボトムアップの草の根型の発展を遂げ、Web をグローバルな巨大なデータベース化するのに成功したように、セマンティクスギャップを許容することによって、ボトムアップの草の根型に意味、知識が蓄積されて、巨大な知識ベースが構築されることを目論んでいる。　次に、図 3.6 に示した Semantic Web アーキテクチャの各レイヤの内容を以下に示す。なお、Logic, Proof, Trust レイヤは、構想段階では何も決まっていないので、省略する。

1) XML Schema

XML Schema[11]は、XML ドキュメントの以下に示すクラス情報を記述する。

・構造情報：どのようなエレメント、属性から構成されているか

・データタイプ：そのエレメント、属性がどのようなデータタイプであるか

<xsd:element name="Price" type=xsd:decimal>

・データの基本情報：エレメントの頻出回数の制約などの情報

<element name="item" minOccurs="1" maxOccurs="unbounded">

これには、意味情報はなく、DTD と同様の機能を提供するが、DTD は XML でないのに対し、XML Schema は XML であるので、XML で使われているツールが XML Schema に対しても使えるメリットがある。

2) RDF(Resource Description Framework)

RDF[12]は、メタデータを表すための枠組みであり、計算機が理解可能とするための情報記述を与える。RDF では、リソースとプロパティと値の三つ組で構造を表す極めてシン

プルな構造化モデルを採用している。

例えば、リソース：http:// www.w3.org/Home/Lassila のプロパティ：著者の値：Ora Lassila は、次のように表される。

```
<rdf:Description about="http://www.w3.org/Home/Lassila">
  <Name>Ora Lassila</Name>
</rdf:Description>
```

値は、リソースを指してよいので、ネットワークとして定義していくことができる。「http:// www.w3.org/Home/Lassila（リソース）の著者（プロパティ）は、個人 http://www.w3.org/staffId/85740（リソース）であり、その個人の名前（プロパティ）は Lassila（値）、Email アドレス（値）は lassila@w3.orgである。」は、次のように記述される。

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <Creator>
      <rdf:Description about="http://www.w3.org/staffId/85740">
        <Name>Ora Lassila</Name>
        <Email>lassila@w3.org</Email>
      </rdf:Description>
    </Creator>
  </rdf:Description>
</rdf:RDF>
```

また、構造を表すモデルとしてコンテナモデルを採用し、<Bag>, <Sequence>, <Alternative>が用いられる。

3) RDF Schema

RDF Schema[13]は、RDF の以下のようなデータタイプを記述する。

・クラス、及び他のクラスとの関係を定義

オブジェクト指向風に<Class>, <subClassOf>が用いられる。

・プロパティ、及び他のプロパティとの関係を定義

<subPropertyOf>で他のプロパティの定義を継承することを示す。

・プロパティの値の範囲、属するクラスなどプロパティの制約を定義

<range>でプロパティの値の範囲を、<domain>でプロパティが属するクラスを指定する。

・意味（ラベルとコメント）を定義

<label>でラベルを、<comment>でコメントを指定する。

4) メタデータの例—Dublin Core

以上の RDF、RDF Schema を用いてメタデータが定義され、各業界、団体で標準化作業が行われている。その主なものには、以下のものがある。

・Dublin Core；書誌情報の基本語彙集

・P3P：プライバシに関するメタデータ

・RSS：サイト情報の要約と公開

・WSDL：リモートプロシジャコール（SOAP）のメタ情報

Dublin Core[14]は、Web や文書の作者、タイトルなどの書誌情報の基本語彙集である。次の 15 の基本エレメントタイプが決められている。

Title Creator Subject Description Publisher Contributor Date Type Format Identifier Source Language Relation Coverage Right

さらに基本エレメントタイプよりもっと厳密なメタデータ記述を行うために、以下のような修飾子が規定されている。

Title.alternative

Description.tableOfContents Description.abstract

Date.created Date.Valid Date.available Date.issued Date.modified

Format.extent Format.medium

Relation.isVersionOf Relation.hasVersion Relation.isReplaceBy Relation.replace Relation.isRequiredBy Relation.requires Relation.isPartOf Relation.references Relation.isFormatOf Relation.hasFormat

Coverage.special Coverage.temporal

5) DAML+OIL Ontology

DAML+OIL Ontology[16][17]は、RDF Schema に加えて、より詳細なクラス、プロパティの定義、制約、表記を記述する.

これらの記述のために以下のタグが規定されている。

cardinality cardinalityQ Class complementOf Datatype DatatypeProperty DatatypeRestriction Datatype value differentIndividualFrom disjointUnionOf disjointWith domain equivalentTo hasClass hasClassQ hasValue imports intersectionOf inverseOf maxCardinality maxCardinalityQ minCardinality minCardinalityQ ObjectClass ObjectProperty ObjectRestriction oneOf onProperty Ontology Property range Restriction sameClassAs sameIndividualAs samePropertyAs subClassOf subPropertyOf toClass

TransitiveProperty　UnambigousProperty　unionOf　UniqueProperty
versionInfo

3.4.4　オープンな協調開発環境

（１）オープンソースソフトプロセス

Linux は、オープンソースソフトプロセス（OSSP）[18]の成果である。OSSP を採用したが故に、Linux は、安定性・スケーラビリティ・カスタマイズ性・性能の向上、開発・バグ修正の素早さ、低コストをもたらし、マイクロソフトを脅かす存在となっている。OSSP は、単にソースを公開した方式にとどまらず、インターネットを最大限に活用した開発方式であり、市場原理に基づく開発方式とは異なるハッカー社会の贈与文化に基づく開発方式である。

[OSSP によるソフトウェア開発の特徴]

OSSP によるソフトウェア開発方法の特徴を以下に示す。

① OSSP の開発組織

- 地域的な広がり
  国際化、標準化に優位性を持つ。
- 万人が認めるリーダの存在
  Linux では Linus　Torvalds の存在
- 巨大な数の貢献者集団
  Linux ではコード提供者 200 人以上、バグ修正者 1000 人以上
- 金銭的動機ではなく、余暇に開発に貢献する個人集団
  （ただし、商業版 Linux が登場して変わりつつある）

② OSSP が成功する開発運営方法

- インターネットの共働技術を駆使
  メーリングリスト、ニュースグループ、Web
- 共通の目標
  「UNIX を作り直せ」＝＞方向性を明確にし、開発チーム全員の意思決定
  （「すばらしい OS を作ろう」といったあいまいな目標はダメ）
  UNIX の経験＝＞何がうまくいって、何がうまくいかないかが分かる。
- 共通の経験、スキル
  UNIX/gnu＝＞参入障壁を低くし、参加できる開発者を増やす。
- 並列競合開発とコンポーネント化の枠組み

コンポーネントの枠組みを確立し、複数の小チーム、個人が個別に開発。
競合した場合は、最もいい実装を選択。

・ 並列デバック

誰かが問題を見つけ、たいてい別の人がその問題を理解してそれを直す。
デバックは、プロジェクトに参加する人の数に正比例して効率が上がる。
これが、ブルックスの法則（開発者の人数を増やしても、管理、調整作業が増大して効率が比例して上がらない）から逃れる鍵

・ 紛争解決

Linux では、「やさしい独裁者」モデル
Linus Torvalds がプロジェクトのリーダ。
大きなコンポーネントを信頼できる副官に移譲。
副官は、サブコンポーネント開発者に移譲。
Apache は、共同開発者による議決委員会方式。

・ 動機づけ

開発者の個人的な悩みを解決
自分のエゴの満足とハッカー社会での評判
ハッカーの社会は、贈与文化。
ものが豊富な社会（ソフトが自由に共有される社会）では、競争的な成功の尺度は仲間内の評判
ノウアスフィアの開墾
未開の土地を開拓し、自分のものにすること
弱者の反発（反マイクロソフト）

・ コードの分裂の回避

万人が認めるリーダの存在：Linus Torvalds
GPL ライセンス（派生したものはフリーにしなければならない）
ハッカー社会での評判が低下する恐れ

[OSSP 開発を効率化する法則]

Linix 上の fetchmail を開発したレイモンドは、その OSSP を実践した経験から、OSSP 開発を効率化する法則を「伽藍とバザール」[19]に書いている。その法則を以下に列挙する。

①よいソフトはすべて、開発者の個人的な悩み解決から始まる。
②何を書けばいいかわかっているのがよいプログラマ。なにを書き直せば（そして使い

回せば）いいかわかっているのが、すごいプログラマ。

③捨てることをあらかじめ予定しておけ。どうせいやでも捨てることになるのだから（フレッド・ブルックス『人月の神話』第11章）

④まともな行動をとっていれば、おもしろい問題のほうからこっちを見つけだしてくれる。

⑤あるソフトに興味をなくしたら、最後の仕事としてそれを有能な後継者に引き渡すこと。

⑥ユーザを共同開発者として扱うのは、コードの高速改良と効率よいデバッグの一番楽ちんな方法。

⑦早目のリリース、頻繁なリリース。そして顧客の話を聞くこと。

⑧ベータテスタと共同開発者の基盤さえ十分大きければ、ほとんどすべての問題はすぐに見つけだされて、その直し方もだれかにはすぐわかるはず。

⑨賢いデータ構造と間抜けなコードのほうが、その逆よりずっとまし。

⑩ベータテスタをすごく大事な資源であるかのように扱えば、向こうも実際に大事な資源となることで報いてくれる。

⑪いいアイデアを思いつく次善の策は、ユーザからのいいアイデアを認識することである。時にはどっちが次善かわからなかったりする。

⑫もっとも衝撃的で革新的な解決策が、自分の問題のとらえかたがそもそも間違っていたという認識からくるということはよくある。

⑬「完成」（デザイン上の）とは、付け加えるものが何もなくなったときではなく、むしろなにも取り去るものがなくなったとき。

⑭ツールはすべて期待通りの役にたたなきゃダメだが、すごいツールはまったく予想もしなかったような役にもたってしまう。

⑮ゲートウェイソフトを書くときはいかなる場合でも、データストリームへの干渉は最低限におさえるように必死で努力すること。そして受け手がわがどうしてもと言わない限り、絶対に情報を捨てないこと！

⑯自分の言語がチューリング的完成からほど遠い場合には、構文上の甘さを許すといろいろ楽になるかもね。

⑰セキュリティシステムのセキュリティは、そこで使われている秘密の安全性にかかっている。見かけだけの秘密は要注意。

⑱おもしろい問題を解決するには、まず自分にとっておもしろい問題を見つけることから始めよう。

⑲開発コーディネーターが、最低でもインターネットくらい使えるメディアを持ってい

て、圧力なしに先導するやりかたを知っている場合には、頭数は一つよりは多いほうが絶対にいい。

[OPPS の強み]

指数関数的性質

- OSSP はインターネットと共に成長
- OSSP では「勝者がすべてをかっさらう」
- 開発者は最大の OSSP プラットフォームに貢献したがる。
- 大規模な OSSP プロジェクトのほうがもっとたくさんの「目先の問題」を解決する。

長期的な信用

- バイナリは死に絶えるがソースコードは永遠なり。
- コード分裂の不在が長期的信用につながる。

並列デバック

並列開発

完璧な API とドキュメンテーション

頻繁なリリース

[OSSP の弱み]

OSSP プロジェクトを始めるのは難しい。

OSSP を信用させ、開発に貢献するには以下のことが基準を満たさないといけない。

大きな将来のノウアスフィア

大きな悩みを解決する

まずは問題のそこそこの部分を解決せよ

飽和点に達した後の開発

飽和点に達すると、OSSP が成功する要件である「共通の目標・スキル」、および「大きなノウアスフィア」が薄れてくる。これをどう克服するか。

非専門家からのフィードバック

OSSP は、「開発者の個人的な悩みを解決」が動機付け

技術水準の高いユーザ向けシステムのなりがち

（2）エクストリーム・プログラミング(XP)

エクストリーム・プログラミング[20]とは、Kent Beck らによって提唱されているソフトウェア開発方法であり、略して XP(eXtreme Programming)と呼ばれている。

XP は、コミュニケーション、シンプルさ、フィードバック、勇気に価値を置く開発手法である。その特徴は、変化を容認（Embrace Change）するとの思想に立ち、その変化に対応できように、初期設計はシンプルに行い、リファクタリングによる再設計を重視し、また、変更コストが時間とともに増大しないように、テストを自動化するなどプログラミング、テストを重視している。また、チームに責任と権限が与えられ、メンバがプロセスを最適化していくセルフオーガナイズされたチームを目指し、人間であるプログラマに大切にする思想に立っている。

[12 のプラクティス]

XP は、これらの思想に立脚し、次の 12 のプラクティス（経験に基づいて有用性が立証された実践項目）を示している。

① 計画ゲーム(The Planning Game)

ビジネス優先度と技術的見積により次回リリースの範囲を早急に決める。現実が計画と変わったら、計画を更新する。

② 小規模リリース(Small Releases)

シンプルなシステムを早急に生産に投入する、それから新バージョンを非常に短いサイクルでリリースしていく。

③ 比喩(Metaphor)

どの様に全体のシステムが機能するかを示すシンプルな喩え話(メタファー)をメンバが共有することで全ての開発を導く（ガイドする）。

④ シンプルデザイン(Simple Design)

いつでもシステムは出来る限りシンプルに設計されるべきだ。余分な複雑さは見つけ次第取り除かれる。

⑤ テスティング(Testing)

プログラマは継続的にユニットテストを書く、それは開発を続けるために完全に動かなければならない。顧客は、機能の開発が終わったことを示す機能テストを書く。

⑥ リファクタリング(Refactoring)

２重コードを取り去り、コミュニケーションを改善し、単純化し、柔軟性を加えるために、プログラマは、システムの動作を換えることなくシステムを再構成する。

⑦ ペアプログラミング(Pair Programming)

全てのコードは２人のプログラマにより一台のマシンで書かれる。

⑧ 共同所有権(Collective Ownership)

誰でも、どのコードでも、どこででも、いつでも、プログラマはコードを修正できる。

⑨ 継続的インテグレーション(Continuous Integration)

システムを一日に何回もインテグレードしビルドし、テストを 100% パスさせる.

⑩ 週 40 時間(40-Hour Week)

週 40 時間以上仕事をしてはいけないのがルール。

⑪ オンサイト顧客(On-Site Customer)

現実のユーザをチームに加えて、フルタイムで質問に答えられるようにする。

⑫ コーディング標準(Coding Standards)

プログラマは、コーディング標準に従って全てのコードを書く。

3.4.5 コンポーネントウェア

(1) デザイン・パターン

デザイン・パターン[21]は、ソフトウェアの拡張性・再利用性を高める構成法のカタログである。

ソフトウェア開発の生産性を高める有効な手段の一つが再利用である。繰り返し使われる機能を集めた関数ライブラリやオブジェクト指向に基づき機能とデータを一つにまとめて内部を隠蔽したクラスライブラリといったコンポーネントは、再利用形態の例である。

この他の再利用の形態として、フレームワークがある。フレームワークは、コンポーネントのように完成品ではなく、半完成品である。繰り返し使われる枠組み(半完成品)だけを提供し、開発者が目的に応じて、その枠組みに拡張を施し、最終的に完成品を効率よく作成することを目指している。通常、互いに関係を持つクラスを集めたクラスライブラリとして提供され、開発者はそれらを継承するなどして独自のクラスを作成し、フレームワークが用意した枠組みに沿ったアプリケーションを作成する。

しかし、フレームワークは、大きな問題を抱えている。それは、その設計アイデアをよく理解し、クラス間の約束事などの仕掛けを理解しなければ、フレームワークを使いこなすことはできないということである。その問題を解決するのがデザイン・パターンである。設計の中で繰り返し使われる設計アイデアをデザイン・パターンとして取り出して名前を付け、ボキャブラリを多くの人々の間で共有することによって、設計アイデアや、それによる構築法の理解を助け、再利用を可能にするのがデザイン・パターンである。

[GOF のデザイン・パターン]

ソフトウェアのデザイン・パターンのアイデアは、1991 年の OOPSLA (Object-Oriented Programming System, Languages and Applications) で出され、Erich Gamma、Richard Helm、Ralph Johnson、John Vlissides の 4 名による『Design Patterns』が出版されて、

世の中に広まった。これには、表 3.3 に示す 23 個のデザイン・パターンが記述されおり、4 人組になぞらえて、GOF(Gang Of Four)の 23 パターンと呼ばれている。

表 3.3　GOF の 23 デザイン・パターン

分類	生成	構造	振る舞い
クラス	Factory Method	Adapter（クラス）	Interpreter Template Method
オブジェクト	Abstract Factory Builder Prototype Singleton	Adapter（オブジェクト） Bridge Composite Decorator Facade Flyweight Proxy	Cham of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

各デザイン・パターンについて、目的、別名、動機、適用可能性、構造、構成要素、強調関係、結果、実装、サンプルコード、使用例、関連するパターンの説明がなされている。例えば、Observer パターンの目的、適用可能性、構造は、以下のように示されている。

[目的]

あるオブジェクトが状態を変えたときに、それに依存するすべてのオブジェクトに自動的にそのことが知らされ、また、それらが更新されるように、オブジェクト間に一対多の依存関係を定義する。

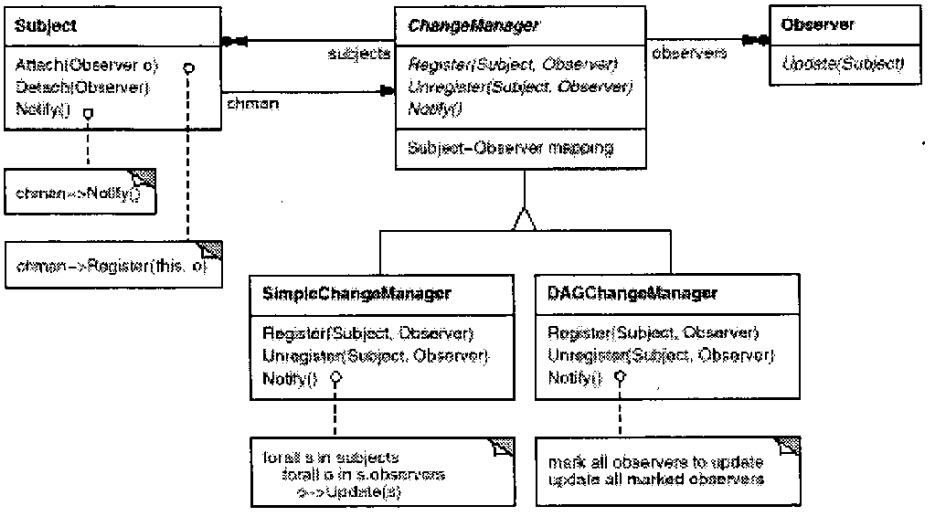
[適用可能性]

次のような状況で Observer パターンを使う。

- 抽象化により、2 つの面が、一方が他方に依存しているという形で現れる場合。これらの面をそれぞれ別々のオブジェクトにカプセル化することにより、それらを独立に変更したり、再利用したりすることが可能になる。
- 1 つのオブジェクトを変化させるときに、それに伴いその他のオブジェクトも変化させる必要があり、しかも変化させる必要があるオブジェクトを固定的に決められない場合。
- オブジェクトが、他のオブジェクトに対して、それがどのようなものなのかを仮定せ

ずに通知できるようにする場合。別の言い方をすると、これらのオブジェクトを密に結合したくない場合。

[構造]



（2）Web サービス

Web サービスとは、広義には、インターネットの標準的なプロトコルを使ってアクセスできるサービス全般を意味する。狭義には、インターネットの標準的なプロトコルとして、サービスの通信プロトコルを規定する SOAP(Simple Object Access Protocol)、サービスのインタフェース定義を規定する WSDL(Web Services Description Language)、サービスディレクトリを規定する UDDI(Universal Description, Discovery, and Integration)を使ってアクセスできるサービスを意味する。

そのコンセプトは、サービス指向アーキテクチャに基づいて、サービスとして表されるソフトウェア・コンポーネントを疎結合でつなぐものである。技術的には、オブジェクト指向、コンポーネントウェア、及び分散処理技術の延長線上にある漸進的な技術であるが、しかし、ソフトウェアアーキテクチャ的には、プラットフォーム独立の実現、クラサバモデルから P2P へ、ビジネス的には、ソフトウェアからサービスへ、ソフトウェアエコシステムの登場、無料 Web から有料 Web 時代へ、といった革新的な変革をもたらすものである。

以下に、Web サービスの基本技術である SOAP、WSDL、UDDI について説明する。

[SOAP]

SOAP[22]は、XML を用いてメッセージ交換するための通信規約であり、サービスを実行するときに用いられる。下位の通信層は通常 HTTP を用いるが、これに限定した仕様で

はなく、例えばSMTPを用いることもできる。

SOAPによって送受信されるXMLは、ルートにEnvelope要素があり、その下位要素に、HeaderとBody要素が存在する構造を持つ。Header部は、主にメッセージの制御のために扱われ、例えば、署名情報などが格納される。Body要素に、通信したい内容が格納される。このBodyの中身は、XML Schemaで定義した任意のXML文書が記述できる。

特にSOAPには、RPC(Remote Procedure Call)を実現するために用いる表現の規約があり、要求側では、Body要素の中に、メソッド名、引数に関するXML情報が記述され、一方、応答側に、結果を表すXML情報が記述される。このRPCに用いる引数や結果の値に利用できる型は、XML Schemaで定義された基本データ型(byte, short, integer, doubleなど)と、その複合型(構造体と配列)である。構造体は、XML Schemaの文書型定義を用いて表現し、配列は、SOAP encodingによる特別な表現が用意されている。

この他、SOAPには、SOAP Messages with Attachmentsという仕様があり、MIMEなどでエンコーディングした複数のパートにまたがる文書を転送することができる。

[WSDL]

WSDL[23]は、Webサービスのサービスインタフェースを XMLで定義するための言語であり、サービスがどのようなものであるかを理解するために用いられる。また、このWSDLによって、Javaのinterface定義等のプログラミング言語からWSDLを自動生成したり、SOAPのクライアントプログラム用プロキシクラスを生成したりする支援ツールが可能となり、ソフトウェア開発が容易となる。WSDLのXML文書は、definitions要素の中に次のような要素を記述する。

- types
  引数や返却値に現れる型を宣言する部分。XML Schemaによる文書型の規定を行う。
- message
  交換する一つのメッセージのフォーマットを規定する。message要素の中にpart要素があり、これが引数のひとつずつに対応する。
- portType
  まとまった操作の集合を表す。portTypeの中には、一つの操作を表すoperation要素がある。さらに、operation要素の中には、input要素、output要素が記述され、これが、それぞれmessage要素と結び付けられる。
- binding
  転送プロトコルへのバインディングを行う。例えば、SOAP bindingでは、WSDL

で定義したインタフェースと SOAP メッセージの関係を定義する。

- service
  service 要素は port 要素を持ち、サービスのエンドポイントなどを指定する。

[UDDI]

UDDI[24]は、Web サービスの公開と検索を行うレジストリの仕様を規定し、主にレジストリのデータ構造とレジストリへのアクセスインタフェースの定義を行っており、どのようなサービスがあるかを検索するために用いられる。

その内容は、businessEntity, businessService, bindingTemplate の 3 階層からなる XML 文書と、tModel を表現する XML 文書からなる。

- BusinessEntity
  サービスを提供する企業に関する情報(企業名、所在地、企業コードなど)を記述する。
- businessService
  サービスの内容や種類に関する情報を記述する。
- bindingTemplate
  サービスに接続するための情報(tModel へのリンク、エンドポイント)を記述する。
- tModel
  サービスにアクセスするためのインタフェース情報(WSDL 等)へのリンクを記述する。

そして、これらの情報を登録、検索するための API としては、次のものが用意されている。(_XXX は検索対象を示す)

- 参照 API (Inquiry API)
  - find_XXX　キーワードなどを用いた検索のための API。検索の結果は対象となった要素のキーがリスト形式で返される。
  - get_XXX: find_XXX で検索されたキーから、その実体（内容）を取り出すときに使用する API。
- 発行 API (Publication API)
  - save_XXX: オブジェクトを登録するための API。
  - delete_XXX: レジストリのエントリを削除するための API。

3.4.6 既存ソフトウェア資産を生かす技術

（1）アプリケーションインテグレーション技術

現在の社会は、膨大な既存アプリケーションを抱えている。しかも、その多くの既存アプリケーションは、脆弱であり、スパゲティ状態である。SDP が目指すものが実現したと

しても、これらの既存アプリケーションを即座に捨てるわけにはいかない。SDP が実現する新しいソフトウェアアーキテクチャを核としたインフラ上に既存アプリケーションを追加・統合し、段階的に消滅させていかねばならない。このために、アプリケーションインテグレーション技術が必要となってくる。

アプリケーションインテグレーション技術は、図 3.7 に示す 5 つのレベルがある。

<table>
<tr><td rowspan="4">統合ミドルウェア</td><td colspan="3">ビジネスプロセスマネジャ</td></tr>
<tr><td colspan="3">インテグレーションブローカ</td></tr>
<tr><td colspan="3">スーパーサービス</td></tr>
<tr><td colspan="3">ゲートウェイ</td></tr>
<tr><td>基本ミドルウェア</td><td>データ管理</td><td>通信管理</td><td>プラットフォーム管理</td></tr>
</table>

図 3.7　アプリケーションインテグレーション技術の階層

基本ミドルウェアのデータ管理は、リモートのファイル、DB を論理的にローカルにあるかのように扱う。たとえば、JDBC、ODBC、NFS、Microsoft の Windows などである。通信管理は、RPC、メッセージ指向ミドルウェアのように、リモートとの通信を同じシステム上の通信のようにする。プラットフォーム管理は、通信管理のスーパーセットで、サーバとサーバ間、サーバとクライアント間のプログラムローディング、メモリ管理等のリソース管理を同一のシステム上の管理のように行う。

統合ミドルウェアのゲートウェイは、異なった DB システム、通信システム、プロットホームを共通のインタフェースで扱えるようにする。スーパーサービスは、異なった複数の OS、サーバにまたがって、ディレクトリ、セキュリティ、トランズアクションマネジャといった共通の機能を提供する。インテグレーションブローカは、二つのアプリケーション間のメッセージの内容を変換したり、コンテンツの内容に応じて宛先を決定したりして、アプリケーションの統合をサポートする。ビジネスプロセスマネジャは、アプリケーション統合をマルチステップまで広げて自動化し、ビジネスプロセスのワークフローをサポートする。

繋ぎ方として、メッセージバス方式とハブ方式がある。統合のパターンには、システムが物理的も論理的にも独立で一方向のデータ同期の方式と、システムが物理的には独立であるが、論理的にはワークフロー的に複数のプロセスが一方向に繋がるマルチプロセス方式と、システムが物理的にも論理的にも繋がり、双方向のデータの同期を取り遅延なく整合性をとる複合アプリ方式とがある。今後は、複合アプリ方式が重要となっていく。

繋ぐ時の通信のモデルには、対話型、リクエスト・リプライ型、メッセージ・パッシン

グ型、ストア＆フォワード型、パブリッシュ・サブスクライブ型の５つがある。

ストア＆フォワード型は、キューを持つことにより、メッセージの完全性、配信保証をし、パブリッシュ・サブスクライブ型は、送信者は宛先を指定しなく、受信者が受信したい情報を論理的に指定する方式で、送受信者の相互作用を最も柔軟にできる。この２つの通信型のサポートによって、イベントベースのビジネスプロセス統合を実現できることに特徴がある。例えば、新しい注文を受けると、販売報告システム、製造システム、請求伝票システムへ即時に通知され、ビジネスプロセスの統合が遅延なくデータの同期を取って実現できる。

以上のようなアプリケーションインテグレーション技術で既存アプリケーションを統合していくには、脆弱でスパゲティ状態の既存アプリケーションを標準化、部品化、抽象化、再構築するのを自動化、支援するツールとして、コンバータ、アダプタ、ラッパー、リバースエンジニアリング等で構成されるアプロケーションインテグレーション環境が必要となってくる。

（２）CBS（COTS－Based Systems Initiative）

CBS[25]は、カネーギーメロン大の SEI(Software Engineering Institute)で研究開発中のプロジェクトで、商業的ソフトウェア・コンポーネント(COTS：Commercial Off The Shelf)、既存ソフトウェア・コンポーネントを用いて、システムを構築するための開発手法に取り組んでいる。CBS は、従来のシステムコンテンツ（要求、価格、スケジュール、OS・サポート環境等のシステムに求められる性格）の選択、アーキテクチャと設計、実装の工程を順次に行うウォータ・フォールモデルではなく、システムコンテンツ、アーキテクチと設計、市場の於ける製品の能力を同時に検討するアプローチを目指している。

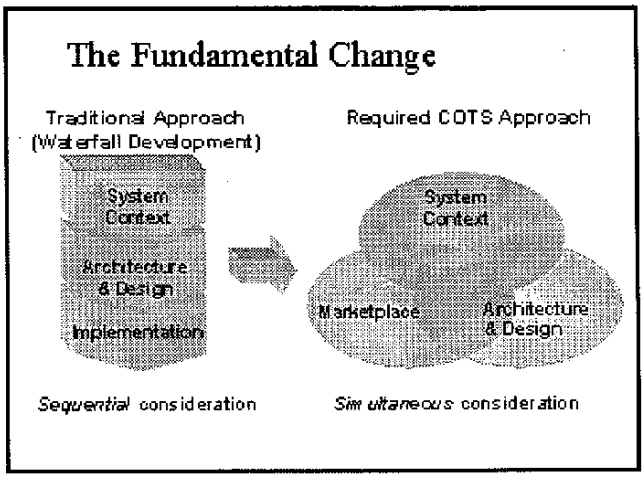


図 3.8　CBS の開発手法のアプローチ

COTS、既存コンポーネントからシステムを構築するには、以下のような質問に答えて行かねばならない。

- どんな技術、製品が適切か？
- 我々のシステムに製品のミスマッチがどの程度あるか？
- 個々のコンポーネントを制御しづらくなるにも拘らず、信頼性、安全性、性能といったシステム特性をどのように設計するか？
- COTS 製品とカスタムコードをどのようにインテグレートしていくか？
- 長いライフタイムに亘ってシステムを出荷する間に COTS の良さをどのように生かすか？

このような難しい質問に答えるために CBS は次の3つの領域をキーポイントとしている。

1）製品と技術の評価の実施

製品と技術の評価に際しては、直接的なシステム要求に応えると同時に、その技術、ソフトウェアの進化が途切れることがないかを考慮しなければならない。そのため、候補となる製品、技術は、関係する製品、技術との関連のなかで評価されなければならない。したがって、問題領域での実験に着手して、クリティカルな質問に答え、COTS の製品、技術の適用に対するガイドラインを作成していくようにしなければならない。

2）入手とマネジメントの実施

CBS では、システムの開発者という立場からシステムの消費者、インテグレータという立場へと変化し、コンポーネントのライセンシング、知的財産権の交渉、開発・メンテナンスコストの見積もり、スケジュールの予測、人材の管理、リスクの同定と軽減といったことに対する戦略が必要となる。さらに、システムに柔軟性を持たせるための知識、いろいろな創造的なソルーションを可能とするドキュメントも要求される。このために、CBS は、CBS をうまく活用した組織の実践例の情報、あるいは、CBS における入手の失敗を避ける情報を提供し、CBS に対する要求の作成、リスクの同定等、CBS プロセスを支援する。

3）設計とソフトウェア・エンジニアリングの実施

CBS システムでは、コンポーネントをインテグレートするために「つぎはぎのコード」が使われ、このため、可読性、発展性、信頼性に欠ける傾向がある。これを克服するため、CBS コンポーネントの人工的な見せかけを記述する参照モデルを開発、提供している。CBS のシステムエンジニアリングでは、図 3.9 に示すようなプロセスを経てシステム開発が行われる。最初はコンポーネントの多くのプロパティは未知であり、ブラックボックス

として扱われ、システマティクな調査によって、プロパティが明らかにされる。プロパティが明らかにされると、他のコンポーネントと矛盾したり、対立したりするものが明らかになり、その矛盾、ミスマッチがコンポーネントの改造プロセスで除かれる。ミスマッチが除かれると、コンポーネントはシステムに組み込まれ、そして、他のコンポーネントと共に再構成されて、進化していく。

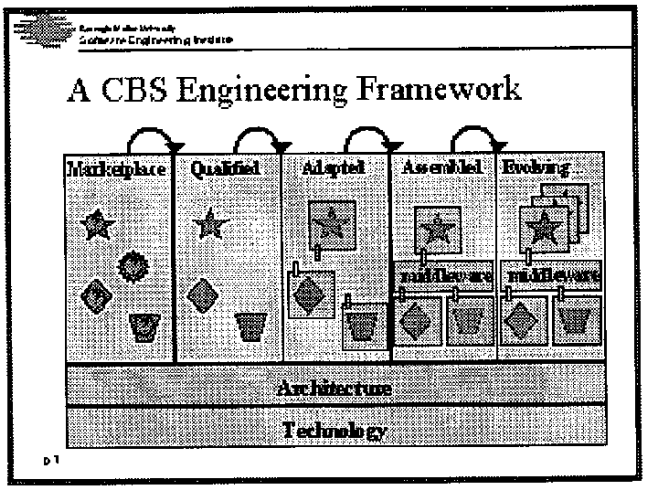


図 3.9　CBS のエンジニアリング・フレームワーク

この参照モデルは、順次的な段階で行われるものでなく、コンポーネントのプロパティによってアーキテクチャ・パターンが決められるものである。

[参考文献]

[1] "Information Technology Research: Investing in Our Future", http://www.ccic.gov/ac/report/(日本語版は http://www.icot.or.jp/FTS/REPORTS/H10-reports/AITEC9903Re1_Folder/AITEC9905R1-a8-fm.html)

[2] "Planning Workshop on New Visions for Software Design and Productivity", http://www.itrd.gov/iwg/sdp/planning/index.html

[3] "Networking and Information Technology Research and Development – Supplement to the President's Budget for FY2003", http://www.hpcc.gov/pubs/blue03/index.html（日本語版は http://www.icot.or.jp/FTS/Ronbun/BlueBook2003-J.PDF）

[4] Simulink, http://www.mathworks.com/products/simulink/

[5] Ptolemy, http://ptolemy.eecs.berkeley.edu/

[6] MDA, http://www.omg.org/mda/

[7] "Business Process Execution Language for Web Services, Version 1.0", http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

[8] Gregor Kiczales,Eric Hilsdale,Jim Hugunin,Mik Kersten, Jeffrey Palm, and William G. Griswold, "Getting Started with AspectJ", Communications of the ACM, Vol.44 No.10 pp59-65(2001)

[9] 軽い形式的枠組み、Benjamin C. Pierce, "Software Research: Where do we go from here?" , http://www.itrd.gov/iwg/sdp/planning/presentations/UPenn-Pierce.pdf

[10] TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA, "The Semantic Web" http://www.sciam.com/2001/0501issue/0501berners-lee.html

[11] David C. Fallside, "XML Schema Part 0: Primer" http://www.w3.org/TR/xmlschema-0

[12] Ora Lassila, Ralph R Swick, "Resource Description Framework(RDF) Model and Syntax Specification", http://www.w3.org/TR/REC-rdf-syntax/

[13] Dan Brickley, R.V. Guha、"Resource Description Framework(RDF) Schema Specification 1.0"、http://www.w3.org/TR/rdf-schema/

[14] "Dublin Core Metadata Element Set, Version 1.1: Reference Description" http://dublincore.org/documents/dces/

[15] "Dublin Core Qualifiers"、http://dublincore.org/documents/dcmes-qualifiers/

[16] "Annotated DAML+OIL (March 2001) Ontology Markup" http://www.daml.org/2001/03/daml+oil-walkthru.html

[17] "Reference description of the DAML+OIL (March 2001) ontology markup language" http://www.daml.org/2001/03/reference.html

[18] オープンソースソフトプロセス、http://www.opensource.org/halloween.html（日本語版はhttp://www.post1.com/home/hiyori13/freeware/halloween.html）

[19] "伽藍とバザール", http://www.tuxedo.org/~esr/writings/cathedral-bazaar/ (日本語版はhttp://www.post1.com/home/hiyori13/freeware/cathedral.html)

[20] ケント ベック, "XP エクストリーム・プログラミング入門—ソフトウェア開発の究極の手", ピアソンエデュケーション

[21] エリック ガンマ, ラルフ ジョンソン, リチャード ヘルム, ジョン ブリシディース, "オブジェクト指向における再利用のためのデザインパターン" , ソフトバンクパブリッシング

[22] "Simple Object Access Protocol (SOAP) 1.1", http://www.w3.org/TR/SOAP/

[23] "Web Services Description Language (WSDL) 1.1", http://www.w3.org/TR/wsdl

[24] "UDDI Version 3.0", http://uddi.org/pubs/uddi-v3.00-published-20020719.pdf

[25] CBS, http://www.sei.cmu.edu/cbs/cbs_description.html

第 4 章

米国の連邦政府 R&D 計画における省庁間の役割分担と連携の仕組み

第４章　米国の連邦政府 R&D 計画における省庁間の役割分担と連携の仕組み

米国連邦政府による科学技術への研究開発投資の規模は、世界最大である。こうした投資は、過去 50 年間、米国の経済、国家安全保障、そして国民の福祉に計り知れないメリットをもたらしてきた。米国政府による研究開発投資の戦略は、民間部門から十分な資金援助が得られない研究活動を対象として複数の政府機関を通じた助成を行う、という点で一貫している。たとえば、リスクの高い新興分野、長期的な取り組みを必要とする研究、前例がなくコストが高い最先端のインフラストラクチャや設備、新しい知識の発見と水平展開を目的とした学術研究プログラムといった活動が助成の対象となってきた。このような省庁連携型のプログラムの最大の特徴は、複数の連邦政府機関の間で対話と連携が展開され、すべての機関が科学技術研究ポートフォリオ全体の作成と維持に参加してきたことである。このアプローチには、連邦政府資金によるすべてのプログラムと同様に、明らかな長所と短所がある。しかし、このシステムが国家目標の達成に貢献してきたことは疑いのない事実である。特に、近年の省庁横断プログラム・イニシアチブは、連邦政府の研究開発ポートフォリオに多大な影響を及ぼしてきた。その結果として、優先順位の変化、基幹インフラストラクチャの構築、新興分野の研究の加速、無駄や重複の減少といったメリットも実現されている。残念なことに、利害が競合する省庁間の連携では、人材と施設の両面で資源を調整することが、難題として各機関の前に立ち塞がるのが常である。しかし、連携が効果的に作用すれば、コストを上回るメリットが期待できることは明らかである。

4.1　目的と範囲

本報告書は、前回の AITEC 調査報告書の続編として、米国において複数の政府機関が合同で実施している近年のプログラム・イニシアチブを支える連携プロセスについて、包括的に調査した結果をまとめたものである。本調査の目的は、そのような連携プロセスが、なぜ有益であり、どのような形で効果を発揮するのかを、省庁間プログラムの運営／管理面に注目しながら検証することである。次に、今回の調査で取り上げる主な問題の一部を示す。

- 省庁間の機関横断的なイニシアチブはどのように構想され、形成されるか。
- 参加機関は共同プログラムの定義と開発にどのような役割を果たすか。
- 複数の省庁や機関にまたがるプログラムとプロジェクトはどのように運営されるか。
- 参加機関の間で資源と責任がどのように委譲されるか。
- 複数の機関による財政支援の中に身を置く研究者や研究施設の側では、省庁間プログラムをどのように見ているか。

- プロジェクトとイニシアチブ全般の成否をどのように評価すべきか。
- 予算管理機関（例：連邦議会や行政府の担当部局）はどのような役割を果たすか。
- 省庁間の調整を担当する部局や、プログラム・マネージャの草の根的な活動の役割は何か。

本調査では、こうした問題について、マクロ・レベルとミクロ・レベルの2つの視点から考察する。

まずマクロ・レベルでは、人材、資金、プロジェクトがどのように各種のプログラムや機関の境界を越えて流れているか、つまり、省庁間の壁がどのような形で取り除かれているのか、という点に注目する。

ミクロ・レベルでは、省庁合同体制の下で、特定のプログラム要素やイニシアチブが、どのような場面でどのように機能しているか、そしてどのような形で調整と連携のプロセスに（それぞれが独自のあり方で）貢献しているかに注目する。この2つの相補的な観点を組み合わせることで、全体として、研究と技術革新を支える1つの特異なプロセスである省庁間連携という仕組みの内部的な働きを、より明瞭な形で表現することが可能になる。

4.2 調査の枠組み

こうした論点をマクロとミクロの両レベルで検討するために、今回の調査では、3つの視点から成るアプローチを使用する。

第一に、米国の研究開発の機構全般を概観するために、調査の背景となる枠組みを設定する。その際は、特に、連邦政府のさまざまな助成組織と、省庁間の調整活動のために設けられた管理構造を中心として考察する。この視点を通じて、どのような要因が（内部と外部の両方の要因を含めて）省庁間の連携を推進し、どのような要因が阻害するのかを、より明確に理解することが可能になる。

第二に、マクロ・レベルで、連邦政府がコア・テクノロジーの開発で果たしている独自の役割、すなわち、大学における基礎研究を立ち上げ、産業界の参加を促し、そうした研究活動が実を結ぶように（つまり、市場化されるように）導く、という役割に注目して分析を行う。複数の機関が（同時期に、あるいは異なる時期に一定の期間にわたって）共同研究に参加することによって形成されている技術革新システムを、本調査ではイノベーション・パイプラインと呼んでいる。

第三に、省庁間プログラムの細部をミクロ・レベルで検討するために、情報技術（IT）分野を詳しく取り上げる。ITは、2004会計年度の予算請求も含めて、過去10年近くの間、常に最優先の省庁横断研究開発イニシアチブの1つとして扱われてきた。情報技術研究開発は、おそらく、過去10年間で、最も順調に発展し、最も適切に運営されてきた省庁間イニシアチブの1つであり、調整と連携の複雑さに伴って発生するすべての課題を具体的に示してくれる事例でもある。

4.3　主な調査結果

以上の枠組みに従って、データを収集し、資料を調査し、実際にプログラムに携わった主要な関係者への面談調査を行った。以下に、今回の調査から得られた主な結果と結論を示す。

(1) ミッションは異なるが、目標と特性は共通している

これは、科学技術領域のあらゆるタイプの省庁間活動を後押しする最大の原動力である。各機関は、それぞれが異なる固有のミッションを遂行するために必要とする研究開発ポートフォリオを作成する任務を担っているが、目標とその達成手段は類似している。具体的には、次のような目標が省庁間で共通している。

- 長期的な基礎研究を支援する。
- 大規模なインフラストラクチャの構築に向けた活動を支援する。
- 知識移転を目的とした産学連携を確立する必要がある。
- 外部の資源を有効利用して、機関固有のプログラムを改善し、プログラム・ポートフォリオ内の助成格差を解消し、新興分野の研究に着手する。
- 補完的な知識と管理形態を取り入れて、組織力を強化する。
- 研究開発の成果を研究開発機関以外にも水平展開して、一般社会での実用化を目指す必要がある。

上記のどの共通目標も、単独では他の機関に協力を依頼するだけの説得力ある理由にはならないが、全体として見ると、適切な機会が生じた場合に省庁間コラボレーションを採用する強力な論拠になる。

(2) リーダーシップは議会、行政府、そして各省庁レベルから発動されている

米国で省庁横断的な研究開発プログラム・イニシアチブが次々と誕生するにあたっては、政府の強力なリーダーシップが大きな役割を果たした。このリーダーシップは、さまざまな方向から発動された。

まず第一に、連邦議会が、省庁間プログラムの発足に結び付く法案を採択したことである。この法案は、複数の機関の参加を必要とするプログラムの発足とそれに対応する管理構造の実現を求めるものであった。現在のネットワーキング／情報技術研究開発プログラム（NITRD）の前身

である高性能コンピューティング／通信イニシアチブ（HPCC）は、議会の立法措置の結果として誕生したものである。

第二に、行政府も行動を起こし、国家科学技術会議（NSTC）を設立し、科学技術政策局（OSTP）と行政管理予算局（OMB）が担っている省庁間調整の役割を強化するという措置を講じた。ここでのキーワードは「政策」、「予算」、「管理」である。この 3 つの要素が適切に結合することで、機関連携プログラムの推進に必要な動因が提供された。

第三のリーダーシップは、機関の管理者層そのものから発動された。上層部からの奨励策や促進策を受けて、さまざまな行政機関の上級管理職や中間管理職は、省庁間の連携を自分の仕事の中心課題と考えるようになったのである。

(3) 複数の機関から、多様性、資源の増強、相互補完性というメリットが得られる

複数の機関が取り組みに参加する最も明白な利点は、選択肢の多様性が得られること、そして、通常は単独の機関では実現できないような新しいプログラムを立ち上げる機会が得られることである。

今回調査した省庁間プログラムの場合、この利点の一部は、実施されたプログラムの幅広さに現れている。しかし、省庁間のコラボレーションには、単なる選択肢の多さや資源の共有を超えた要素も含まれている。常に観察される事実は、参加機関が、個々の機関の能力を超えたレベルに設定された共通の研究課題や実施計画を達成しようと努力することである。たとえば、電子図書館研究イニシアチブ（DLI）では、国立医学図書館（NLM）や連邦議会図書館（LOC）といった機関が参加することによって、全米科学財団（NSF）、米国航空宇宙局（NASA）、国防高等研究計画局（DARPA）などの科学技術専門機関が提供できない能力が補完されている。同様に、NITRD イニシアチブの場合も、政府の主要な研究開発機関が多数参加して大規模な相互補完的体制を形成しているために、資源の共有、研究対象範囲の広さ、管理手法の相違といった問題に対処することが、どの参加機関にとっても非常に大きな課題となった。しかし、連携の方針を慎重に考案し、それを適切に実施したことによって、こうした課題は、連邦政府の研究開発ポートフォリオの大きな財産へと変貌したのである。

(4) 異なる活動モデルがコラボレーションの柔軟性を実現する

今回の調査は、省庁間の取り組みが多様な形態を取ることを示している。まず、連携プロセス自体が 2 つ以上の機関の非公式の話し合いの結果として生じ、それが最終的に共同プログラムへと発展するケースがある。そうした事例は、(通常は) 二者間のコラボレーションから成り立つ小規模なイニシアチブの一部で実際に見られるものであり、人間言語資源イニシアチブ（HLR）に

おけるNSFとDARPAや、計算神経科学プログラム（CNS）におけるNSFと国立衛生研究所（NIH）などがそれに該当する。逆に、さまざまな機関の上層部局の指導の下で公式の折衝が実施された結果として、省庁間の連携が開始されることもある。共同プログラム（例：電子図書館研究イニシアチブ）への助成に合意する正式の文書を取り交わすことによって、各参加機関（DLIの場合は6機関）は、予算、人事、およびその他の資源に関して、一定の義務を負うことになる。この公式の手続きに従うと、ときには労力と時間を取られることもあるが、高い継続性が得られるとともに、プログラムの実施中に不愉快な紛争が生じるのを防止することができる。また、トップダウン方式とボトムアップ方式を対比したモデルも、実情を適切に説明するものであった。むしろ、状況によっては、2つの方式の組み合わせが、省庁間プログラムに最も有利に働くこともある。その最適な例はNITRDプログラムであり、NITRDの方式が成功したことで、その後、他の多くの省庁間プログラムがNITRD方式をモデルとして踏襲しようと試みるようになっている。NITRDの場合、多数の機関にまたがって長い間地道に実施されてきた草の根レベルの活動が、後年の大規模な共同イニシアチブの礎石を築くことになった。こうした長年にわたるボトムアップ活動があったからこそ、その後、この多機関合同のプログラムが確固としたシステムへと発展できたのである。連邦政府の研究開発階層の最高レベルからトップダウンで方向付けと管理が実施されるNITRDの現在の姿は、そうした経緯の結果である。

（5）省庁間連携はコア・テクノロジーの開発とイノベーション・パイプラインの維持に重要な役割を果たす

今回の調査で、米国のコア・テクノロジー開発には省庁間の連携が決定的に重要な役割を果たしていることが明らかになった。これまで、多数の政府機関が、相当な年月にわたって、大学における基礎研究に助成し、大学間の連携強化を支援し、産学の連携を通じた共同研究を実施するように産業界を指導してきた。その結果、運に恵まれれば、研究成果が市場化という形で実を結ぶケースも生じている。こうした場合に政府機関が果たしている役割は、イノベーション・パイプラインを創造し、維持し、拡大するという役割である。

このような連携の取り組みの発端は、意図的な場合、偶発的な場合、ときには御都合主義的な場合さえもある。しかし、どのような形であろうと、その結果として生じる影響は望ましいものであり、しかも長期的に持続するものである。過去30年間にわたるIT分野のコア・テクノロジー開発の歴史を振り返ってみると、この見解が裏付けられる。タイムシェアリングに始まり、コンピュータ・グラフィックス、超大規模集積回路（VLSI）、データベースを経て、インターネット、音声技術に至るまで、技術革新がその着想から数十億ドル規模のビジネスまで発展していく経緯は、複数の政府機関からの助成を（同時に、あるいは異なる時期に別々に）受けながら実施される研究開発と同じ軌跡をたどっている。各政府機関の役割は異なることもあるが、常に相互

補完的な機能を果たしている。たとえば、人工知能（AI）の研究を支援した機関は、当初、DOD（主としてDARPA）だけであったが、その後、NSFが参加したことは、AIを正式な学問分野として存続させる上で、DODとは異なる面で非常に重要な役割を果たすことになったのである。

今回の調査で明らかになったのは、いずれの場合でも、コア・テクノロジーの開発を後押しするこうした機関合同の取り組みが、大規模な先導的イニシアチブの基盤を築き、イノベーション・プロセスを促進してきたという事実である。

(6) 省庁間研究開発ポートフォリオ全体は大小さまざまなプログラム・イニシアチブの混合から構成される

過去10年間にわたる連携プロセスを調べてみると、さまざまな種類のプログラム・イニシアチブ（たとえば、通常は多数の機関が参加する大規模なイニシアチブから、2つの機関あるいは比較的少数の機関だけが関わる小規模なイニシアチブまで）を組み合わせることのメリットが明らかになった。こうした各種のプログラムは、その貢献度や影響力の点でも異なる。大規模なプログラムは、先導的な役割を果たし、資源の有効利用、国家規模のインフラストラクチャの構築、無駄や重複の低減といった効果をもたらすほか、最も重要な点として、研究開発の優先順位と予算配分の変化を引き起こす。大規模プログラムの典型は、NITRDイニシアチブである（詳細は後の章で検討する）。一方、小規模プログラムの活動は、より大規模で長期的な連携を生み出す源泉となるほか、新たに出現する研究ニーズや各機関のミッション要件に対応する役割を果たす。この両極の中間には、電子図書館研究イニシアチブ（DLI）などの中規模プログラムが位置する（DLIについても後の章で詳しく検討する）。中間的なプログラムは、両極のプログラムの長所を兼備する傾向がある。すなわち、小規模イニシアチブに特有の柔軟性と形式主義からの自由、そして通常は管理の経費が少ないという利点がある一方で、適切に運営すれば、各機関や連邦政府の研究開発ポートフォリオに対して長期的かつ多大な影響力を行使できるということであり、DLIがその実例である。

(7) 省庁間連携に寄与するその他の原動力

省庁間連携を推進する原動力の1つは、連邦政府の外部にある要因、すなわち、研究コミュニティ自体である。この力は、主として、連邦政府の研究開発プログラムに施設と人材の両面で資源を提供する研究系大学に由来するものである。実際には、ここでは2つの力が働いている。

まず、大学の研究活動、特に、優秀な中核的研究センターで実施される研究活動は、複数の政府機関から助成を受けているのが普通である。なぜなら、助成機関は、最大限の投資効果が得ら

れるように、最も優れた研究者と研究施設を投資先として選択するからである。このこと自体が、複数の機関の連携を促す力として機能する。

もう１つの要因として、研究大学の側が、必要に迫られて、共同出資や共同管理の可能性を探るために複数の機関から支援を求めざるを得ないという状況がある。このような力は、ほとんどの政府機関内部で個別的プログラムの推進力として働いてきたが、近年では省庁間イニシアチブを推進する力としても機能するようになっている。現在、米国には研究系の大学が約200校あり、そのすべてが省庁間の連携を強化する方向で（大なり小なり）一定の役割を果たしてきている。

(8) 省庁間連携の強化に向けた政府の新しいアプローチ：「アメとムチ」

ミッションや利害が異なり、互いに競合している機関同士が連携することは、本来、難しい試みであり、しかも研究開発に充てられる資源が限られているとなれば、連携はもはや至難の技である。長年の間、省庁間連携は、米国政府にとって「絵に描いた餅」のような目標であり、一部の省庁間イニシアチブで一定の成果が上がるようになったのも、ごく最近のことである。それでも、こうした初期の成功によって、連邦政府は、同様の成功を他のイニシアチブでも実現するための処方箋を見出したように思われる。それは「アメとムチ」のアプローチである。

まず「アメ」の部分は、すべての省庁の予算編成プロセスに OMB と OSTP を深く関与させることによって達成される。OMB と OSTP は、大統領の代理として、各機関の長に対する行政命令を発行し、国家の最優先課題である機関連携プログラムにどのように取り組むべきかを示す指針を提供している。この取り組みに参加しようとしない機関は、あらゆるものを失う立場に立たされる。一方、連邦議会は、研究開発プログラムの策定や管理に省庁間の連携が欠如していることを長年にわたって批判してきた。こうした批判に実効性を伴わせるために、議会は、過去 10 年にわたって一連の立法措置を講じてきた。その 1 つが、1993 年に制定された政府業績成果法 (GPRA) である。現在、GPRA は、長年にわたる政府の逡巡に服してきた後に、ようやく全面的に施行されようとしている。GPRA は、無駄と重複を低減するために、省庁間の連携を要求している。また、GPRA は、各政府機関が、省庁間プログラムを優先的な活動として、また業績の評価基準として自発的に採用することを奨励する方策も提供している。今回の調査で明らかになった限りでは、これが省庁間の連携を「強制」する新しいアプローチの「ムチ」に相当する部分である。

4.4　結論

省庁間プログラム（機関横断プログラム）の創設と運営／管理は複雑な仕事であり、その実施に成功することは、現実には達成できそうもない幻の目標であった。それでも、国家がより賢明

かつ効果的な科学技術投資を実施するための有効な手段になるという点で、省庁間プログラムは、やはり追求するに値する目標である。米国では、省庁間プログラムが必ず成功しているとは言えないが、近年では成功事例も生まれ始めている。今回の調査は、米国の幅広い省庁間活動の経験から教訓を汲み取ろうとする包括的な試みである。こうした活動は、コア・テクノロジーの開発支援から、草の根レベルの活動を通じた小規模プログラムの育成、そして大規模なイニシアチブに至るまで多岐にわたるが、いずれの活動も省庁間連携プロセスに対して独自の形で貢献してきた。省庁間活動業績マトリックスを使用して、このような諸活動の検証と分析を行った結果、効果的な連携を可能にする重要な因子と要素の一部が明らかになった。ただし、連携を確かな成功へと導く魔法の公式や処方箋が見つかったわけではなく、単に、将来同じ道を歩む者が少しばかり確かな足取りで進めるようになっただけであるが。

第5章

わが国IT開発拠点の中国移転に関する調査

第５章　わが国IT開発拠点の中国移転に関する調査

5.1　調査目的

本調査はソフトウェア分野における日本企業による中国での事業展開に焦点を当て、その目的・現状・課題・今後の展開などの分析を通じて、日中間のソフトウェア分野における関係の現状と将来、日本のとるべき方向性・対策を検討するものである。

経済産業研究所 上席研究員 関志雄は、日本と中国の全般的な産業構造を対米輸出データの分析によって検討し、日本は「失われた 10 年」を経ても、アジア諸国の中で依然として最も高付加価値型の輸出構造を有しており、一方中国は未だ雁の行列の後ろを飛んでいると指摘している。

http://www.rieti.go.jp/users/china-tr/jp/020502newkeizai.htm

この主張によれば、日本と中国の産業構造に関してはいまだに大きな格差が存在することになるが、ソフトウェア分野についても同様な見解が成立するであろうか。ソフトウェア分野においては中国はもとより日本においても産業の国際的競争力が弱く、ソフトウェア製品の対米輸出は限りなくゼロに近いと考えられる。一方、両国の国内市場と言う観点で考えると、市場自体の大きさは中国の方が（現時点では潜在的とは言え）日本を凌駕する。また、ソフトウェア産業は、他の産業に見られるような特殊な生産設備は不要で人的資源が基本であるという特徴がある。ソフトウェア人材は主にソフトウェア開発関連人材（アーキテクト、プロジェクトマネジメント、スペシャリストなど）とセールス・マーケティング関連人材（マーケティングセールス、コンサルタントなど）に分けられるが、前者は一般的に先進国と途上国間の格差が小さく、教育で比較的容易に埋めることができる。セールス・マーケティング関連人材はその育成にやや時間がかかるが、この点においても日本と中国の差異は急速に縮まりつつある。このような状況から、ソフトウェア分野において日本が先端的、中国が後発的という図式が当てはまるかどうかについては疑問の余地がある。

日本の情報機器ベンダーや大手のソフトウェアユーザ企業においては、コスト低減を主目的として既に中国への業務委託が相当程度に進展している。この現象の当面の状況は、高付加価値型を有する日本のソフトウェア産業が後発の中国へ技術主導の展開を図っていると言うよりは、むしろ画一的なサービスメニューしか無い市場でコスト削減だ

けが差別性の源泉になっており、日本の各企業は中国の低賃金労働力に依存することで短期的な打開策を講じていると考えることができる。結果的に中国が日本のソフトウェア産業界の市場で一定の競争力を有するに至っており、既に中小のソフトウェアハウスの市場が狭められている。

この傾向がさらに広がると同時に、その延長線上に研究開発分野などの付加価値の高い領域においても中国との関係が深まる傾向が見え、ソフトウェア分野における我が国産業の空洞化に対する懸念を持つ人々が増え、さらには遠からず中国が日本を逆転する可能性を指摘する声もある。

こうした背景にもとづいて、本調査では日本と中国のソフトウェア産業の関係について６つの論点を設定し、文献調査とインタビュー調査によって日本のソフトウェア産業の現状と必要な施策などを浮き彫りにすることを目的としている。

5.2　日本と中国のソフトウェア産業の関係についての主要な論点

日本と中国のソフトウェア産業の関係に関する議論を次のような論点に整理した。

論点１：日本と中国の関係のシナリオ

中国の経済的・技術的発展が新たな市場を生み日本の経済・ソフトウェア産業界にプラスの効果をもたらすとの見方がある一方で、中国の台頭によって日本のソフトウェア産業界は市場を失い空洞化が一層進展するとの観測もある。

論点２：中国市場の可能性

日本のソフトウェア産業は中国の経済発展による市場拡大で恩恵を受けることができるとの期待があるが、別の見方として、中国の市場や商慣行はまだ十分に国際的なレベルに到達しておらず、また物価の面からも市場としての中国には当分期待が持てないとの悲観的見解もある。

論点３：低コスト労働力の永続性

中国の競争力の源泉の一つである低賃金労働力は今後も継続するとの考え方が一般的であるが、既に北京や上海などではこの傾向は正しくないとの主張がある。低コスト労働力を求めて中国内陸部にさらに事業展開する傾向が進み、さらにはベトナムなどへの移転の可能性を指摘する声もある。

論点４：ソフトウェア分野の研究開発力

現時点では日本から中国に委託している領域は比較的仕様が定まった部分の開発が主体であり日中間の技術レベルには一定の差があるとの認識がある。しかし、既に研究開発を中国に委託している企業もあり、膨大な人口と優秀な頭脳で中国は遠からず日本の技術力を凌駕するとの予測もある。

論点５：中国企業による日本進出

日本と中国の関係の多くは現時点では日本企業による中国への事業展開あるいは日本企業による中国人技術者の雇用である。しかし、中国資本による日本のベンチャーの M&A や中国企業の日本進出傾向も加速している。

論点6：日本のソフトウェア産業の高付加価値型への展開

日本のソフトウェア産業では高付加価値型サービスで各企業がその力を競うような方向は残念ながら見られず、受託開発を中心として人件費コスト削減が唯一の競争力の源泉となっている。本質的にはこうした産業形態を改めてより高度で独自のサービスを各企業が展開する方向に向かわなければならない。それが実現可能か否かが問題である。

これらの点を検証するために、日本および中国のソフトウェア産業関係者に以下の項目に関するインタビュー調査を実施した。

a. 日本と中国との提携関係の概要

日本と中国の提携関係の形態、双方の事業目的、対象業務、対象技術分野、マーケットなど。

b. 日本と中国のソフトウェア産業構造

日中のソフトウェア産業の特徴、双方の技術者の特徴など。

c. 日本と中国の提携の今後の展開

今後の事業規模の見通し、対象業務の変化の見込み、事業拡大のための条件など。

d. 日本のソフトウェア産業の発展に向けて

日中のソフトウェア産業の発展のための施策など。

5.3　インタビューにもとづく考察

5.3.1　中国のソフトウェア産業の特徴と課題

前章の項目に基づくインタビュー調査の結果、中国のソフトウェア産業の特徴と課題を次のように整理した。

(1) 急激な成長を支える政策

国の政策実施は即断即決に加えて即実行。独裁体制の国家であることが奏功し非常に効率が良い。その反面、政府や地方政府の腐敗があり、人々の意欲を削ぐ面もあるが、中国ではそうした状況は珍しくない。人々はそれに対応する術も心得ている。

中国の企業には品質に対する意識が希薄であるとの指摘があるが、例えば北京市では企業のISO認証取得やCMMへの取組みに対して奨励金を出すなどの政策を実行している。

こうした国や地方政府の育成政策、発展しつつある国内市場、グローバル化がもたらした外国企業向け市場、積極性に富む人材、などで急激な成長を遂げた。

(2) 人材育成

急激なキャッチアップのために人材育成が重要な課題となっている。

- 若年層からの教育

  小平の政策の結果、若年層からの教育の必要性が認識されている。

- 大学での教育

  大学ではきわめて実践的な知識・技術を主体に教育している。米国企業の資格制度のための授業や大学独自の資格制度などもある。こうした実用的な技術の教育の成果で、たとえばWEBアプリケーション業務で新しいツールを導入して開発する、というような作業では日本より技術力がある。

- IT関連企業内の教育

  急激な社員の増大への対応のため、採用後三ヶ月間はトレーニングのみで人材の即成を実施している企業もある。労働契約は2～3年の時限があるので流動性は避けられず、教育等で投資したが退職されたという例は良くある。これを織り込んだマネジメントが重要。

中国は人材不足の状況を不利な点としてのみ考えず、逆にビジネスの芽と考える力強さを持っている。人材育成法についてのコンサルティングや教材、教育カリキュラムの

提供などの教育市場が生まれ、これらに対応する企業を通して IT 産業発展の要因ともなっている。IT 関連企業が学校を作るようなことまで実施できる仕組みになっており、日本を市場と考え、日本人技術者を対象とした教育ビジネスを検討している企業もある。

(3) 未熟な産業基盤

急激な市場経済化が進められているが、国営企業、民間企業ともに十分にビジネスインフラが整っておらず、ルールを守らない企業も多い。さらに、中国におけるソフト産業がまだ若いので、ソフト開発という個人の能力に依存する産業における会社の占める位置（特に雇用関係）は未成熟である。たとえば、中国人技術者は「意欲」と「向上心」に優れているが、反面、仕様書以外のことに手をつけてはいけないような形態の仕事に満足できず、自分の能力が発揮できる職場に転職していく行動として現れ、しばしば「無責任」と言われてしまう。こういう人材をプラスの方向に向かわせるマネジメントが要求されており、会社の求心力強化のために作業環境改善や従業員に対するストックオプションなどの施策を講ずる企業も出始めている。

ソフトウェア分野では 80 年代から産業化が始まり、成功した企業は最初は会計システムの開発を手がけたところが多い。自前の ERP を開発してもいるが、多くは会計システムをベースにしてインベントリなど付け加えただけで必ずしもユーザのニーズに合ったものとは言えない。ERP などのソフトウェアは文化的な背景もあり外国製のものを使う訳に行かず、問題となっている。

また IT 関連企業の形態として日本などの大手企業と中国側とで合弁で設立した会社が多いが、合弁双方の経営方針の相違と管理問題によって解散に追い込まれた企業も少なくない。

(4) 独自技術・製品開発への課題

一般的に言って中国には独自技術が無い。企業は自分の製品、独自技術を育成しなければならず、政府も独自技術開発の必要性を強く意識している。紅旗 Linux の開発もこの一環（Microsoft 社の Windows を使わないことが決まっている軍事分野では紅旗 Linux が使用されている）。データベースなど自前の技術がまだ十分に育っていない分野もある。

(5) 市場としての中国

中国のソフトウェア企業は自らの国を最大のターゲットを考えている。広大で未発達な内陸部、不安定なビジネスインフラなどを抱えて、いつ市場が具体的なものになるか

は不明であるものの、e-Home などの先進的な分野においても自国をターゲット市場と考えている中国企業があり、市場化への動きは沿海部については相当早いことが予想される。

こうした動きにつれてソフトウェア産業の発展の可能性は特に高い。企業の数が急激に増えつつあり、また市場経済に入って管理レベルが高くなりつつあることも企業向けソフトウェアの大きな需要につながっている。今後５年くらいはニーズが増え、マーケットが大きくなるだろう。さらに良い材料は情報技術分野のテクノロジートランスファーが車の生産などに比較して簡単であり、しかも製造業のように大規模コストのかかる産業集積を必ずしも必要としない点である。

5.3.2　主要な論点に関するインタビュー結果に基づく見解

インタビュー調査に基づいて、先の６つの論点について以下のように分析・整理した。

論点１：日本と中国の関係のシナリオ

中国の経済的・技術的発展が新たな市場を生み日本の経済・ソフトウェア産業界にプラスの効果をもたらすとの見方がある一方で、中国の台頭によって日本のソフトウェア産業界は市場を失い空洞化が一層進展するとの観測もある。

⇒

日本のソフトウェア産業が中国への進出を深めることによって空洞化すると言う仮説を支持する声は少ない。理由は、中国でのソフトウェア開発には以下のような問題点があるためどの企業にも簡単にできると言う訳ではなく、その結果、中国での開発が日本のソフトウェア産業界で支配的な傾向になる可能性が少ないからと言う論拠である。

・文化的な相違。仕様書にすべての情報を明示することは現実的には不可能であり、記述されていない部分の解釈には自由度があるため、文化を異にする日中間で食い違い、これが原因となって発生するトラブルが多い。
・コミュニケーションの困難さ。言語の問題は当然としてビジネス慣行の相違もある。日本で学生時代を過ごし、あるいは業務経験がある中国人技術者がプロジェクトチームに含まれていることが望ましいが、さもなければ一定の期間をかけて教育する必要がある。

・品質に対する認識。中国人技術者の品質意識はまだ低い。日本と同様のテスト環境が用意できにくいことも問題。
・機密保持の問題。日本の発注者の中には、中国への再委託によって知的財産権が流出する恐れを抱く人々もおり、中国への委託を躊躇することも少なくない。

上の指摘は基本的には日本がソフトウェア設計開発の上流部分を担当し、それを受けて中国側が開発を行うことを前提にしている。言い換えると日本のソフトウェア産業が国内市場を対象にしているケースを想定しており、その限りでは中国は大きな脅威にはならないことを意味している。したがって、中国あるいは世界を対象にしたマーケットで日本と中国のソフトウェア産業が競合し、その際に両国の力関係がどのような構造になると想定されるかと言う問題設定は現時点では両国の関係者の意識には顕在化していない。こうした環境変化があり得るかどうかは推測の域を出ない。しかし後述するように日本のソフトウェア産業界の現状から推し量ると、国際展開の結果として中国と覇権を争うような事態の可能性が非常に高いとは考えられず、逆にそのことがより大きな問題と指摘することもできる。

一方、別の意味で日本のソフトウェア産業の空洞化についての懸念を表明する見解がある。それは、単純なソフトウェア開発に限ればコスト低減の要請が決定的な要素であるため、そうした業務が中国などに流れる傾向はさらに続くと考えられ、それによって日本のソフトウェア産業で初歩的なプログラミングを経験する機会が少なくなり、技術者の育成の観点で問題を生じる可能性があると言う指摘である。すなわち「プログラミングを知らないソフトウェア技術者」が生まれ、「生産能力を欠いた製造業」と同様の問題を抱え込む可能性が高まる。

また、ソフトウェア産業に対する日本国内の発注者企業の海外移転が進み、これら海外移転した日本企業が現地のソフトウェア企業に発注する方策を取ることによって、国内のソフトウェア産業のプロジェクト機会が減少することによる影響も憂慮すべき問題と考えられる。

日本のソフトウェア産業の中国進出による国内ソフトウェア産業界の空洞化への懸念と言う問題は、当初設定された問題の枠組み自身の範囲では杞憂に終わるとの認識が支配的であるが、この問題設定の周辺には別の重要な課題があることが明らかになったと言える。

論点２：中国市場の可能性

日本のソフトウェア産業は中国の経済発展による市場拡大で恩恵を受けることができるとの期待があるが、別の見方として、中国の市場や商慣行はまだ十分に国際的なレベルに到達しておらず、また物価の面からも市場としての中国には当分期待が持てないとの悲観的見解もある。

⇒

全体的に中国を市場と認識している日本企業は少ない。潜在的な市場と考えている企業にしても、実際に市場として期待できる時期はまだ読めていない。中国でのソフトウェア事業の市場をパッケージ販売と受託開発に分けて考えると、パッケージ販売は違法コピーされないで商品を流通させるためには数百円程度にまで価格を抑えなければならずまったく収益が出ない。ハードウェアにバンドルして販売する戦略も考えられるがこの場合にはさらに１桁価格が抑制されるため、知名度を上げる程度の意味しかない。

一方、受託開発で日本企業に可能性があるとすれば国営企業であるとの認識を示す企業もあるが、まだビジネスに至っておらず、当面はそれに向けた意欲もほとんど無いかに見える。原因は金額的な格差の問題が大きいが、さらにこれまで日本の民間市場で培ったソフトウェア開発のノウハウが活用できないとの認識も手伝っている。確かに日本の民間企業と中国政府とでは業務プロセスがまったく異なると考えられる。この他、リスク回避の観点から中国国営企業の業務受託を躊躇する見解もある。

一方、中国の企業は自国を最大のマーケットと認識しており、日本市場を向いている企業は部分的にはあるものの、それが全般的傾向とは言えない。日本から中国へのソフトウェア発注が増えることによって日本側が抱く空洞化への危惧は、多くの中国側関係者にとってはいささか的外れな八つ当たり的な訴えに響くようである。中国は膨大な人口を背景に国内需要が爆発的に発展する前段階にあり、そのことへの対処がビジネス的にも社会的にも極めて重要であって、日本向け市場で部分的に収益を上げている企業があったとしてもそれはごく一部の傾向に過ぎないとの認識がある。中国は国を挙げて中国問題に腐心しており、日本が自らの都合で抱えるに至った日本のローカルな問題に重大な関心を寄せる人々は非常に少ない。

中国では多くの企業が勃興しつつあり、しかも市場経済に入って管理レベルが高くな

りつつあることから企業向けソフトの需要は極めて多い。さらに先端的な分野の研究開発を行っているソフトウェア企業でも自国（現時点ではその一部だが、それでも規模は大きい）を第一の市場と考えているところがあるなど、中国が市場として立ち上がる時期は予想より早いと考えられる。この場合も中国全体を一国と考えるのではなく、地方ごとの特色にあわせた柔軟な対応が必要である。

中国が市場として発展した場合、きめ細かいモノ作りと言う日本人の価値観は中国市場に対しても競争力の源泉になるとの指摘がある。中国側の有力企業でも、日本との連携を求めているにも係わらず、そのリスクへの危惧あるいは日本側へのコネクション不足などから二の足を踏んでいるケースもある。中国側は自国に無い技術を求めており、それに応えることのできる日本のソフトウェア企業が中国への展開を実現することが切に期待される。

論点３：低コスト労働力の永続性

中国の競争力の源泉の一つである低賃金労働力は今後も継続するとの考え方が一般的であるが、既に北京や上海などではこの傾向は正しくないとの主張がある。低コスト労働力を求めて中国内陸部にさらに事業展開する傾向が進み、さらにはベトナムなどへの移転の可能性を指摘する声もある。

⇒

日本企業の中国への発注の最大要因は低人件費である。逆に言えば、中国で低コストが実現できなければ、日本のプロジェクトが中国に発注されることにはならない。現にここ１年くらいのIT不況で中国やインド人技術者の代わりに社内の日本人を使って仕事をするようになりつつある、とのコメントを寄せる日本企業もある。人件費は中国国内でも確実に上昇しているため、日本企業からの発注が企業のみならず地域の経済にも少なからぬ影響を与える大連市などはこの問題に重大な関心を寄せている。

今のところは日中間のコミュニケーションコストなどを加えてもまだ国内開発より安いと考えている日本企業が多いが沿海部の人件費上昇によって早晩現在のビジネスモデルが成立しなくなると考える日本企業は内陸部に拠点を展開し始めている。この場合も豊富な人材があることを前提にすると西安あたりが限界と考えられている。

一方、中国企業側にも日本からの仕事がこのまま継続するかどうかについて危機感がある。現時点では日本の発注理由がコストのみであることを中国も認識しており、中国企業の過当競争状態の中でコスト競争力を維持するために上流工程分野に向かおうとする企業もある。設計や場合によっては要求仕様分析なども含むこういうプロジェクトはやはり日本滞在５年以上などのごく一部の中国人技術者にしか対応できないため、大きなマーケットになるかは疑問である。

単に低コストを追求すれば中国以外の国への展開もあり得ないことではないが、無尽蔵とも言える規模のしかも優秀な人材を提供できる国は他になく、日本からすると中国は貴重な良質の労働力供給源であることに変わりは無い。

論点４：ソフトウェア分野の研究開発力

現時点では日本から中国に委託している領域は比較的仕様が定まった部分の開発が主体であり日中間の技術レベルには一定の差があるとの認識がある。しかし、既に研究開発を中国に委託している企業もあり、膨大な人口と優秀な頭脳で中国は遠からず日本の技術力を凌駕するとの予測もある。

⇒

中国の技術力は総体的に考えればまだ日本に及ばない。ソフトウェア開発の作法にしても洗練されておらず、「動けば良い」的なレベルで留まっていることや十分な検証がなされていないことも多い。インド人技術者が拡張性やアーキテクチャ的観点なども考慮してプログラムを書くのと対照的と指摘する声もある。しかしこの点に関しては日本人技術者の発注者能力に問題があるケースも少なくない。中国に限らず、日本がソフトウェアプロジェクトで外国と連携するためにはCMMやISOなどに則った標準的なソフトウェア開発プロセスを身につけなければならないとの指摘はよく聞かれるところである。もっとも、この点が言わば非関税障壁となって中国人技術者が日本市場へ十分に対応できず、結果として「論点 1」で指摘したように中国での開発が大きくは広まらないために過度に日本のソフトウェア開発が中国に流れず、日本は空洞化を心配しなくて済むと言ういささか皮肉な状況もある。

日中の間のソフトウェア開発プロジェクトで発生する多くの問題の原因を、単に中国の技術レベルの低さに帰することは誤りで、日本側に原因があることもあれば、良し悪

しの次元を超えて文化の相違に根ざす問題もある。

ただし、中国国内でも、自国が世界に対して競争力のある独自の技術を持たないことについて危機感を持つ層があることは確かである。中国の大学では、米国企業の資格制度のための授業を行うなどきわめて実践的な教育をしており、最新技術を用いたプログラミングなどの分野では日本と遜色が無いかそれを上回る程度の力を蓄えつつある。しかしその反面、基礎的な力やそれに関連する教育は十分とは言えない。

論点5：中国企業による日本進出

日本と中国の関係の多くは現時点では日本企業による中国への事業展開あるいは日本企業による中国人技術者の雇用である。しかし、中国資本による日本のベンチャーのM&Aや中国企業の日本進出傾向も加速している。

⇒

中国企業による日本企業の買収の例はまだ少ない。しかしもともと投資や賭け事が好きな国民性なので、中国のベンチャーキャピタルが日本のソフト企業を買収するケースは今後増えていくであろう。日本の下請け企業で活路を中国に求めるケースも出始めている。自らの技術が残れば、外資であろうと、技術者を中国へ送ろうと構わないと考える経営者も多い。外資は買収しても基本的にマネジメントと日本のマーケットは日本にまかせるので、抵抗が少ないとの見解がある。

国際的なM&Aの背後にある事情は複雑である。たとえば、日本の中小企業で資金はないが、マーケットと技術力のある会社が中国の大手同業社と50%ずつ資本を出し合って会社を作り、そこが日本の会社を支配する形が出てきている。中国で安く製造したものを日本の会社を通して安価に売る、あるいは、日本の技術を利用して中国市場でも売ると言うパターンである。ソフトウェアではまだこういう例は少ない。

また、米国企業が日本の製造業を買収して、中国に移管するケースも出てきている。米国が資金を出して日本の優秀な技術を買い、中国に工場を作って安価で豊富な労働力を使って安くて良い製品を生産し、米国に供給するという構図である。こうした米国の投資グループには中国系米国人やユダヤ系米国人も多く、彼らとしては特に米国の国益を意識している訳ではなく、基本的には個人的な富の追求の結果

と考えるべきであろう。ビジネスで成功を納める上で国境は決定的な要因にはならない。

日本ではソフトウェアに限らず、M&A に対して十分な認識を持っている組織は極めて少ない。たとえば日本企業は意思決定があまりに遅すぎるとの指摘がある。M&A ではスピードが決定的に重要であるのに、リスク回避、責任回避、安全性重視のための社内根回しに時間を取られすぎた結果、後発の米国企業に先を越されてしまうケースもある。こうした状況から脱却するには経営者の意識改革が非常に重要である。

論点6：日本のソフトウェア産業の高付加価値型への展開

日本のソフトウェア産業では高付加価値型サービスで各企業がその力を競うような方向は残念ながら見られず、受託開発を中心として人件費コスト削減が唯一の競争力の源泉となっている。本質的にはこうした産業形態を改めてより高度で独自のサービスを各企業が展開する方向に向かわなければならない。

⇒

日本が目指すべき方向は概ねこの路線であると考えられるが、具体的な道筋が見えていないことが問題である。基本ソフトは言うに及ばず、主だったミドルウェアやパッケージもそのほとんどが欧米の製品に依存しており、純粋なソフトウェア製品に限定すれば日本の活路は一向に見えない。

本報告書のテーマである中国との関連で言うなら、上の観点からしても、市場として成長し始めている中国に対して日本はもっと積極的に対応すべきであろう。その中から新たな市場を通して日本の技術や製品が有用性を発揮できる分野を見出すことができる可能性もある。中国企業にしても、中国市場への展開に備えて日本企業の助力を得たいと希望するところがあることは既に指摘したとおりである。

本調査の主題であった空洞化への恐れは、前述したように、中国から見ると日本のきわめてローカルな見方から発するコップの中の嵐的現象であり、巨大な国内市場への近未来の対応の必要性に迫られている中国ではほとんど一顧だにされていない問題と言っても過言ではない。現在の空洞化問題は日本のソフトウェア産業がその市場を自ら日本国内に限定してしまった上でなおかつそれを守ることができないとの自信のなさの表れ

でもある。

むしろ日本のソフトウェア産業は中国市場の発展を契機に自らも飛躍する方向に向かいたい。中国市場や世界の市場で中国人技術者と連携しあるいは競合して切磋琢磨する機会を得ることの必要性を自覚する必要があろう。その結果として日本の技術が中国市場あるいは世界市場で中国に破れて危機的状況に直面することがあるとすれば、その時こそソフトウェア産業の空洞化問題が現在の不定愁訴の域を越えて現実的な課題となる。現在の日本のソフトウェア産業は真の意味での空洞化を心配する以前の状態にとどまっていると認識すべきである。

5.4 結論

日本のソフトウェア産業が、高度なレベルの技術が要求されない開発部分を主体とした業務を低賃金で優秀な技術者を擁する中国へ発注する傾向を強めることがソフトウェア産業の空洞化をもたらすのではないか、との懸念に端を発した本調査はより深刻な日本のソフトウェア産業の課題を浮き彫りにした。

当初の関心事であった中国への事業展開に伴う空洞化については、おそらくそのような事態には進展しないとの見方が支配的である。日本のソフトウェア産業はもっぱら国内市場のみを対象にしているため、外国人技術者あるいは外国企業にとっては日本語、あいまいな仕様書、日本の組織に特有の業務プロセス、日本文化などの日本的業務環境が言わば非関税障壁となり、日本市場への参入が容易ではない。したがって中国への委託が日本のソフトウェア産業の大勢を占めるには至らないであろうことがその根拠である。つまり、日本が危機感を感じている空洞化が現実のものとならないであろうとの観測は日本が国内市場に留まっていることが前提になっている。

グローバル化が進展し、また日本の成長産業のほとんどがグローバルマーケットを対象にしていることを考えると、ソフトウェア産業が国内市場のみに目を向けて自らを半ば鎖国状態に置いていることは、中国展開による空洞化への危惧とは比較にならない深刻な事態である。日本のソフトウェア産業は「前門の虎、後門の狼」とも言うべき状況に直面し、立ち往生を余儀なくされている。次代を切り開く先進的な分野では米国・欧州の後塵を拝し、一方定型的開発の領域では低コスト労働力を背景にした中国などアジア諸国が力を発揮しつつある。欧米が世界市場を対象に製品提供を行い、中国は今まさ

に勃興しつつある膨大な自国市場への対応に腐心しているのに対し、日本のソフトウェア産業には世界に対して技術・サービスを提供する力と意欲が欠けている。このような閉塞状況を脱して日本も世界に通用する高付加価値型の分野を是非とも切り開かなければならない。

本調査の主題である中国との関連で言えば、中国を低コスト労働力の供給元としてだけ考えるのではなく、やはり市場と認識して将来への布石を意識した展開が必要である。TD-SCDMA の例が示すように、中国一国の市場を中心的な対象にした規格が国際的標準規格として取上げられるほどの勢力は他には考えられず、このことのみを以ってしても潜在的な市場としての中国の勢いは推して知るべしである。また全世界に広がる華人ネットワークが持つ力もきわめて特異なものであり、早晩中国が世界の市場に成長することは確実である。もちろん、これは全中国が同時期に等しく市場化することを意味しない。その点では中国を一つと考えるのは不適切であり、ケースバイケースの対応が求められる。

日本のソフトウェア企業が中国に展開するパターンとしては、たとえば沿海部の富裕層を対象にした先進的な IT サービスの事業化を計画し、そのために日本の技術力に関心を寄せている中国企業と連携して日本が有する進んだ技術を生かす方策があり得る。あるいは中国人技術者には真似のできないキメ細かなもの作り力を生かして WTO を契機に経営の高度化を進める中国企業へ展開する道もあるかも知れない。こうした例に比べるとやや卑近で発展性に欠けるきらいがあるがリスクが低いものとして、中国へ進出したあらゆる業種の日本企業の情報化に関するニーズを、同じく中国に展開した日本のソフトウェア企業が引き受けるような形態のサービスもあり得る。

中国を市場と考えた事業展開には様々なリスクが伴う。しかし日本にいる我々が忘れがちなのは、日本に進出している中国企業、あるいは日本の業務を下請的に実施している中国企業も彼らなりに大きなリスクを感じ、それを克服して事業展開をしている点である。今回の調査でも、「日本相手のビジネスはリスクが大きく、あえて踏み出す決定ができない」との感想を漏らす中国の有力 IT 企業があった。文化・商習慣の相違に起因する起業へのためらいは片務的なものではなく相互が共に抱える問題である。つまるところ、リスクを取らざるを得ないと考える危機感、リスクを乗り越える戦略構想力、それらを支える起業家精神の有無が問題である。今の日本のソフトウェア産業界には総じてこれらが不足しているが、あながちその力が無い訳ではなく、奮起が切に期待される。

中国の経済成長は 2008 年の北京オリンピックが一つのマイルストーンになる。それまではほぼ現在の趨勢で成長が続くと考える識者が多いが、オリンピック後も同様の高い経済成長が持続できるかは疑問である。中国への事業展開には時間的にも地理空間的にも大きな変動要因が見え、中国から「日本は中国以上に社会主義的だ」と評されるほどに、ある意味で変化が少ない環境に慣れた日本企業にとっては第一歩を踏み出すことを躊躇させるに十分な雰囲気がある。しかしこの問題を避けて日本が安定的成長を続けることはおそらくできない。

わが国が行う情報技術研究開発のあり方に関する調査研究（その７）

© 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
先端情報技術研究所
東京都港区芝２丁目３番３号
芝二丁目大門ビル４階
TEL(03)3456-2511