

CRDS-FY2009-SP-12

戦略プログラム

情報システムに対する要求仕様の変化に対応するソフトウェア技術

STRATEGIC PROGRAM

Software technologies for adapting requirement
changes of information systems



独立行政法人科学技術振興機構 研究開発戦略センター
Center for Research and Development Strategy, Japan Science and Technology Agency

研究開発戦略センターでは、国として重点的に推進すべき研究領域や課題を選び、以下3種類いずれかの戦略プロポーザルとして発行している。

戦略イニシアティブ

国として大々的に推進すべき研究で、社会ビジョンの実現に貢献し、科学技術の促進に寄与する

戦略プログラム

研究分野を設定し、各チームが協調、競争的に研究することによって、その分野を発展させる

戦略プロジェクト

共通目的を設定し、各チームがこれに向かって研究することによって、その分野を発展させると同時に共通の目的を達成する

エグゼクティブサマリー

情報システムを構成するソフトウェアは複雑化、巨大化の一途をたどっている。情報システムが社会のインフラ化している今日において、その事故は非常に大きな影響を社会におよぼす。事故の原因がソフトウェアの不具合である事例も後をたたない。情報システムが、安全で安心してしかも持続的に安定して利用できることが社会の期待として大きい。

情報システムを構築するためのソフトウェア技術に関する研究開発は、古くから実施され、生産性向上や信頼性向上を目的としたソフトウェア開発支援ツールや開発の方法論が世に出されてきた。その中で、一定の成果を収めつつ、成熟期にさしかかったかに見えるソフトウェア技術であるが、オープンネットワークを基盤とする情報システムは、①不特定の利用者が存在し、その振る舞いや要求の変化を捉えることは困難といった課題、さらに、②情報システム自身の経年劣化や、企業合併などによる情報システムの統合といった社会環境の変化なども考慮しなければならないといった課題がある。これらの情報システムに対する要求仕様の変化にどうソフトウェア技術として対応するかという課題は、もはや国内ベンダー各社での地道な取り組みでは、解決できなくなってきており、大手ベンダー各社は、DSF (Dependable Software Forum)¹を設立し、大規模・複雑化するソフトウェアの課題に取り組み始めた。しかし、課題の重要性を考えれば、産業界だけでなく大学の研究者も含めて、日本全体の英知を結集して取り組むべきである。学側ではこれまで基礎研究として、様々な方法論やツール開発が行われてきたが、産側での適用にいたる成果としては十分とは言えない状況であった。学側がこれまで取り組んできた単独の方法論やツールだけで情報システムのライフサイクル全体（設計、開発、運用、保守）をカバーすることはできず、いくつかの研究成果を統合する応用研究が非常に重要となってくる。基礎研究とそれらの成果を統合する応用研究を産学連携により強力に推し進め、日本発のツールおよび方法論を世に問い、様々な産業のインフラとなる情報システムの安定的運用を目指さなければならない。

本戦略プログラムでは、基礎研究の具体的研究課題の例を提案するとともに、基礎研究の成果を統合する応用研究を DSF などの動きと呼応し産学連携により強力に推し進めることを提案する。

情報システムの設計、開発、運用、保守というライフサイクルにおいて、従来の研究は、設計および開発の部分に焦点があてられ、ソフトウェア工学としてソフトウェアの生産性向上、高信頼化に関する研究が主として行われてきた。しかし、今後は運用時における要求の変化をはじめとする情報システムを取り巻く様々な環境の変化に対応し、持続的な情報システムを実現するためのソフトウェア技術が求められる。設計だけでなく、運用、保守までも含むライフサイクル全体に対して、全体的アプローチでの研究を新たに推し進めなければならない。

具体的な研究課題の例として、ライフサイクル全体をカバーすることを目的に、変化する要求の抽出・分析、モデルにおける変化の記述、プログラム検査、モニタリングといっ

1 <http://www.nttdata.co.jp/release/2009/122200.html>

た基礎研究課題を提案する。さらに、こうした基礎研究の成果を現場で適用するために、いくつかの研究成果を統合する応用研究の推進に関する提案をおこなう。

これにより日本発の情報システム構築の方法論やツール群を世界に向けて発信し、各種産業のインフラとなる情報システムが様々な変化に対応し持続的・安定的に運用される状況を実現し、日本の産業競争力優位を実現する。また、実施される研究開発プロジェクトを通じてシステム思考ができる人材の育成を図る。

Executive Summary

Today information systems are the basis of almost all of the infrastructure of our society. Any failure in the information systems will cause tremendous impact on our society. Society requires information systems which are safe, secure and stable. It is widely recognized that the major cause of the information system failures is in software, which is steadily growing larger and getting more complex.

Research and development on software technology for safe, secure and stable information systems has been carried out for many years, and software development tools and methodologies have been developed to improve productivity and reliability. Although it seems as if software technology is approaching a maturity level, software in information systems based on open networks still have problems. For example, it is difficult to adapt to 1) unpredictable needs of unspecified users, and 2) the inevitable environment changes such as aging degradation of the information system itself or user's specification changes. The issue of how software technology could cope with such changes in information systems is no longer solvable only by the patient efforts of individual vendor. In order to deal with the software issues which are getting massive and complicated, the major Japanese vendors recently established Dependable Software Forum (DSF)¹. In academia, various methodologies and tools were developed as the result of basic research. However, the results were not widely applied in industry. Each methodology and tool alone, on which the academia has been working, can not cover the overall life cycle of information systems (design, development, operation, maintenance). Thus, the applied research to integrate multiple research results is of great significance. In order to achieve the goal of stable operation of the information systems which support the infrastructure of various industries, we should strongly pursue the industry-academia collaboration in basic research and applied research.

In this strategic program, we propose basic research on specific subjects, and strong industry-academia collaboration for applied research to integrate the results of basic research.

In the life cycle of an information system, which is its design, development, operation and maintenance, conventional researches have been mainly focusing on the design and development phases, especially on improving the productivity and reliability of the software. However, in the future, software technology will be required capable of realizing a sustainable information system which can respond to the various environment changes including the changes in specifications during

¹ <http://www.nttdata.co.jp/release/2009/122200.html>

operation. Not just design and development phases, but research on holistic approach towards the entire life cycle should be pursued.

For the purpose of covering the entire life cycle, we propose basic research subjects such as extraction and analysis of the changing specifications, description of changes in models, program inspection, monitoring etc. Furthermore, in order to apply the results of such basic researches to real systems, we propose promotion of applied researches integrating multiple research results through industry-academia collaboration.

If all of this goes well, the expected outcome will be as follows: we can contribute to the world by showing methodologies and tools for implementing information systems that can cope with various changes and operate sustainably and stably. We will also be able to nurture researchers and engineers who have system thinking capability.

目 次

エグゼクティブサマリー

Executive Summary

1. 提案の内容	1
2. 研究投資する意義	5
3. 具体的な研究開発課題	7
4. 研究開発の推進方法	11
5. 科学技術上の効果	13
6. 社会・経済的效果	15
7. 時間軸上の考察	17
8. 検討の経緯	19
9. 国内外の状況	21

付録 専門用語説明

1. 提案の内容

情報システムを構成するソフトウェアは複雑化、巨大化の一途をたどっている。情報システムが社会のインフラ化している今日において、証券会社の取引システムの事故や、交通機関での予約システムの停止によるトラブルなど、その事故は非常に大きな影響を社会におよぼす。事故の原因がソフトウェアの不具合である事例も後をたたない。情報システムが、安全で安心にしかも持続的に安定して利用できることが社会の期待として大きい。

情報システムを構築するためのソフトウェア技術に関する研究開発は、古くから実施され、生産性向上や信頼性向上を目的としたソフトウェア開発支援ツールや開発の方法論が世に出されてきた。

こうした中で、一定の成果を収めつつ、成熟期にさしかかったかに見えるソフトウェア技術であるが、今日的オープンネットワークを基盤とする情報システムにおいては課題も多い。今後ますます重要となる大規模でオープンなネットワークを前提とした情報システムでは、利用者像を特定することができない。一般利用者は、思い違い、不注意、悪戯など、気まぐれな振る舞いを示すかもしれない。さらに、悪意を持って「攻撃」することさえある。このような開発時に予測できなかった利用者の振る舞いや要求の変化に対して、システムがどのように振る舞うかも予測できないし、利用者要求の変化をあらかじめ予測することも不可能である。さらに、情報システム自身の経年劣化や、企業合併などによる情報システムの統合といった社会環境の変化なども考慮しなければならない。つまり情報システムに対する要求仕様の変化にいかに対応するかという課題をソフトウェア技術として解決しなければならない。

これらの課題は、もはや国内ベンダー各社個別の地道な取り組みでは、対応できなくなってきたおり、大手ベンダー各社は、DSF (Dependable Software Forum)²を設立し、大規模・複雑化するソフトウェアの課題に取り組み始めた。しかし、課題の重要性を考えれば、産業界だけでなく大学の研究者も含めて、日本全体の英知を結集して取り組むべきである。学側ではこれまで基礎研究として、様々な方法論やツール開発が行われてきたが、産側での適用にいたる成果としては十分とは言えない状況であった。学側がこれまで取り組んできた単独の方法論やツールだけで情報システムのライフサイクル全体（設計、開発、運用、保守）をカバーすることはできず、いくつかの研究成果を統合する応用研究が非常に重要となってくる。

本戦略プログラムでは、基礎研究の具体的研究課題の例を提案し、基礎研究の成果を統合する応用研究を DSF などの動きと呼応し産学連携により強力に推し進めることを提案する。

これまで情報システムは、決められた条件下で設計・開発を実施し、運用時に出た変化への対応は、主に再設計・再開発という形で行われてきた。今後は、運用時における要求の変化をはじめとする情報システムを取り巻く様々な環境の変化に対応し持続的な情報システムを実現するためのソフトウェア技術が求められる。

情報システムの設計、開発、運用、保守というライフサイクルにおいて、従来の研究は、

2 <http://www.nttdata.co.jp/release/2009/122200.html>

設計および開発の部分に焦点があてられ、ソフトウェア工学としてソフトウェアの生産性向上、高信頼化に関する研究が主として行われてきた。しかし、上記観点から、設計だけでなく、運用、保守までも含むライフサイクル全体に対して、全体的アプローチでの研究を新たに推し進めなければならない。

具体的な研究課題の例として、ライフサイクル全体をカバーすることを目的に、変化する要求の抽出・分析、モデルにおける変化の記述、プログラム検査、モニタリングといった基礎研究課題を提案する。欧米では、変化への適応という観点から、生物学的アプローチや人工知能的アプローチからの研究も実施されているが、ここでは数理・論理に基づく方法を中心とした研究を想定している。さらに、こうした基礎研究の成果を現場で適用するために、いくつかの研究成果を統合する応用研究を推進するとともに、研究成果であるツールをできるだけ多くの人に利用してもらいフィードバックを得るために、オープンソースのコミュニティ作りも推進する。

情報システム構築のためのツールとしてのソフトウェアに関する研究成果が市場に出るまでには、以下のフェーズがある。

- ① 基礎研究フェーズ
- ② 産業界のニーズを取り入れ基礎研究成果を統合する応用研究フェーズ
- ③ 実証実験フェーズ
- ④ 産業界が中心となって普及展開する市場化フェーズ

本戦略プログラムで提案している国が行うべき研究開発投資の対象フェーズとしては、①と②の部分である。しかし、情報システムは、社会の重要インフラであり、③においては、継続的な産学連携、④に関しても、研究成果であるツールの利用を公的調達要件とするなど、国としても普及展開する取り組みが求められる。

●コラム 1 今日の情報システムの特徴

ここでいう情報システムは、次のような特徴を持つシステムを対象としている。①レイヤー化された複雑なシステム、②人間の操作による不確実性を含むシステム、③ネットワークで他のシステムと繋がれたシステム、④それぞれのシステムやサブシステムの寿命が異なるものの集合体としてのシステム、⑤要求や環境の変化に対応し、進化し続けなければならないシステム。このような情報システムの例としては、災害通報システム、交通 I C カードシステム、銀行 A T M システム、座席予約システム、証券取引システム、電子申請システム、電子商取引システムなどがある。

情報システムは、以下の困難さをもつ。

仕様／実装の不完全さ、システム完全理解の困難さ

要求、仕様、設計、実装、テスト仕様、テスト実施の不完全さ

構成要素の論理的不透明さ（複雑化、巨大化、ブラックボックス化、レガシーコードの利用 など）

使用状況の不確実さ、システムの挙動予測の困難さ

要求事項・レベルの変化、想定外の使われ方

ネットワークを介しての構成要素の変化、想定外の接続

ネットワークを介した外部からの意図的な攻撃

情報システムは不完全さと不確実さに起因する未来に障害となりうる要因（開放系障害要因）を宿命的に抱えており、それが顕在化した時のシステムへの影響をマネージして（顕在化前もしくは顕在化後に適切に対応し、影響を最小とするように）、利用者が期待する便益をできる限り、安全にかつ継続的に提供しなければならない。

DEOSプロジェクト（CREST「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」）中間報告書より

2. 研究投資する意義

(1) 社会のインフラである情報システムの安定化

国家の基盤、企業活動の基盤、生活の基盤である情報システムは、社会の重要インフラである。その安全性・信頼性を確保し、人や組織の活動が依拠するサービスと情報に対する信用を保証し、情報社会の安全と信頼を持続的に保障するシステムを確立することによって、情報社会の安定化と将来の発展が確保される。本戦略プログラムが提案する研究プロジェクトの成果である情報システム構築の方法論やツールを利用して、情報システムを構築することにより、情報システムの安全と信頼を持続的に保障することが可能となる。情報システムのうち、特に公共の社会インフラにかかわる部分については、国の積極的投資が必要であり、そこでの成果は民間への展開にいかされる。

(2) 国際競争力の強化

上記で述べたように、社会インフラ化している情報システムは、当然その上で企業活動を行う企業にとっての生命線であり、その安定化なくしては、各種産業の発展は図れない。

この分野の取り組みは、欧州が先行しており、産業界への展開が図られている（9.2 節参照）。

本戦略プログラムが提案する研究プロジェクトの成果である情報システム構築の方法論やツール群は、これから社会インフラとして情報システムが展開される国々への輸出が可能である。日本の特徴である非常に高度なシステム運用技術などは、この研究プロジェクトが狙いとする設計から運用まで含む全体的アプローチの成果を用いることでさらに磨きがかかり十分競争力となりうる。

(3) 学際分野「情報社会技術」の発展

情報システムは社会と密接に関連しており、情報技術とともに社会技術と呼ぶべき概念、技術、プロセスが必要である。こうした情報社会システムのアーキテクチャやデザイン、実装を実施するプロジェクトを実施することで、新しい学際分野「情報社会技術」が発展し、人材育成のプラットフォームを確立する効果がある。

(4) 人材育成

従来から日本は部品や材料に関しては競争力があるが、システムの部分が弱いとの指摘もあり、本戦略プログラムが提案する研究プロジェクトを通じて、システム思考のできる人材を育成することが可能となる。

3. 具体的な研究開発課題

従来の情報システム開発プロセスは、設計・実装・テストを順に行い、リリース（顧客への引渡し）するまでが一区切りであり、その後の運用において出てくる変更要求については、小さな変更ならば保守作業として部分的改変を行い、大きな変更ならば設計からやり直して新システムに置き換えるのが普通であった。しかし、情報システムが大規模かつオープンな（他の情報システム／物理システムと接続されている）社会的インフラとなる場合には、このような開発プロセスは困難であると予想される。それは、主として次の理由による。

- 利用者の要求や運用環境は常に変化するが、システムが大規模なため開発・リリースのサイクルが変化に追いつかない。
- 高い信頼性を要求されるが、オープンシステムを完全にテストするのは難しい。

例えばスマートグリッドを考えてみると、CO₂削減に関する法制度の変更や、新しい発電・蓄電技術の登場はいつ起きるかもしれないが、もし部分的改変で間に合わないような大きな変更が必要になったとしても、スマートグリッドシステム全体を作り直すには多くの時間がかかるため、実世界の変化への対応が間に合わない。また、家電制御システムや料金決済システムと接続された場合、システム間の相互作用まで含めたテストを行うのは難しいし、どれかのシステムに変更があるたびに、接続されているすべてのシステムのテストをやり直すのは、コストの面で現実的ではない。

このような問題点を解決し、情報システムの信頼性を保ちつつ持続的に発展させるためには、設計・実装と運用・保守を一体化した連続的な開発プロセスが必要になると予想される（図1）。設計は抽象的なモデル記述を用いて行い、運用中に出てくる要求の変化は、モデルの作り直しではなく、モデルの変化として追加的に記述できるようにする。モデルが変化すると、それを動作中のプログラムに反映させることによって、要求の変化への対応が完了する。その過程では、変化する要求を抽出・分析する要求工学的手法、モデルの変化を正しくプログラムに反映し、高い信頼性を実現する形式的手法が使われる。

以下で例示する研究開発課題のうち、（1）～（4）は4章で述べる基礎研究として実施する課題である。それに加えて（5）で述べるように、既存技術と新しい方法論やツールを統合し、現場での試使用などのフィードバックを受けて基礎研究に反映する応用研究も並行して実施する。

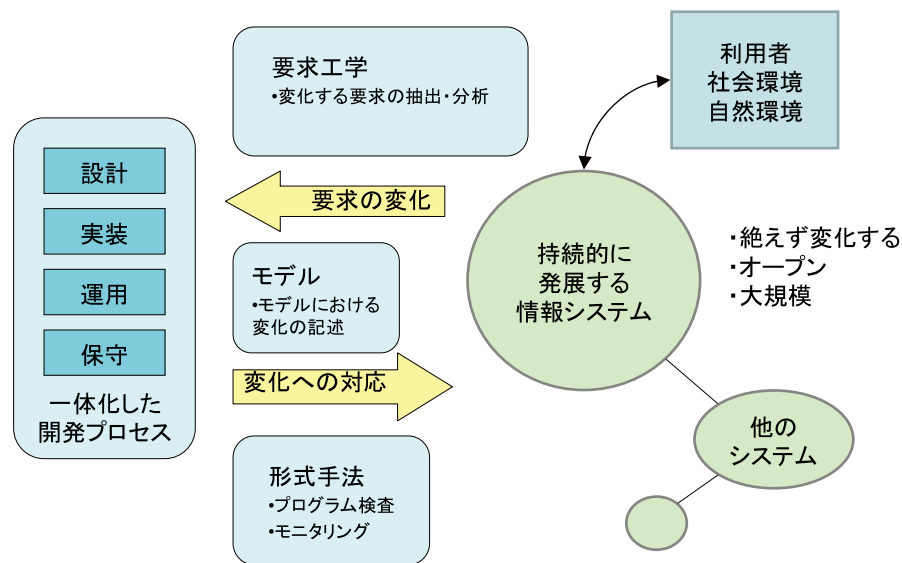


図1 持続的に発展する情報システムの開発プロセス

(1) 変化する要求の抽出・分析

要求仕様は、伝統的なウォーターフォール型では開発の最初の段階で決定されるべきものであったが、近年注目されている反復型およびアジャイル開発においては、開発の過程で徐々に決定されていくものである（コラム2 参照）。さらに、持続的発展可能なシステムを実現するには、システムがリリースされて運用中であっても要求仕様が変化していくことを前提に考えなければならない。

例えば、時間の経過とともに、最初は想定されていなかった利用方法や他のシステムとの連携などが必要になることがあるが、その要求仕様は元々想定されていた利用方法についての要求仕様と整合性があるとは限らない。したがって、矛盾する要求を含む仕様を記述し、それを解決するための意思決定支援の手法が求められる。

現在の開発プロセスでも往々にして開発中に要求が変化することがあり、それに対応するため要求工学ではソフトシステムズ方法論、コンカレント要求工学、アジャイル開発手法などが研究されている。

このような技術はすでに多くの研究や実践が行われているが、まだデファクト・スタンダードとなるような有力な技術は現れていない。また、絶えず変化し続けるオープンシステムに適用した例は少なく、そのような場合での有効性も明らかではない。したがって、ある程度の規模のシステムで実験を行い、その知見に基づいて既存手法の改良や新しい手法の開発を目指す研究開発を行う。

(2) モデルにおける変化の記述

保守作業によってプログラムの修正が繰り返されると、最初の設計からしだいに乖離していく。するとプログラムの見通しが悪くなり、修正の手間の増大やバグの発生といった悪影響が現れてくる。また、差分開発においては、モジュール単体のバグよりも新しいモジュールを組み入れた時のモジュール間の関係によって問題が起りやすいことが指摘されている。このような問題を防ぐには、抽象度の高いモデルによって変更を管理し、それ

をプログラムに正しく反映させることが有効であると思われる。

モデル駆動開発においては、モデル記述からプログラムを生成することが行われているが、モデルの変更を管理するという視点からの研究はあまりされていない。

要求仕様の変化に伴ってモデルがどのように変更されるかという差分を明示的に記述できるようなモデル記述言語（既存言語の拡張、または新規言語）を研究開発する。また、モデルの変更が正しくプログラムの変更に反映されているかを管理・追跡できるようなツール類の開発を行う。

（３）プログラム検査

モデルを数理・論理的手法を用いて記述する利点は、プログラムがモデルに対して正しいかどうかを検証できることである。ただし、厳密な証明は難しいため、完全ではないが簡略化した手法による検査を行う場合も多い。

持続的に発展するシステムでは頻繁にモデルが変更され、それに伴ってプログラムも変更される。この時、モデルの変更とプログラムの変更が正しく対応しているかどうかを検査することは、システムの信頼性を確保するために極めて重要であるが、人間がすべてテストを行うと非常にコストがかかる。特にオープンシステムでは、接続されたシステムにも影響が及ぶので、人力でのテストには限界があり、何らかの検査ツールの利用が欠かせない。

この分野の研究は、検証、モデル検査、テスト自動生成など多岐にわたっており、さまざまな研究成果が出ているが、産業界での応用はようやくいくつかの採用例が出た段階である。したがって、実用に近い規模の実験を重ねることにより、実用化における困難がどこにあるかを究明し、それを解決するための技術を研究開発する。また、先に述べたモデルにおける差分の明示的記述に対応して、以前の検証履歴を再利用して効率よく検証作業を行う、差分検証の技術を研究開発する。

（４）モニタリング

実行中のシステムの挙動を調査する手法として、要求モニタリングがある。これは、性能や信頼性などの要求が満たされているか確認するため、情報収集の機構をあらかじめシステムに組み込んでおくことである。

持続的に発展するシステムにおいては、運用状況を把握し、改善の必要があれば設計にフィードバックすることが、従来のシステムにも増して重要である。しかし、モニタリングによって収集されるデータを人間が直接監視しなければならないのは負担が大きい。そのため、設計で用いられたモデル記述と対応づけられるよう、収集されたデータを抽象的なモデルのレベルで解釈して人間に提供する技術を研究開発する。

（５）一体化した開発プロセス

既存の開発プロセスと上記（１）～（４）の要素技術を統合し、設計・実装・運用・保守を一体化した連続的な開発プロセスについて研究開発する。これには新しい方法論の提案と、実用に耐えるツールの製作が含まれる。

●コラム2 ソフトウェア開発の方法論

ソフトウェア開発のプロジェクトをどのような工程で行うかは、実際の開発現場において非常に重要な課題であり、いろいろな方法論が提案されている。

最も古くからあり、現在でも広く使われているのがウォーターフォール型である。これは、次の工程を順を追って実施していくものである。

1. 要求定義
2. 設計（概要設計と詳細設計に分けることがある）
3. 実装
4. テスト
5. 保守

それぞれの工程が完了するまでは次の工程には入らないし、原則として前工程に戻ることも無いため、作業は滝が流れ落ちるように各工程を進んでいく。しかし、実際の開発ではそのような理想的な進捗になることはほとんどなく、設計の段階で要求の矛盾が見つかったり、テストの段階で設計の不備が見つかったりするので、前工程へのフィードバックを考慮することは不可欠である。

ウォーターフォール型に対する批判として、そもそも顧客が自分自身の要求を知らないのが、最初に要求を固定するのは現実的でないことがある。その結果出てきたのがラピッドプロトタイピングである。これは、本格的な開発の前に簡単なプロトタイプを作り、それを顧客と開発者が検討することにより要求定義や設計の漏れを無くするというものである。

ラピッドプロトタイピングの考え方をさらに押し進めると、反復型（あるいはスパイラル型）になる。最初は、動作するのに最低限必要な部分や要求が明確な部分だけを作り、それをプロトタイプとして顧客が検討し、明らかになった要求を元に再び設計と実装を行うことを繰り返す。つまり、ウォーターフォール型の各工程を反復しながら次第に大きなシステムを作っていく方法である。

アジャイル開発においても反復型の手法が採られている。アジャイル開発とは、単なる開発工程だけの方法論ではなく、顧客と開発者の関係を始めとしてソフトウェア開発プロジェクトのスタイル全般を見直すという提案であるが、実際の開発手法としては、問題を分割し、非常に短い期間（通常は数週間程度）を単位とした反復型の開発を行う。

実装やテストの工程をできるだけ厳密かつ自動的に行い、人為ミスを極力減らそうとするのが形式手法である。形式手法そのものは、形式言語や定理証明を利用した技術全般を指し、開発の方法論に特有のものではない。開発の方法論における形式手法としては形式的仕様記述、モデルからのプログラム生成、プログラムの正当性の証明などがある。

4. 研究開発の推進方法

情報システム構築のためのツールとしてのソフトウェアに関する研究成果が市場に出るまでには、以下のフェーズがある。

- ① 基礎研究フェーズ
- ② 産業界のニーズを取り入れ基礎研究成果を統合する応用研究フェーズ
- ③ 実証実験フェーズ
- ④ 産業界が中心となって普及展開する市場化フェーズ

本戦略プログラムで提案している国が行うべき研究開発投資の対象フェーズとしては、①と②の部分である。

従来、大学での基礎研究の成果が、産業界で適用されなかった大きな理由としては、研究成果である原理や方法論を、実際に使えるソフトウェアとして仕上げる部分での投資が効率よくなくおらず、使えるシステム開発支援ツールとして広まらないことがある。既存の技術や①の基礎研究成果の進捗状況を見ながら、適切な組み合わせを試みる②の応用研究フェーズの研究が重要な役割を果たす。従って、①と②を並行させてファンディングし実施することが必要となる。

また、③の実証実験フェーズでは、①と②の成果が実際の開発現場で適用可能かどうかを検証するために大規模な実証実験を行う。この時、企業研究者、大学の研究者が入り込み、新しい課題を見つけ①の基礎研究にもどるといったフィードバックも重要となる。

ここでは、①基礎研究フェーズと②応用研究フェーズの体制について述べる。

基礎研究フェーズでは、3章で述べた研究課題に対して、方法論やツールのアイデアについては、個人の独創性もあり個人での挑戦も可能であるが、ツールを実際に使えるものとして作り上げ、ツールを統合する応用研究はチームで挑戦することになる。参加が想定される研究者としては、ソフトウェア工学の分野で、要求工学、形式手法、モデル化などの研究者、さらに要求抽出・分析に関連して、シミュレーションや行動分析、データマイニングの研究者などの参画が想定される。

応用研究フェーズに関しては、方法論や原理の研究およびその成果の現場適用を企業でおこなっている企業内研究者との連携が非常に重要となる。情報システムのライフサイクル全般にわたって、1つの方法論やツールでカバーできるものではなく、いろいろなツールを統合する試みが重要となる。また、その原理や方法論をソフトウェア開発やシステム構築・運用の現場に持ち込むためには、使えるツールとしてのソフトウェアが必要となる。このツールの品質がある程度保証され、使い勝手のよいものでなければ、いかに方法論や原理が素晴らしいものであっても、現場では使われないものとなる。ツールの作成は、1社で閉じて実施するには限界があるので、オープンソースとして公開し、ソフトウェアをできるだけ多くの人に利用してもらい、フィードバックを得るためのコミュニティ作りも

推進する。

5. 科学技術上の効果

(1) ソフトウェア工学の新たなフロンティア

これまでソフトウェア工学は、ソフトウェアの設計・開発に焦点をあて、生産性の向上や、信頼性の向上に取り組んできた。

ソフトウェアのライフサイクル全般に関わる変化適応の研究開発により、要求工学やアーキテクチャ、開発方法論などに、これまでのスタティックなモデルからダイナミックなモデルへの転換が必要となり、新たな発展が期待できる。

ライフサイクルに関わる人間観察などの社会技術的観点からの要求分析や予測技術などとの融合という観点からもソフトウェア工学への展開が期待できる。

(2) 情報社会技術の創出

情報システムは社会と密接に関連しており、情報技術とともに社会技術と呼ぶべき概念、技術、プロセスが必要である。こうした情報社会システムのアーキテクチャやデザイン、実装を実施するプロジェクトを実施することで、情報技術と社会技術の融合領域である新しい学際分野「情報社会技術」が発展し、人を含めた情報システムのあり方、社会システムをデザインする中での情報システムのデザインといった研究が進展するとともに、人材育成のプラットフォームを確立する効果がある。

社会技術の分野で取り組まれている人間の行動に関わる研究や、コミュニケーションの研究など、情報技術との融合によってさらなる発展が期待できる。

6. 社会・経済的効果

(1) 社会インフラとしての情報システムの安定化

本戦略プログラムが提案する研究プロジェクトの成果である情報システム構築の方法論やツールを利用して、情報システムを構築することにより、国家の基盤、企業活動の基盤、生活の基盤である情報システムの安全と信頼を持続的に保障することが可能となる。社会の重要インフラの安全性・信頼性を確保し、人や組織の活動が依拠するサービスと情報に対する信用を保証し、情報社会の安全と信頼を持続的に保障するシステムを確立することによって、情報社会の安定化と将来の発展が確保される。情報システムのうち、特に公共の社会インフラにかかわる部分については、国の積極的投資が必要であり、そこでの成果は民間への展開にいかされる。

(2) 産業競争力

社会インフラ化している情報システムは、当然その上で企業活動を行う企業にとっての生命線であり、その安定化なくしては、各種産業の発展は図れない。そうした観点から、この分野の取り組みは、欧州が先行しており、産業界への展開が積極的に図られている(9.2節参照)。

本戦略プログラムが提案する研究プロジェクトの成果である情報システム構築の方法論やツール群は、これから社会インフラとして情報システムが展開される国々への輸出が可能である。日本の特徴である非常に高度なシステム運用技術などは、この研究プロジェクトが狙いとする設計から運用まで含む全体的アプローチの成果を用いることでさらに磨きがかかり十分競争力となりうる。

また、進化容易なシステムにより企業合併に伴うシステム統合なども容易になり、社会の柔軟性やダイナミズムを増大させる。

(3) 管理コストの低減

かなりの人手をかけておこなっていた保守作業や再開発への研究成果の適用により、人手を削減することが可能となり、管理コストが削減できる。

(4) 人材育成

従来から日本は部品や材料に関しては競争力があるが、システムの部分が弱いとの指摘もあり、本戦略プログラムが提案する研究プロジェクトを通じて、システム思考のできる人材を育成することが可能となる。

7. 時間軸に関する考察

第4章の研究推進方法でも述べたが、ソフトウェアの研究成果が市場に出るまでには、以下のフェーズがある。

- ① 基礎研究フェーズ
- ② 産業界ニーズを取り入れ基礎研究成果を統合する応用研究フェーズ
- ③ 実証実験フェーズ
- ④ 産業界が中心となって普及展開する市場化フェーズ

国として研究プロジェクトを実施するにあたり、ワークショップなどを複数回実施し、基礎研究課題の整理や応用研究実施のための体制について議論を深め、企業内研究者を含むコミュニティ作りを実施する。こうした準備期間として1～2年程度を想定する。

一般的にソフトウェア開発支援ツールが研究段階から実際に広く世の中に使われるまでに10年程度かかることを想定し、①と②で5年間程度プロジェクトを実施しひとつの成果の区切りとする。基礎研究フェーズおよび応用研究フェーズを並行に5年間程度実施し、その後、実証実験フェーズを3年間程度実施する。

①と②の間、②と③の間、さらには④から①へのフィードバックサイクルをまわし、研究の持続的ループを構成する。

最終的にはプロジェクトの成果を、新興国を市場とした欧米との競争に打ち勝つために、日本発のソフトウェア開発手法のデファクト化を進め、産業界への展開、教育、普及と展開していく。

また、普及のための関連施策として、以下のような検討も必要となろう。

- ソフトウェア調達における解析可能なモデル記述の推進
 - 現在自然言語で行われている RFP (Request for proposal) や提案書にかかれる要求事項を解析可能なモデルで記述し、双方で理解可能なものとする。そのためのガイドライン作成や公共調達での条件とするなどの取り組みを行う。
- 技術成果に対する公共系システム開発への試行適用を実施

8. 検討の経緯

以下の方々から提案の構想段階で有益なご意見を伺った。

- 所真理雄 所長
 - － ソニーCSL研究所
- 本位田真一 教授
 - － 国立情報学研究所
- 玉井哲雄 教授
 - － 東京大学 大学院総合文化研究科
- 山本修一郎 教授
 - － 名古屋大学 情報連携統括本部 情報戦略室
- 米澤明憲 教授
 - － 東京大学 情報基盤センター長
- 國領二郎 教授
 - － 慶應義塾大学 総合政策学部長
- 笠原裕
 - － NEC 支配人
- 中島震 教授
 - － 国立情報学研究所

9. 国内外の状況

9. 1 国内の状況

ソフトウェアの変化への対応に関する研究としては、1997年度から3年間にわたって科研費重点領域・特定領域研究として行われた「発展機構を備えたソフトウェアの構成原理の研究」がある。主要な研究課題は(1)発展機構を備えたソフトウェアの理論、(2)ソフトウェア発展のための基本機構、(3)発展可能ソフトウェアの構築方法論、(4)ソフトウェア発展方式、(5)既存ソフトウェアの適応的発展である。この研究が元になって1998年から International Workshop on Principles of Software Evolution という国際会議が毎年開かれるようになった。当初は日本で開催され、参加者の大半が日本人だったが、近年はヨーロッパでの開催が多い。

ソフトウェアの信頼性向上という課題に関連し、主要な調査・研究として次のようなものがある。

- 文部科学省では、2003年から5カ年計画で「e-Society 基盤ソフトウェアの総合開発」として、「高信頼組込みソフトウェア構築技術」や「プログラム自動解析に基づく高信頼ソフトウェアシステムの構築技術」および「安全なシステム記述言語および高信頼 OS 記述言語」の開発等を実施した。
- 産業技術総合研究所は2004年から6年間の計画でシステム検証研究センターを設置した。2009年には北陸先端科学技術大学院大学、国立情報学研究所、三菱総研、関西電力、メルコ・パワー・システムズと共同で MCBOK2008（ソフトウェア開発のためのモデル検査知識体系）を発表した。また、2010年4月からは連携検証施設「さつき」でこの方向の研究が続けられる予定である。
- CRDS の戦略プログラム「組込みシステム用ディペンダブル OS」³をひとつのきっかけとして CREST「実用化を目指した組込みシステム用ディペンダブル OS」(DEOS プロジェクト)が2007年度にスタートした。
- IPA-SEC(ソフトウェアエンジニアリングセンター)は、2007年度産学連携ソフトウェア工学実践拠点において、「高信頼ソフトウェア構築技術に関する動向調査」を行った。
- NTT データ、NEC、富士通、日立、東芝などが2010年1月にディペンダブル・ソフトウェア・フォーラムを設立した。これは、ソフトウェアの信頼性を向上させるために形式手法などの設計技術を確立し、産業界への普及と定着を図ることを目標としている。

このように、すでに多くの基礎的研究がなされているが、産業界への応用はあまり進んでいないのが現状である。その原因としては、効果の検証されていない手法を実システムの開発に適用することのリスク、新しい手法を採用するためには人材育成などのコストが大きいこと、ハードウェアの性能による制約やソフトウェア・ツール類の整備不足などが挙げられる。しかし、情報システムの大規模化・複雑化がますます進展した結果、産業界においても、従来の手法だけではもう限界であるという認識が広まってきている。例えば、

3 CRDS-FY2005-SP-05

ディペンダブル・ソフトウェア・フォーラムの設立に見られるように、大学ではなく企業が中心になって、実用化を念頭に置いた研究の取り組みを始めているのは注目すべきである。

9. 2 国外の状況

(1) 国際会議

情報システムの変化に対応する技術は、一般に適応型システムと呼ばれる。これには、自動再構成や自己管理の機能を備え、人間が介在せずに自動的に適応するものも含まれるが、まだ基礎研究や実験段階のものが多い。それに対して、人間がソフトウェアを変更することにより対応させるソフトウェア工学的手法は、より実用に近いレベルの研究が多い。

要求工学の分野では IEEE International Requirements Engineering Conference、形式手法の分野では International Symposium on Formal Methods という、それぞれ5日間に渡る会議が毎年開催されており、活発に研究が行われている。

変化に対して自動的に対応する適応型システムについては、次のような会議が開催されている。

- IEEE International Conference on Self-Adaptive and Self-Organizing Systems は 2007 年に関連する 4 つのワークショップを統合して開催され、それから毎年開かれている。
- ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems は 2006 年から開催されており、上の IEEE の会議に比べてやや伝統的ソフトウェア工学寄りの発表が多い。

(2) EU の FP7

適応型システムに関するプロジェクトとして、例えば次のようなものが挙げられる。

- Security engineering for lifelong evolvable systems (2009-2012) : 長期間使い続けるシステム（モバイルと家庭用が主な対象）のセキュリティ、プライバシー、ディペンダビリティを確保することを目的とする。ソフトウェア工学的な手法・ツールの開発が中心である。
- Dynamic variability in complex, adaptive systems (2008-2011) : 変更の可能性をあらかじめ考えておき、実行時に動的再構成する技術の開発が目標である。Aspect oriented software design と Model driven engineering が中心である。

形式手法に関しては、FP6 RODIN の後継として 2008 年から 4 年間の計画で DEPLOY プロジェクトが行われている。これは、形式仕様言語 Event-B やリファインメントモデリングについて研究し、それを産業界に展開することを目的としている。

この分野の取り組みでは日本・米国よりも EU が先行している。特に、形式手法に関して産業界への展開に力点を置いているのは注目すべきである。今後わが国でこの分野の技術が必要になった場合、海外の研究成果を参照するだけでは人材育成やノウハウの蓄積に長い時間がかかってしまうので、今から産学連携による研究開発を進めておく必要があるのではないかと思われる。

(3) 米国

NSF (National Science Foundation) の FY 2009 Budget Request to Congress の中に Adaptive System Technology という項目がある。これはヒューマン・マシン・インターフェースに焦点を絞り、利用者に合わせて自動的に適応するシステムを研究しようとするものである。NSF が出した 2010 年のファンディングのテーマに Building Engineered Complex Systems がある。これは情報システムに限らず人工物一般を対象としているが、設計に形式手法を用いることが想定されている。NSF の Cyber Physical Systems に関するプロジェクトの中で高信頼性システムや形式手法の研究が行われている。また、マイクロソフト研究所がプログラム自動検査技術の研究を積極的に行っていることが注目される。

(4) 国際標準

情報システムが重要な社会的インフラとなるに従って、安全性に関する要求が厳しくなっている。製品の安全性を保障する手段としては、形式手法による検証が注目されている。電気・電子関連の安全規格である ISO/IEC61508 では、Safety Integrity Level 2 以上で形式手法を推奨している。また、IT 製品のセキュリティに関する国際標準 ISO/IEC 15408 では、Evaluation Assurance Level 5 以上で準形式的表現、Level 7 では形式的表現が必須となっている。

付録 専門用語説明

アジャイル開発手法

アジャイル開発手法とは、顧客と開発者の協働作業や、迅速で短期の開発作業の繰り返しなどを特徴とする複数の手法を組み合わせた開発方法論である。要求抽出に関しては、「要求は変化する」および「問題を細かく分割し、それぞれの問題の開発期間を非常に短くすれば、要求変更コストは高くない」という仮説に基づいており、顧客が開発チームに参画して短期間の開発とレビューを繰り返すことにより、段階的に要求抽出が行われる。

オープンソース

オープンソースとは、ソフトウェアの著作権者の権利を守りながらソースコード（プログラム）を公開することを可能とするライセンス（ソフトウェアの使用許諾条件）を示す概念である。多数の開発者が公開されたソースコードを利用することで多くのフィードバックが得られ、ソフトウェアの改良が加速されることが期待される。

形式手法

数理論理的な言語や証明を利用し、厳密に正しさを保証する手法の総称。

コンカレント要求工学

一般のコンカレント工学は、設計から生産までの部門が協働して同時並行的に作業を行い、開発期間の短縮や、後工程からのフィードバックによりコスト減と品質向上を目指すものである。この考え方を要求工学に取り入れたのがコンカレント要求工学で、要求分析と設計を並行して行う。

差分開発

システムを開発するとき、以前に開発したシステムをベースにして、異なる部分だけを開発する手法。

仕様記述言語

要求仕様やモデルを曖昧にならないように書き記すための言語。数理論理的に厳密なものを形式仕様記述言語と呼ぶ。

ソフトシステムズ方法論

目的が明確に定義されていない状況においては、「そもそも何を解決すべきか」というところから検討しなければならない。ソフトシステムズ方法論は、関係者間の認識の違いを明確にし、解決に向けての合意を形成する手法である。

モデル

一般には、問題となる対象をある特定の形式で表現したものである。ソフトウェア開発におけるモデルは、プログラムよりも抽象的なレベルでソフトウェアを表現したものであり、UML, Z, VDM などで記述されることが多い。

モデル駆動開発

まず問題領域や対象をモデル化し、そのモデルを段階的にプログラムに変換していくことにより実装を行う手法。

要求工学

システム構築における顧客の要求を、科学的に定義する方法や考えの総称。顧客は何が目的なのか、どういった機能が必要なのかといった要求をまとめ、それをシステムに正しく反映させる手法全体を対象とする。

要求仕様

構築するシステムに対する顧客の要求を定義したもの。通常は要求抽出と要求分析によって作られる。

要求抽出

顧客や専門家の知識から、構築するシステムに対してどのような機能や性能が求められているのかを明らかにすること。

要求分析

抽出された要求の関係性や優先順位を明らかにすること。

UML (Unified Modeling Language)

OMG (Object Management Group) が 1997 年にオブジェクト指向開発のための標準的な仕様記述言語として提案したもの。図式による表記が特徴である。

VDM (Vienna Development Method)

数学を基盤として仕様、プログラムの正当性を保証するための手法で、形式手法のひとつ。IBM のウィーン研究所で 1960 年代から 70 年代にかけて開発された。

Z

形式仕様記述言語のひとつ。公理的集合論とラムダ計算、一階述語論理で使われる標準的な数学記法に基づいている。1977 年に Jean-Raymond Abrial が開発した。

■戦略プロポーザル作成メンバー■

勝山光太郎	フェロー	(電子情報通信ユニット)
服部隆志	フェロー	(電子情報通信ユニット)
加藤孝久	フェロー	(環境技術ユニット)
南谷 崇	フェロー	(電子情報通信ユニット)
丹羽邦彦	上席フェロー	(電子情報通信ユニット)

※お問い合わせ等は下記ユニットまでお願いします。

CRDS-FY2009-SP-12

戦略プログラム

情報システムに対する要求仕様の変化に対応する ソフトウェア技術

STRATEGIC PROGRAM

Software technologies for adapting requirement changes of
information systems

平成 22 年 3 月 March 2010

独立行政法人科学技術振興機構 研究開発戦略センター 電子情報通信ユニット
Center for Research and Development Strategy
Japan Science and Technology Agency

〒 102-0084 東京都千代田区二番町 3 番地

電 話 03-5214-7484

ファックス 03-5214-7385

<http://crds.jst.go.jp/>

© 2010 JST/CRDS

許可無く複写／複製することを禁じます。

引用を行う際は、必ず出典を記述願います。

No part of this publication may be reproduced, copied, transmitted or translated without written permission.

Application should be sent to crds@jst.go.jp. Any quotations must be appropriately acknowledged.
