

解説



アジア・太平洋におけるソフトウェア技術

4. 中国におけるソフトウェア工学
—過去、現在、そして未来—†

Yu Chen †† Meiqi Fang ††

(翻訳編集:菅沼 明††)

中国におけるコンピュータサイエンスとアプリケーションは1950年代中ごろに始まっている。しかしながら、いろいろな理由によりソフトウェアの開発は1970年代の終わりまで長い間無視されてきた。その時以来、中国の科学者と技術者はこの分野で多くの結果を輩出した。構造化技術とソフトウェアエンジニアリングの方法論を試した後、オブジェクト指向とCASEツールがこの分野の最も活発に議論のある話題になり、多くの研究と製品がそれらに関わっている。多くの情報や外国との個人的な交流により、この分野の先進国と中国とのギャップは以前より小さくなっている。中国でのソフトウェアエンジニアリングとソフトウェア産業の現状を細部にわたって紹介する。その後、短期の展望と長期の展望との観点からソフトウェアエンジニアリングの発達に関する所見と戦略的提案について述べる。複雑なシステムとしてソフトウェアの全体論的な見方を強調する。

1. 歴史の概況

正式に言うと、中国で最初にコンピュータが作られたのは1950年代の中ごろである。1956年に中国政府はコンピュータを含む先進技術の新しい分野の開発を目指した。多くの優秀な技術者と学生が中国科学院と大学から選ばれ、計算技術(computing technology)と名付けられた組織を起こした。これは、他の国と比べて遅くはない。

当時コンピュータは計算の道具であると考えられており、複雑な数学の等式を解くことにしか使われていなかったことは残念なことである。それゆえ、コンピュータは科学的な研究分野でしか使

われていなかった。当時の政局の結果として、コンピュータは主に軍事目的に使用されていた。上と同じ理由で、技術はソビエト連邦から主に伝えられた。たとえば、ソフトウェア技術は「自動プログラミング」と呼ばれ、事業におけるコンピュータの応用は「自動マネジメント」と呼ばれた(ロシア語ではACY)。西洋諸国との交流はほとんどなかった。この環境に限られていたので、もちろん、西洋諸国とのやりとりは不可能であった。1986年、30年祭の会議の席で30年の歴史を遡ってみる際に、中国政府の高官は以下のように述べた。「ここ20年間で学んだことは、われわれがコンピュータの製造ばかりに留意して、アプリケーションに関しては無視していたことである。つまり、ハードウェアばかりに気をとられて、ソフトウェアには無関心であったと言える。」それゆえ、中国には1978年まで市民が使うようなアプリケーションはほとんど存在していなかった。中国はこの政策のために20年間を浪費した。一方、他の国々ではこの20年間の間に発展が絶え間なく続いていた。

中国において、コンピュータ以外の分野の転換点は1970年代の終わりである。開放政策によって、中国は全世界に対して門戸を開いた。アメリカやヨーロッパ、日本から新しい技術をともなってコンピュータが中国に入ってきて、新しい世代の技術者を養成するためにすぐに使われた。現在、この開放政策はすべての人に対して影響を与えている。たとえば、MISやDSS、DOS、UNIX、その他の関連した話題は中国のエンジニアと学生の間でよく知られるようになった。中国と先進諸国とのギャップは以前よりも小さくなっている。

このように、歴史は二つの期間に分けることができる。1978年以前:孤立、コンピュータを市民が使うことはほとんどない、ソフトウェアは無視、

† Software Engineering in China—Past, Present and Future— by Yu CHEN and Meiqi FANG (School of Information, Renmin University of China).

†† 中央人民大学
††† 九州大学工学部

1978年以降：世界の技術の流れの共有，コンピュータを市民が使用することに留意する，ソフトウェアに留意する。

ソフトウェアエンジニアリングに関する本当の開始点は1978年である。初めの時期には、ほとんどのアプリケーションは、個々のプログラマが研究や軍事的な目的のために開発していた。そのために、ソフトウェアエンジニアリングはまったく存在しなかった。

2. 最近15年間の発達

1978年以来、この分野では多くの変化が起きている。この変化を個々の観点から以下で述べる。

a. ハードウェア

1978年当時、中国にコンピュータはほんの少ししかなかった。統計によると、ほんの数百台であった。それらは、中国製もしくは共産圏の国（ルーマニアと東ドイツ）製のものであった。概して、それらはトランジスタを基本とし、パンチカードを使用する第二世代のコンピュータであった。これらのコンピュータは軍事関係の省庁といくつかの特別な機関（石油製造と天気予報）で使われた。

唯一の例外は中国銀行である。外国の銀行とつなぐ必要があるために、1974年に西洋諸国製のコンピュータを使い始めた。また香港支店ではさらに早くからIBMのマシンを使用していた。

1979年にいくつかの大企業はIBMやCDC、DGなどからメインフレームを購入した。HPやHoneywell、Burroughs、DECもまたソフトウェア部門で中国のコンピュータ市場に参入した。1980年以降にマイクロコンピュータが中国に入ってきた。当初は、CROMEMCOとApple IIであったが、その後IBM PCやPC/XT、PC/ATなどになる。1985年の終わりには、中国のコンピュータは130,000台を越えた。そのほとんどはマイクロコンピュータであり、ミニコンや中型機、大型機、メインフレームはおおよそ9,000台であった。

1986年以降、いろいろな理由で発展のスピードは1980年代の始めのころより遅くなった。しかしながら、1990年代に入るとコンピュータに対する興味が再燃した。これには二つの要因がある。一つは、パソコンの価格がますます下降して、多くの家庭で自分自身または子供のためにコンピュータを購入できるようになったことである。もう一

つは通信の容易さを直接活用できるようになったことである。情報ハイウエーやEDI、EMAIL、公的な情報サービスが中国の学識的な団体の間に現れた。いわゆる情報革命にだれもが直接影響を受けた。政府と企業は情報産業にさらに留意し、投資を行った。1978年以来の第二の大きな好機であったと言われている。中国の情報産業はまさに新しい潮流に乗ろうとしている。

b. 言語

上記のハードウェアのもとで、中国のソフトウェアもここ15年間で大きく変化した。1978年に、中国で最もよく使われていた言語はALGOLといわゆるBCY（ALGOLの簡易版中国バージョン）であり、FORTRANが中国の大学などの機関に紹介され始めた程度であった。たいいていの技術者はCOBOLを知らなかった。その後すぐに、技術者を養成する大学でPASCALの講習が始まり、COBOLが商業学校と多くの経営者養成課程で使われるようになった。PASCALは今日でも使われている。しかしながら、COBOLへの関心はすぐに冷めた。その理由は、COBOLのプログラム中に多くの英単語があり、技術者ほどは英語に親しんでいない中国の経営者にとって難しかったためであろう。

1980年代の中ごろに、中国でADAを紹介する大学や企業もあった（主に軍事関係の省庁において）。しかし、この動きも成功しなかった。今日、中国で最も広く使われている言語は以下の二つのタイプの言語である。第一のグループはdBASE IIIやFoxBASE、FoxProである。これらは、「使いやすい」とパソコンの上で動作することが理由で、エンドユーザに広く使われている。中国の技術者や学生の90%以上はまったくメインフレームを使用したことがない。もし、パソコンの環境で快適には動作しないソフトウェア（言語やDBMS）があれば、それは少なくともここ5年間では中国では民間に広まらなかった。それがパワフルであろうが洗練されたものであろうが、中国では大きな市場を見つけることができない。第二のグループはCとC++である。これらは、能力と互換性のためにソフトウェア技術者の間で広く使われている。CまたはC++を使うことで、有能なプログラマはパソコン上でやりたいことはなんでもできる。Windowsの中国バージョンはより

大きな能力を提供した。もちろん技術的観点から、CやC++は第一のグループより複雑である。しかしながら15年間のプログラミングの経験を経て、中国のプログラマは技術的問題を詳しく述べることができるほど優秀になった。今日の中国には難しいコーディングの仕事をごこなす優秀なプログラマがたくさんいると信じている。アメリカやヨーロッパで立派に活躍している多くの中国人学生がその明白な証拠である。彼らは中国国内で基本的な養成を受けたからである。

c. 方法論

今日、われわれはソフトウェアエンジニアリングと組織化のための方法論、ソフトウェア開発のコントロールに着手している。前述のように、1978年にわれわれは方法論をまったくもっていなかった。計算をすることしか職務としない個々のプログラマにとって方法論が重要でないことはもっともである。応用システムがビジネス環境で製品として使われるようになると、ソフトウェアの質や生産コスト、生産のタイムスケジュールは本気で考えられるであろう。それゆえ、それらの問題を1980年代の初めから考え始めた。

最初に選んだ手法は構造化解析と構造化設計であった。1978年にわれわれの大学(Renmin 大学)で構造化解析と構造化設計の講義を開講しはじめた。われわれはその手法をYourdonとJacksonから学んだ。うまく構造化されたプログラムを設計し、プログラミングのために良いスタイルに仕立てようとした。1982年から1985年までわれわれの学科では構造化プログラミングという名前の特別講義を開講していた。それは流行の始まりであり、別の大学も後を追った。しかし実際のアプリケーションの開発において、要求は予想されたほどは明確ではないし、タイムスケジュールも遅れ、プログラムの質は満足させられない、そして最終的に予算はいつも超過していた。われわれは新しいツールと技術を求めなければならなかった。

2番目に選んだ手法はソフトウェア開発手法である。われわれはライフサイクルやマイルストーン、チェックポイント、アルファ・ベータテスト法、ドキュメント作成の基準化などを紹介した。IBMのAD/Cycleも中国に紹介された。講義のみでなく、ソフトウェアエンジニアリングの研究のための特別機関を設ける大学も現れた(たとえば、

武漢大学)。1985年以来、多くの大学はソフトウェアエンジニアリングの講義を開講した。またソフトウェア会社には、生産過程においてこの方法論を用いようとするものもあった。しかしながら、結果は予想したほどは良くなかった。予算はまだ超過し、スケジュールはますます遅れた。

1990年代に入って、新しいツールが中国に紹介された。それはオブジェクト指向の方法論である。紹介された当初は、プログラミングのツールであり、SMALLTALKやSIMULA IIといった特別な言語に関連付けられたOOPであったが、その後、オブジェクト指向データベース管理システム(OODBMS)やオブジェクト指向分析(OOA)、オブジェクト指向設計(OOD)へと拡大していった。抽象化やカプセル化、継承、メッセージセンディングを用いることによって、ソフトウェア開発の良い方法が見つかり、ソフトウェアの再利用や、J. Martinが想像したようなソフトウェア工場、リエンジニアリング、リバースエンジニアリングを実現できるであろうとして期待された。学術的にみると、その分野のたくさんの研究や論文がある。しかしながら、それら新しいツールの実用上の利益を証明する強力な証拠は存在しない。YourdonやBoochなどの本はすぐに中国に紹介された。3カ月ほど遅れただけである。しかしながら、実際に使ってみると予想したほどではなかった。中国には有能なプログラマがたくさんいる。しかし、良いソフトウェア製品は少ない。

d. 環境とツール

CASEツールや開発環境の概念は1980年代の中ごろに中国に紹介された。前述のように、ソフトウェアの開発中に見つけられた困難に打ち克つために、いくつかの方法論が試された。そして、関連のあるCASEツールも紹介され、実際に使うために発展していった。当然、最初のころは、図やドキュメントを作成してアナリストやプログラマを補助するために、すでにあるソフトウェアを使用していただけである。たとえば、フローチャートやデータフローダイアグラムを作成するのにAUTOCADを使用し、ドキュメントを書くためにWORDSTARを使用し、データ辞書(data dictionary)を構築するためにdBASE IIIを使用するなどである。これは本当のCASEツールではない。

次の段階は特別な目的をもったツールを開発することである。たとえば、清華大学で開発された AUTO-dBASE III は dBASE III のためのコードジェネレータである。実際に、それは dBASE III 上のシェルであり、非手続き的なインタフェースを提供する。そしてそれは dBASE III の(対話画面や報告書の形式、メニュー画面のための)コードを生成するためにいくつかのパラメータを受け取る。この段階ではツールは孤立しており、いくつかの仕事を補助するだけである。それらの中には何も関係はないし、一貫性検査もない。それらのほとんどは、コーディングの際に使用されるだけであり、いわゆる下流工程の CASE である。

1990 年代になると、新しい開発は二つの方向を向いていた。一つは統合的 CASE (I-CASE) である。北京大学には、政府が援助したプロジェクトが存在する。その目的はソフトウェア開発のプロセスをすべてにわたって支援し、ツール内で扱うすべての情報の一貫性を保つような一連のツールを開発することである。それは UNIX 上で開発されている。Renmin 大学のわれわれのグループも I-CASE 上で開発を行っている。われわれの生産物は DOS や Windows 環境上で動作し、オブジェクト指向のアプローチを採用している。われわれのプロジェクトは現在データテスト中である。しかし、それらすべてのプロジェクトは現在進行中である。CASE ツールの本当の市場は、ORACLE, SYBASE, SSA, PowerBuilder, UniFace, PowerHouse などの外国企業にまだ占有されている。

3. 動向と風潮

中国の多くの研究者と技術者を引き付ける人気のある話題がいくつかある。以下に示すものがその中でも特に人気がある。

A. CASE ツールはまだ人気のある分野の一つである。多くのグループがその分野で仕事をしたり、独自の製品を開発したり、外国のツールを紹介したりしている。大学や研究所での興味の対象は、下流工程の CASE から、システム分析とシステム設計をうまく補助する上流工程の CASE に移っている。バージョンコントロールやクオリティコントロール、テストの組織化といったプランニングツール (IBM の言葉) が考慮されている。

これらの話題は長い間無視されていた。2, 3 年後には、中国語の文字を扱える CASE ツールがいくつか市場に現れるであろう。それは中国のソフトウェア産業を劇的に押し進めるであろう。

B. オブジェクト指向のアプローチはまだ興味のある話題である。中国政府はオブジェクト指向の方法論を用いた独自のオペレーティングシステムとデータベース管理システムを開発するための計画を立てている。オブジェクト指向はすでに学部大学生と大学院生の授業になっている。この話題の教科書が数多く出版された。C++ や SMALLTALK のような特殊な言語を使用するにも関わらず、方法論としてコンピュータアプリケーションの多くの分野ですでに取り入れられている。

C. Nolan モデルが示すように、高いレベルの処理は意思決定支援システム (DSS) に向くであろう。中国では、生成段階は簡単なデータプロセッシングに基づいている。先進地域と特別な分野では、DSS は、特に高水準の方針分析と意思決定において、すでに考慮されている。このことは、高水準の方針決定では、コンピュータをコンサルティングツールとして使い始めていることを意味する。この分野のいくつかのシステムが開発中である。プロトタイプまたは試用バージョンはすぐに完成するであろう。この種のプロジェクトの開発については、ソフトウェアエンジニアリングの技術が実際に使用され、成功している。

D. 人工知能 (AI) から始まる不完全な情報の処理と曖昧な情報の処理が中国で活発に議論のある話題となってきた。もちろん、この分野では、根本的な多くの困難がまだ残っている。それゆえこの実際の価値はまだ議論の最中である。ソフトウェアエンジニアリングの観点から、それは特別な分野の知識に基づいた別の種類のソフトウェアである。それらがもつ困難は知識の習得と知識の構築に関わりがある。もちろん、エキスパートシステムのシェルもこの分野で議論されているべき特別なソフトウェア製品である。

E. マルチメディアも活発に議論のある話題である。別の国のように CD-ROM やその他の媒体を使用して音声と画像、動画をコンピュータで扱い、テキストファイルと同じように処理する。この種のソフトウェア製品を開発するために、多く

の新しいパッケージとツールを次々と使用していく。その中には独立したパッケージやソフトウェアもあれば、図や音声、アイコン、その他部品を含んだライブラリであるものもある。それゆえこの分野のプログラマは芸術家や製本所とみなせるであろう。マルチメディアを使用することは、方法論と仕事のスタイルを含めて、プログラマに大きな影響を与えるであろうと信じている。

4. 障害と戦略

上で述べた簡単な紹介によって、中国には優秀なプログラマが数多くおり、ソフトウェアの分野の最新の話題のほとんどすべてに注意を払っていることは明白である。しかし、中国製のソフトウェアの開発は、他国の進歩から取り残されている。われわれは、以下の要因がこの疑問の解答のように思う。

A. 計画指向 vs 市場指向

ソフトウェアのすべての原理は要求に基礎を置いている。中国では今日まで、ほとんどの大きなプロジェクトは政府が決定していた。要求を明らかにすることをせず、ほとんど要求を変えることもなかった。予算は開発チームとは別の人がコントロールしていた。そして、ソフトウェアの品質や生産コスト、タイムスケジュールは開発チームの人とはまったく無関係であった。一言で言うと、ソフトウェアの品質を向上させるため、コストを削減するため、プロジェクトをスケジュールどおりに進めるための動機づけがなかった。新しい私企業の中にはその種の動機づけをもつものもあるが、大きなソフトウェアプロジェクトに投資するほどは力も強くなく、金銭的余裕もない。同じころ、中国は中央集権の計画経済から市場指向に向かおうとしているが、多くの不安定な要因は有能なプログラムチームの作成を不可能にしている。

B. 著作権

今のソフトウェアプロジェクトのほとんどはパソコン上で行われているために、それらを容易にコピーすることができる。一般的に言うと、中国人の収入は他の国と比べてまだ低いし、ソフトウェアの価格に比べても低い。それゆえ、多くの人や企業は倹約したが、法に従うしきたりはそんなに強くないので、ソフトウェアの不法コピーがしばしば起こる。その結果として、多くの技術者

は著作権は保護できないと思っており、別の仕事に移っていく。

C. ハードウェア環境

前述したように、投資は外部がコントロールしているため、ハードウェアがソフトウェア開発には古過ぎることがしばしばある。たとえば、多くのパソコンでは Windows が動かないし、UNIX の操作経験がある人はわずかである。計画立案システムのととでは、装備を新しくするために投資をするのに長い時間かかる。ときには、それは2、3年かかるであろう。急速に成長する分野において2、3年は長過ぎる。多くのプログラマは低いレベルのプラットフォーム (DOS 3.0 で PC 286) 上で仕事をしている。

まとめは明らかである。もし中国が独自のソフトウェア産業を発展させたいのなら、まず最初に以下のことをしなければならない。

- 中央集権計画から市場指向に再編し続け、投資のメカニズムを変え、「要求駆動」の原理を本格的に履行すること。
- 著作権を保護すること。
- ハードウェアを適時に更新すること。

中国は巨大な市場であり、ソフトウェア産業の生産者である。開放政策と政策の再編が続けば、中国は独自のソフトウェア産業を作り、近い将来、高い品質のソフトウェアを供給するであろう。

これらすべては、中国のソフトウェア工学の研究とソフトウェア産業にとって明るく有望な展望を開いている。われわれは二つの観点からそれらの展望 (短期の展望と長期の展望) について述べる。

5. 短期の展望

A. IAB や Apple, Microsoft, DEC, CDC, HP, 富士通, BULL などのような主な企業は中国に支店を設けているし、ハードウェアの製造・組立やソフトウェアの開発のために中国の大学や研究機関、コンピュータ製造業、ソフトウェア会社と契約を交わしている。このことは、中国のソフトウェア工学の研究およびソフトウェア産業の発達を速めるであろう。中国のソフトウェア産業が世界のソフトウェア産業の最も遅い発達と歩調をそろえるのはそう遠くない。

B. 中国でコンピュータソフトウェアの大衆化

に影響を与える大きな問題の一つは漢字を使用できるバージョンの問題である。漢字を扱えるようにすることは大きな労力が必要である。DOSやWindows, WORDSTAR, SAS, SPSSのような一般的なソフトウェア製品には漢字を扱えるバージョンがある。キーボードを使用して漢字を入力することは、この分野の賢人が請け負っている最優先のプロジェクトである。およそ400種類の漢字入力システムが紹介され、大衆に受け入れられるように競争している。それらのうちどれも他のものより優れているとは言えない。ひとたびあるシステムが優勢になり、大衆に受け入れられれば、世界の約4分の1の人口をもつ中国は世界最大のソフトウェア市場である事実からみて、それはソフトウェア分野での進歩とみなすことができる。

C. 上にあげたように、CASEツールの開発は中国でしばらくの間進んだ。CASEツールの開発は全世界においてソフトウェア産業の最初の成長分野である。経済的、技術的な制限のために中国はこの2~3年この分野の先端にはいないかもしれない。しかし、システム開発の自動化の要求が増えているので、外国製のみではなく中国で開発されたCASEツールの中国語バージョンが中国の市場に現れるであろう。

D. エキスパートシステムの基本的・実験的なバージョンが開発され、1980年代の初めから中国で使用され始めている。たとえば、簡単なエキスパートシステムが、伝統的な中国の薬物療法の分野での原因分析のために開発された。いくつかの実用の価値があるエキスパートシステムが開発され、中国でそう遠くない将来に実用に供するであろうと期待されている。

6. 長期の展望

ソフトウェアの生産性・信頼性・正確性の問題は、コンピュータがビジネス・科学・工学の目的のために実用規模で使用されるので、コンピュータ科学者と専門家を長く悩ませている。困難の元凶はシステムとしてのソフトウェアの複雑さにある。この問題に取り組むためにさまざまな試みと努力を行った。最初に構造化技術、次に情報工学(Information Engineering)、最後にオブジェクト指向とオブジェクト技術を試した。G. Boochの本「Object-Oriented Analysis and Excellent with

Application」¹⁾の最初の章で、複雑なソフトウェアの問題について簡潔で優れた説明を与えている。Boochが説明しているように、混沌から秩序への道はオブジェクト指向を通っている。

オブジェクト指向の出現で、実のある前進が起り、ソフトウェア産業に少なからぬ恩恵が得られている。しかし、実際のソフトウェアの生産においてオブジェクト指向によって十分に取組むことができない困難が横たわっていることが予知されている。システムがますます大きくかつ複雑になっている(たとえば膨大なファクタや複雑な関係を含んでいる)ので、膨大な量のファクタや膨大な関係をすべて理解することは、人間の能力をはるかに超えている(いわゆる複雑さを扱う人間の能力の限界)。ファクタの組合せ的爆発は困難をさらに悪化させている。われわれの見地では、ソフトウェア開発の方法論すべての弱点がそこにある。現在活発な議論のあるオブジェクト指向と並ぶ構造化技術の方法論的基礎は分解の原理、またはいわゆる「divide and conquer」の原理であり、システムの全体に対しては、ほとんど注意を払わないか、またはまったく注意を払わない。ここで、われわれは科学的な方法論におけるいわゆる全体と部分の長年の問題(R. N. Langloisの「Systems Theory, Knowledge, and the Social Sciences」, 特に参考文献2)の582~586ページを参照されたい)に再び出会う。Frederrick P. Brooksも彼の著書「No Silver Bullet」¹¹⁾においてこの問題を述べ、本質と偶然の違いに言及している。彼はソフトウェア開発にとって何が本来の難しさを指摘している。

およそ300年前から、人類は科学と技術の分野で進歩を続けてきた。Francis BaconやGalileo Gililei, Issac Newtonなどの偉大な観念学者は近代文明のための基礎を築いた。彼らの方法論の基本的なアイデアは、いわゆる復元の方法論または分解の方法論であり、複雑な物事を簡単に扱いやすい部分に分解することを強調している。有名な内科医I. Prigogineは彼の著書「Order out of Chaos」でこの歴史を振り返っている。彼は、世の中の一部と世の中の全体的な特徴を完全に説明でき、将来において世の中の振舞いが正確に決定できることを理解すればすぐに、この世の中は大きな機械(精巧な時計)として理解されると指摘し

た³⁾。他の多くの専門家たちもこのことを異なる見方で議論している。たとえば、P. Checkland⁴⁾やL. Bertalanffy⁵⁾、P. Feyerabend⁶⁾、I. Lakatos⁷⁾などである。同様に、ある問題はいくつかの低いレベルの理解しやすい部分に分解できる。それらの部分が理解されるとすぐに、全体の問題も解くことができる。この方法論はおよそ40年前から成功してきて、人類にたくさんの利益を与えた。

しかしながら、第2次世界大戦後ますます困難なことが現れた。この方法論は多くの新しい問題を解くには力不足であった。多くの科学者と哲学者が、この方法論は複雑なシステムの全体を無視していると指摘した。つまり、多くの場合に、複雑なシステムを形成するときに新しいいくつかの新しい特徴が加えられることである。そして、その新しい特徴は個々に理解したり、取り扱ったりはできないと指摘した。伝統的な方法論の別の弱点は客観的な見地である。成長の見地もまた無視されている。カオスとフラクタルの研究は自然の広く新しい分野があること、成長の見地または動的な見地から自然をみることを示した。Herbert Simon 教授が指摘したように、自然物は人工物よりパワフルであり、柔軟性に富んでいる⁸⁾。さらに、自然物は環境に適応でき、自分自身を容易に修正する。論理物理学者やシステム科学者から哲学者まで、ここ数年の間に多くの議論が行われた(参考文献3)、5)、6)~8)などを見て欲しい)。言及されなければならない興味深い傾向がある。それは、問題を解くのに完全に異なるアプローチがあり、かつ東洋哲学では全体は固有のものであるので、異なる分野の多くの専門家が東洋哲学からインスピレーションを探そうとしたことである。

ここで、ソフトウェア工学の方法論における全体と部分(全体論と個体主義)の問題に戻ろう。複雑なシステムをもっと十分に扱うことがさらに必要になっていることを考慮すると、伝統的な方法論で成し遂げられるものを基礎にして、システムの全体論的な見方に注意を向け、もっと注意深く考えるときがきたと、われわれは考えている。これは大きな秩序である。つまり、われわれはこの時期にすべての答えをもっているとは考えていない。次に述べることは、複雑なシステムの全体論的な見方の研究に関係があるかもしれないし、うまくいけば激しい議論に導くと思われる関連し

たポイントとしてわれわれが考慮することである。

第一に、われわれの見地で、伝統的な方法論による分析のための基礎のような階層制の原理は全体論的でもあり、個別主義的(分析的)でもある。階層制のそれぞれのレベルまたはレイヤ、サブシステムは全体であるという意味では、それは全体論的であり、階層制のそれぞれのレベルまたはレイヤ、サブシステムをより低いレベルまたは低いレイヤ、サブサブシステムに分解するという意味では個別主義的である。Booch が述べているように、「複雑なシステムの構造は、構成要素間の階層的な関係と同様にシステムの構成要素の機能で実現することが重要である(参考文献1)のp.72)」。ここで、「構成要素間の階層的な関係」は全体論的であると言うことができる。「システムによって付け加えられる価値は構成要素の関係から作られなければならない⁹⁾」。[付け加えられた価値]のアイデアは、システムの全体論的な見地では、「突発的な」特性または「突発的な」振舞いと呼ばれるものである。分解の過程において、システムをすっかり理解してしまうのはとても難しいので、それ以上分解できなくなるところがあるかもしれない。そのような場合、分解できるか否かは、全体論的にシステムをみて研究することである。要するに、階層制の概念は本質的にシステムの全体論的な見方である。もしわれわれがシステムを全体としてみないならば、階層制は存在しないからである。このように、階層制の見地もまた複雑なシステムを扱うための全体論的な方法であると言うことは正しい。

第二に、ニューラルネットワークの理論の基礎をなす学習と適合性のアイデアはシステムの全体論的な見地に応用することができる。システムが調整されなければならない、またはシステムの操作の過程において誤りが生じる場合、システムは学び、新しい状況に適応し、エラーを修正し、最終的にシステムを正しい道に戻ることができる程度に、システムは利口でなければならない。この順応性と自動修正の能力は、全体としてシステムの特性であり、システムの突発的な振舞いである(参考文献3)のp.583)、また、その能力はシステムの個別の部分果たす適当な機能に完全に依存しない。たとえば、人間は、体の一部が傷ついて

も家に帰れるし、鳥は、かぎづめを抜かれても飛ぶことができる。

第三に、成長のアイデアはシステムの全体論的見方である。発達の初期やプロトタイプから始めて、システムは全体として成長する。ユーザと製作者が共同して、関係者すべてが満足する最終形態にまでシステムをもっていく。システムの進化の途中で、始めに決めたことが正しくなくなるかもしれない。われわれがシステムの実際の振舞いを観察するので、修正や適合はシステムを構造から改善することになる。ここで、われわれは前節で述べた修正と適合のアイデアを再びみる。しかし、重要な区別が存在する、つまり、前節の修正と適合はシステムの内部的な振舞いであったのに対して、ここで述べる修正と適合は、たとえばそれを外部のエージェントをとおして行うように、システムの外部的な振舞いである。

第四に、複雑なシステムでカオスを扱うことで、カオスの理論は助けを乞うための最初の一つとなりうる。われわれは、ここで、この分野の研究の方向づけをするために基本的な問題を提示する。混沌としたすべてのシステムに基礎をなす秩序のパターンとしてアトラクタが現れるのなら、複雑なシステムとしてのソフトウェアのアトラクタは何であろうか。

今までに述べてきたことは、この章の最初に掲げた問題の適切な答えであるとは言えない。しかし、それはこの分野の将来の研究方向を暗示するものである。中国はコンピュータ科学者とソフトウェアの専門家の比較的大きなチームをもつ国であるので、世界におけるソフトウェア工学とソフトウェア産業の発展に寄与すると思っている。

参 考 文 献

- 1) Booch, G.: Objected-Oriented Analysis and Design with Application, 2nd ed., Benjamin Cummings (1994).
- 2) Machlup, F. and Mansfield, U. ed.: The Study of Information: Interdisciplinary Messages, John Wiley and Sons (1983).
- 3) Prigogine, I. and Stengers, I.: Order out from Chaos, Bantam Books (1984).
- 4) Checkland, P.: System Thinking, System Practice, John Wiley and Sons (1981).
- 5) Bertalanffy, L.: General System Theory (1973).

- 6) Feyerabend, P.: Against Method, 3rd ed., Verso (1992).
- 7) Lakatos, I. ed.: Criticism and the Growth of Knowledge, Cambridge University Press (1970).
- 8) Simon, H.: The Sciences of the Artificial, The MIT Press, 2nd ed. (1981).
- 9) Reichtin, E.: The Art of Systems Architecting, IEEE Spectrum, pp. 6-9 (Oct. 1992).
- 10) Herath, J.: Computer Architectures for Intelligent System, IEEE Computer (May 1992).
- 11) Frederrick P. Brooks, Jr.: No Silver Bullet: Essence and Accidents of Software Engineering, IEEE Computer, pp. 10-19 (Apr. 1987).

(平成 6 年 11 月 21 日受付)



Yu Chen

1944 年生。1965 年数学学士(首都師範大学)、1981 年コンピュータサイエンス修士(中国人民大学)。1986 年中国人民大学助教授、1990 年同教授、1992 年中国人民大学情報学部長。著書「Information System Analysis and Design」、「Dialog on System」など。主にソフトウェア工学や情報システム解析・設計の方法論、システムサイエンスに関する 50 編以上の論文がある(そのうち約 10 編は英文)。1988 年より IFIP WG 8.1(International Federation for Information Processing のワーキンググループ 8.1: 情報システムの理論的な解釈に取り組む WG) のメンバ。概念のモデル化および CASE ツール、GDSS, CSCW, カオス、フラクタルに興味を持つ。



Meiqi Fang

1942 年生。1965 年コンピュータサイエンス学士(清華大学)。1981 年コンピュータサイエンスと工学修士号(中国人民大学)。1986 年中国人民大学助教授、1993 年同教授。1986 年より中国人民大学情報学部情報システム工学研究グループディレクタ。著書「Computer Simulation」、「Decision Support System」など 3 冊。主にソフトウェア工学やコンピュータシミュレーション、システムサイエンスに関する 45 編以上の論文がある(そのうち約 10 編は英文)。概念のモデル化および CASE ツール、GDSS, CSCW, コンピュータシミュレーション、コンピュータネットワークとその応用、システムサイエンスに興味を持つ。