

# ソフトウェア工学の 40 年とこれから

玉井 哲雄†

## Four Decades of Software Engineering and its Future

TETSUO TAMAI†

1968 年にドイツのゲーミッシュで「ソフトウェア工学」をテーマとする NATO の会議が開かれてから、今年で 40 年を迎える。本講演ではこれを機会に、ソフトウェア工学の 40 年の歴史をざっと振り返り、また今後について若干の展望を行ってみたい。

### 40 年の回顧

ソフトウェア工学の 40 年を振り返る文章は、すでに「情報処理」の特集号のために書いてそれが印刷に回っており、本講演に前後して発行されるはずなので、講演を聴かれない方は興味があればそれを読んでいただきたい<sup>4)</sup>。以下はその文章の始めと終わりの部分を取り出して、それに手を加えたものである。

1968 年 10 月にドイツのゲーミッシュで Software Engineering という名を初めて冠した会議が開かれた。ソフトウェア工学の歴史はこのときから始まる。そして今年 (2008 年) はそれからちょうど 40 年。ソフトウェア工学も不惑の年を迎えたわけだ。

ゲーミッシュの会議は NATO の科学委員会が主催したもので、約 50 名の会議参加者はすべてヨーロッパと北米から呼び集められた。同じ NATO の会議は翌 1969 年 10 月にもイタリアのローマで開かれ、やはり西欧諸国から約 60 名が参加した。

ありがたいことに、この両会議の後に出版された 2 つの会議録が、pdf ファイルとして復刻され、それが両会議録の編集者の一人である Brian Randell のホームページ上で公開されている<sup>1),2)</sup>。現在の国際会議で通常出版される論文集は、事前に投稿され審査された論文を集めて編集したもののだが、この NATO 会議録

には論文形式で書かれた短い報告が一部として掲載されているものの、主要部分は会議の録音テープを起こして編集されたもので、それだけに当時の熱気をそのまま伝える優れた読み物となっている。とくに 1968 年の最初の会議は、会議としても成功し、編集もうまくいったと Randell が自賛するだけのことはある。

NATO 会議の精神を受け継いで、ソフトウェア工学の国際会議 (International Conference on Software Engineering, 通称 ICSE) が始まったのは 1975 年である。それ以来この分野で旗艦会議と位置づけられ、例年 5 月前後に開催されている。ゲーミッシュから 40 周年となる 2008 年は、恰もよく誕生の地ドイツに戻り、5 月中旬にライプツィヒで開かれた。その会議の初日、開会直後に「SE の 40 年」というセッションが企画され、会議録の編集者であった Peter Naur と Brian Randell が招待されて、元気な姿を見せた。また、この会議の計画の中心人物であった Friedrich L. Bauer は、84 歳の高齢ながらもまだ元気とのことであったが、さすがに今回は顔を見せなかった。しかし代わりに、当時の弟子でその後もずっと Bauer とともに活躍した Manfred Broy が登壇して、Naur, Randell 等とのパネル討論に参加した。

ゲーミッシュ以降のソフトウェア工学の動きをかいつまんで言えば、次のようになるだろう。1970 年代は構造化プログラミングから構造化分析・設計が発展した「構造化」の時代、1980 年代は管理技術にシフトした時代、1990 年代は「オブジェクト指向」の時代、そして 2000 年代は多様化の時代である。

### ソフトウェア工学の現在

21 世紀に入ってからのソフトウェア工学で目立つ動きは、プログラムや文書の分析技術の高度化であろう。

---

† 東京大学  
The University of Tokyo

プログラムの静的解析技術には長い歴史があるが、その手法はますます洗練化されてきている。さらに、モデル検査に代表されるようなモデルレベルの解析、プログラムの構造より振舞いの解析、という方向への進展も目覚ましい。UMLの普及によりUMLで記述されたモデルの解析も、適用範囲が大きく広がっている。

手法の進化はもちろんのこととして、このような分析技術の発達を助長する要因として、少なくとも2点を挙げることができるだろう。一つには、さまざまなオープンソース・ソフトウェアが公開されていることである。これは解析対象としうる膨大なデータを提供している。公開されているのは原始コードだけでなく、設計文書、メールのやり取り、バグ報告など多様である。これは定量的なデータ解析の適用を試みる研究者にとって、垂涎の宝庫といってよい。皮肉に言えば、学会論文や学位論文を量産するための恰好な材料ともいえる。

もう一つはハードウェアの性能の並はずれた向上である。ちょっと前ではお手上げだったような大量のログデータの解析も、簡単にできるようになった。モデル解析の問題を充足可能性問題(SAT)に帰着させて、汎用的なSAT解法器にかけて解くなどという強引な方法が通用するようになったのも、アルゴリズムの工夫だけでなくやはりハードウェアの速さの賜物である。

このような背景で、経験的(empirical)ソフトウェア工学が強調されるようになり、定量的なデータによってソフトウェア工学のプロセスや手法を評価しようという動きが盛んになった。一方で、プログラムの精密な解析というアプローチは、ソフトウェア工学をプログラミング言語分野に再び接近させる契機ともなっている。最近のICSEなどの論文を見ると、プログラミング言語関係の会議で発表されてもおかしくないようなものが少なからずある。

ソフトウェア工学誕生から40年経った現在、40年前にも言われた「現代社会の中心活動にソフトウェアの果たす役割が大きい」という事実は、ますます確固としたものとなっている。しかしそのソフトウェアは目に見えない。その上に、コンピュータそのものが見えなくなっている。現代のクルマには数10のコンピュータが載っているが、クルマのドライバーにそれが見えないのはもちろん、クルマの整備士もそれらを直接見ることはなく、コンピュータを組み込んだ部品ごと点検したり交換したりする。

そこでソフトウェア工学の果たす役割も、ますます増大する一方で、一般社会の目からは隠れてしまいがちである。ソフトウェアが新聞やTVに取り上げられ

るのは、空港のチェックイン・システムで障害が発生して8万人が影響を受けたとか、駅の自動改札機の障害で260万人の足が乱されたといった不祥事ばかりである。ソフトウェア工学のお陰でこれほど社会が助かっているというような話は、記事にならない。

しかし、1968年に提起されたソフトウェア工学の必要性が、過去のものとなったわけでは決していない。この40年の間にソフトウェア工学は着実な成果を上げてきてはいるが、たとえばハードウェアの生産性や性能の目覚ましい進歩に比べるとそれほど目立たないのは、同じ土俵の上で数値的な比較がしにくいとはいえ、アピール不足の感も否めない。その意味ではソフトウェア工学の当面の課題の一つは、いかに社会にその意義を認知させるかにあるといえるだろう。

### そしてこれから

ソフトウェア工学のこれからについては、やはり以前に別の文章を書いたことがある<sup>3)</sup>。そこで技術的な発展の可能性として挙げたのは、以下のような項目である。

- 大量なコンポーネントのもたらす複雑性への対処
- ソフトウェア開発の自動化
- 自律性をもち環境に適応して自己組織化を行うソフトウェアの実現
- 検証技術の発展

ここではこれらの詳細には立ち入らない。

さらにソフトウェア工学の理論の発展として、「連続時間を導入した計算理論」と「連続空間を導入した計算理論」を挙げておいた。これらは大風呂敷の類である。さらに、生物学、文献学、考古学、社会学、文化人類学など他分野との融合が進むことが予想される。

### 参 考 文 献

- 1) Buxton, J. N. and Randell, B.(eds.): *Software Engineering Techniques—Report on a conference sponsored by the NATO Science Committee*, Rome, Italy (1969). <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1969.PDF>.
- 2) Naur, P. and Randell, B.(eds.): *Software Engineering—Report on a conference sponsored by the NATO Science Committee*, Garimisch, Germany (1968). <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>.
- 3) 玉井哲雄: ソフトウェアエンジニアリングの発展シナリオ, *SEAMAIL*, Vol. 14, No. 9-10, pp. 1-7 (2006).
- 4) 玉井哲雄: ソフトウェア工学の40年, 情報処理, Vol. 49 (2008).