

CPS のソフトウェア工学:ソフトウェアは対象か手段か

中島 震

Software Engineering of CPS : Engineering for/with Software

Shin Nakajima

1. はじめに

Cyber-Physical Systems (CPS) は、NSF の H. Gill による造語である。自然、人工の違いを問わず、多数の構成要素からなる「システム」一般を論じるサイバネティクスに触発された。コンピューティングの技術、特にソフトウェア技術が、さまざまな応用分野に浸透していく。その時代にあって、その共通性として、サイバネティクスと同様に、フィードバックに注目する。特定の技術というよりも、パラダイムと理解できよう。

CPS は、その具体的な実現の形から、組込みシステムと関係が深い。欧州のフレームワーク・プログラム (FP) では、FP5 からの流れで、組込みシステムとしての見方が中心をなす。FP6 の ESD や Artemis, FP7 の Networked ESD, ドイツ Acatech の Agenda CPS を経た。2014 年から始まった Horizon2020 の中、Artemis の後継 ECSEL では、Smart CPS を研究開発支援のキーワードとする。ソフトウェア以外の技術も関わることから、CPS の特徴は multi-disciplinary と云われる。

本稿では、CPS とソフトウェア工学の関係について考えてみたい。

2. ソフトウェア工学再考

ソフトウェア工学の発端を振り返ってみよう。1960 年代後半、開発対象の大規模複雑化と共に、プログラミング中心の技術だけでは、開発に不十分なことが強く認識された。そこで議論されたのが、ソフトウェア開発への工学的アプローチの導入である。工学的アプローチとは、対象の問題を明らかにし、解決の標準的な方法・メソッドならびにツールを整備すること。基本的な教育を受け

た技術者が自ら問題解決できるようにする。これらは技術者の資格認定試験につながる。

工学のはじまりは、Civil Engineering と云われる。軍事 (Military) に対するもので、「民生工学」とも訳せる。ところが、日本では、導入時から「土木工学」と呼ぶ。民生の対象、道路・架橋・建造物などの実現法の多くが「土木」であったことが理由である。実際は、鉄骨・コンクリート等も使うが、今でも「土木工学」と呼ぶ。

ここに、工学あるいはエンジニアリングには、2つの意味があることに気づく。「何々工学」は、Engineering for X なのか、Engineering with X なのか。自動車工学や航空工学は for であろう。そこでは、イノバティブな X を得ることが目的であり、多様な手段を組み合わせる。本質的に、multi-disciplinary である。一方、電子工学や制御工学は with といって良い。X の科学的な知見に基づき、工学的な手段として有用な技術体系を得る。Civil Engineering は民間の為 (for) であるのに、土木という手段 (with) に「翻訳」された。これによって、何かを見失ったのではないだろうか。

さて、ソフトウェア工学は 2 つの見方が成り立つ。With としてのソフトウェア工学は、コンピューティング科学を基礎とし、ソフトウェア技術に基づくシステムを構築する方法の体系。生産性や信頼性の向上が主要な関心事である。一方、for としてのソフトウェア工学は、発想 (Imagination) を「ソフトウェアだからこそ」実現できるシステム化の技術を取り扱う。発想から導かれる要求仕様は性質・規則の集まりになり、A.Simon らの「人工物の科学 (Science of Artificial)」に基づく工学という側面も持つ。その最も基礎的な部分は形式科学とされる。ソフトウェアの特徴は外界と相互作用を持つことであり、形式科学は、その本質をとらえきれない。

ソフトウェア工学は、歴史的に、with から出発し、for に目を向けるようになった。With が中心の頃、何をつく

†国立情報学研究所

National Institute of Informatics

「ウィンターワークショップ 2015 イン宜野湾:テーマ T6」

るかは「外側」にあるシステム論の問題であるとした。要求仕様は与えられており、これを満たすソフトウェア・システムを開発する技術体系である。その後、要求獲得の重要性が認識され、for の比重が高まった。今、ソフトウェア技術が多様な応用領域で使われる時代が到来した。CPS は for を指向する自然な延長線上にある。

3. CPS 再考

3.1. CPS Flower

CPS をコンピューティング技術によるサイバネティックスの具体化と考える。要点は、フィードバックの考え方やコンピューティング科学の融合である。J. Wing による CPS Flower は、さまざまな応用の基礎となる CPS を 3 つの特徴からなると整理した。ハイブリッド、不確実さ、コネクティビティと云える。

3.2. ハイブリッド

離散量 (booleans) と連続量と (reals) という異なる性質を持つ。時間の取り扱いという観点で、リアルタイム・コンピューティングと関連が深い。With としては、ハイブリッド・システムの表現と性質解析の技術が中心となる。伝統的に、コンピューティング科学と制御理論という 2 つの分野で研究が進められた。Multi-disciplinary という CPS の特徴を端的に表すと云えよう。

ソフトウェア工学では、機能外要求 (non-functional requirements) に注目する。時間が影響する性質があり、ハイブリッド・システムとして論じることが可能な観点も多い。時間応答性、実行性能などに加えて、電力消費が最近の話題のひとつになっている。

3.3. 不確実さ

「不確実さ」は多様な観点から考えることができよう。コンピューティング基盤に近い層では、外部からの入力データ値や時間的なゆらぎに起因する不確実さがある。一方、for の観点では、フィードバックと密接に関連する興味深い問題を含む。

教科書的なソフトウェア開発は、要求仕様ありきで話が進む。一方、現実には、要求仕様を捉えきれないことが不具合につながる。完全に予測できないこと、つまり、不確実なことを前提としなければならないこともある。たとえば、自己適応ソフトウェア・システムでは、自身の状態と取り巻く外部環境から、次の振る舞いを決定する。フィードバックが基本である。

さて、通常、不確実さはリスク要因である。一方、不確実性は新しいチャンスと考えることもできる。いずれであ

っても、対象を取り巻く環境を把握する手段が必要になる。J. Stermann は、システム・ダイナミックスのモデリング手法を提案した。N. Leveson は、これを宇宙ロケットの安全性解析に用いた。

また、M. Jackson の問題フレーム (Problem Frames) では、不確実さが何処に入り込むのか、何が所与の条件で、何が制御可能なのか、を分析する。Stermann はフィードバックループの本質である動的な側面を重視する。Jackson は静的な関係への関心が強い。

3.4. コネクティビティ

インターネットの普及と共に、コンピュータをつなぐ情報伝送の技術から、「つながる」ことによって生じる人間の行動まで、多段「スタック」を形成するようになった。

CPS は、ドイツ発の Industrie 4.0、北米の Industrial Internet と関係が深い。CPS Agenda は Networked ESD の延長として CPS を見ていた。Industrial Internet では CPS Flower のように幅広い産業 (農業などを含む) の変革を目指す。サプライチェーンを超えたビジネス・エコシステムに注目する。「発想 (何を)」と、それを実現する「規則の体系 (広い意味でのプロトコル)」が中心になり、まさに、「for ソフトウェア」の技術を必要とする。

なお、IoT (Internet of Things) はネットワーク視点の用語であり、あくまでも「黒子」の技術と云えよう。一方、Industrial Internet には、一般の人をつなぐ現在のインターネット技術を次世代の産業に役立てたいというニュアンスが込められている。インターネットの単なる応用ではないし、IoT が実現すれば全てでもない。ビジネス・エコシステムの観点から、System-of-Systems (SoS) を見ていると云うべきだろう。

4. おわりに

本稿では、CPS を取り巻く動向を概観した。ソフトウェア工学との関係では、フィードバックに着目したモデリング手法と、そのモデリング結果を、with の手法に受け渡す方法の確立が今後の課題と考えている。

最後に、第2節の問いへの個人的な答を記す。For から with への翻訳は、ものづくりで大切な新しい発想を見失ったのではないか。そのかわりに、細かい改良の積み重ねを得た。これは、その後の産業発展につながった。一方で、with へのこだわりが強すぎると、世界で同時進行する変革から取り残されることになるかもしれない。つまり、「発想より改良」という風潮が、CPS あるいは Industrie4.0 で繰り返されることを危惧している。