AI-1-3

サイバーフィジカルシステムの数理

Towards Mathematical Foundation of Cyber-Physical Systems

田崎勇一
Yuichi Tazaki

名古屋大学 工学研究科
Nagoya University, Department of Engineering

1 背景

サイバーフィジカルシステム（Cyber-Physical Systems, CPS）とは，複数の計算機がネットワークにより結ばれ，物理系と相互作用するシステムの総称であり，ここ数年の間に主に米国を中心として大きな注目を集めている研究領域である．その潜在的な応用範囲は高信頼な医療機器，インテリジェントな交通システム，プロセス制御，アビオニクス，電力ネットワーク，分散ロボティクス等と多岐にわたる ([1]).

CPSはリアルタイム組み込みシステムやハイブリッドシステムなど，その前身である既存の研究領域と多くのコンセプトを共有する．このため，CPSの位置づけを明らかにするには周辺領域との差異や共通点を把握することが重要である．CPSの特徴や要求事項は様々な文献において無数に挙げられているが，その中から (筆者の興味にもとづき) 特にCPS固有のものを以下に抜粋する:

(1) 物理系 (physical element) と計算要素 (cyber element) との相互作用を意識する
(2) 物理系の空間的な広がりとネットワークトポロジを考慮する
(3) 大規模系へのスケーラビリティを有する
(4) 予測可能性と構造的適応性を有する

(1) に関して，組み込みシステムの理論は物理系と組み込み制御系が成す閉ループの挙動を陽に考慮しないという点でCPSと異なる．(2) における「空間的」とは我々の住む物理空間の（2/3次元的）広がりである．例えばハイブリッドシステム理論ではシステムは抽象化された n 次元状態空間上で記述されるが，CPSではあくまで物理空間上に分布する連続体としてプラントを意識する．制御系に対しても，複数の小さなノードが物理空間上に分布し，ネットワークを構成するという構造を陽に意識する．(3) は，CPSの応用領域がいずれも家庭規模から地域規模，地球的規模のシステムを想定していることに由来する．ただし (4) にあるように，単に大規模化が可能なのではなく，同時にシステム全体の挙動が予測可能 (Predictable) でなければならない．さらに，根本的に予見できない事象に対する適応性も兼ね備えていなければならない．具体的には，センサノードの故障時のネットワークの再構成などである．

現在までのCPSに関する研究動向の多くは応用重視であり，CPSに特化した基礎理論の構築において目覚しい進展は見られない．しかしながら，その中にもいくつか潜在的に有用と思われる研究成果が表れ始めている．本稿ではそれらのいくつかに注目し，CPSの数理の有り様を模索する．

2 CPSの数理モデル

CPSを構成する個々のネットワークノードに割り当てられるリソース (使用可能電力や通信帯域) は非常に限られている場合が多い．このような要求から，計算コスト，通信コストを最低限に抑えられるイベント駆動型もモデルがCPSに有効であると考えられている．文献 [3], [4] において提案されたCPS Event Modelでは，識別子と生起時刻の対からなる従来のイベントに対し，ユークリッド空間上の事象の生起座標と事象の観測者の情報を加えたCPSイベントが定義されている．従来のイベント駆動システムの理論では，イベントの生起順序 (＝時間軸上の前後関係) に基づく因果関係の解析が行われてきた ([5]) が，生起時刻のみならず生起座標を含めた拡張情報を元に，事象の何らかの前後関係を議論することで新たな解析論が導かれる可能性がある．

3 CPSの設計論

冒頭で述べたように，CPSの応用分野では非常に大規模な系への適用可能性と同時に高い信頼性が求められる．現在，大規模システム開発の効率化のためにモデルベース開発 (Model-Based Development, MBD) が提唱され，広く利用されている．MBDはプラントの数理モデルを用いて制御系設計や制御コード・テストコードの生成を自動化する一連の技術である．モデルベース開発において利用されるモデル記述形式は多様であるが，CPSのように複数の異なる形式の構成要素が階層的に連なるシステムを統一的に表現するには，それに適したアーキテクチャ記述言語 (Architecture Discription Language, ADL) を用いることが重要である．各種ADLの分類・比較については，最新ではないが文献 [9] が詳しい．現時点で，CPSへの応用を陽に意識して設計された言語は現れていない．多くのアーキテクチャ記述言語は，文字通りシステムの構造（コンポーネントとそれらの接続関係）を記述することを目的に設計されており，それ自体はシステムの性能や安全性に関して何らかの保証を与えるものではない．実際，アーキテクチャ記述言語で文法上適切に記述されたシステムであっても，それが要求仕様を満たすかどうかは別の手法により検証する必要がある．例えば，リアルタイム組み込みシステムの記述言語であるAADL(Architecture Analysis and Design Language, [8]) で記述されたモデルに対して，モデル検査を行う試みがなされている ([10]).

CPSが対象とするような大規模システムでは，シミュ

レーションやモデル検査の計算複雑性のために，システムを実装した後にテスト・検証するという現状の方法論自体が通用しなくなると考えられる．その一例として，文献 [6] は CPS の大きな応用領域の一つである航空機開発について述べており，MBD は個々の構成要素 (コンポーネント) の開発の効率化には寄与するが，開発後期におけるシステムの統合の効率化には寄与しないという問題点を指摘している．遅れて発見される不具合ほどその修正に膨大なコストを要するため，これらの潜在的不具合を初期のコンポーネント開発の段階で発見し，除去することが求められている．このことから，個々の分野に特化した (domain-specific な) 現状の MBD だけでなく，複数のドメインにまたがったシステム統合をも視野に入れたモデルベース開発の方法論が求められている．

システム統合において生じる問題の多くは，それぞれのコンポーネントが正常に動作するための周囲環境に関する条件が正確に仕様化されていないことに起因する．そのため，各コンポーネントが暗に想定する所定の動作条件への依存性およびそれが破られることで生じる不具合は個別テストでは発見されにくく，システム統合時に初めて露見するケースが少なくない．この状況を改善するには，コンポーネントの実装を開始する前に，それらが想定する動作条件や他のコンポーネントとやりとりされる情報 (＝インタフェース) の仕様を，開発初期段階において漏らさず仕様化しなければならない ([7])．この際，単に仕様書を作成するのではなく，前述のアーキテクチャ記述言語の書式に従い仕様を記述することが重要である．そうすることにより，コンポーネント間のインタフェースにおいて仕様が満たされているかを機械的にチェックすることが可能となり，その結果システム統合時の不具合の発生率を劇的に低減できることが期待される．欧州における SPEEDS プロジェクト ([11]) で策定されたコンポーネントベースの組み込みシステム記述様式である HRC(Heterogeneous Rich Components) は，コンポーネント間インタフェースを契約 (Contract) と呼ばれる形式で記述する．契約は想定 (Assumption) と保証 (Guarantee, Promise) の対からなる．各コンポーネントの実装が契約を満たし，かつコンポーネント間インタフェースにおいて所定の互換性 (Compatibility) 条件が満足されればシステム全体で仕様が満たされることが保証される．Assume/Guarantee の数理については [12] を参照されたい．契約のコンセプトに基づくコンポーネントベース設計は CPS の理論的基礎を与え得ると期待されるが，CPS において焦点となるリアルタイム性や物理系との相互作用に適した契約の数学的表現方法は明らかになっていない．また，現状では各コンポーネントの契約を与え，それを元にシステム全体の契約を導出するというボトムアップアプローチを採っているため，システムの契約が組み合わせ論的に複雑化し，それが要求仕様を満たすかどうかの検証のための計算コストが膨大になるという根本的な問題がある．今後は，逆にシステム全体に課せられた仕様を分解し，各コンポーネントが満たすべき契約を割り当てていくトップダウンアプローチを模索する必要があると考えられる．

4 まとめと今後の展望

サイバーフィジカルシステムは未だ黎明期にあり，確固たる理論は未だ現れていない．CPS の数理モデルとしては，従来のイベント駆動モデルに事象生起座標と観測者情報を加えた拡張モデルが提案されている．また，スケーラブルで高信頼な CPS の設計論には，システムの構造のみならずコンポーネント間インタフェースの仕様をも表現できるアーキテクチャ記述言語が必要である．これには，インタフェースの記述様式や，複数のコンポーネントの結合や分解などの操作とインタフェースとを結びつける数学的枠組みの整備が不可欠であると考えられる．コンポーネントベース開発はソフトウェア工学で生まれた概念であるが，リアルタイムソフトウェアと物理系が相互作用する CPS にも自然に拡張可能であり，またそこに CPS 理論の独自性が生まれる期待がある．

参考文献

[1] E. A. Lee; Cyber Physical Systems: Design Challenges, 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), pp.363-369, 2008.

[2] T. A. Henzinger, J. Sifakis; The Embedded Systems Design Challenge, Formal Methods, Vol.4085 in LNCS, pp.1-15, Springer, 2006.

[3] Y. Tan, M. C. Vuran, S. Goddard; Spatio-Temporal Event Model for Cyber-Physical Systems, 29th IEEE International Conference on Distributed Computing Systems Workshops, pp.44-50, 2009.

[4] Y. Tan, M. C. Vuran, S. Goddard; A Concept Lattice-Based Event Model for Cyber-Physical Systems, Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems, pp.50-69, 2010.

[5] C. Talcott; Cyber Physical Systems and Events, Software Intensive Systems and New Paradigms, Vol.5380 in LNCS, pp.101-115, Springer, 2008.

[6] D. Redman, D. Ward, J. Chilenski, G. Pollari; Virtual Integration for Improved System Design, 1st Analytic Virtual Integration of Cyber-Physical Systems Workshop (AVICPS2010), pp.57-64, 2010.

[7] L. Sha, J. Meseguer; Analytical System Composition, 1st Analytic Virtual Integration of Cyber-Physical Systems Workshop (AVICPS2010), pp.17-22, 2010.

[8] Architecture Analysis and Design Language, www.aadl.info.

[9] N. Medvidovic, R. N. Taylor; A Classification and Comparison Framework for Software Architecture Description Languages, IEEE Transactions on Software Engineering, Vol.26, No.1, pp.70-93, 2000.

[10] C. Ölveczky, A. Boronat, J. Meseguer; Formal Semantics and Analysis of Behavioral AADL Models in Real-Time Maude, Formal Techniques for Distributed Systems, Vol.6117 in LNCS, pp.47-62, Springer, 2010.

[11] SPEEDS: European Union 6th Framework Project in Embedded Systems Development, www.speeds.eu.com, May 2006 - April 2010.

[12] L. Benvenuti, A. Ferrari, E. Mazzi, L. Sangiovanni Vincentelli; Contract-Based Design for Computational and Verification of a Closed-Loop Hybrid System, Hybrid Systems: Computation and Control, Vol.4981 in LNCS, pp.58-71, Springer, 2008.