コンテキスト指向ソフトウェア開発方法論 RT-COM 構想の概要

渡辺晴美†1 今村誠†1 海老原秀亮†1 小倉 信彦†2 久住憲嗣†3 菅谷みどり†4

本稿では，RT-COM と呼ぶ，環境に応じて振る舞いを切り替えるような横断的関心事に効果的な開発方法論の構想について概説する．横断的関心事は，環境に応じたサービスの提供，非正常系対処の両方の課題と関係することから，自動運転等の新しい組込みシステムと従来システムの両方の開発に効果が期待できる．

1. はじめに

様々な環境に応じたサービスの提供は，IoT，Industry4.0，スマートロボット，自動運転等の新しい組込みシステムの特徴の一つである．従来の組込みシステムにおいても，様々な環境を非機能要件あるいは非正常形として扱ってきた．これらの要件がソフトウェアを複雑にし，理解性を損ね，保守・テストを難しくしてきた．

何か一つの異常信号を受信した場合，振る舞いの変化は受信したモジュールのみに留まらず，他のモジュールに影響を及ぼす場合が多々ある．例えば，自動走行しているロボットが，カメラと距離計の両方で障害物を探知している際に，カメラの信号が途絶えたとする．この場合，カメラのモジュールのみの処理が変わるのではなく，障害物の探知方法も変わり，走行スピードを遅くすることになる．すなわち，一つの信号が，横断的に複数のモジュールに影響を与えることになる．このような事柄は，横断的関心事と呼ばれ，ソフトウェアを複雑にする事柄として知られている．

新しい組込みシステムにおいて，環境に応じたサービスの提供は，非正常系以上に複雑な横断的関心事といえる．横断的関心事を解決する方法として，コンテキスト指向ソフトウェア開発技術がある．本技術により，オブジェクトに横断的なサービスや非正常系を扱いやすくできる．すなわち，「カメラが壊れた場合のサービス」という単位で記述，保守，テストが可能になるということである．以下，コンテキスト指向ソフトウェア開発技術，我々が取り組んでいる RT-COM について概説する．

2. コンテキスト指向ソフトウェア技術

コンテキスト指向ソフトウェア開発技術(COS)とは，コンテキスト指向プログラミング技術(Context-Oriented Programming: COP)[1][2]の概念を取り入れたソフトウェア開発技術である．COP の多くは，環境の変化に応じて，振る舞いを変化させるために，レイヤと呼ぶ概念を取り入れている．図 1 に示すとおり，レイヤは，環境に応じたサービスや非正常系をレイヤで管理する．レイヤは複数のオブジェクトから構成される．図 1 は自動掃除機の例である．自動掃除機は，基盤レイヤにあるとおり，走行，位置獲得，掃除の 3 種類のオブジェクトからなり，複数台で掃除を行う場合は，協調レイヤを活性化し，基盤レイヤに上書きする．この図では，走行と掃除が，協調走行と協調掃除に書き換わる．このように，環境（コンテキスト）に応じて，複数のオブジェクトに対し横断的に振る舞いを変更することが可能である．

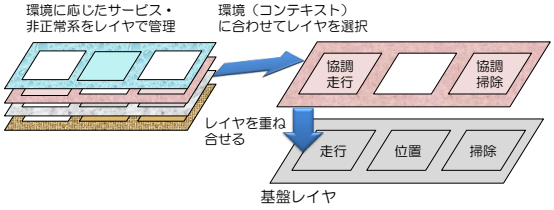

図 1 コンテキスト指向ソフトウェアの概要

3. 組込みシステムのための COS

様々な環境に応じたサービスの提供，あるいは，横断的関心事となる非機能要件の問題を緩和するために，我々は RT-COA(Real-Time Context-Oriented Architecture)[3]とよぶコンテキスト指向ソフトウェアのアーキテクチャに基づいた RT-COM(Real-Time Context-Oriented Methodology)[4]と呼ぶ開発方法論に取り組んでいる．以下，RT-COA および RT-COM の概要について述べる．

3.1. RT-COA

(1) 導入容易性： 既存資産の問題，新規パラダイムの学習コストを軽減するために，コンテキスト判断部分，すなわち環境を識別する部分と，サービス提供部分を分離する．図 2 では，Layer Manager と Layers を分離し

†1 東海大学, †2 東京都市大学, †3 九州大学, †4 芝浦工業大学

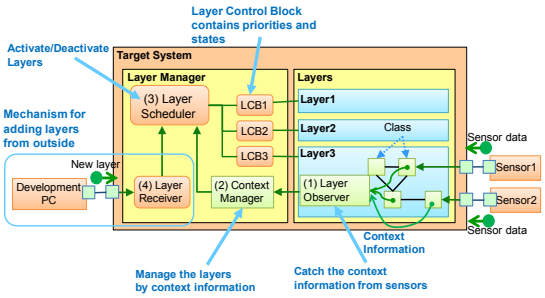

図 2　RT-COA の構成

ている．分離を可能にするために，図 2 の(1) Layer Observer により，各レイヤ内で発生したコンテキストの変化を捉え，(2) Context Manager で判断をする．開発者は Layer Observer と Context Manager を記述する必要があるが，レイヤのサービス部分の記述と分離できる．この分離により既存資産を利用可能にし，開発者の役割も分担できる．

(2) 安全なサービスの提供：資源競合，実行順序の問題を扱うために，OS のスレッドと同様に，レイヤを状態管理する．図 2 の(3) Layer Scheduler で行う．Layer Control Block がレイヤのメタ情報を保有し，どのレイヤを活性化するかを決定する．

(3) 新規サービスの追加：保守性を高めるために，稼働を止めずに新規サービスを追加可能にする．具体的には，サービスに対応するレイヤを実行時に追加可能にするために，図 2 の(4) Layer Receiver でレイヤを受け取り，(3) Layer Scheduler で管理する．

3.2. RT-COM

図 3 に我々が目指しているコンテイスト指向ソフトウェア開発方法論(Real-Time Context-Oriented Methodology)を記す．図に示す通り，ソフトウェアプロダクトラインのフィーチャモデルとUML に加え，レイヤ構造図，レイヤ相互作用図，レイヤ振る舞い図[4]を用いたモデル駆動開発を目指している．これらのモデルから COP プログラムを生成する．また，ペトリネットによるシミュレーションも目指している[5]．また，実践的な組込みシステムに適用を目指し，ROS への応用も行なっている[6]．

4. おわりに

本稿では，コンテキスト指向ソフトウェア開発方法論である RT-COM について概説した．これまで，RT-COA の個々の要素やペトリネットのシミュレーション，モデリング言語，ROS への応用などを行なって来た．

今後，個々の研究を充実さ合し，RT-COM の実現に向けて研究を進めたい．

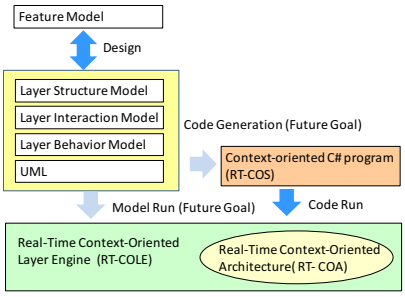

図 3　RT-COM の概要

参考文献

[1] G. Salvaneschia, C. Ghezzia, M. Pradellaa: Context-oriented Programming: A Software Engineering Perspective, Journal of Systems and Software archive, Vol.85 Issue 8, pp. 1801-1817, 2012.

[2] R. Hirschfeld, P. Costanza, and O. Nierstrasz: Context-oriented Programming, Journal of Object Technology, Vol. 7, No. 3, pp. 125-151, 2008.

[3] H. Watanabe, M. Sugaya, I. Tanigawa, N. Ogura, and K. Hisazumi: A Study of Context-Oriented Programming for Applying to Robot Development, Proceedings of the Workshop on Context-oriented Programming (COP) 2015, ECOOP 2015,2015.

[4] H. Watanabe, I, Tanigawa, M. Sugaya, N. Ogura, K. Hisazumi: A Layer Structure Diagram and a Layer Interaction Diagram towards a Context-Oriented Development Methodology for Embedded System, The Workshop on Live Adaptation of Software Systems(LASSY'16), 2016.

[5] H. Watanabe, I. Tanigawa, N. Ogura, M. Sugaya, K. Hisazumi and A. Fukuda: Coloured Petri-Nets Framework for Simulating Method Invocations on Context-Oriented Software, Proceedings of the Workshop on Meta-Programming Techniques and Reflection (META) 2016, SPLASH 2016,2016.

[6] Y. Saeki, I. Tanigawa, K. Hisazumi, A. Fukuda: ContextROS: Context-Oriented Programming for the Robot Operating System, Proceedings of the Workshop on Context-oriented Programming (COP) 2017, ECOOP 2017,2017.