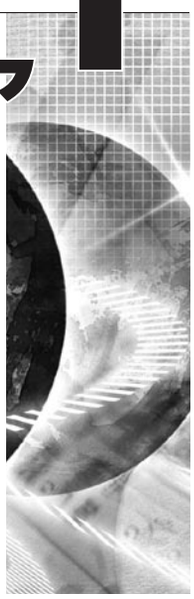
1

高信頼性組み込みソフトウェア開発のための技術課題



片山 卓也
北陸先端科学技術大学院大学情報科学研究科
katayama@jaist.ac.jp

高信頼組み込みソフトウェア開発の問題点

組み込みソフトウェアは，家電製品，自動車，通信機器などに組み込まれ，これらの製品や機器の品質や機能を決める最も重要な要素である．したがって，組み込みソフトウェアの開発技術を高く保つことは，我が国の製造業の競争力の維持にとってきわめて重要である．現在，組み込みソフトウェアの品質や信頼性は，組み込み技術者の高い技能によって支えられており，この人的な要素の重要性は今後とも変わらないが，それと同時に，大規模化・複雑化している現在の組み込みシステムには，新しい技術開発が必要である．

従来，組み込みソフトウェアはそのサイズがあまり大きくなかったこと，また，その機能が比較的単純であったことなどもあって，開発には最新のソフトウェアテクノロジーが用いられてこなかった．しかし，高度なユーザインタフェースや通信機能など製品に要求される機能が高級化すると同時に，利用可能なCPUやメモリなどのハードウェア資源が高性能化したことなどによって，ソフトウェアが大規模化・複雑化し，これまでの開発方法論が十分に機能しなくなりつつあるといわれている．組み込みシステムについては，外界との強いインタラクション，実時間性，限られた利用可能資源，高い信頼性，短い製品開発サイクル，などの厳しい要求があり，これらを満たす開発方法論の研究開発が強く求められている．本稿では，ソフトウェア開発方法論に関連したいくつかの技術課題について述べる．

オブジェクト指向組み込みソフトウェア開発

オブジェクト指向方法論は，オブジェクト指向の概念，すなわち，内部が隠蔽され，公開されたインタフェースのみを通して連係動作するオブジェクト集合，によってソフトウェアを記述しようとするものである．オブジェクト指向開発方法論は，ビジネスシステムにおいては標準的開発技術として認知され，特に大規模なソフトウェアの開発や保守・再利用に有効であることが実証されてきた．これは，オブジェクト概念に基づくソフトウェアにおいては，実世界の構造に近い形でそのモデル化が行われ，高い理解性や変更容易性，再利用性が得られるからである．

これに対し，ソフトウェアの動作は，オブジェクト間の連係動作として影に隠れてしまい，手続き型計算モデルのように，それを陽に表現されない．この特徴のために，ソフトウェアの構造とその動的振る舞いを直接に結びつけることが困難で，ハードウェア資源の制約や実時間性要求などが厳しい組み込みシステム開発においては，これまでオブジェクト指向技術は本格的には使われてこなかった．

しかしながら，コンポーネント技術による複雑システムへの対応可能性，UML記法やUMLツールの普及，クラスライブラリやコンポーネントの流通などによる高い再利用性など，オブジェクト技術の持つ技術的利点は大規模化した今後の組み込みソフトウェアの開発には必要であり，その利用は世界的な流れでもある．信頼性の観点からも，信頼性の高いコンポーネントやクラスの流通，再利用による信頼性の向上などの利点が多い．弱点とされてきた実時間性や性能面の改善に関しては，実時

間Java技術や実時間オブジェクト指向設計・解析技術の研究開発も活発であることから[1), 2)]，近い将来，実時間要求の厳しい特殊なものを除いては，組み込みシステムの開発にオブジェクト指向技術が利用されることが標準的になると予想される．

先進再利用開発：プロダクトライン開発，モデル駆動開発

多数の類似製品に組み込まれるソフトウェアの開発にとっては，再利用技術はきわめて重要である．プロダクトライン開発は，共通の特徴や性質を持つ一連のソフトウェアプロダクト群を，基本アーキテクチャを共有する再利用資源を利用して構築し，それによってプロダクト群の開発コストを低減させる技術である[3)]．既存ソフトウェアを利用した開発は日常的に行われているが，プロダクトライン開発として整備をするには，その内容や開発プロセスを明確化し，体制や環境を整備することが必要である．

これには，予想されるプロダクト群全体に対する機能や特性などに対する要求の明確化，それに基づいた基本アーキテクチャの決定，プロダクトライン資産となるべき共通コンポーネントの集合の設計と開発，個々のプロダクトの構築に合わせた共通コンポーネントの変更や再利用方式などを事前に計画し，体系化しておく必要がある．また，それに合わせたプロセスや環境の整備などが必要である．

プロダクトライン開発の基本は，一連のプロダクト群に対する要求の明確化である．その1つの方法として，ソフトウェアのフィーチャー（機能や品質特性）によるモデル化や，それに基づく変更点の設定などに関する研究が行われている．フィーチャーの形式的な記述やフィーチャー間の関係，特に整合性や依存性などの明確な定義や検証方法，それに基づくコンポーネントの系統的な設計と変更管理法の確立などが望まれる．また，特に組み込みシステムとの関連では，性能や実時間特性などの動的特性に関して研究すべき課題も多い．今後の組み込みシステム開発における再利用技術の中心課題であり，その研究開発を理論と実務の両面から着実に推進することが必要である．

プロダクトライン開発とならんで注目を集めている再利用開発方法論は，モデル駆動開発（MDD）である[4), 5)]．プロダクトライン開発が機能の追加や増強のための技術であるのに対して，モデル駆動開発は，アプリケーションを抽象モデルによって記述することにより，アプリケーションモデル（プラットフォーム独立モデルPIMとも言う）と実装プラットフォームの分離を行い，それによってアプリケーションモデルの再利用やアプリケーションモデルからのシステムの自動生成を行う技術である．従来，組み込みシステムの開発では，性能やハードウェア資源を重視するあまり，アプリケーションのロジックとデバイス・CPUの制御などが絡み合ってプログラムが作られ，その結果，保守性や移植性に問題が生じることが多かった．MDDの採用によりこれらを分離すれば，組み込みソフトウェアの移植性を上げるうえで，大きな効果が期待できる．

すでに制御系を対象にしたシステムでは，ブロック線図によって記述されたアプリケーションモデルから制御プログラムを自動生成する技術が実用化されつつある[6)]．より一般のシステムにMDD技術を適用するためには，対象ドメインの絞り込みや明確化によるアプリケーションロジックの記述可能性の確保，組み込みシステム用のプラットフォームの技術開発や，実時間性などの性能上の問題の解決などを図る必要がある．MDDは従来のコードを中心に置いた開発から，抽象度の高いアプリケーションモデルに開発の焦点を移そうとするものであり，モデルの検証やコードの自動生成などによって，信頼性の向上と開発コスト削減をもたらすことが期待される．

形式検証技術

組み込みソフトウェアにとって，その信頼性は最も重要な特性である．ソフトウェア品質の低さはそれの組み込まれた製品の価値を下げ，リコールなどによる経済的損失をもたらすのみでなく，機器の誤った制御などにより人命などへの危険をもたらす可能性もある．この点はビジネスソフトウェアとは大きな違いがある．ソフトウェアの信頼性は，その設計方法論や開発プロセスなどにも依存するが，最終的にはプログラムの正しさの検証によって保証されなければならない．

現在，プログラムの正しさは，テストによって検査するのが普通である．これはテストデータをプログラムに入力し，その出力を観測することによりプログラムの正しさを確認しようとするものである．テストでは，テストデータの選択が最も重要であり，これにはテストのカバー率などに関して多くの基準が考えられ，入出力関係によってプログラムの正しさを保証できるビジネス系のソフトウェアの場合には，これらのテスト基準に従うことにより，テスト集合を合理的に決定することが行われてきた．一方，組み込みシステムのように外部環境とのインタラクションによって動作するプログラムに関しては，簡単で合理的なテスト基準を設けることが一般には困難であり，テストの設計にはノウハウと経験が必要となる．その結果，場合によっては膨大で高コストな網羅的実機テストを行うことになる．もちろん，実機テスト

はプロダクト開発の最終段階では必要であるが，プログラムの検証に関しては，もう少し組織的で工学的なアプローチが求められる．

テストとともに古くから研究されてきた検証方法が形式検証である．これは，プログラムの仕様を形式的に与え，プログラムが仕様を満たすことを機械的に検証する方法である．プログラムの形式検証については，これまでに非常に多くの研究開発がなされてきたが，多くの方法が定理証明技術によるものである．これは，仕様をある種の論理式で与えることにより，プログラムの正しさを論理式の正しさに還元し，それを定理証明技術により確立しようとするものである．このような方法で実際のプログラムの検証を行う先進的な試みが数多く行われてはきたが，検証コストの高さなどから，一般に普及するまでには至っていない．

これに対し，現在では，定理証明より簡便なモデル検査技術に注目が集まり，組み込みシステムなどへの適用が試みられている．この技術は，プログラムの動作空間を有限状態モデルによって表現し，このモデルの全探索によってプログラムの正しさを保証しようとするものである[7]．ハードウェアの検証に用いることをターゲットとして発達してきた技術であるが，ソフトウェアに対しても適用され，成功例も報告されている．

ハードウェアに比べ状態空間の大きいソフトウェアにモデル検査を適用するには，解決すべき課題も多い．プログラムを直接対象とすると状態爆発を起こすことから，抽象化された検証モデルを別途作成し，それに対してモデル検査を適用するのが普通である．このための技術としてはデータ抽象化や述語抽象化技術などが知られている．問題は，抽象化されたモデルに関しての検証から誤りが検出されなくても，元のプログラムが正しいことが結論できないことである．抽象化によって誤りが隠れてしまった可能性があるからである．どこまで抽象化を緩めて再検査を行うかが問題で，抽象度の調整と検査対象部分の絞り込みには，システムの深い理解が必要である．

この問題の1つの解決法は，プログラムより抽象的な設計モデルを対象にモデル検査を適用することである．これには実行可能な設計モデルを構築する必要があるが，これにより状態空間を抑えて設計の正しさを検証することが可能になる．MDDにおけるアプリケーションロジックのモデル検査を行うこととも関連する手法である．

現在のモデル検査技術は，ソフトウェアへの適用に関しては技術的には未熟である．たとえば，大きなシステムを部分に分割し，各部分の検証結果を利用して全体の検証を行うモジュラーモデル検査や，システムの拡張に対応して増分的に検証を行うためのインクリメンタルモデル検査法の開発が不十分であり，大きなシステムを分割して検証する方法論が十分には確立されていない．その一方，実時間制約を対象にしたモデル検査や，連続状態と離散状態の混合を許すハイブリッドモデル検査技術などの種々のモデル検査の研究が活発に行われており，将来的には組み込みシステムのための総合的な検証・解析技術基盤となる可能性が高い．欧米の先進的研究開発機関や産業界ではこの技術に大きな研究開発投資を行っており，我が国においても理論および応用の両面から十分な研究開発を行う必要がある．

文部科学省e-Societyプロジェクト「高信頼組込みソフトウェア構築技術」

ソフトウェアにおいては，主に大学などで行われている研究と産業界の開発現場で使われている技術の乖離が大きいといわれている．このため，最先端の研究成果が開発現場で使われることが少なく，また，開発現場で真に必要とされる課題が研究されていないということが強く指摘されている．この問題を解決するために，文部科学省リーディングプロジェクト「e-Society 基盤ソフトウェアの総合開発」は2003年度より開始され，現在，8大学および十数社の企業が緊密に連携し，次の2つの技術領域，（1）高い生産性を持つ高信頼ソフトウェア作成技術の開発，（2）情報の高信頼蓄積・検索技術の開発，を重点的に推進している[8]．本プロジェクトでは，産業界からのニーズに基づき，大学等が持つ研究ポテンシャル，人材養成機能を最大限活用し，社会の基盤となるソフトウェアの研究開発と研究者養成を一体的に推進している．このプロジェクトの最も大きな特徴は，大学と企業が密接に連携し，現実問題の解決を目指して研究開発を推進していることである．e-Societyプロジェクトにおいては，高信頼ソフトウェアに関して現在7課題の研究が行われているが，このうちの1つが「高信頼組込みソフトウェア構築技術」である．オブジェクト指向技術や形式検証技術を用いた高信頼組み込みソフトウェア構築法の研究が進められており，本特集記事のうち，検証技術，オペレーティングシステムに関する記事は，このプロジェクトの研究成果に関するものである．

参考文献

1）ボレラ他：Javaリアルタイム仕様，ピアソンエデュケーション．
2）http://research.sum.com/project/mackinac
3）岸　知二，野田夏子，深澤良彰：ソフトウェアアーキテクチャ，共立出版．
4）Frankel, D. S. ： MDA モデル駆動アーキテクチャ，エスアイビーアクセス．
5）http://www.org/mda/
6）真田幸俊：MATLAB/SimulinkによるCDMA，東京電機大学出版局．
7）Clarke, E. M. 他：Model Checking, MIT Press.
8）http://cif.iis.u-tokyo.ac.jp/e-society/

（平成17年4月10日受付）