

組み込みシステムの動向

青 木 利 晃†

ウィンターワークショップ・イン・伊豆のアーキテクチャ・組み込みグループでは、ソフトウェア工学の立場で、現在の組み込みシステム開発の問題点と解決策について議論した。本稿では、この議論により浮き彫りになった組み込みシステムの現状と取り組むべき今後の課題を紹介する。

Trend of Embedded Systems

TOSHIAKI AOKI†

In Winter Workshop in Izu, we discussed about problems and their solutions of current embedded system developments. In this paper, we show current status that we have identified and future works in the embedded system developments.

1. はじめに

組み込みシステム開発は、現在、世界中で最も盛んに研究されている分野の1つであり、日本の学会や研究会においても組み込みシステム開発に注目したセッションやパネルが頻繁に行われている。ソフトウェア工学研究会では、組み込みオブジェクト指向ワーキンググループを設立し、オブジェクト指向シンポジウムや定例研究会でこの話題について積極的に取り上げている。しかしながら、取り組むべき具体的な課題や進むべき方向性が見えてこないのが現状である。そこでウィンターワークショップ・イン・伊豆では、「組み込みシステム開発とソフトウェア工学」と題して、組み込みシステム開発の現状について、ソフトウェア工学の観点から議論を行った。

本稿では、2節で組み込まれるソフトウェア(以下、組み込みソフトウェアと呼ぶ)の特性について、3節で議論した内容について紹介する

2. 組み込みソフトウェアの特性

組み込みシステムとは、機器を制御するために組み込まれた計算機システムのことである。従来、組み込みシステムは、一部の電気製品に組み込まれる小規模のシステムであった。現在は、プラント制御システム、自動車エンジン制御、携帯電話、オーディオ機器、と

挙げれば切りが無いほどあり、その応用範囲は非常に広がってきている。

組み込みソフトウェアの主な特性としては、以下のものが挙げられる。ただし、すべての組み込みソフトウェアが以下の特性を持つのではなく、開発対象となるシステムにより考慮する特性、及び、それらの重みが異なる。

- 高いパフォーマンス

家庭電化製品の組み込みシステムでは、非常に限られた計算機資源しか使用できない場合が多い。そのため、組み込みソフトウェアに高いパフォーマンスが要求されることがある。

- 実時間性

組み込みソフトウェアでハードウェアを制御する際、特定の時間内に処理を完了させたり、一定周期毎に処理を実行しなければならない場合がある。

- 高い信頼性

組み込みソフトウェアの誤りは、製品のリコール等により莫大な損失を発生させてしまう。このため、高い信頼性が要求される。また、航空機システム等では人命に影響を与えることがある。このようなシステムでは、機器が故障しても動作し続ける耐故障性の実現をソフトウェアに要求されることがある。

- ハードウェア依存

組み込みソフトウェアを実際に動作させるためには、対象となるハードウェアが必要である。組み込みシステム開発ではハードウェアも開発対象で

† 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

あるため、それが完成するまでソフトウェアを実際の環境で動作させることができない。

- 市場動向依存
携帯電話等の市場では、その動きが非常に流動的である。市場動向に合わせて製品を開発する必要があるため、開発中に仕様変更を要求される場合がある。

3. 組み込みシステム開発の現状

ワークショップでは、現在の組み込みシステム開発が抱える問題と解決策をソフトウェア工学の立場で明確にすることを試みた。これにより、現在の組み込みシステム開発の現状及び問題点が、一部ではあるが、浮き彫りになった。以下では、仕様記述、アーキテクチャ設計、オブジェクト指向開発の現状と、組み込みシステムの分類について紹介する。

3.1 仕様記述

仕様記述に関する重要な問題の1つは、仕様を早期に固めづらいことである。その理由としては、大きく分類して2つある。1つ目は、組み込みシステム開発は市場動向に大きく依存しているため、開発中に仕様変更が発生することである。その誘因としては以下のものがある。

- あるシステムの開発中に、他社が同様のシステムでより多くの機能を備えたものを開発、又は、発売した場合、それより魅力のある仕様に変更しなければならない。
- アイデアが鍵を握る製品では、すべてのアイデアを早期に洗い出すのは不可能である。開発中でも新しいアイデアを思い付いた場合、それを実現しなければならない。
- 法律改正や契約改正等により、システムに規制がかかる場合がある。

2つ目はハードウェア上で色々実験してみないと仕様を固めることができないということである。PCやワークステーション上の純粋なソフトウェア開発の場合、RUP(Rational Unified Process)¹³⁾やXP(eXtreme Programming)¹⁴⁾のような繰り返し型プロセスにより、上流工程と下流工程を往來しながら仕様を固めることが可能である。しかしながら、組み込みシステム開発の場合、評価機の完成は開発工程の終盤である場合が多い。

開発中の仕様変更や、予期されなかった問題点が発生した場合、その解決をソフトウェアにより行うことがしばしば発生する。これは、ハードウェアの変更はほぼ不可能なためである。現在では、FPGA(Field

Programmable Gate Array)のようなプログラム可能集積回路を用いると、前もってある程度のプロトタイプリングは可能であるが、消費電力等を完全に評価することは不可能である。

その他の仕様記述に関する問題点としては、現在提案されている仕様記述言語では組み込みシステム特有の要件を記述しづらいことが挙げられる。2節で述べたように、組み込みシステム特有の要件として実時間性、信頼性等がある。例えば、UMLを用いる場合、これらの要件は、自然言語によるメモ等により記述することになってしまう。

3.2 アーキテクチャ設計

組み込みシステムのアーキテクチャ設計を行う際、ソフトウェアとハードウェアの境界線を決定する必要がある。そのためには、ソフトウェアとハードウェアの知識が必要となるが、両方を持ち合わせたアーキテクトは稀であり、また、両方をアーキテクトに求めるのは要求過多であり避けるべきである。この問題を解決するためには、2つの技術が必要となる。

1つ目は、ソフトウェアエンジニアとハードウェアエンジニアがその知識を共有するための共通言語である。このような共通言語は、ソフトウェアとハードウェアの両方の要件を十分に記述できる必要がある。現在、両方を記述可能な言語としてSpecC¹⁵⁾やSystemCといった仕様記述言語が注目されているが、ハードウェアアーキテクチャの表現を重視しており、ソフトウェアアーキテクチャの記述が十分に出来ないとされている。

2つ目は、アーキテクチャの評価やトレードオフを行うための技術である。現在、パフォーマンスや信頼性といった非機能的要件に注目してソフトウェアアーキテクチャの評価やトレードオフを行う技術が盛んに研究されている。このような技術が組み込みシステム開発に適用可能かどうかの調査が必要である。

3.3 オブジェクト指向開発

現在、組み込みシステム開発においてもオブジェクト指向開発が注目されている。その背景には、組み込みシステムの規模が一般に大規模システムと言われているものに切迫している、積極的な再利用が必要になった、等が挙げられる。

オブジェクト指向開発手法の特徴として、上流工程におけるUML等のダイアグラムを用いたシステムの記述がある。この記述は、分析モデルと呼ばれる。分析モデルでは、開発対象システムや対象領域に出現する概念を、実装を考慮せずに整理する。この工程が大規模システム開発に非常に有効である。一方、組み

込みシステム開発の特徴に、実時間性や信頼性といった厳しい非機能的要件の取り扱いがあるが、従来のオブジェクト指向開発法ではこれらを具体的に取り扱っていない。そこで、組み込みシステム開発に特化したオブジェクト指向開発法として、ROOM¹⁶⁾、OCTOPUS¹⁷⁾、Real-Time UML¹⁸⁾等が提案されている。しかしながら、組み込みソフトウェア開発において困難な、スケジューリング法、タスクへの詰め込み、処理の最適化といった問題を具体的に扱っていないため、どの開発法も十分に現実的なものとなっていない。

3.4 組み込みシステムの分類

すでに述べたように、現在「組み込みシステム」と呼ばれるシステムは挙げれば切りが無いほどの種類がある。それぞれの規模及び制御対象は異なるため、システム開発における方法論、プロセス、モデルも異なるはずである。しかしながら、現在、これらは総じて組み込みシステムと呼ばれており、研究会等でも同一の領域として扱われている。このため、技術の発展の妨げになっているように思われる。そこで、組み込みシステムと呼ばれている領域を、さらに細かな領域に分ける必要がある。

次に領域の分け方であるが、その軸が明確でない。今までのソフトウェアは、それが提供する‘サービス’により領域が分類されている場合が多い。これは、システムリソースが柔軟であるため、ソフトウェアアーキテクチャや方法論への影響力が大きい‘サービス’の視点が注目されたからである。一方、組み込みシステムの場合、CPUが8bit~32bit、Memoryも数KBオーダー~数百MBオーダーのものがあり、これらに依存してソフトウェアの作り方が異なる。CPU 8bit、Memory 32KBであれば、アセンブラでプログラムを組んだり、タイミングチャートを書いたりしてI/Oへのアクセスを気にしなければならない。一方、CPU 32bit、Memory 256MBくらいのシステムであれば、OSを使い、I/Oの制御はデバイスドライバとしてモジュール化すれば良い。将来的に、ハードウェアの価格が下がりそのサイズが小さくなれば、現在の‘サービス’による領域分けが適切になるかもしれないが、現状では、システムリソースやハードウェア構成に注目した領域分けが必要である。その他にも、方法論等に影響を与える要因として、信頼性、実時間性等があり、組み込みシステムが持つ特性をどのように整理して、どのように領域分けをするかを考えなければならない。

4. ま と め

ワークショップでは、3節で取り上げた問題点の解決

策についても議論した。挙げられた主な解決策は、パターンによる解決やベストプラクティスの蓄積であった。もちろんこれらの蓄積は重要であり1つの解決策である。一方、本質的な解決策がまだ見当たらないとも考えられる。問題点の本質に近付くためには事例を蓄積して焦点を絞って深い議論をすることはもちろん必要だが、そのための場を提供する研究会等のコミュニティも重要である。従来は、企業の秘密主義により、技術者の研究会等への参加に消極的であった。しかしながら、現在は、組み込みシステムの多様化や大規模化に伴って様々な問題を抱えるようになり、そういった場への企業からの参加者が増えている。その受け皿として、SWEST^{*1}、CEST^{*2}、Emblem^{*3}といった研究会や団体も設立されつつある。こういった団体では、技術者の交流やセミナーによる教育活動が行われており、技術者の企業間、産学間の連携を促している。

その他に、小人数の組み込みシステム開発のエキスパートと研究者により構成される技術的な先導をする団体が必要であると思われる。具体的には、以上で述べた団体と連携して現状を把握し、組み込みシステムに適用可能な技術を探し適用事例を示したり、現在の問題点を解決する技術を開発する。このような活動は、小人数でないと行えず、また、日本の組み込みシステム開発技術の向上には不可欠であると考えられる。

最後に、活発に議論して下さったアーキテクチャ・組み込みシステムグループの参加者^{*4}の皆様へ感謝いたします。

参 考 文 献

- 1) 青木利晃: オブジェクト指向組み込みシステム開発における科学的手法
- 2) 荒井玲子: 現実味あるアーキテクチャ議論へのアプローチ
- 3) 岡崎光隆: 形式的手法による組み込みシステム設計モデルの獲得
- 4) 佐藤啓太: 組み込みシステムアーキテクトについて
- 5) 沢田篤史: ハードウェア/ソフトウェア協調設計ツールの統合環境
- 6) 鈴木正人: コンポーネント接続の形式的検査方法
- 7) 豊島真澄: 組み込みシステムにおける耐故障性に関する一考察
- 8) 七島功一: システムバスを用いた反応的システムの構築法
- 9) 根岸雅子, 井上健: オブジェクト指向の適用によ

*1 組み込みシステム技術に関するサマースタッフ

*2 組み込みソフトウェア技術者研究会

*3 組み込みシステム業界交流会

*4 参考文献1~12の著者が参加者。

るデータ管理の再設計

- 10) 橋本隆成:組み込みシステムへのオブジェクト指向分析設計の適用
 - 11) 松井聡一:状態遷移図の段階で抽出すべき不完全な記述
 - 12) 渡辺晴美:ネット指向言語に基づくリアルタイムシステム開発支援
 - 13) I.Jacobson, et.al: The Unified Software Development Process, Addison-Wesley, 1998.
 - 14) K.Beck: eXtreme Programming Explained, Addison-Wesley, 1999.
 - 15) D.Gajski, et.al: SpecC: Specification Language and Methodology, Kluwer Academic Publishers, 2000.
 - 16) B.Selic, G.Gullekson and P.Ward: Real-Time Object-Oriented Modeling, John Wiley & Sons, 1994.
 - 17) M.Awad, J.Kuusela, and J.Ziegler: Object-Oriented Technology for Real-Time Systems, Prentice Hall, 1996.
 - 18) B.Douglass: Real-Time UML developing efficient objects for embedded systems, 1997.
-