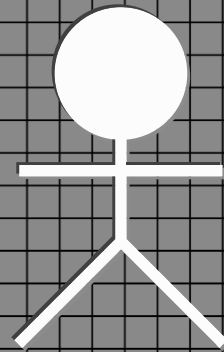


風評を鵜呑みにせず、きちんと自分の手で評価しよう

なぜUMLの
敷居は高いのか？

南角 茂樹，長濱 美保

IT (Information Technology) 分野の業務系などでは広く普及している UML だが、組み込みシステムに適用しようとするとしっくりこないことが多い。ここでは業務系システム開発の視点も交えながら、組み込みシステム開発に UML を導入する際の課題について明らかにする。筆者は、UML によるモデリング手法を丸ごと導入するのではなく、まずは効果のあるところに限定して導入することを提案する。(編集部)

以前から、ソフトウェア開発の効率化、および設計品質向上の手段として、モデル・ベース開発の有用性が語られてきました。しかし現実には、組み込みシステム開発の現場に必ずしもこの手法が浸透しているとはいえません。

ここでは、「なぜモデル・ベース開発は現場で浸透していないのか」を、「そもそもモデル・ベース開発が組み込みシステム開発に向いているのか」も含めて、考えてみたいと思います。なお、以後、モデルを表現するための記法としては UML を中心に考えます。また、論点を絞るため、オブジェクト指向アプローチを前提とします。それ以外の手法については、必ずしも以下の説明が当てはまらないかもしれないので、注意してください。

● 新しい手法は、なぜ開発現場に根付かないのか

実際の組み込みシステム開発には、流用開発と新規開発があります。それぞれの場合について考えてみます。

1) 流用開発の場合

従来から製造している組み込みシステム製品の品ぞろえを増やしたり、派生製品を発売したりするような場合は、流用開発が行われます。新規事業の開始など、まったく新しい市場へ進出する場合を除くと、組み込みソフトウェアの開発プロジェクトの多くはこのケースに相当すると思います。

流用開発では、現存しているソフトウェア資産を生かした開発を行う必要があります。従来の手法で開発することが多く、通常は新しい手法を導入することはありません。その製品ラインが昔から続いているものであればあるほど、新しい手法を導入しにくいという状況があります。

また、新しい開発手法に不慣れなために生じる設計品質の低下(デグレード)や開発期間の長期化も問題となりま

す。さらに、組み込みソフトウェア開発においては、「出来上がったもの」そのものが仕様となっている場合があります。これは、仕様書に書かれていない、あるいは書ききれないシステムの細かい振る舞いについては、既存のシステムそのものが仕様(リファレンス)として扱われるということです。ですから、仮に新しい手法で開発しようとする、その互換性を保つために、最初に既存のソフトウェアの調査が必要になります。従来と同じ手法で開発するのであれば、この調査は不要です(右掲の**コラム1**を参照)。

このように、一つの製品ラインに対して継続的に流用開発が行われている状況では、新しい手法を導入するケースは多くありません。もしあるとしたら、新しい手法による利点が相当大きく、なおかつそのことが証明されている場合に限られるでしょう。なお、例外として、それまで使用していたマイコンが製造中止になるなどの外的要因により、やむなく手法を切り替えるといったケースもあります。

2) 新規開発の場合

ここでいう新規開発とは、従来製品に使用していたソフトウェアを流用しないという意味です。従来のソフトウェア資産を流用しないということは、従来のソフトウェアを流用するメリットよりも、使い続けることによるデメリットの方が大きいということになります。例えば、「従来のソフトウェアの品質が悪い」、「改造も含めて開発効率が悪い」など、重大な問題がある場合です。このような場合には、新しい開発手法の導入は比較的スムーズに進みます。

しかし、それだけ問題が切迫しているわけですから、速やかな問題の解決が求められ、さらに新しい手法の導入の成果を短期間に示すことが求められます。従来手法による

開発をモデル・ベースに切り替えるようなケースの多くは、このような状況に相当すると思います。

新しい手法を導入する際には、社外のコンサルタントの助けを借りることが多いのですが、これが問題になるケースもあります。組み込みシステム開発においては、手法以上に製品固有の知識が必要です。社外のコンサルタントは製品知識をあまり持っていないことが多く、そのため、コンサルタントが手法の導入の足を引っ張ることがあります。むろんきちんと対象製品を勉強しようとするコンサルタントの方も多いのですが、契約して教えてもらはずが、逆に教える一方になってしまったという経験もあります。

● ET ロボコンに見る技術者の取り組み

2007年7月14日～15日に、ETロボコン2007の本選大会が東京で開催されました。「ETロボコン」とは愛称であり、正確には「ET (Embedded Technology) ソフトウェアデザインロボットコンテスト」と呼ばれます。これは、デンマークのLEGO社の教育用ロボット「LEGO MINDSTORMS RCX」を使用し、**写真1**に示すような決められた車体で決められたコースをラインに沿って走行する競技です。

見かけ上は同一の車体なので、ソフトウェアのみの差で走行性能が決まることになります。ただし現実には、センサやモータの個体差が大きく、理想どおりにはいかないようです。これも現実のものを制御対象とする組み込みの宿命なのかもしれません(苦しい言いわけだが…)。

この大会には、「組み込みソフトウェア・エンジニアの新人教育と技術伝承」という目的があります。この目的を達成するために、ETロボコンには2種類の競技が用意さ

れています。一つは「ライン走行のタイム・レース」、もう一つは「走行システムの分析・設計のモデリング」です。参加者は年々増えており、今年は全国3地区(関東、東海、関西)に分かれて予選大会が行われるほどでした。**写真2**は、参加者が実際にソフトウェアを開発している様子です。

通常、システム開発には工程があります。例えば、要求定義、分析、外部設計、内部設計、実装、テストといった工程です。ライン走行のタイムには、実装された結果が表れます(ただしLEGOの場合、先に述べたようにハードウェアの個体差にもかなり影響される)。

その実装は、こうしたいくつもの工程を経て、でき上がったものです。開発のフェーズが進む中でモデリングを



写真2
ETロボコン参加
チームの開発風景

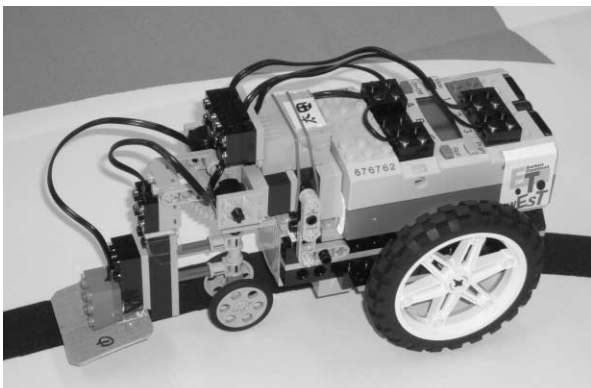


写真1 ETロボコンで使われている自律走行ロボット

教育用ロボット「LEGO MINDSTORMS RCX」をベースとしている。ETロボコンでは、これをバス・ファインダまたは走行体と呼んでいる。

コラム1 丸めの問題はどうモデリング？

組み込みシステムにおいて、ソフトウェアで実行していた浮動小数点演算に対して、浮動小数点演算プロセッサ(FPU)という専用のハードウェアが使われ始めたころの話です。

ソフトウェアの大部分の開発はC言語で行っていました。問題が発生したのは浮動小数点数を整数に代入(キャスト、丸める)したときのことです。当時、コンパイラのメーカーが浮動小数点数の扱いに慣れていなかったためか、丸め(四捨五入)にFPUの備えている丸め機能を使っていました。ご存じの方もいると思いますが、整数に丸める場合、整数までの距離が同じ場合は偶数側に丸めるといふ、四捨五入とは異なる定義になっています(現在では、IEEEの規格になっている)。そのため、1.5を丸めると2になるのは良いとして、2.5を丸めても2になってしまいました。

このときは、さすがに「出来上がったもの」が仕様というわけにはいかず、コンパイラ・ドライバに手を加えて切り抜けました。このようなケースは、どのようにモデリングすればよいのでしょうか？