

10年で変わった「組込み」、変わらない「組込み」

独立行政法人情報処理推進機構

ソフトウェア・エンジニアリング・センター 組込み系プロジェクト
プロジェクトサブリーダー

田丸 喜一郎

1 こんなに変わった「組込みへの認識」

この10年で「組込み」に対する社会の認識も大きく変わった。最近では「組込み」の言葉を新聞や雑誌などで目にするのも珍しく無くなった。

政府の認識も変わってきている。今年の4月には、産業分類コードに「組込みソフトウェア業」が新設された。総合技術会議が5月にまとめた革新的技術戦略案では、世界トップレベルにあり経済社会に大きな波及効果をもたらすことが期待される技術（＝革新的技術）として「組込みソフトウェア技術」が挙げられた。

2 こんなに拡大した「組込み産業」

我が国の組込み産業は拡大の一途を辿っている。組込みシステムの開発費もこの5年間で2004年度の5兆7千億円から2008年度では8兆3千億円に増加しており、その4割以上を占める組込みソフトウェアの開発費も2004年度の2兆円から2008年度には3兆5千億円に増加している。

組込みソフトウェアを搭載した組込み機器の生産額は、製造業の生産額の半分以上を占める状況になった。GDPに占める割合も13%を超えた。

3 こんなに増えた「組込み技術者」

組込みソフトウェア開発費の約8割は人件費に充てられており、開発費の増加は、開発に従事する技術者の増加を意味している。2004年度の推定約15万人に対して、2008年度は推定約24万人となっており、5年間に約9万人が新規に参画したことになる。

それでも経営者の多くは技術者の不足感を感じている。不足感の調査を始めた2005年度では不足感は41%。現状10人の技術者を抱える経営者は、4人足りないとの認識である。日本全体で換算すると約10万人不足に相当する。

4 いまだに変わらない

「ハードウェア・トラブルとの戦い」

その一方で組込み開発の現場では変わらない面も多い。

組込み開発の特徴の一つとして、ハードウェアとソフトウェアの並行開発が挙げられる。PCなどをハードウェア・プラットフォームとする業務系のシステム開発では、ソフトウェア開発に着手する段階では、基本的なハードウェアは稼動している。

多くの組込みシステムの開発では、システムLSIなどのハードウェア開発とソフトウェア開発が並行に進められる。このため、業務系のソフトウェア開発では遭遇しない様々なトラブルに遭遇する。

「ハードウェアの不具合をソフトウェアで回避」「（開発着手後の）ハードウェアの仕様変更に対応」など、半数のプロジェクトが遭遇する。全くハードウェアに起因するトラブルに遭遇しないプロジェクトは1割にも満たない。いまだに9割以上のプロジェクトは、ハードウェアのトラブルと戦いを余儀なくされている。

5 いまだに変わらない「勘と経験」

組込みソフトウェア開発はマイクロプロセッサの登場により、ハードウェア開発に付随した作業として現場発生し、肥大化してきた。このため、大規模化した現状の組込みソフトウェア開発においても工学的な手法が根付いている現場は少ない。

大規模開発で重要となるプロジェクトマネジメントでさえ、多くの現場は、いまだに「勘と経験」が支配している。例えば、工数見積もりにおいても、「個人の経験」「過去の実績」をベースとする割合は合わせて7割にもなる。計画書の作成方法でも同じく5割、テスト完了基準でも6割が「勘と経験」である。

6 いまだに変わらない、いや、こんなに変わったZIPC

最後に、20年近くも変わらずに我が国の組込み開発で使われ続けているZIPCに触れておこう。と思ったのだが、残念ながら紙面が尽きてしまった。これについては別稿に譲ることにしよう。

ご存知ですか？ EHSTM/ZIPC 豆知識

STMの駆動型にはイベントドリブン型（E型）とステートドリブン型（S型）がありますが、その違いは何でしょうか？

■E型

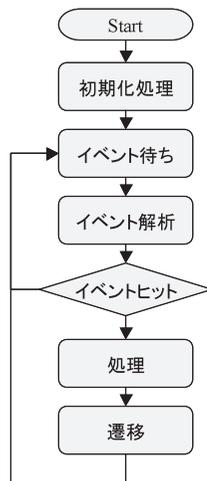
イベントが入力されることで動作する型です。イベントドリブン型はイベントをプログラムの一箇所待ち、イベントが入ったらイベント解析関数を起動してイベントを解析します（図A）。その結果、STMごとに割り振られた事象Noが算出され、現在の状態Noとクロスする部分のアクションが呼び出されます。

■S型

状態がアクティブになることで、イベントの解析を開始する型です。ステートドリブン型は状態がアクティブになるごとに、状態に結びついたイベント開析関数を起動します（図B）。イベント解析をした結果、現在の状態Noとクロスする部分のアクションが呼び出されることとなります。

図A) E型駆動の動作原理

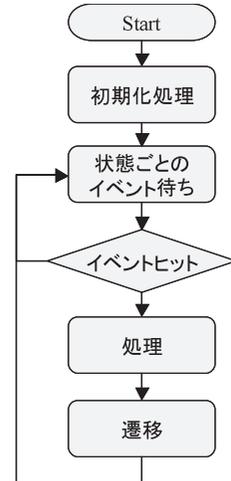
- 1) メイン関数が発行されたイベントを受け取る。
- 2) 受けたイベントを解析して一致するイベントを調べ、割り振られた事象Noを算出する。
- 3) 状態を解析し、現在の状態Noを算出する。
- 4) 割り振られた事象Noと現在の状態Noからクロスするアクションを呼び出します。



図A

図B) S型駆動の動作原理

- 1) メイン関数が発行されたイベントを受け取る
- 2) 状態を解析し、現在の状態Noを算出する。
- 3) 受けたイベントを解析して一致するイベントを調べ、割り振られた事象Noを算出する。
- 4) 割り振られた事象Noと現在の状態Noからクロスするアクションを呼び出します。



図B