

通信系組み込みソフトウェア開発経験から

藤森 隆

日本電気(株)

和文抄録

信号処理および通信分野での組み込みソフトウェア開発の経験に基づいて、ソフトウェア工学的な観点から同分野でのソフトウェア開発上の問題点とその発生原因の分析を行うとともに、問題解決への糸口を探る。とりあげる問題点としては (i) 組み込みソフトウェア開発で生じるリソース制約を回避するための修正が及ぼす開発工程間の依存性の増大とソフトウェア部品の再利用性の低下、(ii) 信号処理ソフトウェア開発における実装工程と信号処理系設計および評価工程の結合強化の効果、(iii) 通信系ソフトウェアの開発における接続性の評価とソフトウェア実装工程間の結合の発生についてである。

Issues on Embedded Software Development

- based on our experiences on communication software development -

Takashi Fujimori

NEC Corporation

英文抄録

In this paper, we point out some technical issues on embedded software development based on our experiences on developing signal processing and communication software. We also examine the clues to improve the issues. The major issues discussed are as follows:

- 1) As we need to tune up the software to attain severe restriction on resources, development phases tend to strongly depend on each other. It also prevents reusability of software assets.
- 2) Though we have to adequately trace the signal processing design to its implementation, it is not easy under the current tool environment.
- 3) Tight coupling of implementation phase and evaluation phase of communication software development.

1.はじめに

組み込みソフトウェアの開発においては、プロセッサの処理能力の向上やメモリの大容量化といったハードウェア面のキャパシティの増大とともに、ネットワーク機能の搭載や操作性の改善のためのヒューマンインタフェース機能の強化といった機能要求の高まりを受けて、その規模および複雑さの増加傾向が顕著になっている。その一方で製品の開発に充てられる時間は実質的に短縮されている。本報告は、小型の機器における通信系の組み込みソフトウェアの開発の事例を取り上げ、そこでの具体的な問題の分析を試み、上述のソフトウェア開発の状況をふまえてこの分野で今後求められる事柄について本研究会での議論の端緒のひとつにさせていただくことを目的としている。

2. 本ソフトウェア開発の概要

本節では、組み込みソフトウェアの開発方法に関する議論を深めるにあたり具体的な対象を明確にするために以降の節に関連のある事柄を中心に本ソフトウェア開発の概要を述べる。

対象となるシステムのハードウェア構成を図 1 に示す。基本的な機能としては、ネットワークインタフェース機能、ディスプレイとキーボード/キースイッチで構成されるヒューマンインタフェース、半導体不揮発メモリまたはディスクドライブで構成されるストレージ、それに機器に付属するメカニズムまたはコントロールサブシステムとのやり取りをするためのローカルインタフェースが含まれている。システム全体の管理を行うメインプロセッサがあり、性能要求とコスト制約をはじめとした製品仕様決定の判断として前記の機能ブロックに専用のプロセッサが配置されることがある。

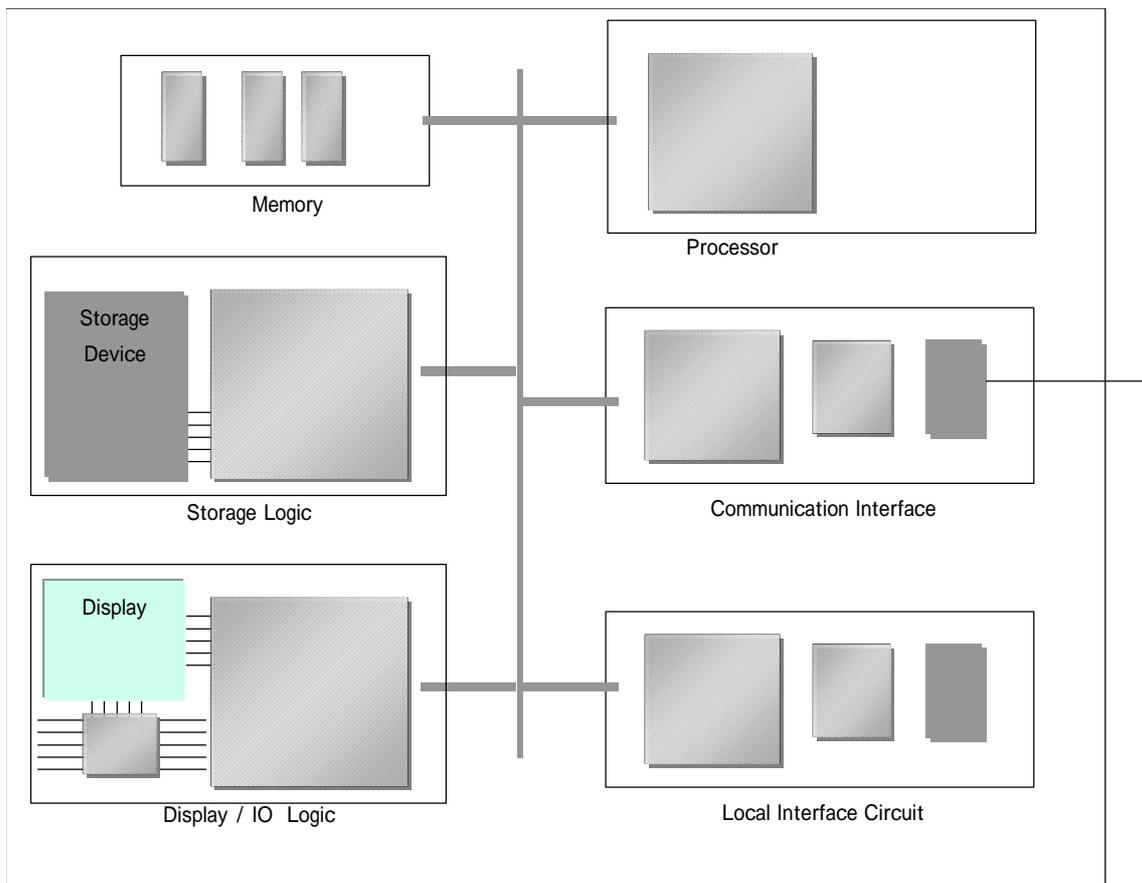


図 1 設計対象の構造(ハードウェア)

WAN¹ダイヤルアップ型のネットワークインタフェースの場合には図 2に示されるように変復調や回線適応等のデジタル信号処理専用のプロセッサと通信プロトコルの処理を行うコミュニケーションプロセッサをネットワークブロック専用で設けるケースや通信プロトコルの処理に関してはメイン(アプリケーション)プロセッサに割り当てるケースまた信号処理を含めてすべてメイン(アプリケーション)プロセッサが実行するケースなどのプロセッサ構成のバリエーションが存在する。上記の信号処理プロセッサおよび通信プロトコル処理プロセッサに関しては、搭載するソフトウェアを開発するケースと専用ハードウェアとしてソフトウェアがすでに組み込まれている製品を購入して使用するケースがありうる。

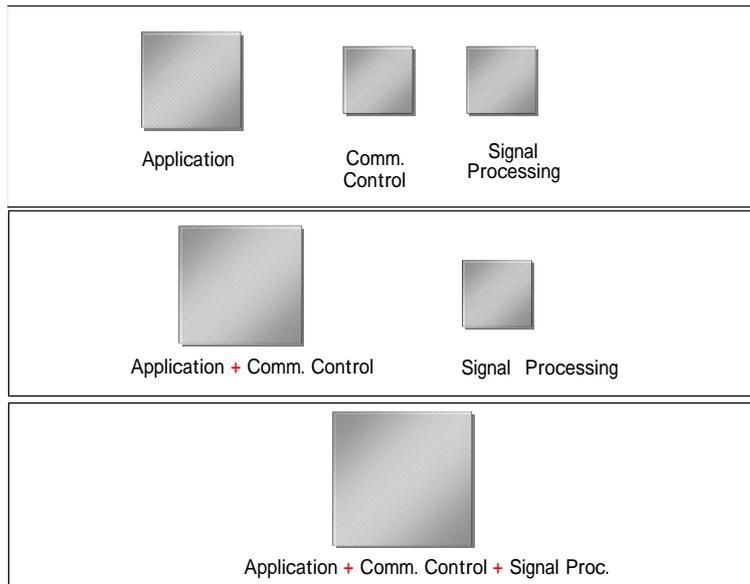


図 2 プロセッサ構成のバリエーション

通信系(WAN)を中心とした対象システムのソフトウェアの機能構成を図 3に示す。この図で上下方向は機能階層に対応させている。

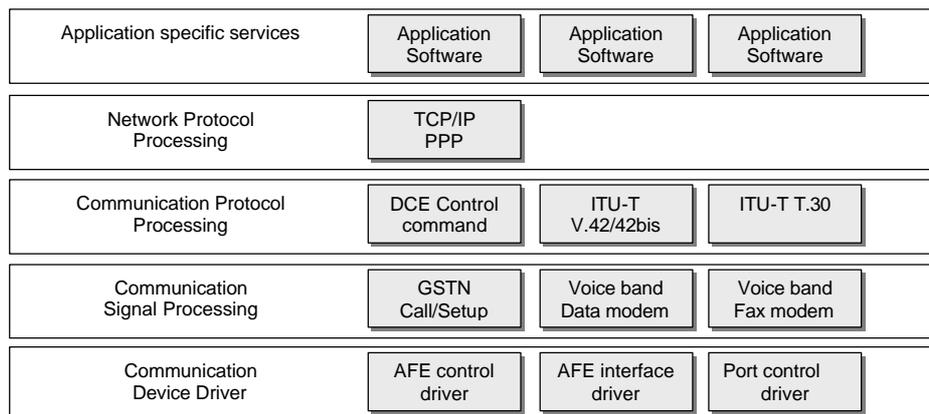


図 3 ソフトウェア機能

¹ WAN: Wide Area Network

3. 組込みソフトウェア開発の問題点

本システムの開発で経験したソフトウェア工学的な問題点を紹介し、それらに対する分析を試みる。ここで述べる問題の領域としては通信系ソフトウェア開発の問題点のうち、特に(i) 組み込みソフトウェアの開発に特有の問題、(ii) 信号処理ソフトウェア開発に伴う問題点、(iii) 通信ソフトウェア開発の側面の問題点とする。これらはソフトウェア開発の観点からは再利用技術に関するものと最適化技術に関するものとに分類される。

3.1 組込みソフトウェアとしての課題

完全なブラックボックス化の困難なソフトウェア(ライブラリ)部品

組込みソフトウェア開発で起こりがちなことは、各機能ブロックをシステムとして統合する工程で、リソース制約が顕在化することである。これらの制約に対処するためにリソース要求の最適化²を図るのであるが、その適用範囲がブラックボックスとして使用されるのが理想的であるソフトウェア部品にも及ぶことがある。典型的なリソース要求としてはメモリ使用量、プロセッシングタイムなどがある。

ソフトウェアライブラリの品質上の課題

上記の問題の再利用の観点からの問題として、以下のような組込みソフトウェアのライブラリ部品化における品質の確保の難しさがあげられる。第一は部品として再利用における汎用性を優先した場合、それを使用する際には上述の理由から最適化の作業が発生する機会が少なくないこと。次は通常最適化を施した後検証を行うことから、検証がなされたものをライブラリ化すると部品としての汎用性が乏しいものとなること。第三は検証されたプログラムの一部を汎用性のある部品としてライブラリ化するには、しかるべき作業時間を割り当てる必要があるが、製品を作ることに主眼が置かれている場合には、スケジュールの圧力のもとでそれを徹底することが容易でないこと。

品質が高く再利用可能な部品が形成されにくいことに伴う生産性低下の悪循環に陥るのはこのためである。

3.2 信号処理ソフトウェアとしての課題

信号処理ソフトウェアの開発では採用する信号処理アルゴリズムの検証とパラメータの設計が実装の前段階にある。

開発ツールによってはこの段階で生成された信号処理のモデルからターゲットプロセッサのコードまたは高級言語のソースコードに変換する機能をもっている。ただし現状においてもプログラム実装レベルの細部の指定も含めた最適化コードの生成に関しては、中間に手作業が入らざるをえない状況は依然として存在している。信号処理のモデルまたはそのパラメータを変更するたびに、ソースコードに加えた修正を繰り返すのは生産性の点で好ましくない。

ターゲットシステムの外部環境をモデル化して信号処理系のシミュレーションにおける相互作用の対象に含めることは有効な方法であるが、精度の高いモデルを作ることの難しさやそれを用いてシミュレーションを実行するのに要する時間的限界などから、完全にその方法を導入することができるとは限らない。図4に信号処理ソフトウェアの設計フローをしめす。ここでは外部環境モデルが信号処理モデルのシミュレーション段階で取り入れられているが、上記のような困難があるので近似モデルであることば注意されたい。

² 最適化:ここではよりよい結果を得るように調整を行うことをさしており、厳密な意味である規範に照らしてその最大値または最小値を与えるようにすることを必ずしも意味してはいない。これ以降でも同様の意味で使用している。

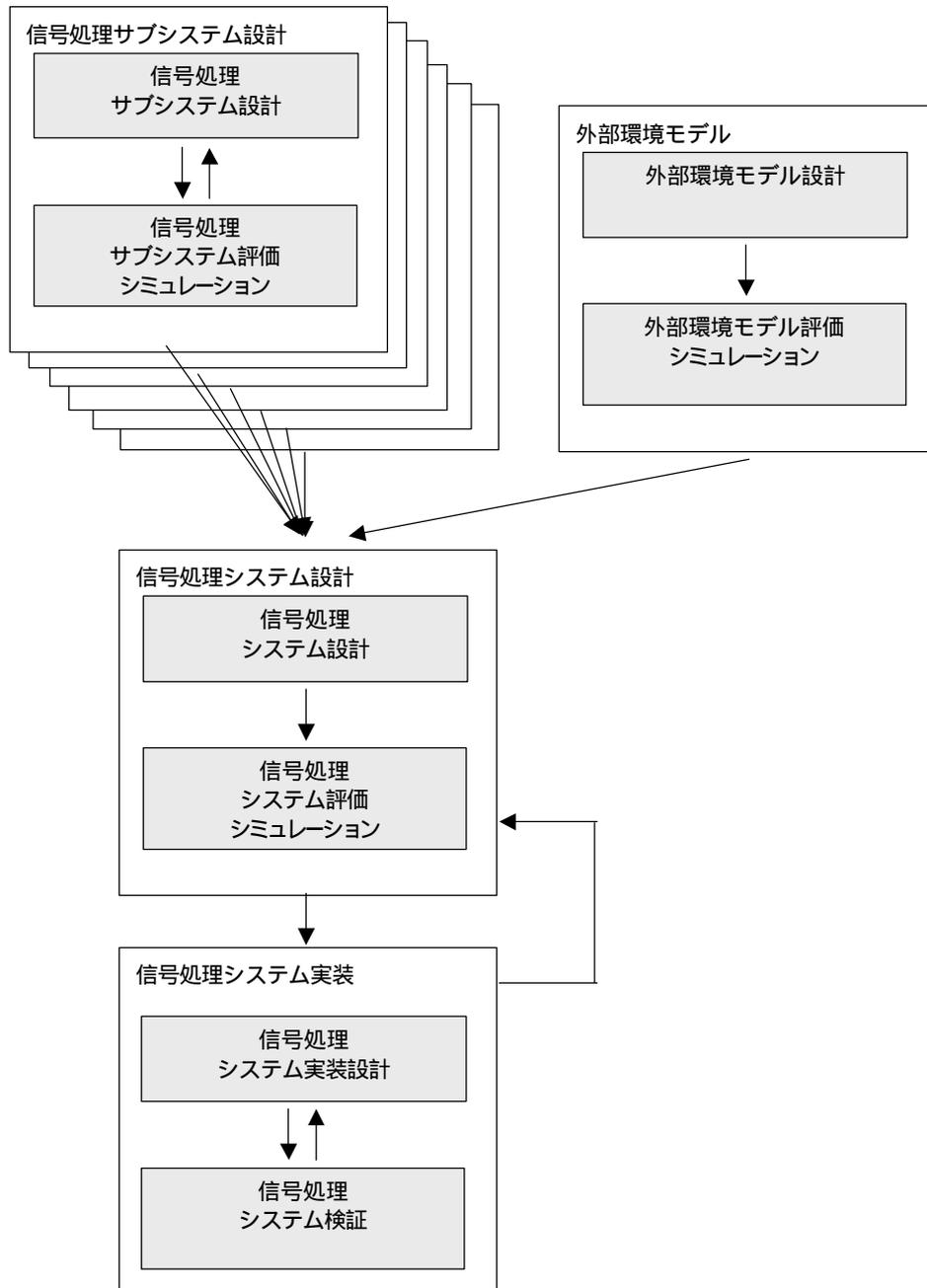


図 4 信号処理ソフトウェアの設計フロー

処理時間と性能³のトレードオフの解決

プロセッサの提供する処理能力に対して実装したソフトウェアの処理量が超過する場合、それを低減するためにリソースの仕様効率の改善とともに、性能マージンの引き下げを行う必要に迫られる場合がある。通信

³ 性能: 処理の品質の意味で用いている。以降も同様

性能を決定するファクタは一般に多岐にわたり特定の部分の性能の引き下げが全体の性能に与える影響を予測することは必ずしも容易でない。一例として受信信号の伝送路適応を行うイコライザの処理時間の低減を図る場合を考えると、適応フィルタの段数を切り詰めることで処理時間を短縮することはできるが補正可能な範囲がその分狭まることになる。伝送歪の分散の少ない回線に対してはそれによる性能の低下は無いが広い範囲にわたって伝送歪が生じるような回線を経由した通信においては受信性能の劣化として表面に現れる。

3.3 通信ソフトウェアとしての課題

通信路モデルと相互運用性の検証

通信路を含めた通信端末間の接続能力と相互運用性の検証は図 5に示すような実際の端末装置と通信路のシミュレータを用いて行うことが多い。

通信機能を提供するためには通信路を介して接続される可能性のある端末装置との接続性と相互運用性を仕様で規定される水準以上にする必要があり、この検証をソフトウェア設計の前段階で行うことは難しい。それは、ソフトウェア設計前段階でシミュレーションによる接続性および相互運用性の検証を行うための対向端末装置のモデルを得ることが実際上困難であるためである。たとえば、すでに稼働している端末装置との間での接続動作の確認のための試験を行うのであるが、通信路状態に依存する試験項目などでは非常に多くの組み合わせが発生することから、試験のカバレッジを十分に確保しようとすると多くの時間を要することになる。特に動作タイミングやタイムアウト判定閾値に関する検証等においてその点が著しい。

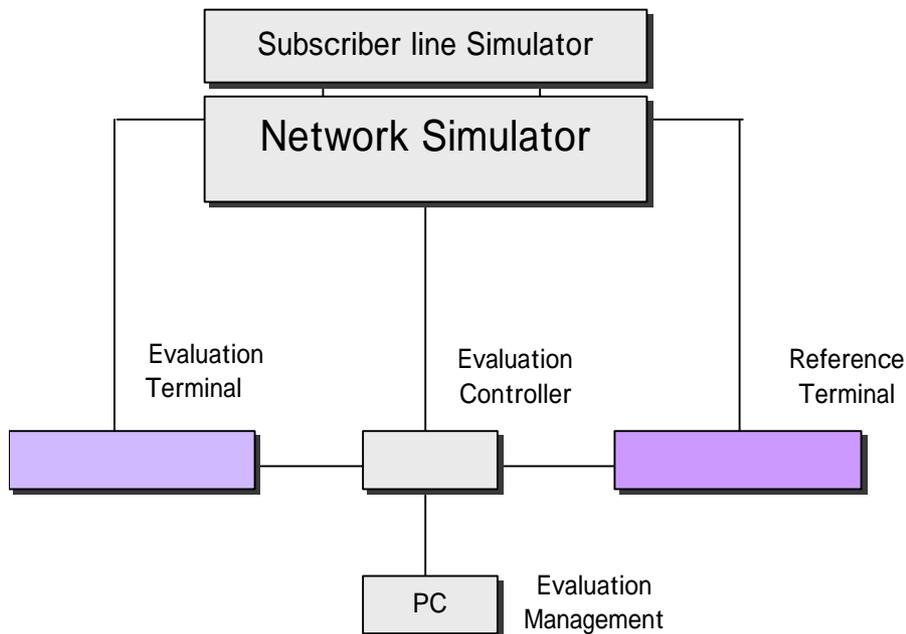


図 5 通信機能の評価システム

4 組込みソフトウェア開発に関する考察

前節で述べてきた課題に関して、組込みソフトウェア開発の手法を中心に考察を加える。

組込みソフトウェア開発における再利用の方法について

ソフトウェア開発における実装工程以降のフェーズで不可避免的に発生する最適化の工程を認めた上で、再利用による生産性の向上を図ろうとすると、再利用のモデル/コードライブラリとして位置付ける複数の段階の設定が必要である。たとえば、再利用のレベルとして、信号処理アルゴリズムのブロックのレベル、高級言語のソースコードのレベル、ターゲットプロセッサのソースコードのレベルなどである。これは、再利用部品の抽象度のレベルでもある。同一のプロセッサでコードレベルの最適化をする場合には上記第三のレベルの部品化を使用するのが適当であり、別のプロセッサ上を実装する場合には第二のレベル、処理のモデルパラメータレベルでの修正が必要と判断される場合には第一のレベルの部品をベースとするといった使い分けができることが望ましく、シームレスにそうした再利用部品の製作/保守/利用をサポートする開発環境を備えることが有効と考えられる。

潜在リスクの早期発見とスパイラル開発

オブジェクト指向開発プロセスでも採用されているスパイラル型の開発手法は、下流工程での制約により生じる上流工程に対する修正箇所を早期に発見しやすくする。その点では同手法の有効性を認めることができるが、こうした小型機器における組込みソフトウェア開発においては各フェーズ間の移行を適切に管理することが一層重要となる。それが不十分であると、作業が開発工程の多くのフェーズに無秩序に拡散する危険性がある。また、ある開発フェーズで発生した変更が確実に他のフェーズおよび開発に携わるメンバーに伝達され、それに対応した更新が行われる必要がある。

ソフトウェア設計モデルと信号処理モデルの統合化

組込みソフトウェアにおいても、特に機能階層において上位階層に位置付けられるソフトウェアではオブジェクト指向のソフトウェアモデルを採用することで、開発効率の改善を期待できる。一方信号処理ソフトウェアをはじめとした下位階層の部分については、議論の中心は主として手続きにおける処理アルゴリズムと実装の最適化にある。これらを有効に統合するようなソフトウェアモデルが望まれるところである。

通信路モデルと接続性/相互運用性の評価と問題解析

接続性/相互運用性の問題には通信系のソフトウェアおよびハードウェアの信号処理のレベルおよび通信プロトコル処理のレベルさらにアプリケーションのレベルまで関わってくる。検証のカバレッジを上げることに関しては自動化によって比較的容易に実現できるのであるが、問題個所の特定と対処は設計/実装工程の担当者の直接的な関与を必要とすることが多い。したがってこの工程の効率の改善としては、現象としての問題個所を的確に発見し伝える機能を開発環境に付与することなどが考えられる。

5. むすび

本報告では、通信系の組込みソフトウェアの開発において見られた開発上の問題のいくつかを紹介しそれらを改善する方法について考察を加えた。これらの中には、現時点ではほぼ問題の解決に役立てられると見られる技術が開発ツールとして提供されるようになってきているものもあるが、それに代わらず今後のソフトウェア工学的な研究と開発ツールとして製品化が待たれるものも含まれている。

参考文献

- [1] Bruce Powel Douglass; Real-Time UML ,Second Edition Addison-Wesley ISBN 0-201-65784-8, 2000
- [2] 日経エレクトロニクス 2001年5月7日号 no.795, pp117-137 日経 BP