組み込みシステムを対象にしたアーキテクチャ指向開発手法の提案

2R－6

川口 晃†、元木 誠‡、岸 知二†
NEC †マイコンソフト開発環境研究所、‡C&C 研究所

1. はじめに

近年、マイコン組み込み品の高機能、短納期化が進んでおり、組み込みソフトウエアも肥大化、短納期化している。一方、オブジェクト指向開発はその再利用への適合性や修正改造に対する強さの点で、組み込みソフトウエアにも有効だと思われるが、組み込みソフトウェアは、実時間性や実装制約が厳しいことが多く、上流工程の考慮は行いにくかった。そこで我々は、オブジェクト指向開発のメリットを生かしながら、実時間性や実装制約にも対応できる組み込みソフトウェア向けの開発手法の検討を進めている。本稿では検討のねらいや、基本的なアプローチについて述べる。

2. 組み込みソフトウエアの特性

組み込みソフトウエアには次の特性がある。

1. 実時間性

組み込みソフトウエアは、ハードウエアとのインタラクションが多く、実行時に実時間性が要求される場合が多い。この場合、オブジェクト指向設計で用いられる静的なオブジェクト間の関連を基本とした設計手法は、ソフトウエアの実時間性を議論するには不十分である。

2. 実装制約

処理速度、実装容量等の制約が厳しく、オブジェクト指向分析、設計で得られた再利用や修正の容易なモデルを素直に実装することが難しい。

3. 実行構造の共通性

組み込みソフトウエアの実行構造は共通化されておらず、実装環境によって独自の構造が使用されているが、実装環境、および実装環境の制約を適切に取捨選択したうえで抽象化すれば、いくつかの主要な構造で代表することが可能である。

3. ねらい

1. ユースケースによる設計

設計手法のベースとして、実時間性やインタラクションを議論し易いユースケース [1] を活用する。

2. 実行構造モデルの導入

設計ユースケースを実行構造モデルに対応づける。実行構造モデルは実装の基本モデルであり、ユースケースの参加オブジェクトとコーディネータで構成される。コーディネータは、参加オブジェクトの活動の調整を行うもので、基礎アーキテクチャで組み立てられる。基礎アーキテクチャは、組み込みソフトウエアの実行構造を構成する基本的な要素で、特定の実装環境に依存しないように抽象化されており、次節で述べる実装アーキテクチャに対応した形であらかじめストックされている。組み立てられた実行構造モデルは、特定の実装環境に依存せずにソフトウエアの実行構造を議論できる抽象化モデルである。

3. 実装アーキテクチャの導入

実行構造モデルをさまざまな実装環境に適用するため、基礎アーキテクチャに対応した実装アーキテクチャを各実装環境毎に整備、ストックする。実装アーキテクチャは、各実装環境の定石やテクニックを採り入れて、その環境で基礎アーキテクチャを実現するための最適なアーキテクチャである。

4. アーキテクチャ指向開発手法

図1を用いて我々が提案する設計手法を説明する。

"Architecture-Oriented Development Approach for Embedded System Software", Akira KAWAGUCHI,Makoto MOTOKI,and Tomoji KISHI, NEC Corporation

4.1 ユースケースを用いた設計モデルの構築

ユースケースを用いてオブジェクト間のインタラクションに着目して分析、設計を行い、必要な処理内容と時間制約を明らかにする。なおユースケースは参加者の抽象度のレベルで階層化できる。

4.2 最適な実行構造モデルの組み立て

前節で述べた階層化設計モデルから、その処理内容、時間制約に則した実行構造モデルを組み立てる。コーディネータの組み立ては、処理内容、時間制約の実現に適切な基礎アーキテクチャを用いて行う。基礎アーキテクチャの具体例としては、並行動作、順列動作、同期通信、非同期通信などが挙げられる。

4.3 実装アーキテクチャを用いた実装

基礎アーキテクチャの実装方法は実装環境に依存しているので、ストックされた実装アーキテクチャから、実装環境に適したものを選択することで、実行構造モデルを実装環境に実装できる。従って同一の実行構造モデルから種々の実装環境に適した実装が行える。代表的な実装環境としては、ハードウエア制御のように時間制約を重視する環境、また携帯電話のように実装メモリ容量を重視する環境、さらに機能追加、再利用性を重視する環境等があげられる。また代表的な実装アーキテクチャとして、同期通信の例を挙げると、時間制約を重視する場合はCPUの割り込みによる実装、実装メモリを重視する場合はフラグによる実装、機能の追加を重視する場合はリアルタイムOSによる実装などが挙げられる。

5. 事例

図2に本手法をコードレスホンに適用した事例における実行構造モデルの例を紹介する。この例は、外線着呼を検出する着信回線ユースケースから組み立てられたモデルである。用いられている基礎アーキテクチャは、着呼信号検出に非同期イベント検出、タイマリセットと外線着呼メッセージの送信のために同期メッセージの送信、着信回線オブジェクトと2.5秒タイマの実行のために並行動作の各アーキテクチャによって組み立てられる。(なお図の記法は暫定的なものである。)

6. おわりに

本稿では、組み込みソフトウエアのアーキテクチャ指向開発手法を提案した、この手法により組み込みソフトウエア特有の実時間性、実装制約の問題を考慮しながらオブジェクト指向開発の持つメリットを享受することができる。今後は、基礎アーキテクチャ、実装アーキテクチャの整備、適切な基礎アーキテクチャの選択と実行構造モデルの組み立てメカニズムの開発、実行構造モデルの表現法、支援環境、実装作業との連係法等の問題を解決して行く予定である。

図 1: 設計手法の概念図

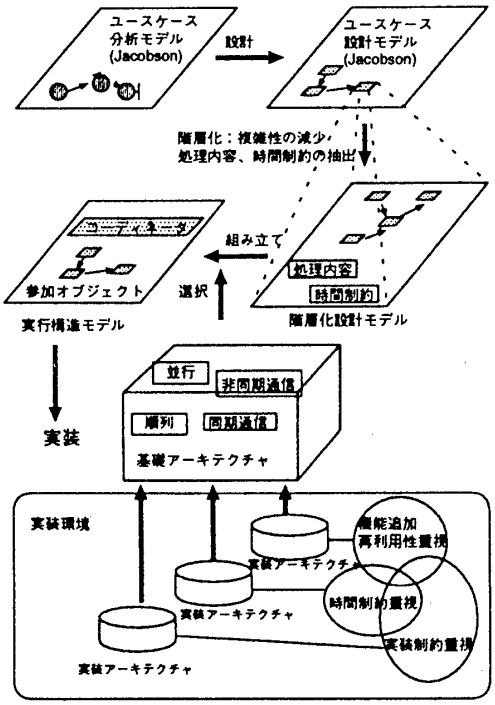
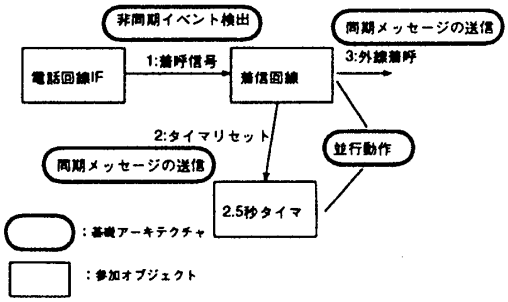


図 2: 実行構造モデルの例



参考文献

[1] Jacobson,I. et al.:Object-Oriented Software Engneering, Addison-Wesley,(1992).