

Title	SES アプローチに基づいた実時間制約を考慮した組み込みシステムの実装法
Author(s)	森本, 大喜
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1329
Rights	
Description	Supervisor:片山 卓也, 情報科学研究科, 修士

SESアプローチに基づく

実時間制約を考慮した組み込みシステムの実装法

森本 大喜

北陸先端化学技術大学院大学 情報科学研究科

2000 年 2 月 15 日

キーワード：SES アプローチ、アスペクト指向プログラミング、組み込みシステム、リアルタイム・オペレーティングシステム

各種の機器に組み込まれてその制御を行うシステムを組み込みシステムと呼ぶ。近年における半導体技術の進歩に伴うマイクロプロセッサやメモリの低価格化により、組み込みシステムの応用分野は拡大している。現在、このようなハードウェア技術の進歩が制御対象となる組み込みシステム機器の規模を劇的に増大させている。そのため、システムの大規模化・複雑化に対して従来の開発手法による組み込みシステム開発が非常に困難になってきている。こういった状況では、システム全体の構造をどのように設計して実現するかが重要であり、この領域においてもオブジェクト指向開発手法が注目されている。しかし、組み込みシステム開発では、実時間性、リアクティブ性、ハードウェア構成などの非機能的要件と呼ばれる厳しい制約が存在するため、現在提案されているオブジェクト指向方法論ではこれらの要件を十分に扱うことができない。よって、これらの制約を扱う適切な設計手法を提案した。

本研究では、青木により提案された SES アプローチに基づき、下流工程においてどのように非機能的要件を取り扱うべきかについて提案する。SES アプローチは、組み込みシステムを対象としたオブジェクト指向開発手法である。このアプローチでは SES(Synchronized execution sequence)と呼ばれる処理列を単位とした SES モデルを構築する。現状では SES アプローチは実際のシステム開発に適用するまでには至っていない。本研究では、電話機の組み込みシステム開発という特定のシステム開発に焦点を当て、これについて SES アプローチに基づく実装を行った。組み込みシステムではリアルタイム・オペレーティングシステム (RTOS : Real-Time Operating System) が使用されるため、電話機組み込みシステムの実装には RTOS を用いて行った。RTOS 上に実装されるソフトウェアは、タスクにより構成される。タスクは RTOS が持つシステム

コールを用いて実行制御、及び、通信を行う。**SES** モデルには、これらの要件は記述されない。そのため実装を行うためには、個々の **SES** の処理を実装すること以外に、**SES** をどのようにタスクへ詰め込むか、**SES** の実行をどのように制御するか、処理の間の通信をどのように行うかを決定しなければならない。そこで、アスペクト指向プログラミング (**AOP: Aspect-Oriented Programming**) の手法によってこれらの要件を扱う。

AOP ではコンポーネントプログラムと複数のアスペクトプログラムにより対象システムを記述する。これらのプログラムは **Weaver** と呼ばれる合成器を用いて合成することにより、目的のプログラムを獲得する。

今回の実装では、**SES** のタスクへの詰め込み方、タスクのスケジューリング、タスク間通信の 3 点をアスペクトとして捉えた。これらに対して、タスクへの詰め込み規則記述言語、スケジューリングアスペクト記述言語、タスク間通信アスペクト記述言語を決定し、手作業で実行可能なプログラムまでの過程を示した。

最後に、**SES** アプローチに基づき **AOP** によって実装した実時間制約のある電話機組み込みシステムに対して、これら 3 つのアスペクトについてどのような関係があり、互いにどういった影響を及ぼすのかについて考察を行った。その結果、タスクの詰め込み方により、スケジューリングやタスク間通信の実現方法に影響を及ぼすことがわかった。例えば、同一タスクに詰め込まれた **SES** 同士が通信を行う場合と、異なるタスクに詰め込まれた **SES** 同士で通信を行う場合では、その通信の実現方法が異なる。また、タスクのスケジューリング方法を変えることで、**SES** のタスクへの詰め込み方に規制が働く。この例では、時間的要因によって起動するタスクを同一タスクにまとめるか、まとめないかによって、スケジューリング方法が異なる。これら 3 つの内でどれに重点を置いて **Weave** するかでトレードオフの関係を生じることがわかった。