
セミナー名「モデル駆動開発を組み合わせたソフトウェアプロダクトライン開発入門」

講師

セイコーエプソン株式会社 / SESSAME 島 敏博 氏

概要

近年の大規模化・複雑化する組込みソフトウェア開発の改善手法として注目されている MDD(モデル駆動型開発)と SPL(ソフトウェアプロダクトライン)。方法論やツール紹介ではなく、実際の現場ではこのアプローチをどのように理解し、活用・実践しているのか。難しい学術的な内容ではなく、本質的に何が取り組みとして重要なポイントなのか。これらのポイントを第一線で実践している現場から、わかりやすく解説します。

対象者

組込みソフトのモデル駆動開発、プロダクトライン開発に興味のある方で、以下のような取り組みを行っておられる方、または以下のような課題認識を持っておられる方。

- 組込み機器開発を行っている (C でも C++でも Java でも)
- 派生機種開発に取り組んでいる
- ソースコードの行数が50万行を超えた
- 大規模になりすぎて、全体がわかりにくくなってきた
- よく似た不具合が、派生機種にまたがって出てしまう
- 試作コストを下げたい
- メカ屋さん、エレキ屋さんにふりまわされている
- オブジェクト指向設計の経験がある
- UML の読み書きができる
- アーキテクトになりたい / アーキテクトの養成に苦労している

モデル駆動開発(MDD)ツールのユーザーである必要はございませんが、セミナーの中では、astah*、Rhapsody を使用した図面で説明いたします。

内容

第1部

ソフトウェア設計方法論の成功と限界

- 構造化設計の成功と限界
どうして限界が来てしまうのか。
- マルチタスク設計の成功と限界
タスクに分けると見通しがよくなった。
しばらくはマルチタスク設計でうまくいったけど、、、
- オブジェクト指向の成功
なぜうまくいったのだろう。
- そして、オブジェクト指向の限界
“オブジェクト指向的“にはよいモデルだが、、、
プロダクトライン開発を考慮するタイミングとは

第2部

プロダクトライン開発入門

- クラス設計からアーキテクチャ設計へ
凝集度と結合度
インターフェースと実装を分離する
固定部と変動部
PC シミュレーションする
OS 依存を吸収する
プラットフォームとプラットフォーム非依存アプリケーション

- 固定部と変動部を分けて開発する
 - フレームワーク+アプリケーション
 - 何がアーキテクチャなの？
 - フレームワークの構築
 - 2つの概念に分けて開発する
 - ソースコード量削減
 - 変動部の中に固定部を見つけ出す
- モデルで設計する。
 - モデルの位置づけ
 - モデリングツールを使う
 - これまでの『手作り開発』
 - これからの『モデル駆動開発』
 - モデルで設計し、ソースコードは自動生成
 - もちろんテスト駆動開発
 - モデル駆動開発の流れ (Rhapsody の例)
 - 両輪のマネジメント
 - 品質の向上効果

第3部

プロダクトライン開発・モデル駆動開発 FAQ

- 導入プロセスについて
 - いつとりかかればいいですか？
 - どこに適用できますか？
 - オブジェクト指向開発、モデル駆動開発といっても上司に理解してもらえないのです。アーキテクトの養成がたいへん難しいです。
- モデルからのコード生成について
 - コードサイズが大きくなるか心配です。
 - ほんとうにモデルからソースコード全体が出てくるんですか？
 - どうして不具合がなくなるんですか？
- モデルの作成について
 - 既存資産のリバースエンジニアリングできますか？
 - 分析モデルから設計モデルにどうやって移していますか？
 - 状態マシンはどこまで複雑になっても大丈夫ですか？
 - モデル駆動開発の部分的な適用は可能ですか？
- モデル検証、デバッグについて
 - モデルはどのように検証するのですか？
 - シーケンス図の役割ってなんですか？
 - ソースコードデバッグできますか？

スケジュール

開催日	時間	内容
2月21日(月)	13:00~17:00	モデル駆動開発を組み合わせたソフトウェアプロダクトライン開発入門
	17:00~17:10	休憩
	17:10~18:00	交流会(希望者のみ)