

拡張可能なグラフィカルエディタへの多階層モデリングフレームワークの適用に関する評価

株式会社富士通研究所 木村 功作 kimura.kosaku@jp.fujitsu.com

開発における問題点

背景: グラフィカルエディタの開発
 ■ モデル駆動工学(MDE)のモデル変換, コード生成による開発自動化
 ■ プラグインを用いたエディタの使い勝手向上によるシェア獲得
 問題点: グラフィカルエディタが扱うモデル(例: データフローモデル)をうまく定義できない

手法・ツールの適用による解決

多階層モデリングの適用
 ■ 任意の階層数のモデルをシンプルに定義
 以下のフレームワークの中でグラフィカルエディタ開発にはどれが適しているか評価
 ■ Eclipse Modeling Framework (EMF)
 ■ Melanee
 ■ MetaDepth

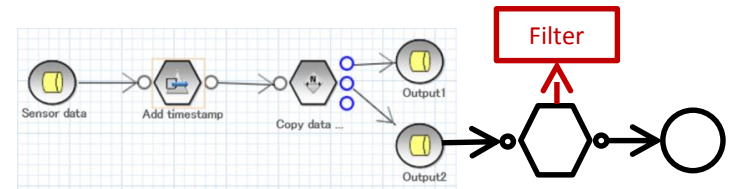
評価の観点と評価方法

どれがデータフローモデル(DFM)を最も忠実に定義できるか?
 → 各FWでの定義と元のモデルとの相違点をカウント

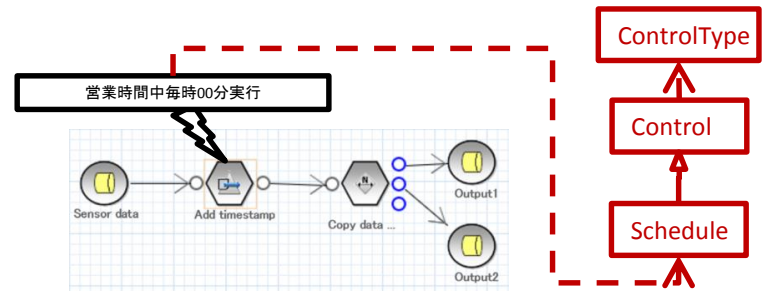
どれがモデルを最も容易に拡張できるか?
 → 以下の拡張での既存要素の変更箇所をカウント

通常階層	DFM階層		Class (EClass)
M3	M4	Meta Object Facility (MOF)	Class (EClass)
M2	M3	DFM要素の定義	Element
M2	M2	パレットに陳列される部品の定義	ProcessType, Process, Duplication, AddTimestamp, EventData, TemporaryData, SVMModel
M1	M1	キャンバスで編集されるインスタンス	Sensor data, Add timestamp, Copy data ...
M0	M0	実際のプログラムコード	タイムスタンプ付与, データ複製

M2: 処理部品の追加



M3: (処理, データ以外の)要素定義の追加



結果

フレームワーク	DFMとの相違点	既存要素の変更箇所		
		M2	M3	
EMF	5	0	0	😊😊
Melanee	0	0	1	😊😊😊
MetaDepth	3	0	4	😊😊😊

まとめ

- DFMを相違点無く定義でき, 既存要素の変更箇所も少ない**Melanee**が優位
 - ・ 片方向参照と参照のインスタンス化が定義できることに起因
- 実際の適用にはまだ足りないものが多い
 - ・ プラグインの仕組み
 - ・ エディタコード上でのモデル操作用API
 - ・ 既存MDEツールの対応強化