

Asakusa Frameworkへの取り組み

2014年11月28日

東芝ソリューション株式会社
ソリューションセンター
ソリューション生産技術部
生産技術担当

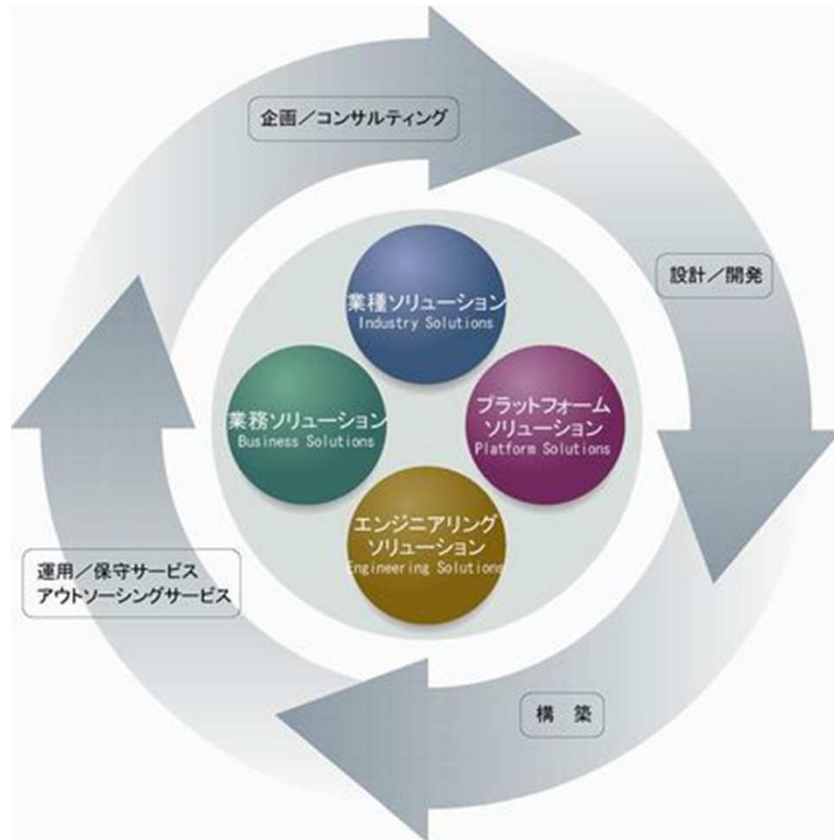
陸 振宏 (Riku Shinkoh)

会社概要

社 名	東芝ソリューション株式会社
本社事務所	神奈川県川崎市幸区堀川町72番地34
設立年月	2003年10月
資本金	235億円（東芝100%）
従業員数	4,300人（2014年10月現在）
代表者	取締役社長 錦織 弘信



会社概要



業種ソリューション

幅広い業種に対するコンサルテーション実績と現場の実務ノウハウをもとに、お客様の経営課題を解決するシステムインテグレーションを行います

業務ソリューション

東芝自身や業界のベストプラクティスをもとに、豊富な経験・ノウハウを活用し、業務を改革・改善するソリューションやサービスをご提供いたします。

プラットフォームソリューション

お客様の要件に最適なサービスとプロダクトを組み合わせ、システムの基盤を支えるプラットフォームをソリューションとしてご提供いたします。

エンジニアリングソリューション

最先端の技術と豊富な経験・実績をもとに、さまざまな産業分野に向け、新しいソリューションとシステムの形を創造します。

所属

ソリューション生産技術部

- 生産技術によるモノづくりの変革を使命とする部隊
- システム開発基盤「CommonStyle[®]」の開発・サポート
- Asakusa FrameworkはCommonStyle[®]の一つとしてサポート

アジェンダ

取り組み その1

顧客PoC事例

～製造業原価計算への適用～

取り組み その2

Asakusa Frameworkの生産性向上

～コード生成ツールの開発～

取り組み その1

顧客PoC事例 ～製造業原価計算への適用～

プロジェクト概要 某社 原価計算機能開発

現状

- ロット毎、商品毎の精緻な原価算出ができてない
- 品種の生産構成を変更した場合の原価シミュレーションが困難
 - ⇒ 低収益品からの撤退、高収益品への注力等の経営判断が後追いとなるリスク
 - ⇒ 意志決定や指示内容の妥当性を示すことができない

目指す姿

- 製造プロセスの実態を反映した、スピーディーかつ精緻な一品別原価の算出
 - ⇒ 損益把握のスピードアップと精度向上
- 構成変動に追従したシミュレーション環境の実現による、品種の最適な取捨選択
 - ⇒ より実際に近い情報に基づく意思決定

実行施策

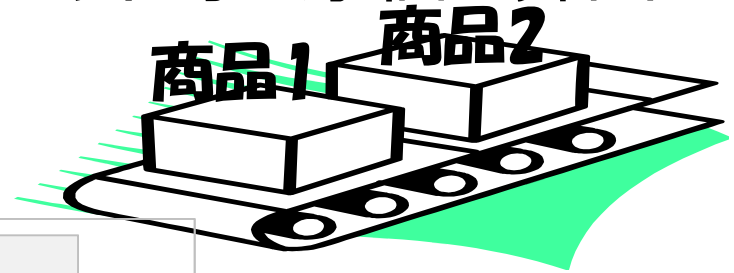
- 製造プロセスに沿った製造原価の積み上げ
 - ✓ 現行 : 実績トータルコストを標準原価で按分
 - ✓ 施策実行後 : 各製造工程毎に直接材/間接材といった要素の原価を計算し、積算
- 実際の使用度合いに応じた、装置原価配賦
 - ✓ 現行 : 各装置の減価償却費を標準原価で按分
 - ✓ 施策実行後 : 各装置の減価償却費を、実際の装置使用度合いを元に按分

目指す姿 To-Be

製造プロセスの実態を反映した商品毎、ロット毎の原価の算出

生産にかかった量だけを按分

各月の購入単価が変動するため、各月の費用からそれぞれ算出



		材料A			
直接材	商品1	4/1購入	100円	5/1購入	300円
	商品2	4/1購入	600円	5/1購入	700円
	商品3	4/1購入	500円	5/1購入	600円

		電気 1200円	ガス 1500円
間接材	商品1	200円	600円
	商品2	600円	500円
	商品3	400円	400円

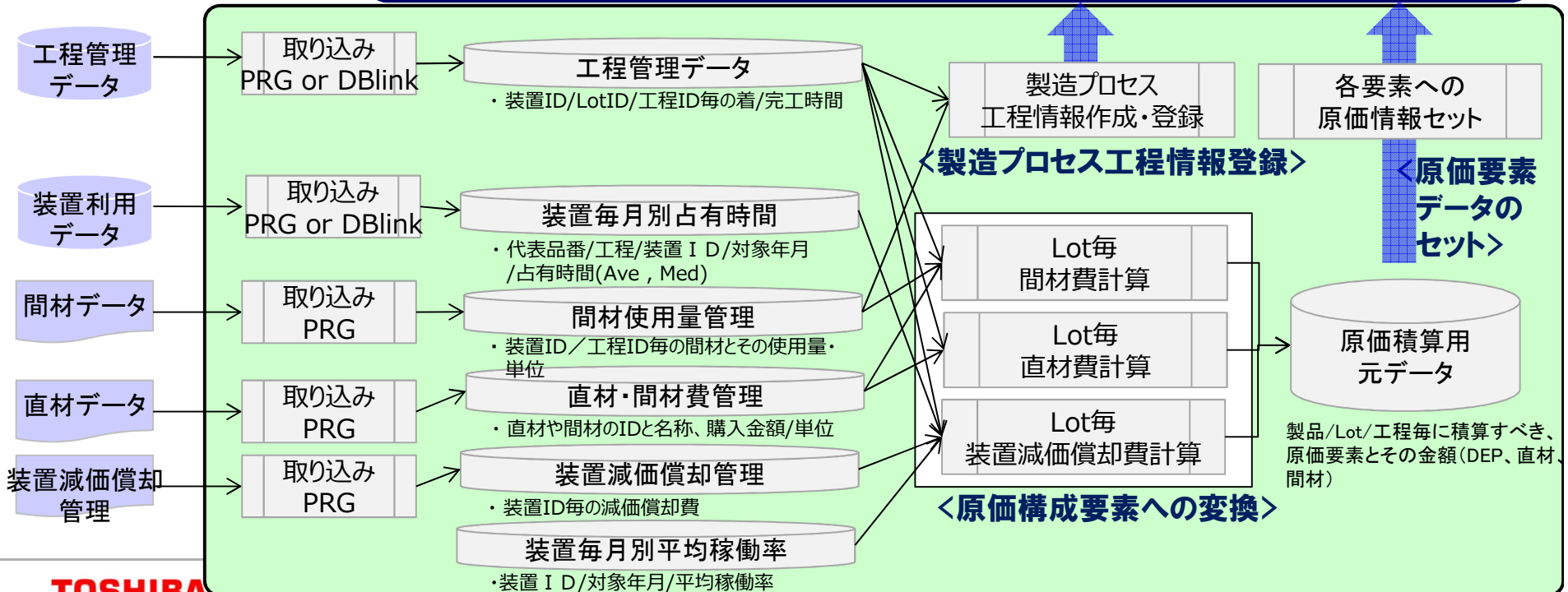
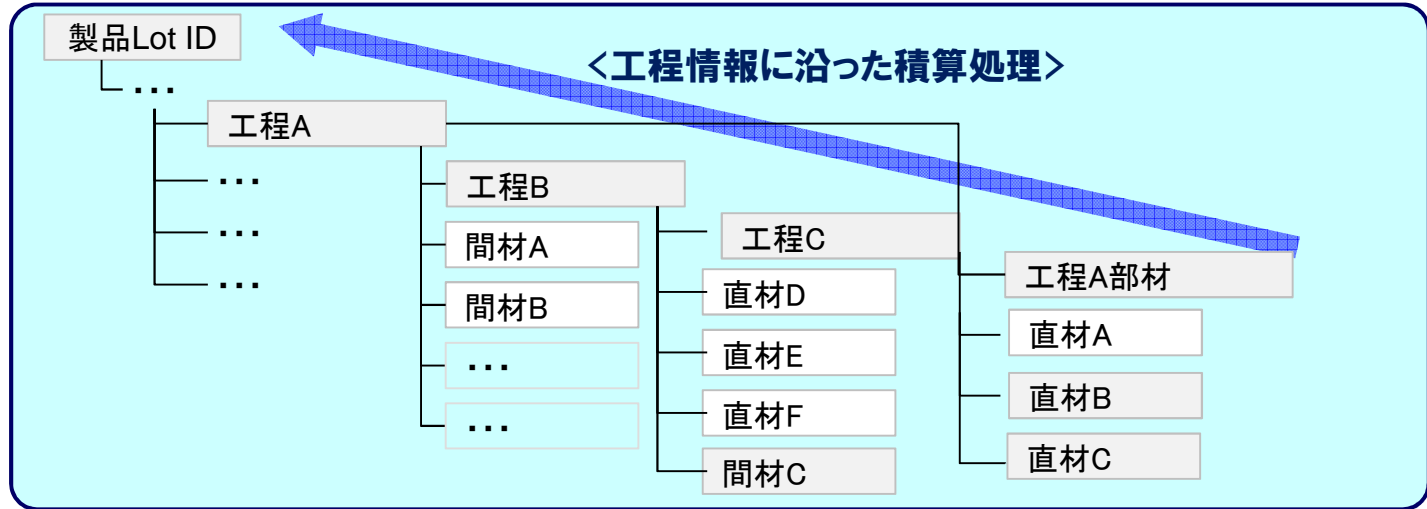
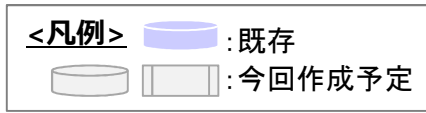
		装置A 1000円	装置B 1200円
装置減価償却	商品1	300円	300円
	商品2	500円	500円
	商品3	200円	400円

製造原価	
商品1	商品2
直接材	直接材
間接材	間接材
装置減価償却費	装置減価償却費

生産にかかった量だけを按分

生産数に対して按分

プロトタイプ概要



検証結果

■データ量

製造履歴（6ヶ月分）、装置占有時間、歩留り、装置稼働率など
計1億3千万レコード（約7GB）

■検証環境

以下スペックを4台

CPU : Xeon X5687(3.6GHz/4コア) x 2

MEM : 72.0[GB]

Disk : 1.0[TB]

OS : Linux

DBMS : Oracle DB

■性能

当初使用予定のパッケージ製品 : 5ロットで 約36分
Asakusa/Hadoop : 12,305ロットで 約28分

大幅に処理時間を短縮

PoCからの課題

- フロー設計からDSLによる実装が煩雑
- 仕様変更が発生した際の処理フローの変更が煩雑
- デバッグのために処理途中のデータを参照したいが、並列分散環境であるため困難
- DB設計の作業がデータモデル定義作業と重複するため、作業量が増加

⇒ 上記の課題を解決するため、
コード生成ツールを開発

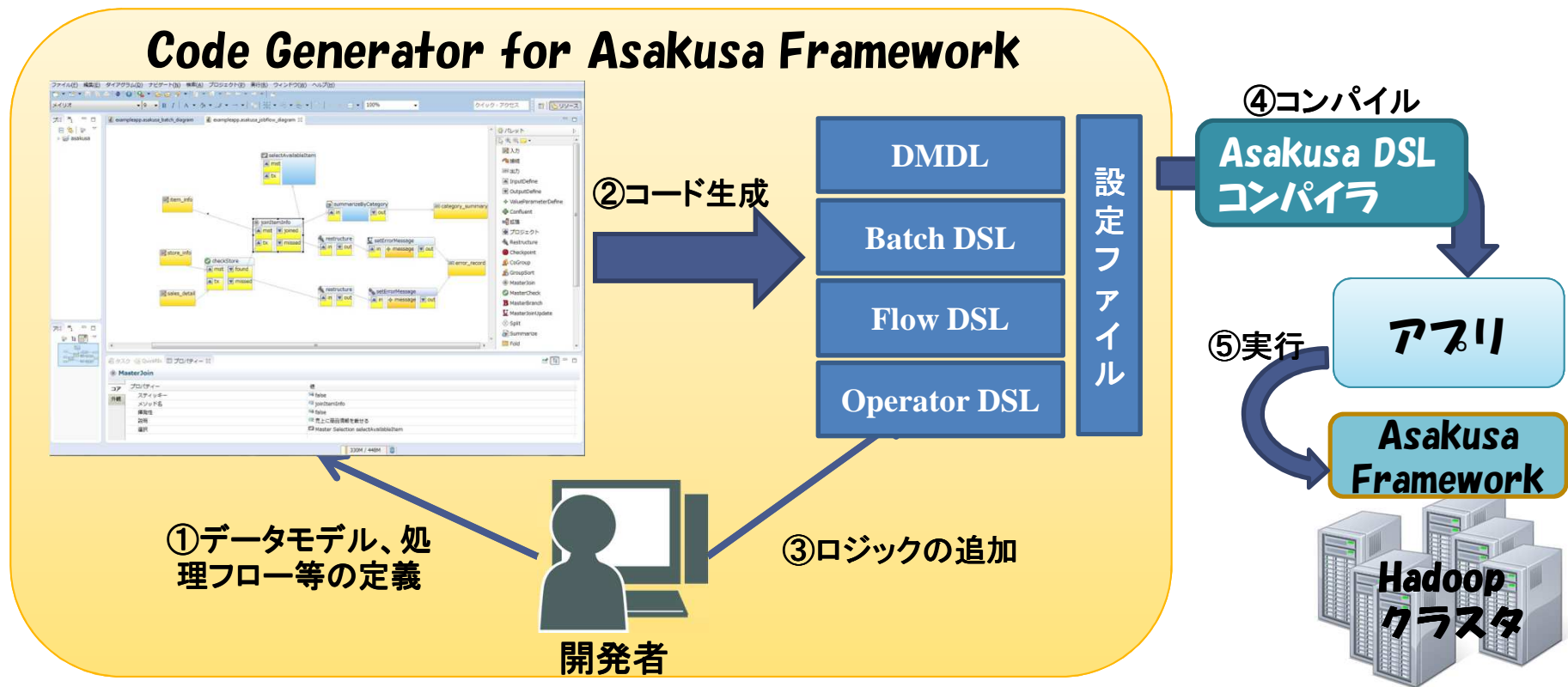
取り組み その2

**Asakusa Frameworkの生産性向上
～コード生成ツールの開発～**

Code Generator for Asakusa Framework

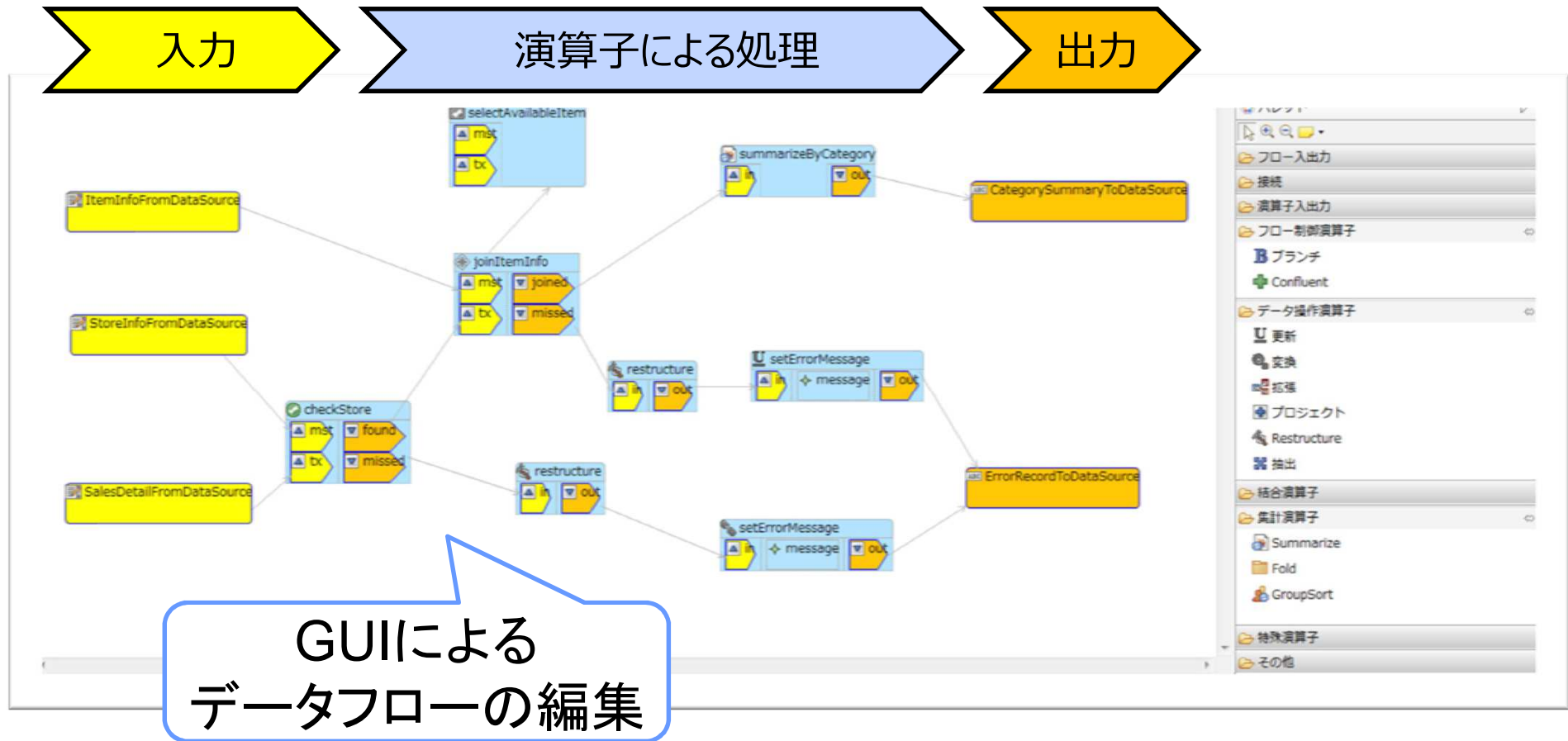
GUI上でデータフローを定義し、コードを自動生成するツールを開発

- Flow DSL、Batch DSLのデータフローの編集
- データモデルの生成
- デバッグ支援機能



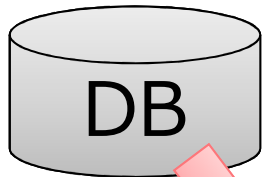
データフロー編集機能

- データフロー定義用のアイコンを利用し、入力→演算子処理→出力のように、GUI上でデータフローを定義し、コードを自動生成
- 仕様変更の際に、データフローを変更することで対応可能



データモデル生成機能

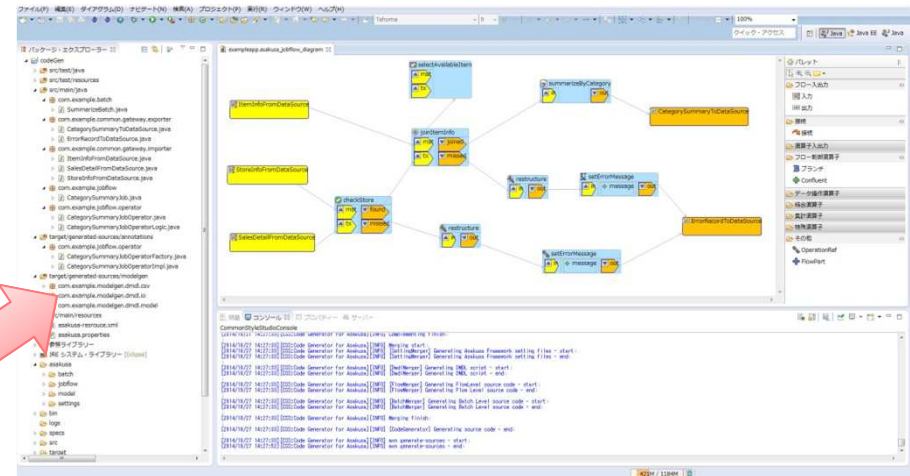
- DBスキーマからデータモデルを自動生成
- 自動生成により、データモデルの作成コストの削減と人為ミスによる不具合混入の防止



①既存のDBスキーマから
定義シートを生成

tableA			
tableB			
tableC			
tableD			
tableE			

DMDL
レコードモデル



③データモデルクラス等を自動生成

②レコードモデルをベースに、
結合モデル、集計モデルを定義



デバッグコード生成機能

- プロパティに設定するだけで、各演算子のOutputをCSVとして出力するデバッグ用コードを自動生成

CSVの出力を指定

プロパティ	値
Enable Debug All	## false
Jobflow Complete Class	## jobflow.CategorySummaryJob
Jobflow Type	## true
Operation Signature	## /** Operation DSLのシグネチャ **/
シグニチャー	## /** Flow DSLのシグネチャ **/
説明	## カテゴリ別に売上の集計を計算する。
名前	## byCategory

実行時に生成

演算子の出力結果 (CSV)

デバッグで利用



コード生成ツールの期待効果

**「取り組み その1」の事例で換算すると、
実装工期を1/3まで短縮できることが確認できた**

- **フロー設計から実装までの工期を削減できる**
- **仕様変更柔軟に対応できるようになる**
- **データモデルの作成工期を短縮し、人為的なミスを減らすことができる**
- **デバッグが容易になる**

まとめ

- **処理の高速化により、これまで出来なかったことが実現できるようになることを期待**
 - スピーディかつ精緻な（ロット別／個品別の）原価計算
 - 損益把握の速度および精度向上
 - 生産構成を変化させた場合の原価シミュレーション
 - 速い意思決定
 - 指示内容の具体化
- **PoCの課題から、コード生成ツールを開発し、Asakusa Frameworkを使ったアプリケーション開発の生産性を向上**

弊社のAsakusa Frameworkの情報ページ

Asakusa Framework関連サービスに関する情報は、当社の共通基盤
CommonStyle[®]情報ページをご覧ください

http://www.toshiba-sol.co.jp/core_tech/commonstyle/index_j.htm

ご清聴ありがとうございました

TOSHIBA

Leading Innovation >>>