

APS Summit 2015

生産システムのための 階層間連携言語（iHCL）の提案 ー システム間連携を模索したMESXの歩み ー

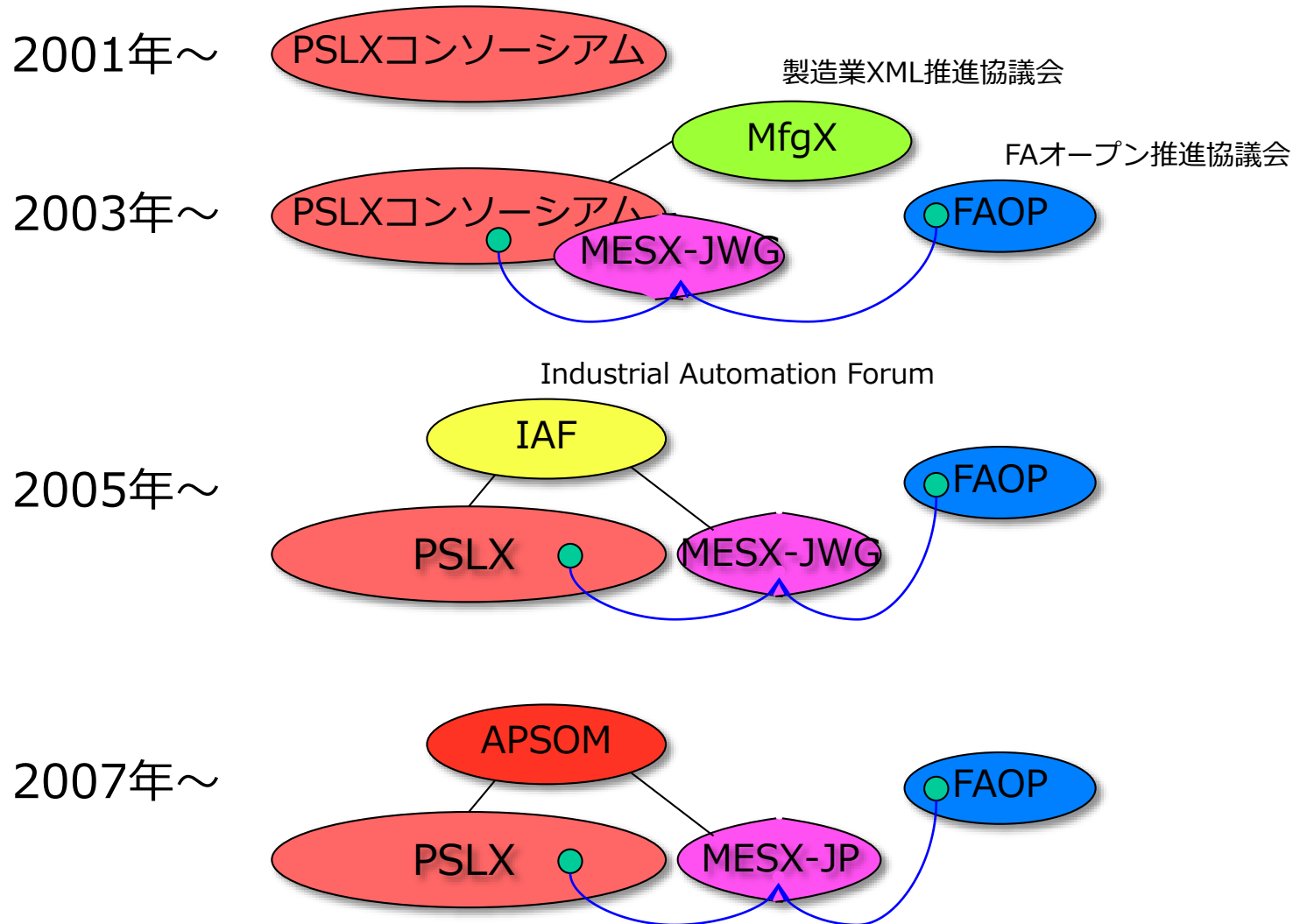
2015年11月25日

（株）情報システム総研
児玉公信

お話の流れ

- PSLXとMESX
- MESX4の基本原則
- MESX3との違い
- 実証実験
- まとめ

- プロジェクト組織の変遷

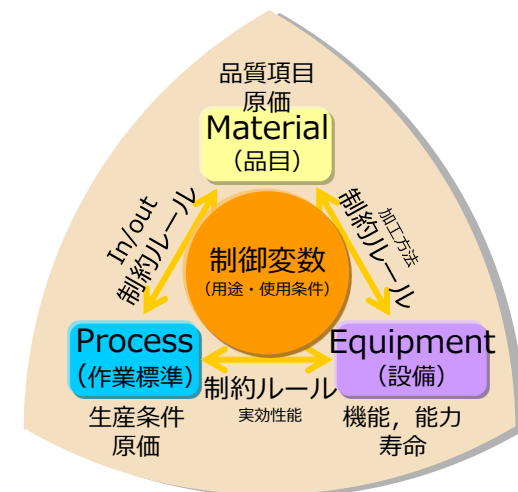


• プロジェクトの変遷

- PSLX1（2001～2003年）PSLXコンソーシアム
 - 受発注・生産管理システムのモデルおよびシステム間インタフェースを検討
 - 参照モデル（オブジェクトモデルとユースケース記述）を仕様書（勧告版）として公表
- PSLX2（2005～2013年）→APSOM（2007～）
 - IEC 62264の議論を経て、オントロジの概念を導入
 - アクティビティに注目して、PSLX1の仕様書（勧告版）を全面改訂
 - OASIS PPS：XMLメッセージ交換の委員会標準（2009～2011年）
 - PSLX2プラットフォーム（連携基盤）の提供
- PSLX3（2014～2015年）APSOM
 - エンジニアリングチェーン、デマンドサプライチェーン、組織階層でセグメント化して、アクティビティをモデル化
 - ドメインモデルではなく業務オブジェクトをモデル化
 - PSLX3プラットフォーム（連携基盤）の提供
- PSLX4（2015年～）APSOM

• プロジェクトの変遷

- MESX（2003～2005年）PSLXコンソーシアム+FAOP
 - 生産管理システムの垂直連携（生産計画からPLCまで）
 - XMLメッセージで一気通貫
 - MESの役割の再定義^[1]
- MESX2（2005～2007年）MfgX（MESX-JWG）
 - IEC 62264の機能階層モデル準拠
 - OASIS PPS準拠のメッセージ
 - PPS準拠で制御機器とのメッセージ検討
 - MESXプロトコルブック，実証評価^[2]
 - 動詞と制御メッセージの必要性
- MESX2.5（2007～2009年）APSOM（MESX-JP）
 - 制御機器を含む業務（故障，予防保守）対応
 - マス・カスタマイゼーションの検討→製造知識のモデル^[3]
- MESX3（2010年～2014）
 - OASIS PPS非準拠
 - クラウド対応
 - 製造ケイパビリティの試設計^[4]（ISO 16100）



- そしてこれから,
 - **MESX4** (2015年～) APSOM (MESX-JP)
 - 管理階層と業務相に基づくシステム分割
 - Customer-Performerモデルに基づくシステム間連携
 - シンプル化したメッセージとプロトコルの設計：iHCI
 - 計画・実行相は一個流し（バッチはバッチ制御機能を付加）
 - 計画・実行相－報告相とのインタフェースは対象外
 - 知識相－計画・実行相とのインタフェースは対象外
 - クラウド上での実証実験

MESX4の基本原則

- システム分割の原則

- 巨大なシステムを分割統治する（システムの階層化）

- サブシステムの単位

- 関心対象のライフサイクルを管理する→高い凝集性

- システム間連携

- 非同期一方参照メッセージによる連携→低い結合度

- 分割の基準：管理階層と業務相[5]

- 関心の階層

- 依頼（商い）

- 指示（資源）

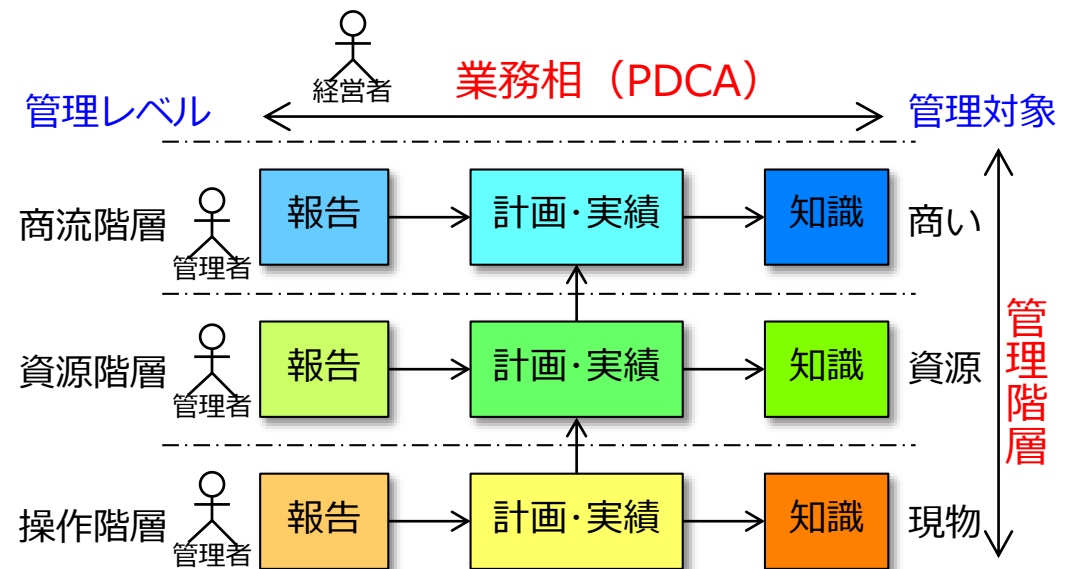
- 操作（現物）

- PDCAサイクル

- 知識（未来）

- 計画・実績（現在）

- 報告（過去）



念のために

- システム分割のよさ
 - モジュール性^[10]
 - 凝集性
 - データと機能の凝集
 - 同じ機能/データが散在しない
 - 余計な機能/データが混入しない
 - 結合度
 - データと機能の依存性
 - 依存性を最小に
 - データ表現からの独立
 - 言語による通信
 - DBのテーブル渡しは外部結合

偶発的凝集 (Coincidental)	低い
論理的凝集 (Logical)	
時間的凝集 (Temporal)	
手続き的凝集 (Procedural)	
通信的凝集 (Communicational)	
逐次的凝集 (Sequential)	
機能的凝集 (Functional)	高い



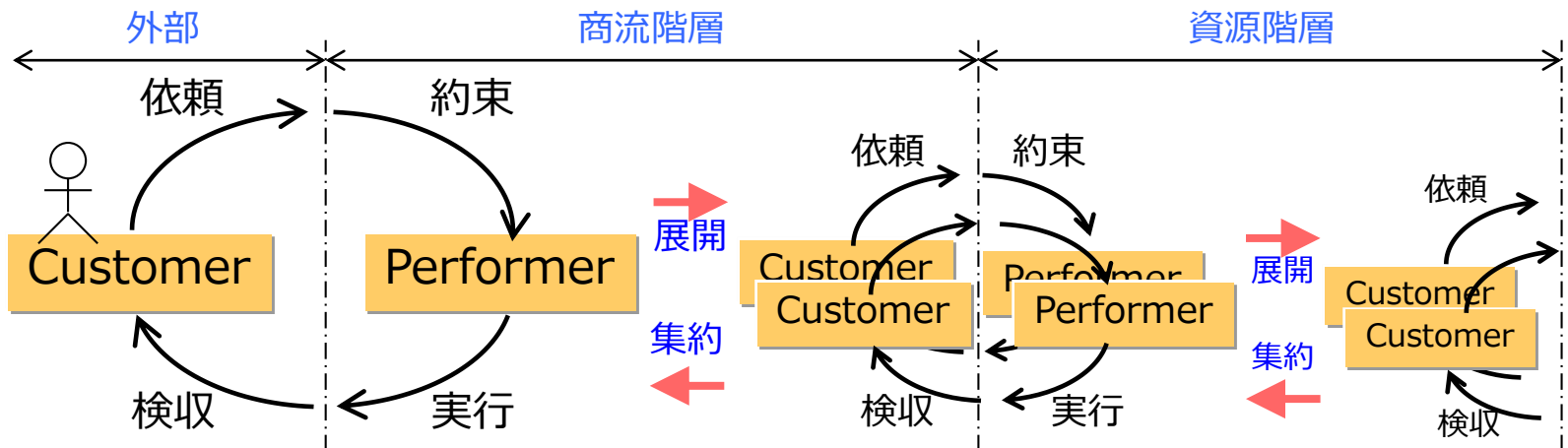
内容結合 (Content)	高い
共通結合 (Common)	
外部結合 (External)	
制御結合 (Control)	
スタンプ結合 (Stamp)	
データ結合 (Data)	
メッセージ結合 (Message)	低い



MESX4の基本原則

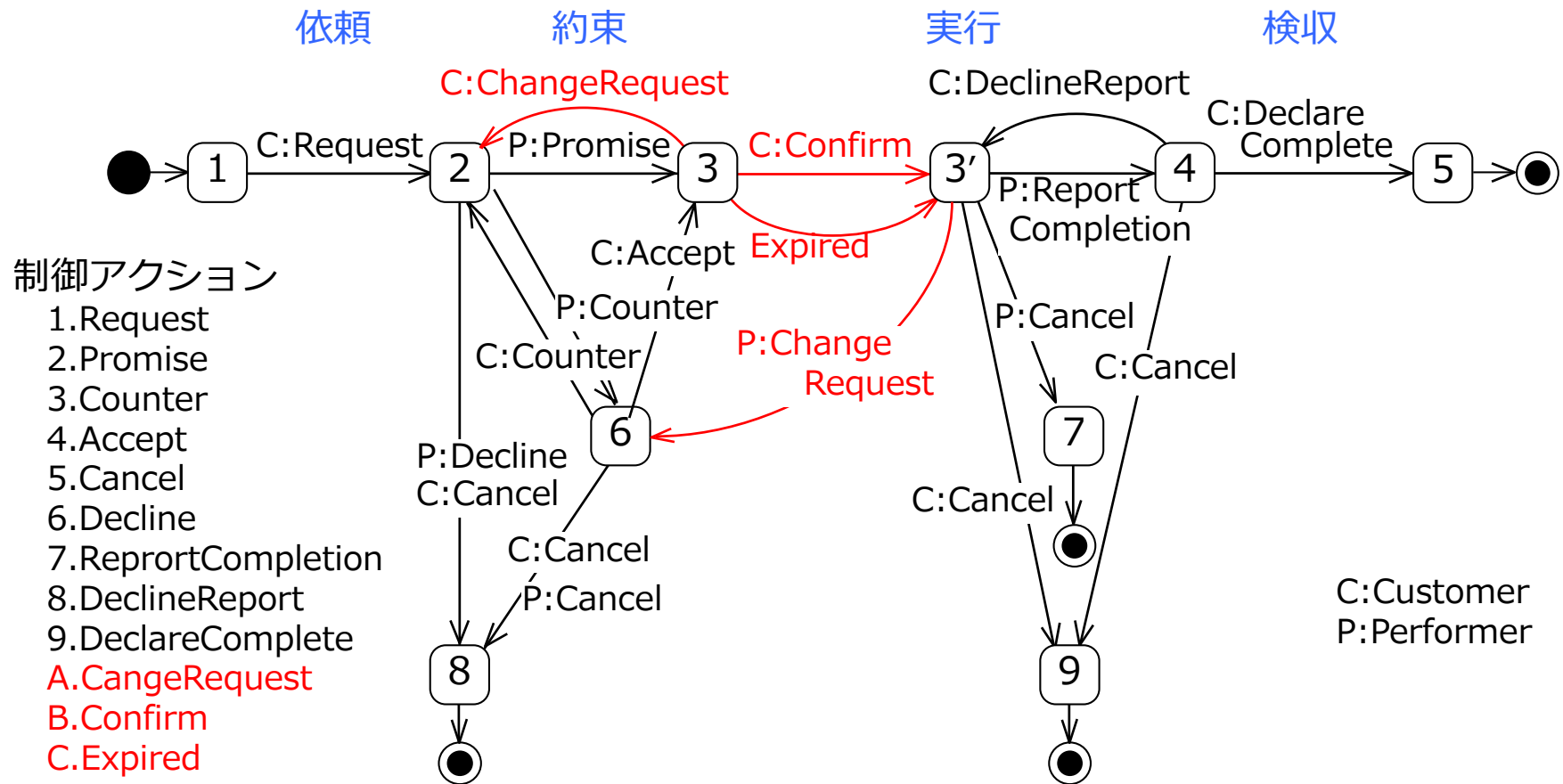
- システム間連携の原則

- Customer-Performerモデル^[6]に基づく
- 対話の再帰構造：依頼→約束→実行→検収
 - Customerのアクション
 - 依頼する, (報告を待って) 検収する
 - Performerのアクション
 - 約束する, 実行して (結果を報告する)
 - 作業展開して, (今度はCustomerになって) 依頼する
 - 結果を検収し, 結果を集約して報告する
 - 作業展開・集約の知識が必要



MESX4の基本原則

- 二者間で起こりうるプロトコル：9状態モデル
 - Customer-Performerモデルに基づく対話の構造^{[7][8]}
 - 制御アクションと状態遷移



MESX4の基本原則

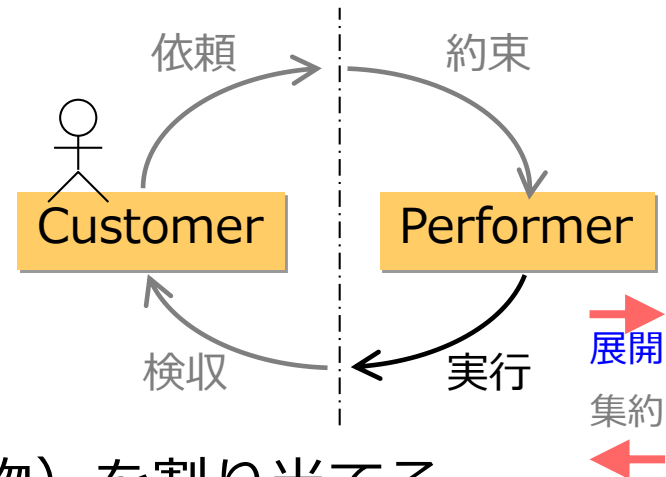
- 注文と作業展開の原則

- 作業展開はプランニング

- 部品展開, 所要量展開
 - 工程展開, 相対的順序関係

- 引当はスケジューリング

- 作業ごとの資源の取り合い
 - 未来の資源（時間, 空間, 現物）を割り当てる
 - どの階層にもプランニングとスケジューリングがある
 - スケジューリングは常に, 実績と新規を合わせたリスク
 - 資源管理層では, スケジューリングと呼ばれる
 - 資源のavailabilityに引き当てる
 - 操作階層では, シーケンシングと呼ばれることがある
 - 操作階層では, 動的リスクが行われることもある
 - 実施結果を基に状況を判断し, 作業を計画してリスク
 - 時々刻々変化する未来状況
 - 段取り最小になるように順序決め



MESX4の基本原則

- 注文と作業展開の原則（続き）

- マス・カスタマイゼーション

- 製品のバリエーションを決めるのは作業
- 作業のバリエーションを決めるのは知識
 - 品目（品目名＋仕様と仕様値）
 - 作業の仕方：作業内容と所要量
 - 資源の状況に制約される作業

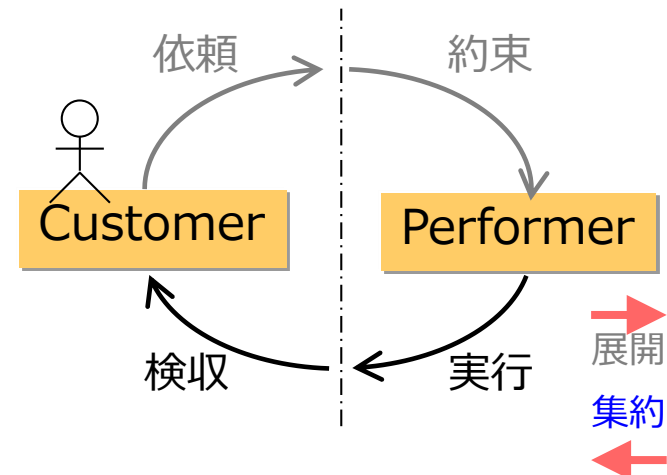
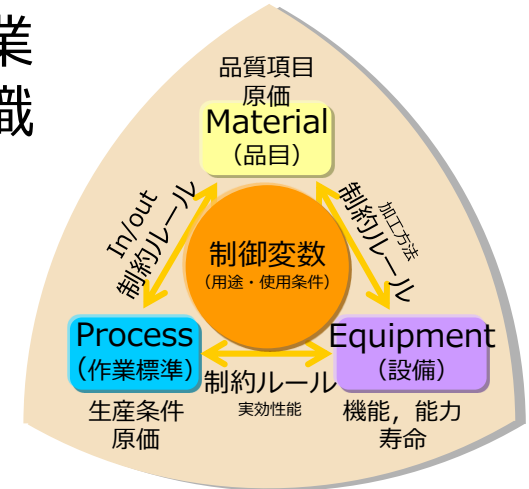
- 知識化

- 品目の変数，作業の変数，設備の変数
- 変数間の関係を定義（関数）

- 作業結果の集約

- 最終結果のまとめ

- 最初の注文に対する結果
 - 品質，納期，数量
 - 原価情報



MESX4の基本原則

- 通信の原則：非同期，一方向参照

- 下から上を呼ぶ

- notify + down (C→P) : pull

- up (P→C) : push

- 制御アクション

- request

- promise

- counter

- accept

- cancel

- ：

- 動詞（業務機能）

- 出荷：ship

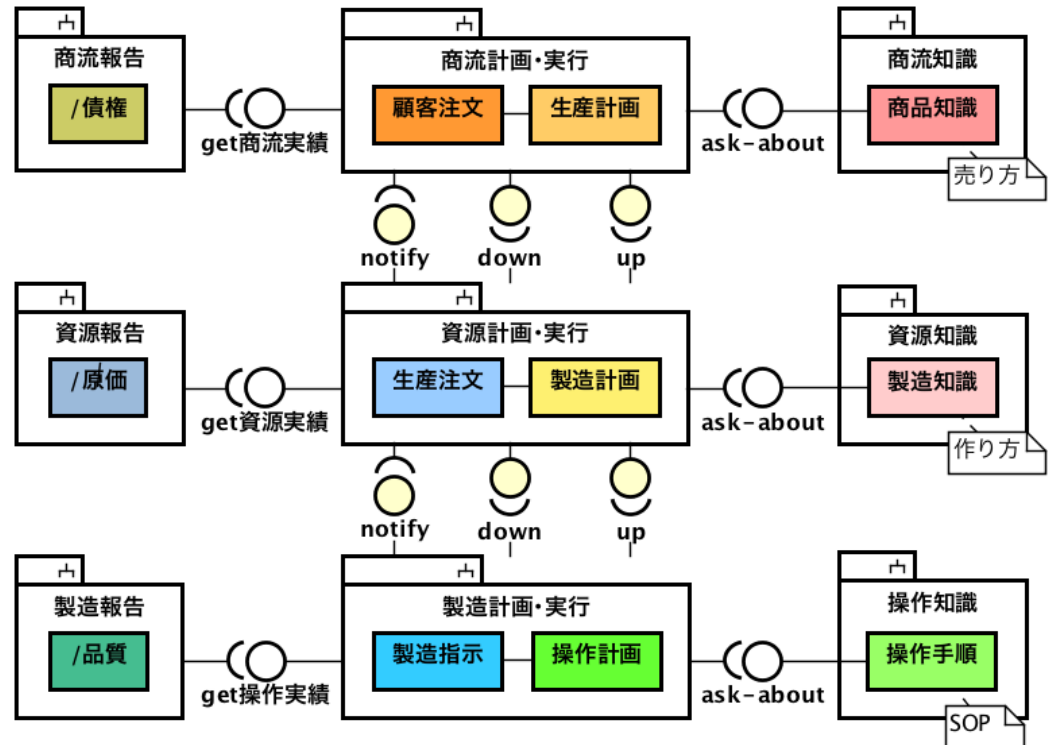
- 購買：buy

- 製造：make

- 検査：test

- 請求：claim

- ：



MESX4のシステム間インタフェース

- iHCL
 - inter-Hierarchy Collaboration Language
 - APSOM標準から国際標準へ
- メッセージとプロトコル
 - DSL (ドメイン特化言語)
 - KQML^[9]
 - Speech Act理論 (J. L. Austin, 1962)
 - プロトコル
 - 9状態モデル

<expression> ::= <演算式> | <論理式>

<演算式> ::= <object> | "(" <演算式> ")" | <演算式> <演算子> <演算式> | <function>

<object> ::= <value> | <expression>

<演算子> ::= "+" | "-" | "*" | "/"

<論理式> ::= <object> <比較記号> [<object>] | <論理式> <論理記号> [<論理式>]

<比較記号> ::= "=" | "<" | "<=" | ">" | ">=" | "!=" | "has" | "isEmpty"

<function> ::= <verb> "(" <arglist> ")"

MESX4のシステム間インタフェース

- メッセージの例
 - make-request

制御情報 (KQML)

制御アクション (performative)

request

```
:sender  商流計画・実行      //Customer
:receiver 資源計画・実行      //Performer
:time-stamp 2015-10-10,10:15:45
:message-id 37812309854398
:language iHCl
:content (
```

動詞

本文 (DSL)

```
  order(make, (
    (識別子:#00004,品目:箱,仕様:(W(100),H(20),D(30)),
      qty:10,納期(2015-10-20)),
    (識別子:#00005,品目:箱,仕様:(W(110),H(30),D(30)),
      qty:15,納期(2015-10-20))
  )
)
number-of-records(2)
)
```

MESX4のシステム間インタフェース

- メッセージの例
 - make-promise

promise

```
:sender &P
:receiver &C
:time-stamp 2015-10-10,10:20:03
:in-reply-to 37812309854398
:message-id 37812311059863
:language iHCl
:content (
  order(make, ((識別子:#00004), (識別子:#00005)),
  number-of-records(2))
)
```

MESX4のシステム間インタフェース

- メッセージの例
 - make-reportCompletion

```
reportCompletion
```

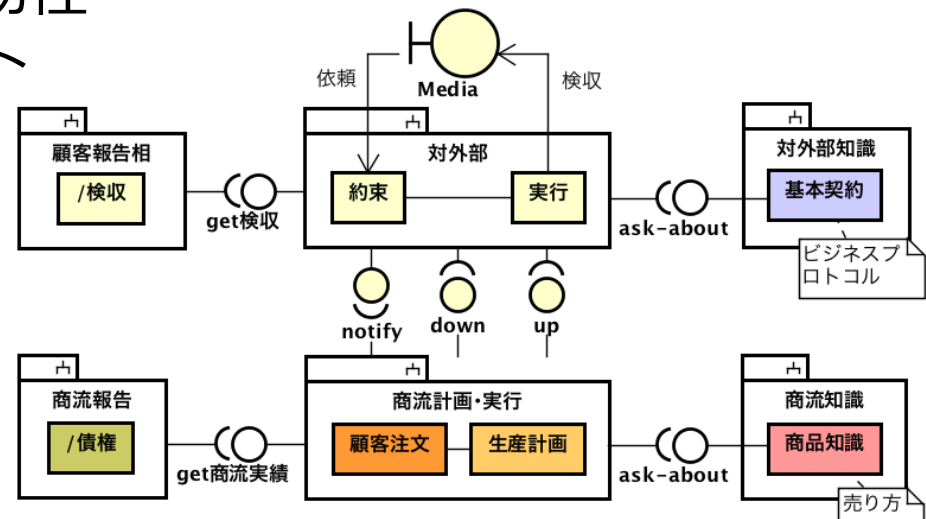
```
:sender &P  
:receiver &C  
:time-stamp 2015-10-13,14:16:54  
:message-id 37812312234458  
:language iHCl  
:content (
```

```
  order (make, (  
    (識別子:#00004,品目:箱,仕様:(W(100),H(20),D(30)),  
      qty:10,納期(2015-10-13)))
```

```
)
```

MESX3との違い

- 基幹システムでの構築実績からのフィードバック
 - 多階層システム連携
 - 商流から機器制御まで（JSONで実装）
 - システム分割基準とCustomer-Performerは強力
 - 外部複雑性の対処
 - 「基本契約」に条件を記述
 - 対外部システムで入力複雑性を除去
 - 対外部システムで出力に複雑性を付加
 - クラウドでもハイトランザクションに耐えられる
 - 並列・分散処理の有効性
 - オートスケールアウト
 - 規模によらない



MESX3との違い

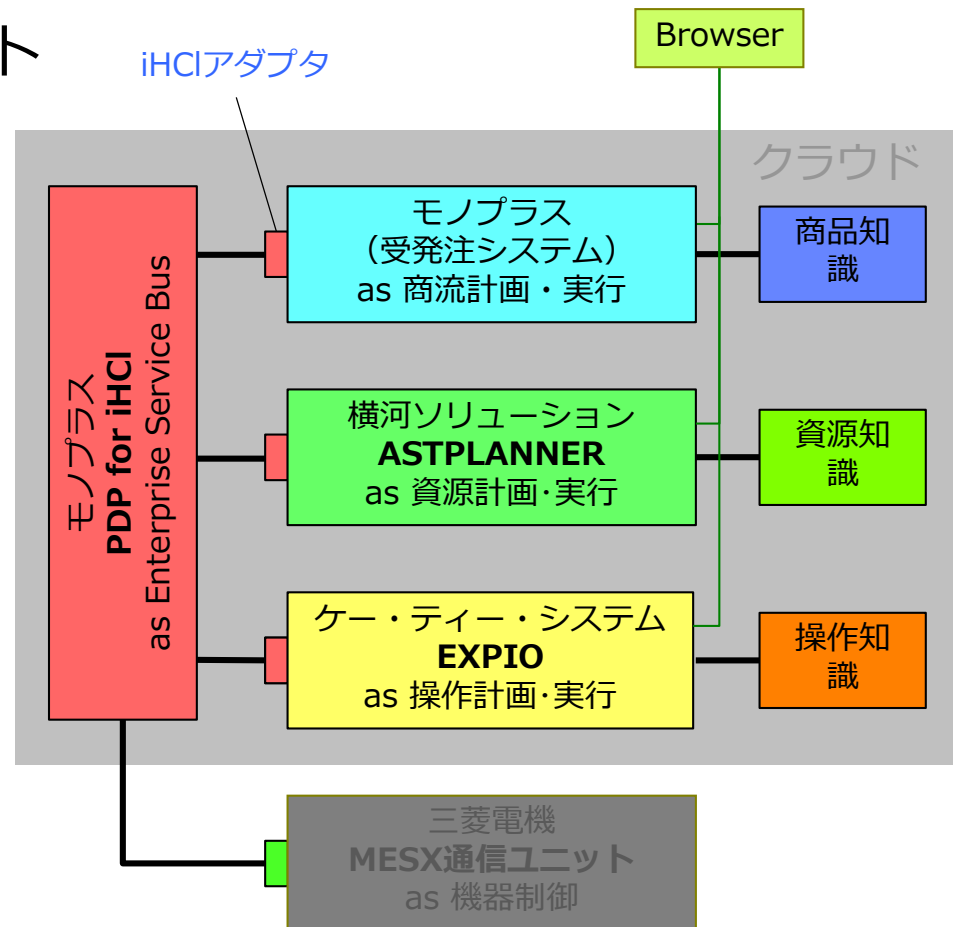
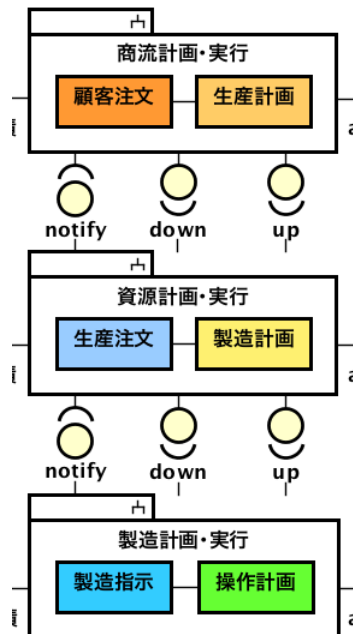
- 階層間メッセージは常に注文に関する対話
 - 注文の内容
 - いつまでに, 誰が, 誰に, 何を (品名, 仕様リスト),
 - どれだけ, いくらで
 - どうする (動詞: 売る, 買う..)
 - 応答の内容
 - 結果どうなった (注文なし実績も考慮)
 - サブシステムのドメインモデルは問わない
 - メッセージがすべて
 - 階層がいくつあっても同じ (拡張容易)
 - SCMの構成
 - サブシステムのケイパビリティプロファイルが必要
 - 呼び出し側からケイパビリティをマッチングする
 - ケイパビリティは「動詞」の組で表現できる
 - 「もの」との対話もOK (IoT)

- アーキテクチャ

- Enterprise Service Busを利用

- 待ちなし, データロストなし
- 並列・分散処理
- オートスケールアウト

- iHCIアダプタ



まとめーMESX4のメリット

- 変化に強い工場の実現を目指して
 - － MESXとは
 - システム分割およびシステム間インタフェースの基準
 - 既存のソフトウェア製品または新規作成の要素システム
 - メッセージ言語とプロトコル：iHCL
 - － MESX4の採用によるメリット
 - 利用者
 - － リアルタイム連携
 - － システムの拡張性（生産システム以外にも）
 - － 段階的構築，短期立ち上げ可能
 - － 機能拡張・縮小が容易
 - － コンポーネントのとっかえひっかえ
 - ベンダ
 - － よいコンポーネント→品質の向上
 - － 機能に集中→投資の効率
 - － 接続性の向上→ビジネスチャンス
 - － SaaSとしての新規ビジネス

参加者募集

- MESX-JPへの参加者募集

- ユーザ

- 生産の基幹システム再構築を考えている
 - グローバル展開を考えている
 - マス・カスタマイゼーションを考えている
 - プロセス型生産＋ディスクリート型生産

- ベンダ

- ユニークな機能パッケージを持っている
 - ドメインモデル主導設計をやりたい
 - 将来のビジネス展開

- 学界

- ケース研究
 - 技術標準
 - モデル化

参考文献

- [1] MESX-JWG, MESXホワイトペーパー, 2004/4/16
- [2] 高橋達也, 武藤, 児玉, 藤田, 大竹, 「標準技術の相互活用による工場内情報連携 (MESXプロトコルによる製販一体化)」, 日本機械学会論文集 (C編), Vol.76, No.772, pp.58-63, 2011
- [3] 児玉公信, 「日本の製造業のための製造知識表現とそのモデル (OMSB) について」, スケジューリング学会 創立10周年記念シンポジウム, 2008/9/20
- [4] 児玉公信ほか, 「生産管理ソフトウェア製品をクラウド上で動的に連携するためのアーキテクチャの試作」, 情報処理学会 IS研究会, 2013/9/12
- [5] 児玉公信, 「企業情報システムのための早期アーキテクティングの一方法」, 情報処理学会 IS研究会, 2012/6/4
- [6] Medina-Mora, Winograd, et al, "The Action Workflow Approach to Workflow Management Technology," Proc. of CSCW 92, pp1-10, 1992
- [7] Winograd, T.: A Language/Action Perspective on the Design of Cooperative Work, H-C-I, Vol.3, pp.3-30 (1987-1988)
- [8] 児玉公信, 「MES (製造実行システム) のモデルに関する一考察」, 情報処理学会 IS研究会, 2012/12/3
- [9] Finin, Fritzson, McKay, and McEntire: "KQML as an agent communication Language," Proceeding CIKM '94, p456-463, 1994
- [10] Yourdon and Constantine: "Structured design: fundamentals of a discipline of computer program and systems design," Prentice Hall, 1979