

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

児玉公信^{†1} 松田三知子^{†2} 野口智史^{†3} 小松原宏祐^{†4} 堀北拓也^{†5}

NPO 法人 APS 推進機構 (APSOM) の研究プロジェクト MESX ジョイントプロジェクトでの活動経過を報告する。ここでは、ISO-16100 のソフトウェアケイパビリティプロファイルを利用し、その機能とインタフェースが適合する生産管理ソフトウェア製品を探し、クラウド上で動的に連携させる方式を検討している。疎結合インタフェースおよびケイパビリティプロファイリングの妥当性について、試作によって検証しようとしている。これが検証できれば、必要な生産管理システムが適時に組み立てられ、増設や廃止が容易となる。

1. はじめに

近年、製造ビジネスにおいて製品の多様化とライフサイクルの短縮化が進んでいる。これと並行して、改善プロセスも行われるため、製造設備、生産管理システム、営業、生産技術、作業員等を含めた製造情報システムは、常に変化し続ける。このため、製造情報システムには、早期立上げ・立下げ、動的変更といった即時実行 (プラグ&プレイ) が可能となる新しい開発法が必要となっている。

この一つの方法として、「NPO 法人 ものづくり APS 推進機構 (APSOM)」では、その設立の 2003 年以来、仕入-製造-販売の水平連携の推進 (PSLX) と並行して、生産計画から製造実施までの垂直統合 (MESX) のための柔軟なシステム連携プロトコルの制定と、試作による実証を行ってきた¹⁾⁵⁾。これは、IEC 62264 (経営と工場の連携)⁶⁾⁷⁾ による機能階層に基づいて、製造情報システムを構成するソフトウェア製品を配置し、その間を疎結合するアーキテクチャとメッセージプロトコルである²⁾³⁾。しかし、プラグ&プレイを実現するには、機器の調達や設置場所の確保などがボトルネックとなる。また、生産ラインを構成する機器やソフトウェアは、ベンダ独自の設計思想に基づく製品であり、自由に組み合わせることは困難とされてきた。

本報では、クラウドコンピューティングを活用することによって、より柔軟なシステム構築を実現する方法を提案する。併せて、一定の基準に照らして機能的に同等なソフトウェア製品ならば、容易に連携できるようにする方法を提案し、その実現性について検討する。

2. ソフトウェア製品のプロファイリング

ソフトウェア製品が機能的に同等かどうかを判定するための国際標準として ISO 16100 シリーズ⁸⁾¹⁰⁾ (製造用ソフトウェア相互運用のためのケイパビリティプロファイリング) がある。これは、製造ビジネスの文脈に限定して、製造ソフトウェアユニット (以下、MSU という) の特性をプ

ロファイリングすることで、組合せ可能な MSU かどうかを識別できるようにするための仕様の国際規格である。

このプロファイルとアーキテクチャを指定することによって、MSU を自由に組合せて、要求に応じた製造情報システムを迅速かつ自在に構成することを考える。

2.1 製造ドメインモデル

ISO 16100 では、まず、MSU のベンダが想定するケイパビリティのオントロジを設定する。オントロジは、製造ドメインモデルおよび製造ドメインデータからなる。製造ドメインデータは、リソース、オペレーション、交換インフォメーション、リレーションなどを表現するプリミティブである。ただし、これらは、ISO 16100 では例示されているものではない。MSU を提供する側が自由に決められる。ただし、決められた製造ドメインモデルが、製造情報システムを構築しようとする利用者が想定するモデルと異なっていると、採用されなくなるので、製造ドメインモデルは、専門家のコミュニティなどと共同して十分検討し、共通の概念に基づいて設定される必要がある。

2.2 アクティビティ・ツリー

次に、MSU のベンダが想定する MSU のケイパビリティが使われる文脈を、アクティビティの階層 (ツリー) に対応づける。これは、上で決めた製造ドメインデータを用いてなされるケイパビリティの意味を明確化するためである。どのようなアクティビティ・ツリーを設定するかは、MSU を提供する側が自由に決めてよい。

また、MSU を利用する側も、実現したい製造情報システムにおけるケイパビリティの構造をアクティビティ・ツリー中のアクティビティと対応づけることで表現する。アクティビティ・ツリーも、共通の認識に基づいて設定される必要がある。

2.3 ケイパビリティプロファイル

2.3.1 ケイパビリティプロファイル・テンプレート

ISO 16100 では、上述のアクティビティ・ツリー上の各アクティビティに対応して MSU がもつ一般的特性と、機能、

†1 (株)情報システム総研
†2 神奈川工科大学
†3 三菱電機(株)
†4 (株)ケー・ティー・システム
†5 横河ソリューションサービス(株)

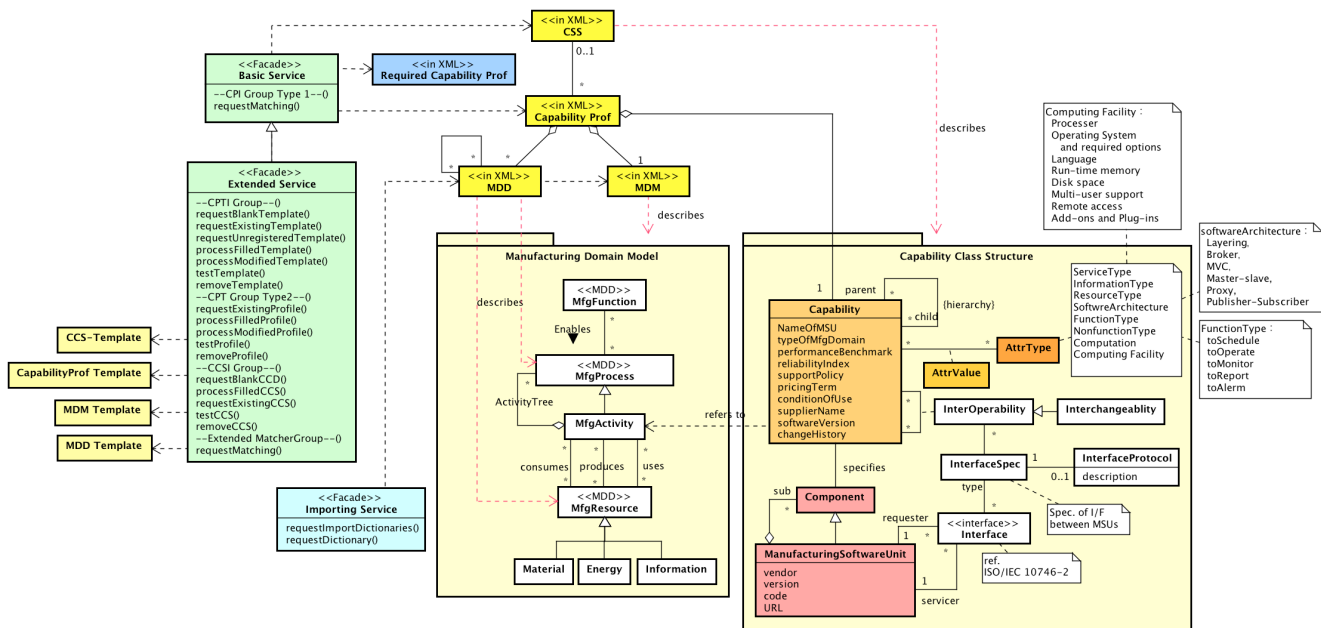


図1. ケイパビリティプロファイルに関する概念モデル

属性を定義するタグを規定したXMLのテンプレート(xsd)を提供する。このテンプレートを、ケイパビリティプロファイルテンプレートと呼び、このテンプレートを利用してMSUの特性は、製造アクティビティに対応する機能として限定的に表現される。

2.3.2 ケイパビリティプロファイル

MSUのベンダは、上述のケイパビリティプロファイルテンプレートを用いて一つひとつのMSUの特性を記述する。記述したXMLファイルをケイパビリティプロファイルと呼び、これを登録したものをケイパビリティプロファイル・ライブラリと呼ぶ。

2.3.3 要求プロファイル

次に、製造情報システムを構築しようとするクライアントは、機能要求の組を、ケイパビリティプロファイルテンプレートを用いて記述し、要求になるべく合致するMSUを、プロファイルマッチャを利用して選択する。この際、構築する製造情報システムの製造ドメインモデルと製造ドメインデータ、およびアクティビティ・ツリーが、MSUのケイパビリティプロファイルのそれらと異なっていると、ヒットするものが少ないということになる。結局、これらの基本情報は、クライアントとベンダの間で共有しなければならない。

2.3.4 プロファイルマッチャ

要求プロファイルに対して、MSUのケイパビリティプロファイル・ライブラリを検索し、一致するMSUを示す機能をプロファイルマッチャと呼ぶ。照合のレベルには、単純に文字列での一致と、オントロジを考慮して意味的な照合を行うものがある。その試実装した内容は文献11)に一部記載されている。

以上のケイパビリティプロファイルの定義と利用に関わ

る概念の関係を図1に示す。

3. 動的連携の仕組み

本報で提案する製造情報システムの基礎となる生産管理システムの構築および運用の仕組みの概略を図2に示す。

MSUのベンダはまず、①MSUのケイパビリティプロファイルを記述し、ケイパビリティプロファイル・ライブラリに登録する。生産ラインを管理したいクライアントは、②生産管理システムに要求する機能を要求プロファイルとしてまとめ、システムコンフィギュレータに送る。システムコンフィギュレータは、要求プロファイルと、複数ベンダのMSUケイパビリティプロファイルからなるライブラリ内のデータとの照合を行うプロファイルマッチャをもつ。この照合によって、クライアントが要求する機能を満たすMSU群が選択できる。④システムコンフィギュレータは、構築される生産管理システム(ターゲットシステム)と、選ばれたMSUのインタフェースとを、アプリケーション・バスを経由して接続する。このとき、MSUをターゲットシステム上に複製するのではなく、③別のクラウド空間上に

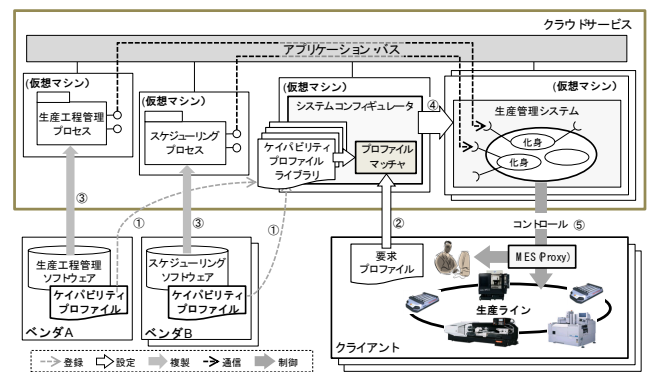


図2. MSUの連携による生産管理システムの構成

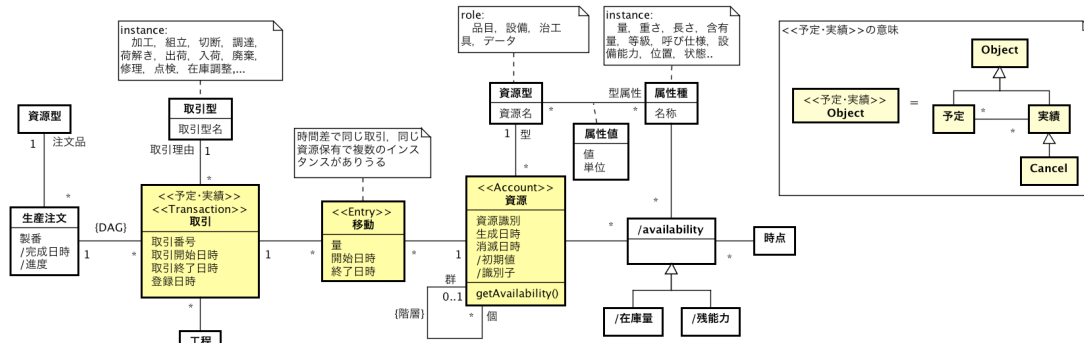


図 3. 製造ドメインモデル

起動しておく。ターゲットシステムには、MSU とのインタフェースを仲介するだけの“化身”が用意される。こうしてターゲットシステムが構成されると、システムコンフィギュレータは、⑤生産管理システムの運用を開始し、生産計画の投入、計画展開、スケジューリング、クライアントのオペレーターや生産ラインへの指示が行われていく。

4. 基本データ

本報の試設計で想定する製造ドメインモデル、アクティビティ・ツリー、ケイパビリティプロファイルのサンプルを次に示す。

4.1 製造ドメインモデル

想定する製造ドメインモデルを図 3 に示す。これは図 4 に示すアクティビティ全体に共通する概念モデルの一部である。アクティビティ・ツリーの中でも、A2.0 (生産計画および資源の引当) およびその下位のアクティビティ (A2.2, A2.3) は製造情報システムの中心的機能を担う。ここでは、生産計画を予定の製造作業単位 (取引-移動-資源) に展開して、品目、設備、時間などの資源が引き当て可能であることを確認して、引き当てる。資源が引き当てられた製造作業単位を、同じ工程で分類すれば、その工程への作業指示になる。また、指示に対して実績も同じ構造で記録する活動となる。

4.2 アクティビティ・ツリー

ISO 16100 では、製造アプリケーションは、一連の情報交換を通して製造資源の集合により実現され、制御され、自動化された製造プロセスの集合としてモデル化される。そして、製造プロセスはスケジュールされた製造アクティビティの連続としてモデル化され、それぞれが (製造システムの) 製造機能に関連づけられる⁹⁾。つまり、製造アクティビティは、製造現場の活動なのではなく、製造情報システムにおける機能利用のグループ化になっている。

ここで使用する製造アクティビティ・ツリーを図 4 に示す。これは UML クラス図を使って、各ドメインと連携インタフェースの構想を表している。

4.3 ケイパビリティプロファイル

ケイパビリティプロファイルの例を図 5 に示す。このプロファイルは、“A2-0001”なる MSU が、上記のアクティビティ・ツリー (A2.2, A2.3) に対応する機能を持つこと、その接続インタフェースについての呼称などが記述されている。

ここで参照されている製造ドメインモデルと製造ドメインデータは <Application protocol id=MESX> として別途定義され、ケイパビリティプロファイルではいちいち指定しないものとする。

4.4 要求プロファイル

要求プロファイルは、基本的に、MSU のケイパビリティプロファイルの上から 4 行目を

```
<Type id="Requirement Profile"/>
```

に変えた上で、<Activity>タグで、必要な機能と、接続インタフェースを追加する。MSU が稼働する Operating System や占有する記憶域のサイズ、性能情報なども記述できるが、今回は検証の対象外とする。

ここで指定するプロファイルは、個別の MSU のそれではなく、ターゲットシステムが必要とするケイパビリティプロファイルであることに注意する。システムコンフィギュレータを前提とするマッチャは、この要求を実現しうる MSU を単独または/および複数の組合せで候補を挙げる。

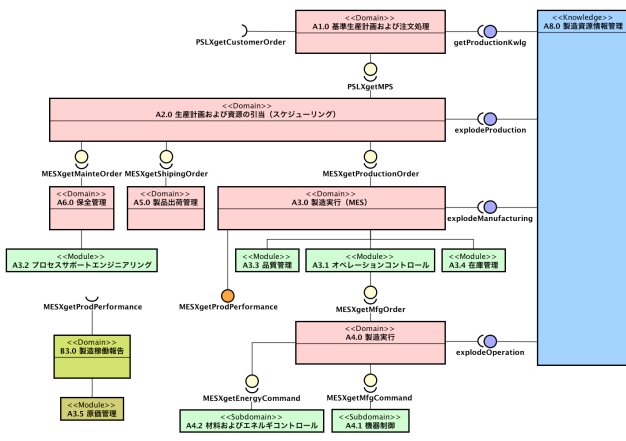


図 4. アクティビティ・ツリー

```
<?xml version="1.0" encoding="UTF-8"?>
<CapabilityProfiling xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Template id="MESX_CapabilityProfilingTemplate.xsd" name="Capability Profiling"
  <Type id="MSU_profile"/>
  <CapabilityProfile date="2013-03-26">
    <Pkgtype version="1.0.0"/>
    <Common>
      <MSU_Capability>
        <ID>A2-0001</ID>
      </MSU_Capability>
      <ReferenceCapabilityClassStructure id="CCS-MESX-A" name="経営と制御の"
      <Capability_Class_Name name="" />
      <Reference_Capability_Class_Structure_Name name="" />
      <Version minor="0" major="1" />
      <Owner>
        <Name> -- Owner's Name-- </Name>
        <Street> -- Address3 -- </Street>
        <City> -- Address2 -- </City>
        <Zip> -- zip code -- </Zip>
        <State> -- Address1 -- </State>
        <Country> JAPAN </Country>
        <Comment> -- String -- </Comment>
      </Owner>
      <ComputingFacilities>
        <Language name="Japanese" />
      </ComputingFacilities>
    </Common>
    <Specific>
      <Activity id="A2.2" name="製造スケジュールを決定する">
        <InformationExchange>
          <ListenTo id="A1.0" name="基準生産計画および注文処理" />
          <MediaProtocol id="Ethernet" />
          <BasicProtocol id="PSLX-KVS" />
          <Encoding id="PPS" version="2" />
          <ApplicationProtocol id="MESX" />
          <Comment> "" </Comment>
        </InformationExchange>
      </Activity>
      <Activity id="A2.3" name="製造実行を指示する/実績を管理する">
        <InformationExchange>
          <TalkTo id="A3.0" name="製造実行 (MES) " />
          <MediaProtocol id="Ethernet" />
          <BasicProtocol id="PSLX-KVS" />
          <Encoding id="PPS" version="2" />
          <ApplicationProtocol id="MESX" />
          <Comment> "" </Comment>
        </InformationExchange>
      </Activity>
    </Specific>
  </CapabilityProfile>
</CapabilityProfiling>
```

図 5. ケイパビリティプロファイル例

4.5 メッセージ設計とプロトコル

本試設計で採用する MSU は MESX プロトコルブック¹²⁾ で定めたサブドメインとその間のメッセージ交換の規程に準拠していることを前提とする。

これは、ケイパビリティプロファイルの記述では、

```
<ApplicationProtocol id="MESX"/>
```

で表されている。そのメッセージのコーディング形式は、OASIS の PPS¹³⁾ という規格に準拠し、通信媒体は Ethernet で、通信のフレームワークは PSLX-KVS と呼ぶクラウド上の一種の黒板アーキテクチャ¹⁴⁾ に基づくメッセージ交換機能を用いるとしている。

このインタフェースはきわめて限定的な組合せといえる。つまり、MSU が連携指定動作するためには、少なくとも、ここまでの取り決めが必要なのである。MSU ベンダが当初からこれらの連携のインタフェースを備えていると期待することはできない。ここでは、各ベンダが MESX 対応のインタフェース変換機能 (Wrapper) をそれぞれの MSU に付加することを想定している。

5. システムコンフィギュレータの試設計

次に、システムコンフィギュレータを試設計し、動的連携の仕組みの課題について検討する。

試設計においては、業務フローを想定し、そこで必要となる機能を取り出してユースケースを記述し、方式設計まで行う。

5.1 業務フロー

まず、業務フローを想定する。業務の主体 (アクタ) として、主催者側、ベンダ、クライアントを想定する。そのうち、アクタがクライアントの業務フローを図 7 に示す。

抽出された機能は、ステレオタイプ《usecase》を付したノートシンボルにユースケース名を記して示す。

5.2 ユースケース

主催者、ベンダ、クライアント別に業務フローを書き、機能を抽出し、ユースケース名を付して一覧にしたものが

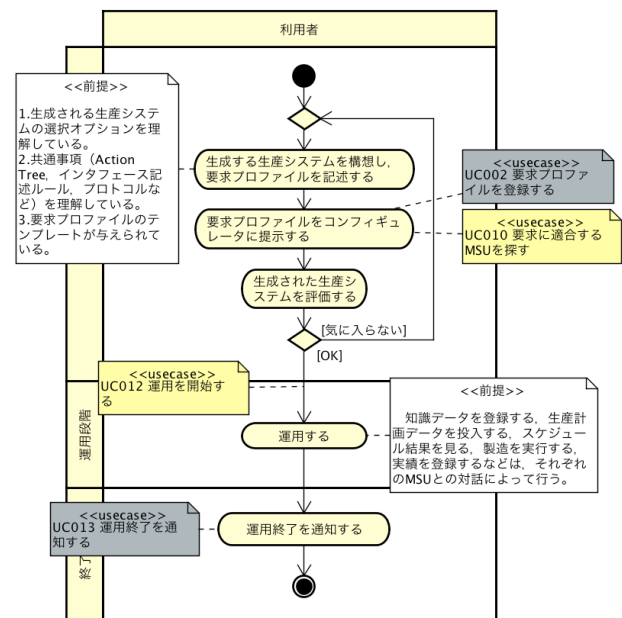


図 6. クライアントの業務フロー図

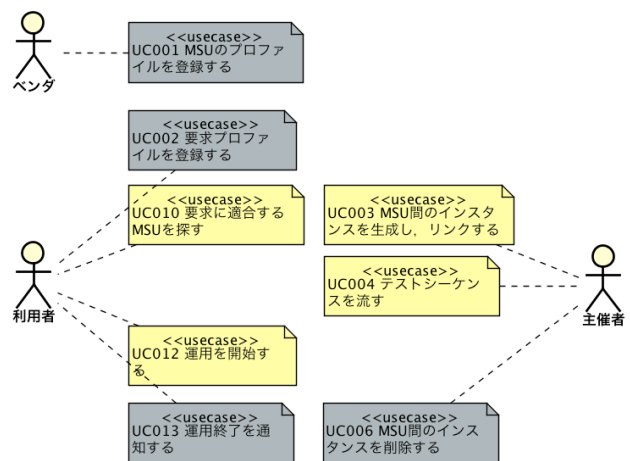


図 7. ユースケース図

表 1. ユースケース記述の例

番号	UC003
名称	MSU 間のインスタンスを生成し、リンクする
概要	利用者が指定した MSU の構成に基づいて、生産システムを生成する
アクタ	主催者
目的	利用者の要求に適合する生産システムを生成したい
事前条件	生成するシステムに関する MSU の構成が保存されている
事後条件	MSU が連携して動作可能な生産システムが生成されている
基本系列	① UC010 がこのユースケースを起動する。 ② システム (configurator) は MSU の構成を参照して、それぞれのケイパビリティプロファイルの記述を基に、MSU のインスタンスを VM 上に立ち上げ、QS とのリンクを確立する。すべての MSU についてこれを繰り返す。
代替系列	<省略>
備考	① Configurator は、MSU ごとに VM を取得し、そこにインスタンスを立ち上げ、他の MSU とのインターフェースは、QS を通して所定のメッセージ形式で行う想定。 ② MSU の構成の妥当性の保証については未検討。今回は、つながれば OK とする。
シナリオ	<省略>
更新履歴	2013.01.26 新規作成 (児玉)

図 7 のユースケース図である。これらは試実装する計画であるが、グレースアウトしたものは実装の対象外とし、登録などの機能はマニュアルで行う。

5.2.1 ユースケース記述の例

ユースケースのうち、主催者が提供するシステムコンフィギュレーションの中心機能である「UC003 MSU 間のインスタンスを生成し、リンクする」の内容を表 1 に示す。

記述内の VM はクラウド上の仮想マシン、QS はキューイングサービス機能によるアプリケーション・パスを表す。

この機能は、プロファイルマッチャによって選択された MSU (一つ以上) を、それぞれ VM 上にアップロードして起動し、キューサービスとのリンクを確立することを行う。

5.2.2 開始のトリガ

すべての MSU が起動され、相互に通信可能となった段階で「UC004 テストシーケンスを流す」を行い、あらかじめ決めておいた一連のメッセージが予定どおり流れることを確認して、稼働を開始する。稼働開始のトリガは利用者の機能である「UC012 運用を開始する」をもって行う。

5.2.3 マスタデータの設定

最初の作業は、共通データや知識データ(“マスタ”と呼ばれることがある)の登録などから始まるが、これらはそれぞれの MSU に実装されている機能を用いて行い、必要に応じて MESX プロトコルにより伝播される。

5.3 概念モデル

次に、ユースケース記述やプロトコルブックから必要な概念を抽出し、整理してクラス図を用いて概念モデルを作

成する。得られたものが図 8 である。製造情報システムが扱う概念は、すでに図 3 の製造ドメインモデルで表現されており、図 8 のモデルは、概念の整理よりも、MSU 間の連携方式に重きが置かれる。

このモデルは、現場の利用者を識別する“Client”クラス(実体はブラウザ)、Client と直接対話する“MSU 化身”クラス、クラウド上の VM に展開される“MSU_Instance”を指すクラス、および“MSU 本体”の情報を保持するクラスを識別し、それらのインターフェースを規定している。

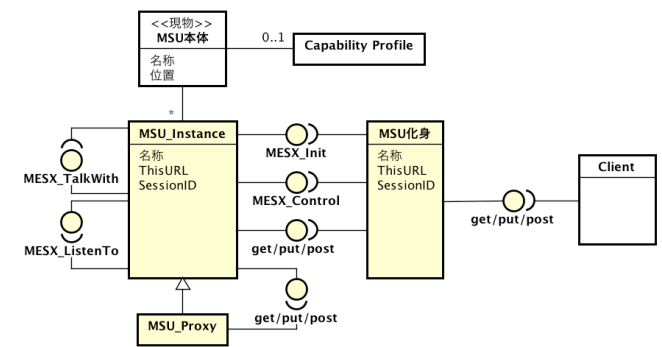


図 8. システムコンフィギュレータの概念モデル

6. 考察

試設計の過程で気づいた検討すべき課題について述べ、以降の構想および設計の見直しに反映する。

6.1 ケイパビリティプロファイリングの要件

第一の課題は、ケイパビリティプロファイルとプロファイルマッチャは、本来どのような利用を想定している、それにふさわしい効果を提供できるのかという点である。

部品表を例に取り上げる。これは図 4 のアクティビティ・ツリーでは A8.0 の活動となっている。部品表が生産管理システムの心臓部であることは明らかである。一般に、部品表は“部品”クラス (PN, Part Number) と“構成”クラス (PS, Part Structure) からなる。ある製品を作るためにどのような部品がいくつ必要となるかは、この部品表を基に所要量展開を行うことで得られる。近年、部品の一つとしてソフトウェアが加わることが多くなった。しかし、他の部品と同様に、部品はプロファイルではなく部品番号で識別されることが多い。これは部品表の作成者が、部品表を作る段階で、プロファイルに基づき、部品検索をして、部品番号にわざわざ変換していることを意味する。

部品は、頻繁に行われる設計変更や調達先の変更などに伴って、その仕様はたびたび変更され、部品と部品の接続インターフェース、性能、制約などの組合せは爆発し、すべてを確認するには膨大な工数が必要となる。マッチャはこのような状況で利用されると想定できるが、それにふさわしい属性が挙げられているのであろうか。部品にできてい

ないことがソフトウェアでできるのか^a, 実践を通したさらなる研究が求められる。

6.2 プロファイリングとコンフィギュレータ

次の課題は、クラウド上に生産管理システムを動的に構成しようという本構想とケイパビリティプロファイリング技術の組合せは適切なのかという点である。

ここには、上で述べたケイパビリティプロファイリング定義の問題が影響している。すなわち、アクティビティ・ツリー、製造ドメインモデル、製造ドメインデータによる定義の限界問題である。製造ビジネスが、見込みによる大量生産の体制においては、MRP-II (Manufacturing Resource Planning) が共通言語、共通概念(常識)であった。しかし、個別受注生産、変種変量生産、多オプション生産体制に切り替わりつつある今日において、いまのところ共通言語、共通概念は存在しないと言ってよいだろう。本試設計においては、MESX ジョイントプロジェクトが提唱するMESX アーキテクチャの機能構造を前提とし、ドメインモデルもその想定モデルを前提としたが、これはモデルとしては“常識”に比べてニッチなものである。製造のソフトウェアのマーケットに流通する通常のソフトウェア製品は、当然、MESX アーキテクチャを採用していない。

ケイパビリティプロファイリングとシステムコンフィギュレータの構想は、アクティビティ・ツリー、製造ドメインモデル、製造ドメインデータに強く依存している。この依存先がきわめて脆弱であり、本構想の成功と製造業の“常識”とは、ニワトリとタマゴの関係にある。試設計と試実装を通して、モデルの洗練、そして成功例の積み重ねが必須である。

7. おわりに

本報では、さまざまなベンダが提供するMSUの連携により、要求に合った生産管理アプリケーションを自在に構成するためのシステム開発運用環境の全体構成を示した。また、このシステム開発運用環境の実現に必須の機能であるMSUのケイパビリティプロファイルと要求プロファイルの照合を行う機能と、照合により選択されたMSUをクラウド上で連携するシステムコンフィギュレータの構想と試設計例を示した。

なお、本報の3章～5章は、独立行政法人科学技術振興機構(JST)「研究成果展開事業研究成果最適展開支援プログラム(A-STEP)【FS】ステージ 探索タイプ」の支援を受けて進められている「ソフトウェアのケイパビリティプロファイリング技術による生産管理システムのクラウド上での実現手法の検討」の研究プロジェクトに協力して得た研究成果の一部を公表するものである。

^a 著者の一人は、属性に基づく部品表システム¹⁵⁾を研究しており、企業が属性つき部品表の採用に消極的であることも感じている。

参考文献

- 1) MESX ジョイントワーキンググループ: MESX ホワイトペーパー, 製造科学技術センター (2004).
- 2) 高橋達也ほか: 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, 76(772), 3240-3245 (2010).
- 3) 岩津 賢ほか: 標準技術の相互活用による工場内情報連携(第6報)—MESX による製販一体化 2層 3層, 生産システム部門講演会講演論文集 2010, 53-54 (2010).
- 4) 児玉公信: MESX における生産管理のモデルと原価管理, 企業会計 63(6), 835-844 (2011).
- 5) 児玉公信: MES(製造実行システム)のモデルに関する一考察, 情報処理学会研究報告, IS-122(4), 1-5 (2012).
- 6) IEC 62264-1: Enterprise-control system integration — Part 1: Models and terminology (2003).
- 7) IEC 62264-3: Enterprise-control system integration — Part 3: Activity models of manufacturing operations management (2006).
- 8) ISO 16100-2: Manufacturing software capability profiling for interoperability Part 2: Profiling methodology (2003).
- 9) ISO 16100-3: Manufacturing software capability profiling for interoperability Part 3: Interface services, protocols and capability templates (2005).
- 10) ISO 16100-5: Manufacturing software capability profiling for interoperability Part 5: Methodology for profile matching using multiple capability class (2009).
- 11) 松田三知子ほか: ケイパビリティプロファイルを利用した生産管理ソフトウェアの連携, 精密工学会秋季大会講演集, M68 (2013).
- 12) MESX ジョイントプロジェクト: MESX プロトコルブック 第8版(未公開), APSOM (2011).
- 13) OASIS: PPS (Production Planning and Scheduling) Part 2: Transaction Messages, Version 1.0, Public Review Draft 01, <http://www.oasis-open.org/committees/pps/>
- 14) Buschmann, F., et al: Pattern-Oriented Software Architecture - A System of Patterns, John Wiley (1996). 金澤ほか訳: ソフトウェアアーキテクチャ, トッパン (1999).
- 15) 児玉公信, 水野忠則: 部品表の統合に関する一考察, 情報処理学会全国大会論文集, Vol. 70, 4.557-558 (2008).