

リアルタイム・サプライチェーンを実現する
iHCl 言語に関する白書

～IoT で世界をつなぐ共通言語について～

特定非営利活動法人 ものづくり APS 推進機構
(APSOM: Advanced Planning and Scheduling Organization for Manufacturing)



目次

1. まえがき	3
2. iHCl の概要	4
3. iHCl の説明	5
3.1 シンプルな対話型メッセージ言語.....	5
3.2 リアルタイム性の実現.....	6
3.3 AI との相性について	7
3.4 保全について	7
4. iHCl 言語の導入手順.....	8
4.1 iHCl アダプタを組み込む方法	8
4.2 既存の業務アプリケーションを iHCl で対話できるように変更する方法	9
5. 実施例.....	11
5.1 社内生産システム	11
5.2 企業間連携システム	12
6. 用語の説明.....	13
7. 参照文献	14

1. まえがき

本白書は、特定非営利活動法人 ものづくり APS 推進機構（略称：APSOM） MESX ジョイントプロジェクトにより策定されました。

世界中の企業や工場がつながる「**Connected Industries**」時代になり、様々な経済社会要素が目的に合わせて柔軟に連携ネットワークを構成し「繋がる」ことによって、新たな付加価値の創出や社会課題の解決を行うことが要求されています。生産分野においても、日本として、このような分散連携システムの構成について迅速な対応が迫られています。分散連携システムに対応するために解決すべき課題としては、以下のようなものが挙げられます。

- a) 商品の受発注等におけるやり取りにおいて依頼内容の変更があった場合、現状では生産現場での“擦り合わせ”や“作り込み”など非自動化手法により柔軟な対応をしている。しかし、「**Connected Industries**」時代になり分散連携システムとして生産システム全体が自動化された場合、日本の強みであるこのような柔軟な対応を維持する手段がない。すなわち、柔軟な対応を取り込んだ分散連携システムにおけるコミュニケーション手段がない。
- b) 商品の受発注、生産、配送、保守などの依頼は、系列会社間や決められた取引先で固定されているが、分散連携システムとして世界がつながったときに、不特定多数の相手と 24 時間対応の商取引をするための共通のメッセージやプロトコルがない。
- c) 日本の国内企業が分散連携システムの構成に迅速に対応でき、国際競争力を維持・強化するためのインフラがない。さらに言えば、このインフラ構築の基盤となる共通のメッセージやプロトコルがない。
- d) 柔軟な分散連携システムの構成に対応するインフラは、クラウドコンピューティング、エッジコンピューティングなどを活用して構築されるが、中小企業など日本全体の企業がこれを利用するためには、共通のメッセージやプロトコルの規定が必要である。
- e) 柔軟な分散連携システムを構築するにあたって、そこで使用されるメッセージやプロトコルが共通化されていなければ、システムの開発・運用・保守に大きな労力（人のコスト）が必要となる。
- f) 「**Connected Industries**」時代へ円滑に移行していくためには、日本の商習慣（例えば、内示、曖昧な受発注、擦り合わせ前提の注文、後払い、掛け売り）を取り込める共通のメッセージやプロトコルが必要である。

本白書で述べる iHCl を活用することにより、以下が期待できます。

- a) 依頼内容の変更を吸収し、日本の強みである生産現場での“擦り合わせ”や“作り込み”などの柔軟な対応を維持できる手段としてのメッセージとプロトコルを利用できる。
- b) 商品の受発注、生産、配送、保守の高機能化、高性能化、自動化が実現され、24 時間リアルタイムで実行でき、柔軟な分散連携システムの構成に対応できる。
- c) 日本企業の「**Connected Industries**」化を促進し、より一層の高付加価値化・差別化ができるようになるため、国際競争力の強化に繋がる。
- d) クラウドコンピューティング、エッジコンピューティングなどのインフラの利活用が可能となり、中小企業などでも世界中から来る仕事の受託や依頼ができるようになる。
- e) 分散連携システムの開発・運用・保守が容易になり、システム採用・立上げの円滑化が可能となる。
- f) 日本の商習慣を取り込んだ分散連携システムの構築が可能となり、日本が一体となった国際対応が可能となる。

2. iHCl の概要

iHCl は、インターネットを介してソフトウェアどうしが対話するための共通言語です。対話は、Customer（発注者）に相当するソフトウェアが、Performer（受注者）に相当するソフトウェアに対して、何かの実行を依頼するやりとりです。依頼の内容は「販売せよ」でも「製造せよ」でもかまいません。

既存のソフトウェア^{※1}を対話できるように設定します。そのケイパビリティプロファイルを記述し、iHCl の文法に従うメッセージを組立/分解するようにします。対話できるようになったソフトウェアを製造用ソフトウェアユニット（MSU: Manufacturing Software Unit）と呼びます。

iHCl の文法に従うメッセージの組立/分解は、APSOM から提供される iHCl 用アダプタモジュールを使うことで容易に実現できます（4 章参照）。

この対話の連鎖によって、世界的なサプライチェーンも実現できます。MSU が対応できれば、注文の内示やオーダ変更、欠品の発生などの異常事態に関する対話も可能です。また、リアルタイムで自動的かつ柔軟な対話が実現できます。受発注システムも、生産管理システムも、部分的かつ段階的に構成できます。

従来、受発注システムには下記の課題がありました。

- 1) 非リアルタイム処理で、深夜などの時間指定によるバッチ処理が必要
- 2) 特定ベンダが提供する受発注の一括システムは、信頼性は高いが高価
- 3) 受発注システムを安く構成するために、安価な業務アプリケーションをさまざまなベンダから入手すると、アプリケーション間のデータ連携でカスタマイズ(時間&費用)が必要

これに対して iHCl は、以下のメリットを提供します。

- a) リアルタイムな受発注システムを構築可能
- b) さまざまなベンダが提供する安価な業務アプリケーションに簡単に組み込み可能
- c) クラウドからエッジコンピューティングや制御機器（例：PLC やロボットなど）まで、受発注データ、製造指示データ、コントロールデータとその結果データを交換可能
- d) クラウド間や、スケジューラ間、MES 間などの連携も可能であり、Customer からの注文に、ある Performer が受注できないときに他の Performer が注文を受け付け可能
- e) 発注取り消しなどにも対応可能
- f) その他、柔軟な受発注システムが構築可能

※1 クラウドサービス、MES、スケジューラ、SCADA、ERP、BOM、エッジコンピューティング、制御機器など。

3. iHCI の説明

3.1 シンプルな対話型メッセージ言語

iHCI は、さまざまな製造機能を、Customer の MSU と Performer の MSU という 2 者間のシンプルな対話に還元し、その組合せによって多くのシステム要素からなる統合的な製造システムを実現します。

図 1 は、Customer と Performer 間で交わされる対話の構造を状態遷移図（状態機械図）で表現したものです。2 者間の対話はこの状態遷移図に従って次のように進みます。

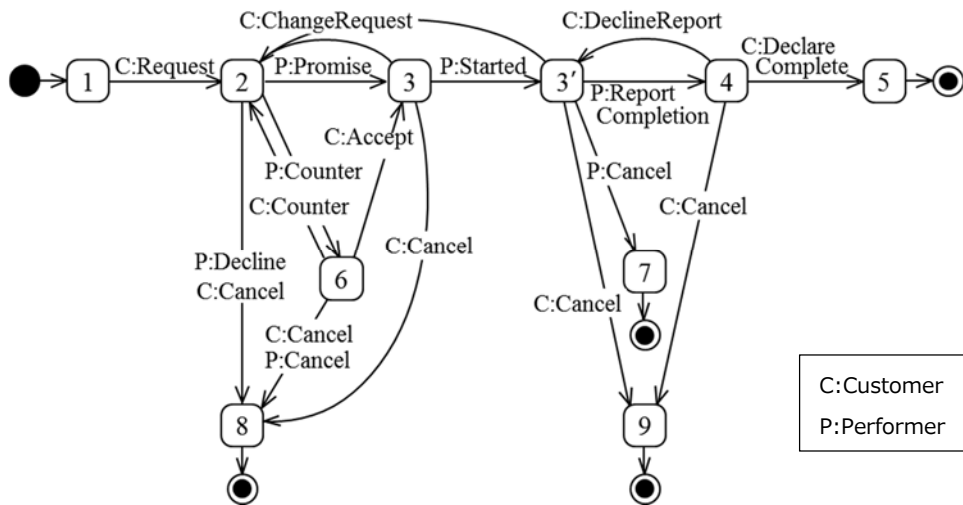


図 1 Customer と Performer 間の対話（状態機械図）

1. Customer は Performer に「注文提示 (Request)」メッセージ [1→2] を送信
2. Performer はその注文内容が実行可能かどうかも含めて妥当であれば、「約束 (Promise)」メッセージ [2→3] を返信
3. Performer は作業に着手したことを「着手した (Started)」メッセージ [3→3'] で通知
- 3'. 作業が終了すると Performer は Customer に「終了報告 (ReportCompletion)」メッセージ [3'→4] を送信
4. Customer はその終了報告を評価して、満足すれば「注文完了宣言 (DeclareComplete)」メッセージ [4→5] を送信
5. 対話を終了

上記の対話はすべてが順調に進んでいる場合です。たとえば、注文品が製造できない、あるいは注文の納期に間に合わないなどの理由により、Performer が「対案提示 (Counter)」メッセージ [2→6] または「辞退 (Decline)」メッセージ [2→8] を送る場合があります。

Performer がすでに約束メッセージを送ったにもかかわらず、Customer が一方的に「注文変更 (ChangeRequest)」メッセージ [3→2] を送ってくる場合があります。また、Performer が作業を実行中であるにもかかわらず一方的に「中止 (Cancel)」メッセージ [3→8] を送ってくることもあります。

Customer は、終了報告の内容に不満があれば、終了報告の「拒否 (DeclineReport)」メッセージ [4→3'] を送って再作業を要求したり、注文を「取消 (Cancel)」メッセージ [4→9] を送ることがあります。

実際の対話は、業務の特性に合わせて、Customer と Performer の両者の合意により、図 1 に示した対話構造の一部を使うことも可能です (図 2)。

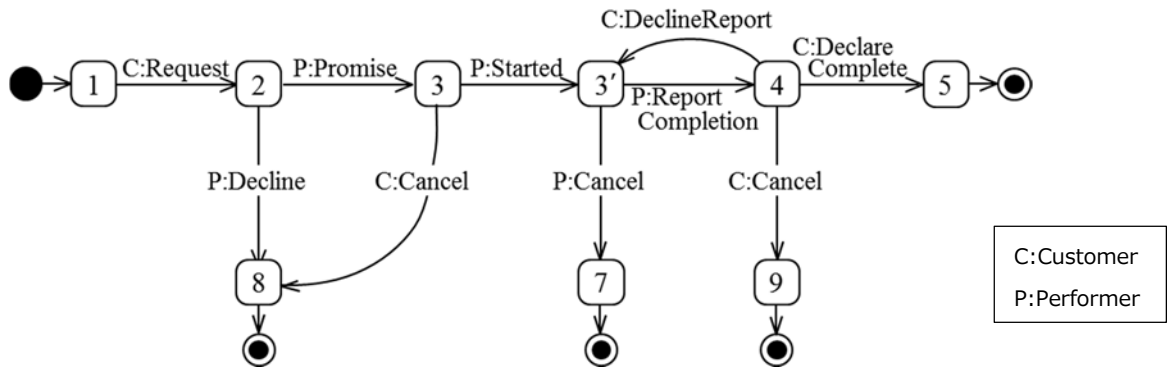


図 2 Customer と Performer 間の対話のサブセット (状態機械図)

3.2 リアルタイム性の実現

リアルタイム応答性が高い受発注システムを構築できます。MES と制御機器間での対話を想定した実証実験において、MES 側の iHCl アダプタ (Customer) がメッセージを受けてから制御機器側の iHCl アダプタ (Performer) に渡すまでに要した時間、およびその逆の処理で要した時間を計測しました。その計測結果の分布を示すのが図 3 です。この図から判るとおり、Customer と Performer 間でやりとりするメッセージの処理に要する時間は概ね 100ms 以下^{※2} であり、遅延の少ない受発注システムを実現可能であると考えます。

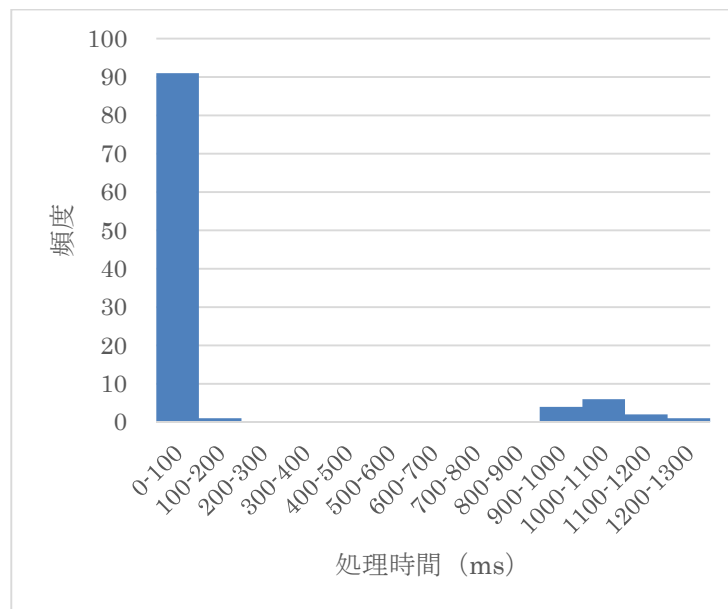


図 3 Customer と Performer 間でやりとりするメッセージの処理時間

※2 この処理時間に通信時間は含んでおりません。また、処理時間は環境にも依存しますので、100ms 以下の処理を確約するものではありません。

3.3 AI との相性について

AI で学習しやすく、人間の判断を仰がず、自動的にかつ自律分散的に受発注可否判断や、変更応答性が良い受発注システムが構築できます。

生産用途では、在庫品の自動発注、設備消耗品の自動発注、設備故障時の交換部品の自動発注など、24 時間 365 日稼働し続ける自動化工場では、人の判断を待たずに工場外への指示や発注が必要になってきます。

また、発注先も納期対応が良い、価格が安い、品質に対するクレームが少ないなどの情報を AI 分析して、最適な発注先を自動的に選択できるようになります。

なお、注文を受けた場合には工場内の空き具合を AI で自動判断し、工場が混んでいれば注文を受けないなど、受注可否判断を自動的に行います。

但し、高額な受発注など経営に影響を与えるものまで AI で自動化しないよう、システム構築時にどこまで自動化するか、また、誤受発注時に取消しをどうするかなどのリスク管理も必要です。

3.4 保全について

制御機器やエッジコンピューティングで分析した保全情報を基に、交換部品や消耗品を機器側から自律的にリクエストして自動発注するシステムが構築できます。このため、上位のシステムで保全品手配のための改造が不要になります。

4. iHCl 言語の導入手順

iHCl による MSU 連携を実現するには、APSOM が提供する iHCl 用アダプタモジュール（以下「iHCl アダプタ」という）を組み入れるか、既存の業務アプリケーションを iHCl で対話できるように変更するか、2つの方法があります。

4.1 iHCl アダプタを組み込む方法

図4は、2つのMSUがiHClアダプタを介して連携するためのメッセージの流れを図式的に表したものです。

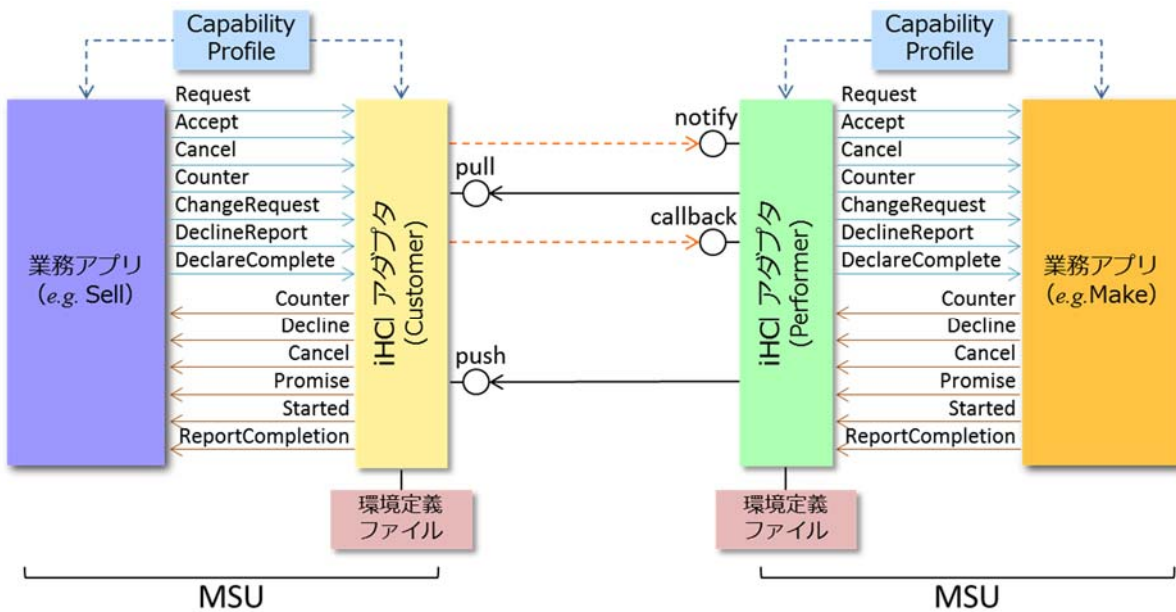


図4 iHCl アダプタを介したメッセージ連携

4.1.1 iHCl アダプタの役割

iHCl アダプタは、図1の状態遷移図に従ってMSU間のメッセージを仲介します。

[下り (C→P) メッセージの受信]

1. Performer の iHCl アダプタは、Customer の iHCl アダプタの pull チャンネルに対して、自分へのメッセージを要求します。
2. Customer は、送信したいデータがあれば、それを引数に入れて iHCl の Customer 用メソッドを呼びます。
3. メソッドはそれを iHCl 形式のメッセージに変換し、通信を介して Performer の iHCl アダプタに渡します。
4. Performer の iHCl アダプタはメッセージを内部形式に変換して、それを Performer MSU の公開メソッドに渡します。

逆に、Performer MSU から Customer MSU へのメッセージは、

[上り (P→C) メッセージの送信]

1. Performer の iHCl アダプタのメソッドに送信したいデータを渡します。
2. メソッドはそれを iHCl 形式のメッセージに変換し、通信を介して Customer の iHCl アダプタ

タの push チャンネルに渡します。

3. Customer の iHCL アダプタは受け取ったメッセージを内部形式に変換して、それを Customer MSU の公開メソッドに渡します。

ともに Performer からの起動（一方向参照）であり、疎結合を実現していることに注意してください。

2 者間の通信先、注文提示の内容、バリデーションのルールなどを、MSU に対応するケイパビリティプロファイルに記述しておきます。iHCL アダプタは対話の開始に先立って、MSU との対話内容、メッセージの通信先などをケイパビリティプロファイルから取得して、自分の振る舞いを決定します。

4.1.2 iHCL アダプタの入手方法

iHCL アダプタは、APSOM の Web サイトから簡単に入手できます。

4.2 既存の業務アプリケーションを iHCL で対話できるように変更する方法

既存の業務アプリケーションを、直接 iHCL アダプタの機能呼び出して対話するように変更できれば、生産システム全体の分散連携システム化はすぐに実現できます。しかし多くの場合、既存の業務アプリケーションを大きく変更することは困難です。そこで、中間的に対話を媒介する Proxy モジュールを置く方法をご紹介します。

4.2.1 業務アプリケーションに Proxy モジュールを用意する

所有する業務アプリケーションが、たとえばファイルインタフェースのみを備えた非常駐型のアプリケーションの場合の対応方法です。これは、業務アプリケーションはできるだけ変更せずに、別に Proxy モジュールを作成するものです（図 5）。

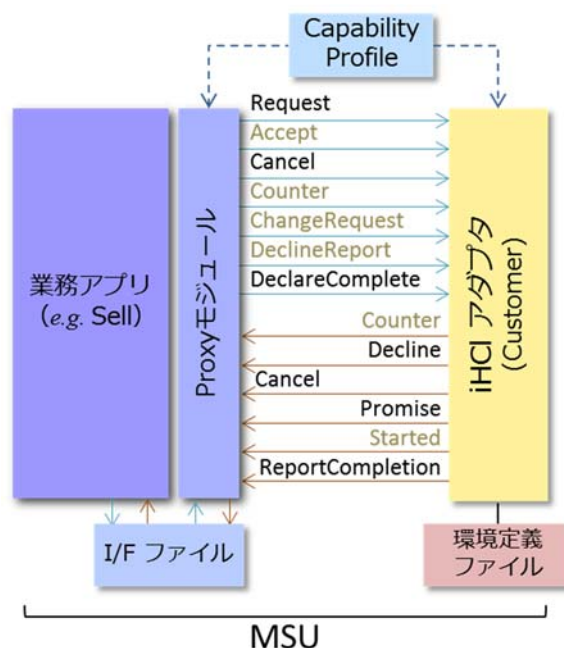


図 5 Proxy モジュールから iHCL アダプタを使用する

たとえば、業務アプリケーションが CSV ファイルを使用したファイル授受のインタフェースしか備えていない場合は、以下のように Proxy モジュールを作成します。使用する Performative も限定します。

[送信処理]

1. 業務アプリケーションは所定のフォルダに送信用 CSV ファイルを書く
2. Proxy モジュールは所定のフォルダを監視し、ファイルが書かれたことを察知して CSV ファイルから、呼び出す iHCl アダプタのメソッドに合わせて引数に整形してメソッドを呼ぶ
3. iHCl アダプタのメソッドは受け取った引数から iHCl 構文に従ってメッセージに変換して蓄積し、Performer に Notify 制御メッセージを送る

ここから先は iHCl アダプタの仕事になりますが、RSVP メッセージを待つて未送信だったメッセージを Performer の iHCl アダプタの callback チャンネルに向けて一斉に送信します。

[受信処理]

1. iHCl アダプタは Performer からの着信メッセージを受け取って引数に変換して、Proxy モジュールの該当するメソッドを呼び出す
2. Proxy モジュールは、引数から CSV ファイルを成形して、所定のフォルダに書き出す
3. Proxy モジュールは、運用ルールに従って業務アプリケーションを起動し、受信処理を開始する

こうして、CSV ファイルを用いて適切な相手先と iHCl メッセージが送受信されるようになります。ただし、通信のリアルタイム性は損なわれますし、送受信ファイルが置かれる所定のフォルダに対するポーリング処理の分だけオーバーヘッドが発生します。これに対しては、運用上および実装上の工夫が必要となります。

4.2.2 iHCl 対応の業務アプリケーションの入手方法

購入した業務アプリケーションが、iHCl に対応しているか否かは、業務アプリケーションを提供しているベンダへ直接対応可否をお問い合わせください。

また、自社の業務アプリケーションを iHCl に対応させる場合^{※3}は、APSOM の Web サイトから直接お問い合わせ下さい。

< <https://apsom.org/contact/index.html> >

窓口：一般財団法人 製造科学技術センター内 APSOM MESX-JP 担当者宛

※3 現時点（2019/1）では、ライセンスの使用権は APSOM 会員に入会することが条件となります。

5. 実施例

5.1 社内生産システム

実施例として、A 社内の業務アプリケーション(注文管理アプリケーション、生産管理アプリケーション、MES、生産ライン)それぞれが iHCl により対話するケースを示します。

本実施例における業務アプリケーション間での対話の関係の一例を図 6 に示します。

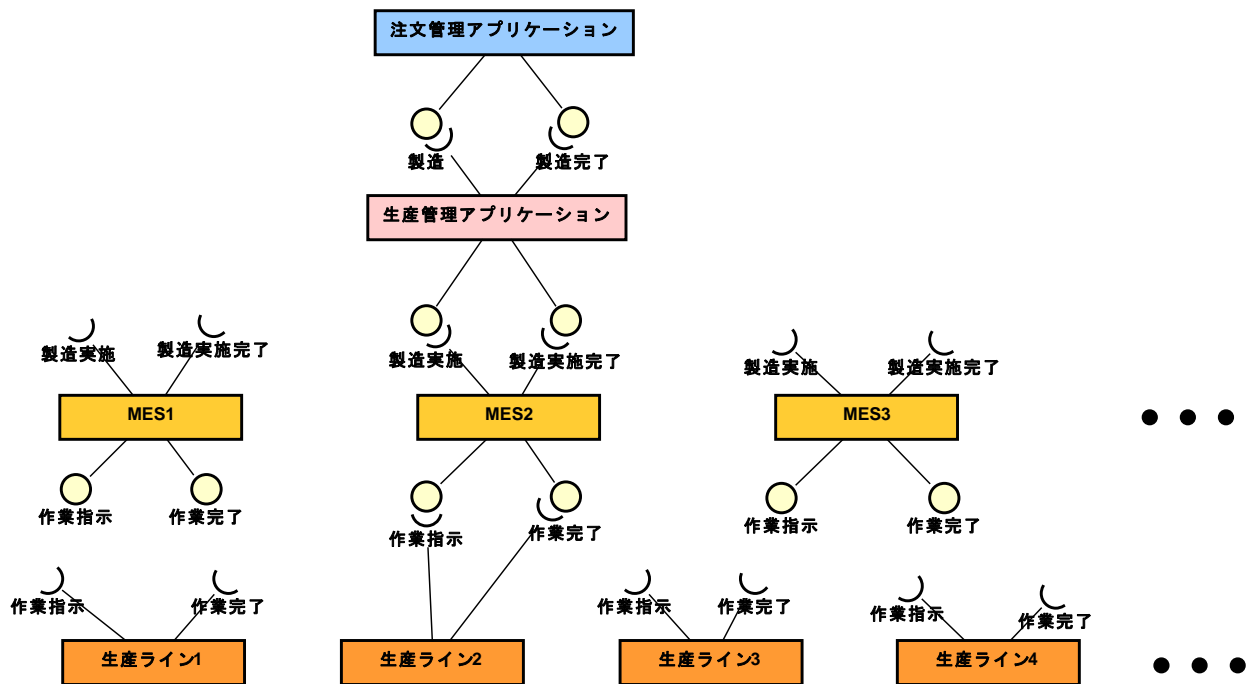


図 6 社内生産システム例

注文管理アプリケーションでは、顧客からの注文を受けて、生産管理アプリケーションに製造注文を送って製品の製造を依頼し、製造完了報告を受け取って、注文の在庫を引き当てます。

生産管理アプリケーションでは、注文管理アプリケーションからの製造注文を受けて、部品展開・作業展開・資源展開を行って、必要な資源(品目と設備、治具、作業者)を引き当ててスケジュールし、製造実施可能な MES を探してその MES に製造実施注文を送ります。その後製造実施完了報告を受け取って、製造完了報告を注文管理システムに返します。

MES は、工程グループごとに設営されます。生産管理アプリケーションからの製造実施注文を受けて、1 ステップずつ作業展開を行って、作業実施可能な生産ラインを探してその生産ラインに作業パラメータと共に作業指示注文を送ります。その後、作業実績と共に作業結果を受け取って、その結果に基づいて次のステップで行う作業を決定します。すべての作業が完了したところで、生産管理アプリケーションに製造実施完了報告を送ります。

生産ラインは作業内容に応じて設営され、生産ライン内のエッジコンピュータが MES から受け取る iHCl メッセージを解釈し、各装置・コントローラへの制御指示を行います。MES からの 1 ステップごとの作業指示注文を受けて作業を実施し、作業完了後 MES に対して作業実績と共に作業完了報告を送ります。

5.2 企業間連携システム

実施例として、A 社内の受発注システムをクラウドから各種業務アプリケーションやエッジコンピュータを接続。また、A 社と B 社間をクラウド接続し、共通言語である「iHCL」を活用することで、専用のインタフェース開発が不要となります。また、A 社と B 社間だけでなく、A 社と C 社、A 社と D 社といった柔軟な受発注システムを構築できます。

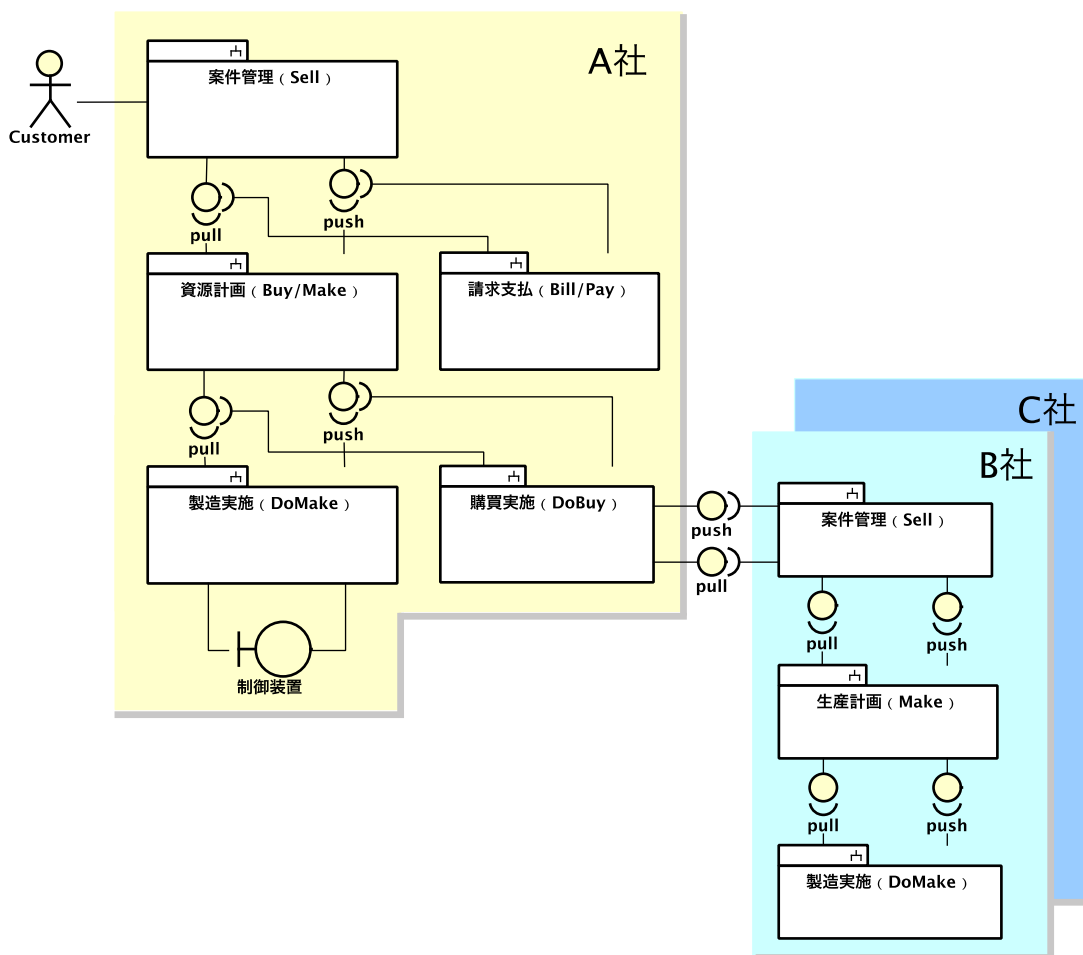


図7 クラウド間連携システム例

6. 用語の説明

Connected Industries

2017年3月に開催されたドイツ情報通信見本市にて、日本の経済産業省が「目指すべき産業の在り方」として提唱した概念

iHCl (inter-Hierarchy Collaboration Language)

ソフトウェアエージェントが互いに協調してシステム全体の機能を創発するためのメッセージ言語とプロトコルを規定している

ケイパビリティプロファイル (Capability Profile)

サブシステムが実行できる機能のプロファイルデータ

業務アプリケーション (Management applications)

生産管理、受発注、MES、スケジューラ、SCADA、MRP、BOMなどのアプリケーションを示す

Customer

顧客であり、発注者側を示す

Performer

受託者であり、顧客に対して付加価値を提供する受注者側を示す。

たとえば、案件管理者 (Deriver)、資材計画者 (Maker)、製造実行者 (Executer)、請求者 (Claimant)、販売者 (Buyer) など。

ただし、Performerである受託者が次の工程を発注する場合は、発注者側 (Customer) となる

「注文提示 (Request)」メッセージ (Request message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「約束 (Promise)」メッセージ (Promise message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「着手した (Started)」メッセージ (Started message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「終了報告 (ReportCompletion)」メッセージ (ReportCompletion message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「注文完了宣言 (DeclareComplete)」メッセージ (DeclareComplete message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「対案提示 (Counter)」メッセージ (Counter message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「辞退 (Decline)」メッセージ (Decline message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「注文変更 (ChangeRequest)」メッセージ (ChangeRequest message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「中止 (Cancel)」メッセージ (Cancel message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「終了報告の拒否 (DeclineReport)」メッセージ (DeclineReport message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

「注文を取消 (Cancel)」メッセージ (Cancel message)

「業務上の意思 (例：注文、処理結果) を伝えるメッセージ」のひとつ

iHCl 用アダプタモジュール (iHCl adapter module)

『iHCl の解釈と符号化・復号化を担うモジュール』と『実際に送受信を行うモジュール』を合わせた概念

PROXY モジュール (PROXY module)

業務アプリケーションと iHCl 用アダプタモジュールの間で、中間的に対話を媒介するためのモジュール

7. 参照文献

- ・ IEC 62264:2013 Enterprise-control system integration - Part 1-Part 2
- ・ ISO TR 16161:2019 Automation systems and integration – Use case of capability profiles for cooperation between manufacturing software units
- ・ ISO 16100 Industrial automation systems and integration -- Manufacturing software capability profiling for interoperability
- ・ JIS B 3900 産業オートメーションシステム及びその統合—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

禁無断転載

**ものづくり APS 推進機構 (APSOM : アプソム)
(MESX ジョイントプロジェクト)**

リアルタイム・サプライチェーンを実現する

iHCl に関する白書

発行日 2019年 1月 22日

発行者 APSOM

東京都港区新橋3-4-10

一般財団法人製造科学技術センター内

APSOM 事務局：山本 明人

編集者

児玉 公信 (情報システム総研)
松田 三知子 (神奈川工科大学 名誉教授)
加藤 潤三 (技術データ管理支援協会)
浅野 寿勝 (横河ソリューションサービス)
堀北 拓也 (横河ソリューションサービス)
山田 浩史 (横河ソリューションサービス)
吉田 和弘 (モノプラス)
小林 毅 (三菱電機)
浅野 義智 (三菱電機)
藤島 光城 (三菱電機)
小松 昭英 (APSOM 主任研究員)
山本 明人 (APSOM 事務局)