

インダストリー 4.0 へのサイバーフィジカル システムのアーキテクチャの枠組み

周 碩彦, Anindhita Dewabharata

インダストリー 4.0 の動向の背後にある社会経済的な原動力には、これまで以上にシステム要素同士が統合されてはいるものの、柔軟性に富みかつ迅速な生産システムが必要である。オペレーション技術と情報技術の間のシームレスな統合を実現するためには、異なるビジネスの階層間でデータを自由にやり取りすることのできる環境を確立する必要がある。本稿では、実践的な開発のためのシステム統合と情報の参照を促進するための、リファレンスアーキテクチャモデル・インダストリー 4.0 (RAMI4.0)、セマンティック技術、そしてサービス指向アーキテクチャとマイクロサービスアーキテクチャ、といったアーキテクチャの全体像に関するサイバーフィジカルシステム (cyber-physical system; CPS) の枠組みについて論じる。

キーワード：インダストリー 4.0, サイバーフィジカルシステム, RAMI4.0, セマンティックモデル, サービス指向アーキテクチャ

1. 緒言

従来、工場における生産システムは、あらかじめ設定された少数の条件が引き金となる離散的な時点に限定して IT システムとの情報交換が行われていた。高レベルに自動化された生産システムであっても、従来どおり受動的でありかつインテリジェントとはいえなかった。

市場の関心が機能よりも製品の利用を通じて得られる体験に移ってきており、生産戦略は大量生産からマスカスタマイゼーション、さらには個別対応へとシフトすることになる。しかし、多様な製品やスピードが要求される生産システムは複雑になりすぎるため、人間が効率的に管理するには困難である。

国際計測制御学会 (The International Society of Automation) は、あらゆる産業のあらゆるプロセスにおいて、グローバルな製造企業が適用するための国際標準である ANSI/ISA-95 を制定した [1, 2]。しかし、実際の (物理的な) 生産システムから IT システムを切り離したために、製造中に生じる過渡的もしくは偶発的な問題の多くを検出することができなかった。つまり、予定されていたスケジュールから大きくずれたり、製造が中断したときに容易には回復もしくは復帰することができなかった。これに対して、IoT (Internet of Things) はこのようなギャップを埋めるためのインフ

ラストラクチャを提供できる。

現実世界のマネジメントにおいて最も影響力のある IoT の機能は、現実世界とデジタル世界の結合を通して、現実世界の状態を感知し取得できることであり、また取得した情報全体をデジタル世界で観察できることである。CPS は、IoT と同様の基本アーキテクチャをもつが、現実世界とデジタル世界を結び付け調整することに重点を置いている [3, 4]。CPS により、IT とオペレーション技術 (OT) がうまく統合するように、現実世界に人体における神経のような役割を導入することができる。近年では、より進化し機械学習などの高度な処理が可能なセンサーと組み合わせることにより、CPS に生産システムに求められる柔軟性と迅速性について、より厳しい要求に対処できるような分散処理能力および自律能力を備えることができるようになった。

本稿では、このような状況を念頭に、CPS の枠組みを紹介する。この枠組みは、リファレンスアーキテクチャモデル・インダストリー 4.0 (Reference Architectural Model Industrie 4.0; RAMI4.0)、セマンティック技術、サービス指向アーキテクチャ (Service-Oriented Architecture; SOA) / マイクロサービスを踏襲するものである。RAMI4.0 は既存の標準を再構成したものであり、標準間で欠けているリンクを見つけ出し、さらに標準化が必要な分野について言及する [5-7]。RAMI4.0 はドイツ電気・電子工業連盟 (ZVEI) が提唱したモデルであり、標準化された方法により、新規のインダストリー 4.0 の応用に対する枠組みを提供するものである。RAMI4.0 によって、CPS の開発はより詳細にかつシンプルなクラスターへと分割され、相互

しゅう せきげん, アニンドヒタ デワブハラタ
国立台湾科学技術大学工業管理学部
台湾台北市 106 大安区基隆路 4 段 43 号
sychou@mail.ntust.edu.tw

運用性 (interoperability) が促進される [8]. さらに、異なるステークホルダー同士の情報共有もできる [9].

理想的な CPS では、すべての要素が情報をやり取りする [10]. 生産工程における自律的な決定のために、通信は中央サーバーを介して、あるいは参加者間で直接処理される [11]. また、個々の要素の機械可読な構造化情報はセマンティック技術を用いて表現される [12]. その中で最も代表的なものとして、Active Semantic Product Memory がある [13, 14]. この方法は、個々の要素のデータの逐次的に収集し、それぞれの要素の情報をコンテキストに応じて取得する [15].

さらに、産業における CPS の開発では、SOA が用いられる [16]. SOA を採用することで、CPS 全体を小規模な単位に分解して、個々の単位の集合によって大規模な問題のソリューションを得ることができる [17]. 最近、SOA の後継として、マイクロサービスアーキテクチャが提唱されている [18]. SOA とマイクロサービスはいずれも、ネットワーク全体に対して有効なサービス (機能) としてシステムを分解し、異種のプラットフォームを統合する. しかし、それぞれの目標を達成するのに、異なる方法を用いている. SOA は複数のサービスを仲介する技術 (スマートルーティング) により各サービスが統合されるため、システムは単純な機能のみを提供するサービス群へと分割される. 他方、マイクロサービスは各サービスが一对一に連携するため、個々のサービスはより複雑な機能を有することとなる [19]. こうすることにより、サービスはシステム全体の取り決めに従う必要がなくなるため、より高いサービス自律性と分解へとつながることになる. 本稿では、マイクロサービスがいかに CPS の開発に役立つのかについても述べる.

次節では、RAMI4.0 について詳述する. そして、3 節では、セマンティック技術について論じ、4 節で SOA およびマイクロサービスについて述べる. 最後に、5 節で本稿を総括する.

2. リファレンスアーキテクチャモデル・インダストリー 4.0 (Reference Architectural Model Industrie 4.0; RAMI4.0)

RAMI4.0 はインダストリー 4.0 にとって極めて重要なフェーズを三つの次元で表現するシステムである. このモデルの目的は、相互に複雑に関連する産業プロセスをより小さくかつ簡潔にまとめることにある. RAMI4.0 はサービス指向アーキテクチャであり、レイヤーやライフサイクル上にすべての要素や IT コン

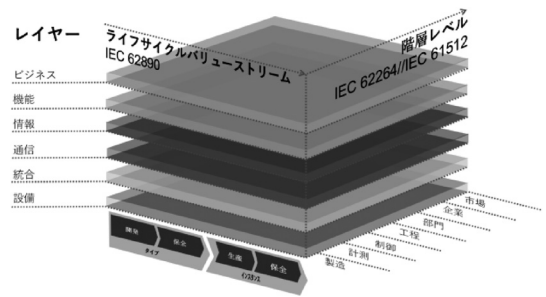


図 1 RAMI4.0 [6, 20]

ポーネントを結び付ける.

図 1 に示すように、縦軸は階層レベルを表している. 左横軸はライフサイクルとバリューストリームを表している. 右横軸はインダストリー 4.0 におけるコンポーネントの機能を位置づけている.

それぞれの内容は次のとおりである [6, 20, 21]. 縦軸には六つのレイヤーがあり、機器を機能別に分解して表現する. すなわち、機器のバーチャルマッピングである. これは ICT の文脈とあわせたものであり、通常、複雑なシステムの属性がレイヤーに分解される. 六つのレイヤーは、ビジネス、機能、情報、通信、統合、設備でありそれらはそれぞれ、1) 企業とビジネスプロセス、2) 設備の機能、3) 必要とされるデータ、4) 情報にアクセスする通信、5) 現実からデジタル世界への移行に関する統合、6) 現実世界の物理的な設備、を示している.

左横軸は、製品ライフサイクル管理を表しており、この分類は IEC 62890 に基づいている. そして、「タイプ (型)」と「インスタンス (実体)」の区別がなされている. さらに、デザインとプロトタイプングが完了し、実際の製品が生産されると、「タイプ」は「インスタンス」に変換される. タイプは、設計オーダー、開発、テスト、そして生産プロトタイプを含んでいる. さらに、コンポーネント、機械、ハードウェア/ソフトウェアなどのタイプが連続生産のための基盤を形成している. 生産された各製品はそのタイプのインスタンスを表している. 顧客に販売されたインスタンスは単にタイプである. それらがシステムに実装されるとき、インスタンスになる. タイプからインスタンスへの変化は何度も繰り返される. ライフサイクルとバリューストリームの詳細な構造が RAMI4.0 の左横軸に示されている.

右横軸では、工場やプラント内での役割と責任が表されている. この分類は、企業の IT や制御システムに対する国際標準シリーズである IEC 62264 の階層

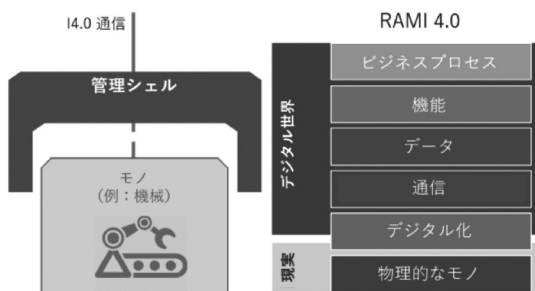


図2 インダストリー 4.0 のコンポーネント：設備と管理シェルの統合 [24]

構造によるものである。この階層は工場や施設における機能を表しており、次の七つのレベルで構成されている。すなわち、1) 市場、2) 企業、3) 部門、4) 工程（つまり機械）、5) 制御、6) 計測（センサーやアクチュエータ）、7) 製造、である。インダストリー 4.0 環境を特徴づけるために、これらの機能性は、「生産」レイヤーの製品から IoT やサービスへの接続に対する「市場」レイヤーを包含するように拡張されている。

RAMI4.0 には、CPS がインダストリー 4.0 で満たすべき詳細な特性が描かれている [22]。「管理シェル」は、インダストリー 4.0 において、機器や設備をデジタル世界と接続可能なコンポーネントに変える鍵となる要素を表している [23]。それはバーチャル世界における特性や状態の保存や、インダストリー 4.0 通信を通して、データをほかのコンポーネントと共有することを可能にする。図 2 では、インダストリー 4.0 のコンポーネントの一例として、設備の管理シェルへの統合を表す。図中、管理シェルは、通信レイヤー、データレイヤー、機能レイヤー、ビジネスプロセスレイヤーにおいて要素を展開している [24, 25]。

RAMI4.0 は業界ごとに異なる視点を統合し、インダストリー 4.0 技術の共通の理解を提供するものである。生産オートメーションや機械工学の業界における要件について、共通の理解を提供する RAMI4.0 を使用することが、産業界の標準化委員会で示されている。さらに、RAMI4.0 はインダストリー 4.0 の 3D マップも提供し、インダストリー 4.0 を定義したり、さらに発展させたりするために、国内ならびに国際標準に各業界の要件を描くための指針を示すことができる。こうして、標準の重複やギャップを識別し解決することができる。

RAMI4.0 は、産業ネットワークを介して、各コンポーネント間の意思疎通を図るために、インダストリー 4.0 のコンポーネントで構成される CPS に対する要件

を指定する。RAMI4.0 は、各コンポーネントが機械可読なデジタル表現をもち、セマンティック技術を用いた SOA を用いることを示唆している [26]。

3. セマンティック技術 (Semantic Technology)

近い将来には、さらに多くのデバイスが一つに接続されるようになるであろう。このような機能は、製品のライフサイクルのはじめ、すなわち実体化する段階から、生産中においても配置することができる。各デバイスには、必要に応じてそれぞれを識別し、センシングや通信するために Active Semantic Product Memory のような技術が埋め込まれ、逐次的にすべての種類のセマンティックデータが蓄積される [13, 14]。生産中に個々のデバイスは、柔軟性や迅速性を保ちながら生産システムと通信しており、プロダクトメモリに対するセマンティック技術は、これらのデバイスの全ライフサイクル期間中、その相互運用性を保障しなければならない。

セマンティックモデリングはデータと、エンティティ間の関係を定義する。情報モデルにより異なる種類のデータを抽象化できるようになり、データ要素間の関係性がわかるようになる [27]。そして、セマンティックモデルにおけるエンティティのすべての集合は、現実世界を表現するためのクラスのタクソノミー（分類基準）を含んでいる。

セマンティックモデリングは公開された標準であるため、複数の方法によってドメインベースのオントロジー（概念体系）が開発されている。セマンティックウェブサービスに対する OWL (Web Ontology Language) ベースマークアップである OWL-S は、プロセスのモデリングと同様にサービスをセマンティックに表現することができる。オントロジーを使用するオーケストレーション・アプリケーションがあり、たとえば、SOCRADES プロジェクトは生産サービスのセマンティックな知識発見やオーケストレーションのためにオントロジーを使用している [28]。

CPS の重要な特徴に、イベント、センサー、工場・機械、製品を表現することがある。最近のセマンティック技術の進展により、それらについてサービスの発見、選択、組成を自動化する際のセマンティック情報の利用が推進されてきた。このような技術は、CPS における協調問題の解決にも援用される。

セマンティックモデリングはエンティティのシステムでの役割を定義するのにも用いられる。これには、シ

ESBはプロバイダとサービスのユーザ間のさまざまな二地点間接続で構成される共通の通信バス経由での通信を可能にするインテグレーションアーキテクチャである [35]。このソリューションは、標準的なデータモデルの導入に向かう傾向がある。ここで重要なことは、インテグレーションのプラットフォームは高度な処理を行えるが複雑ということである。

しかし、異なるチームのサービスにプロセスが横断してまたがると状況が悪化し、システム開発とデプロイの一貫性が必要になる。さらに、SOA サービスインフラストラクチャのすべての変更を管理する単一の集中管理ユニットが保守的なポリシーをもつ、あるいは個々のサービスの制限につながる可能性もある。こういった問題を回避するためにアーキテクチャとして、本稿では CPS のマイクロサービスアーキテクチャを紹介する。

4.2 マイクロサービスアーキテクチャ

マイクロサービスアーキテクチャ (MSA) は、独立して展開できる小規模のモジュール方式の一組のサービスとしてアプリケーションを構造化するアーキテクチャである。各サービスは独自のプロセスを実行し、ビジネス目標にかなうようにうまく定義された軽量のメカニズムで通信を行う [18, 36]。MSA は UNIX の三つの考え方を基にしている [37]。すなわち、1) プログラムは一つのタスクのみを実行してそれを正しく処理すべきである、2) プログラムは同時に複数に動かせるようにすべきである、3) プログラムはユニバーサルインターフェースを使用すべきである、という3点である。これらの考え方は、モジュール化をサポートする再利用可能なコンポーネント設計へとつながっている。重要なことは、サービスが互いに独立して提供されるということであり、このことは、多くの SOA ソリューションとの主要な相違点の一つである。これはシステムのデプロイだけでなく、その進化や改良への努力に対しても影響を及ぼす。

MSA は大規模かつ複雑なアプリケーションの連続的なデプロイを可能にする。Amazon, eBay, Netflix のような多くの大企業は、アプリケーションやプロダクトをサポートするために MSA を用いている [38]。MSA は、それとは真逆のモノリシックアーキテクチャを通して理解することができる。モノリシックアーキテクチャに基づくアプリケーションは単一の自律的ユニットとして開発される。クライアント・サーバーモデルでは、サーバー側のアプリケーションは、HTTP のリクエストを実行し、データベース内のデータを処

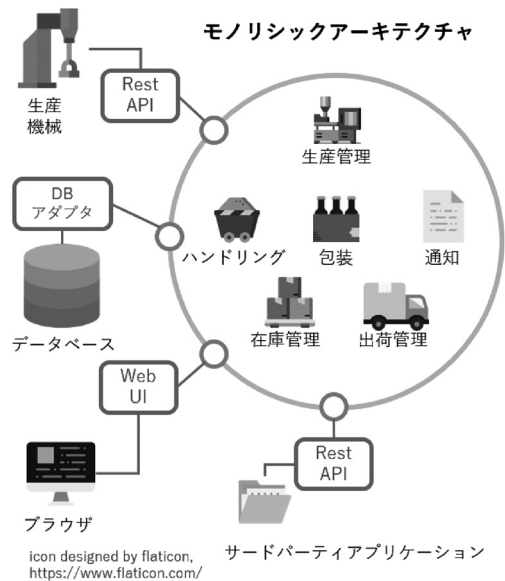


図4 モノリシックアーキテクチャの例

理した結果がブラウザへ送られる一塊のシステム、つまりモノリスである。工場のサイバーフィジカルシステムに対するモノリシックアーキテクチャの例を図4に示す。

しかし、図4からわかるようにモノリシックアーキテクチャの問題は、あらゆるパーツ間に相互依存関係があり、もし生産マネジメントやハンドリングモジュールの小部分に変更が加えられると、すべての機能に修正が必要となったり、さらには全く新しいバージョンを構築もしくはデプロイする必要があることである。この理由は、すべてのモジュールが同じプロセス内で作動しているからである。さらに、あるモジュールにバグがあると、アプリケーションのすべてのインスタンスは同一であるため、バグがアプリケーション全体に影響を及ぼし、場合によってはプロセス全体をダウンさせることがある。

さらに、モジュールの特定の機能を拡張するためには、対象とするコンポーネントだけではなく、モジュール全体を調整しなければならないかもしれない。たとえば、図4において、在庫モジュールについてデータベーステーブルをスケールアップする必要があるとき、ほかのモジュールのデータベーステーブルも同じデータベース内にあるため、それらにも影響するかもしれない。

これとは対照的に、MSAはシステムをより小さく相互に連結したサービスの集合に分割する。それぞれのサービスは、生産管理や在庫管理などの一組の明確

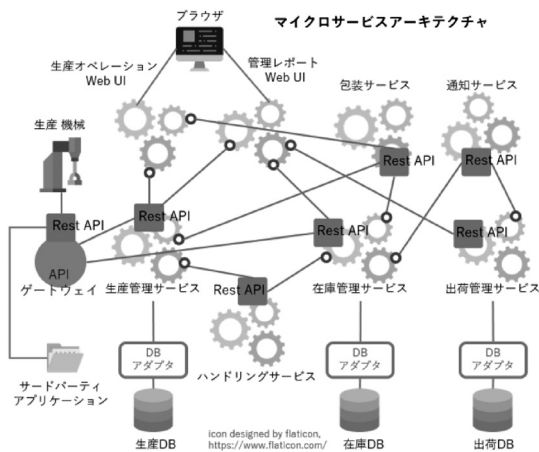


図5 マイクロサービスアーキテクチャの例

な機能を実行する。図5に示すように、MSAでは工場のCPSにおけるそれぞれの領域においては自身のマイクロサービスによって実行される。

いくつかのマイクロサービスはAPIを公開しており、ほかのマイクロサービスやシステムクライアントでも使用できる。さらに、バックエンドサービスはREST APIを公開していて、ほとんどのサービスはほかのサービスにより提供されるAPIを利用している。たとえば、包装管理を例にすると、包装プロセスで何らかの問題や異常を検出すると、オペレータにアラートを出すために通知サービスを用いる。UIサービスは、生産オペレーションのウェブページを配信するために、ほかのサービスを呼び出す。このとき、いくつかのREST APIは生産機械やサードパーティアプリケーションへ公開されている。通信はAPIゲートウェイを介して受け渡される。APIゲートウェイはロードバランシング、キャッシング、アクセスコントロール、APIメタリング、モニタリングなどのタスクを担っている[39]。SOAと異なり、MSAはウェブサービス仕様(Ws-*)やエンタープライズサービスバス(ESB)を用いていない。その代わりに、MSAはRESTのような簡潔で軽量のプロトコルを奨励しており、マイクロサービス自身でESBのような機能を実行する。

MSAにおけるサービスの多くは、独自のデータベースをもっている。サービスはニーズに最も合った異なるデータベースを使用できる。たとえば、自律的な倉庫システムにおいて、無人搬送車(AGV)のジョブサービスのログは、テキスト・ドキュメント指向データベースを用いている。これに対して、出荷管理では、配送先の場所に関する地理情報のクエリを扱うデータベースを用いたほうがよい(図6)。

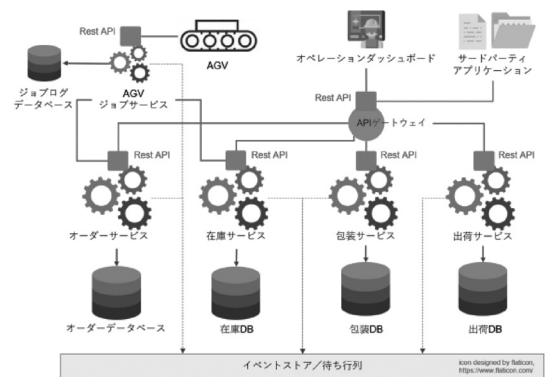


図6 イベントソーシングを用いたマイクロサービスアーキテクチャの例

しかし、マイクロサービスによって異なる種類のデータベースを用いることは、データ管理の問題を起すことになる。たとえば、システムがAGVとそれがオーダーを実行したジョブを表示する必要があるとする。オーダーテーブルとAGVテーブルはそれぞれのサービス専用であるため、AGVジョブサービスはオーダーテーブルに直接アクセスすることはできず、オーダーサービスによって提供されるAPIのみを使用することができる。そのため、1台のAGVがすでにオーダーを実行していて、AGVジョブサービスで状態の変化があるなら、そのイベントを公開するのは難しい。この問題を解決するにはすべてのイベントを記録するイベントソーシングを用いることである[40-42]。ビジネスエンティティの状態が変わるとき、新規のイベントがイベントのリストに追加される。アプリケーションは、イベントのデータベースであるイベントストア内にイベントを保存する。ストアにはエンティティのイベントを追加したり検索したりするためのAPIがある。イベントストアはメッセージブローカーのようにも機能し、サービスをイベントに予約するAPIを提供している。サービスがイベントストア内にイベントを保存すると、すべての関係のあるサブスクライバーへ伝えられる。図6はAGVとオーダーサービスに対するイベントストアの用法を示している。

十分に独立したマイクロサービスコンポーネントは、迅速性と生産性を両立させた自律性をもつ。開発チームはプログラムを完全に理解でき、そして、はるかに速いサイクルでほかのコンポーネントについて独立に構築・デプロイし、テストすることができる。マイクロサービス開発チームは、負荷量の変化に対して、リソースを効率的に利用することができ、また、迅速に対応できる。すなわち、実行時にほかのマイクロサー

ビスのコンポーネントとは独立にそのコンポーネントのスケールアップをすることができる。また、ほかのコンポーネントでの故障とは独立し対応力や復旧力をもつという特徴もある。

SOAとマイクロサービスは、それぞれ異なる方法でシステムをサービスに分割する。SOAはデプロイの観点から、モノリスとして考えることができる [37]。それに対して、マイクロサービスは独立したデプロイを目指しており、簡潔に自動化されたデプロイが可能な自律的なサービスとして考えられている [43]。したがって、SOAが企業活動全体を包括することを目指したものであるのに対して、マイクロサービスは「一つの業務に対する処理をうまく実行する」ことを意図しているという違いがある [38]。

SOAは集中管理方式からシステムの柔軟性を得ることができ、さらにマイクロサービスは分散処理へ向かっている。マイクロサービスはクラウドコンピューティングのコンテキストとうまく適合している [18]。サイバーフィジカル生産システムへの産業界のニーズを考えれば、今後はますますマイクロサービスが指向され、SOAはレガシーとなるだろう。

5. 結言

本稿では、ITとオペレーション技術を結び付けるための、CPSの役割に関する枠組みについて述べた。また、RAMI4.0、セマンティック技術、SOA/マイクロサービスについて論じた。RAMI4.0は、情報を交換、利用する性能をもっている。それにより、異なるステークホルダーがニーズ、課題、関係性について明確に説明したり、意思疎通ができるようになる。

CPSにおいて、セマンティック技術により個々のコンポーネントやデバイスが識別、通信、そして自動的に蓄積されたあらゆる種類のセマンティックデータを認識することができる。さらに、接続されたデバイスはオブザーバーとして役割を果たしている。そして、デジタルツインの概念を実現するために、デバイスがどのように、またどのような環境で使われるのかを知るのに役立つとともに、より簡単に、またより効果的に製品を修正、改善することにつながる。

インダストリー 4.0をサポートするために、CPSにおいてSOAやMSAを用いることにより、CPSにおいて規模や複雑さが増大するという問題への解決策を得ることができる。これによって、新しいレベルの融通性のあるスケーラビリティや障害許容設計で、一層迅速な開発様式とアプリケーションを組み込んだCPS

を構築することができる。

付記 本稿は、高桑宗右エ門（中央大学理工学部）、野村淳一（星城大学経営学部）、三輪冠奈（名古屋学院大学商学部）、譚奕飛（中央学院大学商学部）が原著者との協議に基づいて日本語原稿に翻訳した。

参考文献

- [1] D. He, A. Lobov, and J. M. Lastra, “ISA-95 tool for enterprise modeling,” *ICONS 2012, The Seventh International Conference on Systems*, 2nd edition, H. Kaindl, L. Koszalka, H. Mannaert and M. Jäntti (eds.), IARIA, pp. 83–87, 2012.
- [2] D. Brandl and B. Consulting, “What is ISA-95,” *Industrial Best Practices of Manufacturing Information Technologies with ISA-95 Models*, BR&L Consulting, 2008.
- [3] H. Kagermann, J. Helbig, A. Hellinger and W. Wahlster, “Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry,” Final report of the Industrie 4.0 Working Group, Forschungsunion, 2013.
- [4] R. Drath and A. Horch, “Industrie 4.0: Hit or hype? [Industry Forum],” *IEEE Industrial Electronics Magazine*, **8**(2), pp. 56–58, 2014.
- [5] M. Weyrich and C. Ebert, “Reference architectures for the Internet of Things,” *IEEE Software*, **33**, pp. 112–116, 2016.
- [6] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple and M. Hankel, “Status Report–Reference Architecture Model Industrie 4.0 (RAMI4.0),” VDI-Verein Deutscher Ingenieure eV and ZVEI-German Electrical and Electronic Manufacturers Association, 2015.
- [7] H. Fleischmann, J. Kohl and J. Franke, “A reference architecture for the development of socio-cyber-physical condition monitoring systems,” In *Proceedings of 2016 11th System of Systems Engineering Conference (SoSE)*, pp. 1–6, 2016.
- [8] M. Hankel and B. Rexroth, “The Reference Architectural Model Industrie 4.0 (RAMI 4.0),” ZVEI, 2015.
- [9] K. Suri, J. Cadavid, M. Alferéz, S. Dhoubib and S. Tucci-Piergiovanni, “Modeling business motivation and underlying processes for RAMI 4.0-aligned cyber-physical production systems,” In *Proceedings of 22nd IEEE International Conference on Emerging Technologies and Factory Automation*, 2017.
- [10] Bundesministerium für Wirtschaft und Technologie, “Machine-to-Machine Kommunikation—eine Chance für die deutsche Industrie,” *AG2 Kompetenznetzwerk Machine-to-Machine-Kommunikation*, 2011.
- [11] D. Strang, N. Galaske and R. Anderl, “Dynamic, adaptive worker allocation for the integration of human factors in cyber-physical production systems,” In *Advances in Ergonomics of Manufacturing: Managing the Enterprise of the Future: Proceedings of the AHFE 2016 International Conference on Human Aspects of Advanced Manufacturing*, C. Schlick and S. Trzcieliński (eds.), pp. 517–529, Springer, 2016.

- [12] W. Wang, A. Pfortner, K. Lindow, H. Hayka and R. Stark, "Using ontology to support scientific interdisciplinary collaboration within joint sustainability research projects," In *Proceedings of the 11th Global Conference on Sustainable Manufacturing-Innovative Solutions*, 2013.
- [13] W. Wahlster, *SemProM Foundations of Semantic Product Memories for the Internet of Things*, Springer, 2013.
- [14] W. Wahlster, "The semantic product memory: An interactive black box for smart objects," *SemProM: Foundations of Semantic Product Memories for the Internet of Things*, W. Wahlster (ed.), Springer, 2013.
- [15] W. Wahlster, "Semantic technologies for mass customization," *Towards the Internet of Services: The THESEUS Research Program*, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich and T. Widenka (eds.), Springer, pp. 3–13, 2014.
- [16] L. D. Lago, O. Ferrante, R. Passerone and A. Ferrari, "Dependability assessment of SOA-based CPS with contracts and model-based fault injection," *IEEE Transactions on Industrial Informatics*, **14**, pp. 360–369, 2017.
- [17] A. Gruettner, J. Richter and D. Basten, "Explaining the role of service: Oriented architecture for cyber-physical systems by establishing logical links," In *Proceedings of the 25th European Conference on Information Systems (ECIS)*, pp. 1853–1868, 2017.
- [18] T. Cerny, M. J. Donahoo and J. Pechanec, "Disambiguation and comparison of SOA, microservices and self-contained systems," In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pp. 228–235, 2017.
- [19] A. Balalaie, A. Heydarnoori and P. Jamshidi, "Migrating to cloud-native architectures using microservices: An experience report," In *Proceedings of Advances in Service-Oriented and Cloud Computing: Workshops of ESOC 2015, Revised Selected Papers*, pp. 201–215, 2016.
- [20] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple and M. Hankel, "Reference Architecture Model Industrie 4.0 (RAMI4.0)," Status Report, ZVEI and VDI, 2015.
- [21] F. Zezulka, P. Marcon, I. Vesely and O. Sajdl, "Industry 4.0: An introduction in the phenomenon," *IFAC-PapersOnLine*, **49**(25), pp. 8–12, 2016.
- [22] M. Hoffmeister, A. G. Festo and C. KG, "The Industrie 4.0 Component," Working paper, ZVEI, 2015.
- [23] P. Adolphs, S. Auer, H. Bedenbender, M. Billmann, M. Hankel and R. Heidel, "Structure of the administration shell continuation of the development of the reference model for the Industrie4.0 component," Status Report, ZVEI and VDI, 2016.
- [24] K. Schweichhart, "Reference Architectural Model Industrie4.0 (RAMI4.0) an introduction," *Plattform Industrie4.0*, ZVEI, 2016.
- [25] J. D. Contreras, J. I. Garcia and J. D. Diaz, "Developing of Industry4.0 applications," *International Journal of Online Engineering (iJOE)*, **13**(10), pp. 30–47, 2017.
- [26] F. Pauker, T. Frühwirth, B. Kittl and W. Kastner, "A systematic approach to OPC UA information model design," *Procedia CIRP*, **57**, pp. 321–326, 2016.
- [27] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott and S. McIlraith, "OWL-S: Semantic markup for web services," *W3C Member Submission*, 2004.
- [28] J. L. M. Lastra and M. Delamer, "Semantic web services in factory automation: Fundamental insights and research roadmap," *IEEE Transactions on Industrial Informatics*, **2**(1), pp. 1–11, 2006.
- [29] T. Dillon, E. Chang, J. Singh and O. Hussain, "Semantics of cyber-physical systems," In *Proceedings of Intelligent Information Processing VI: 7th IFIP TC 12 International Conference*, Z. Shi, D. Leake and S. Vadera (eds), pp. 3–12, 2012.
- [30] Z. Yajing, D. Jing, H. Jian, Z. Yansheng, I. L. Yen and B. Farokh, "Abstract service for cyber physical service composition," *Cyber Behavior: Concepts, Methodologies, Tools, and Applications*, IGI Global, pp. 2056–2076, 2014.
- [31] S. K. Mohalik, N. C. Narendra, R. Badrinath, D. H. Le, "Adaptive service-oriented architectures for cyber physical systems," In *Proceedings of 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 57–62, 2017.
- [32] L. Hu, N. Xie, Z. Kuang and K. Zhao, "Review of cyber-physical system architecture," In *Proceedings of 2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, pp. 25–30, 2012.
- [33] J. Morgan and G. E. O'Donnell, "The cyber physical implementation of cloud manufacturing monitoring systems," *Procedia CIRP*, **33**, pp. 29–34, 2015.
- [34] A. V. Feljan, S. K. Mohalik, M. B. Jayaraman and R. Badrinath, "SOA-PE: A service-oriented architecture for planning and execution in cyber-physical systems," In *Proceedings of 2015 International Conference on Smart Sensors and Systems (IC-SSS)*, pp. 1–6, 2015.
- [35] J. Kress, B. Maier, H. Normann, D. Schmeidel, G. Schmutz and B. Trops, *Enterprise Service Bus*, Oracle, 2013.
- [36] C. Richardson, *Introduction to Microservices*, NG-INX, 2015.
- [37] E. Wolff, *Microservices: Flexible Software Architecture*, Addison-Wesley Professional, 2016.
- [38] S. Newman, *Building Microservices: Designing Fine-grained Systems*, O'Reilly Media, 2015.
- [39] A. Kharenko, "Monolithic vs. Microservices Architecture," <https://articles.microservices.com/monolithic-vs-microservices-architecture-5c4848858f59> (2017年11月1日閱覽)
- [40] C. Richardson, "Pattern: Event sourcing," <http://microservices.io/patterns/data/event-sourcing.html> (2017年11月1日閱覽)
- [41] B. Familiar, "Microservice architecture," *Microservices, IoT, and Azure*, Springer, pp. 21–31, 2015.
- [42] M. Macero, "Starting with microservices," *Learn Microservices with Spring Boot*, Springer, pp. 99–177, 2017.
- [43] N. Kratzke and P.-C. Quint, "Understanding cloud-native applications after 10 years of cloud computing: A systematic mapping study," *Journal of Systems and Software*, **126**, pp. 1–16, 2017.