

# マイクロプロジェクト型サービスとしてのIoTシステムの開発・実装・運用のためのフレームワーク

○出口弘（東京工業大学）

## Framework of IoT Systems Development, Deployment and Management for Micro Project Services

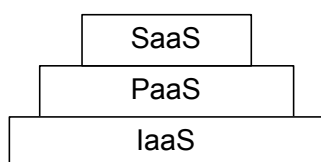
概要— 本稿では、IoT時代のマイクロサービスの粗結合型のシステム開発・実装・運用のフレームワークについて論じる。既にDevOps型開発によるクラウド上でのマイクロサービスの粗結合による情報システム開発は広がりつつある。だが我々が本稿で論じるのはクラウド上でのオープンプラットフォーム上でのひと・もの・ソフトを含む実世界ノードからなるマイクロサービスからなるマイクロプロジェクトという形での自律分散・粗結合型の実世界システムの開発のフレームワークである。これをまずこれをシステム開発のモデリングサイクルの視点からこれを論じ、更に工場のIoTシステム開発の事例を延べ、その上でマイクロプロジェクト型の粗結合システム開発を支援するオープンプラットフォームであるリアルワールドOSについて述べる。

キーワード： IoT, プラットフォームロックイン, マイクロプロジェクト, マイクロサービス, 粗結合, DSEMDCATS, リアルワールドOS

### 1. はじめに

今日の情報システム開発は、一体型（モノリシック）なシステムを運用環境も含めて開発するウォーターフォール型開発から、粗結合なシステムを運用側からのフィードバックを受けながらマイクロサービス単位で開発するDevOpsによるマイクロサービス型の開発へと変化しつつある。これはまたIoTのシステム開発にも大きな影響を及ぼしている。しかし他方で今日のマイクロサービス開発は、SaaS(Software as a Service)をPaaS(Platform as a Service), IaaS (Infrastructure as a Service)と呼ばれる fig. 1.1 で示される様なクラウド上の階層的プラットフォーム構造を前提として行われている。

fig. 1.1 SaaS, PaaS, IaaSの階層プラットフォーム構造



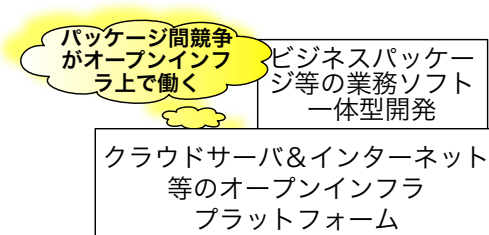
これはインターネットの勃興期にブラウザ、検索サービス、SNA等のコミュニケーションサービスのプラットフォーム（B2Cプラットフォーム）が階層的な構造を持ちつつロックインしていったのと同様のロックインが、我々のビジネスの根幹となる様々な機能的なシステムに関するB2Bプラットフォームに関しても生じる可能性があることを示している。

実際従来の情報システムの一体型の開発であれば、fig. 1.2に示す様に、インターネットやクラウドのようなオープンプラットフォーム環境の上で、業務パッケージ間競争の市場はそれなりに働いてきた。しかし近年のマイクロサービス化とDevOpsなどの新しい開発方式の進展は、この競争環境を一変させ、新たなロックインをB2C領域で惹起する可能性をはらんでいる。

B2Cのプラットフォームのロックインでは、プラットフォーム上へのサービス提供者とプラットフォーム上でそのサービスを選択する顧客が、ともに顧客数の

多いプラットフォーム、サービスの豊富なプラットフォームを選択しようとする選好を持つ事で、相互に強化しあう形で勝ち組のプラットフォームへロックインが進む傾向が顕著に生じた[Deguchi, 2000, 2004]。

fig. 1.2 オープンインフラ上の一体型ビジネスソフトの競争構造



これはインターネット革命以前の経済学で課題とされていた、収穫逓増に基づく独占（或は寡占）と、それを規制する為の独占禁止法という制度的枠組みとはことなるロジックで、独占的な構造が生じてきたことを意味している。この新たな独占構造は、現在までに少なくとも制度規制面を含めて十分議論されてきたとは言いがたい。さらに近年のデータ処理能力の増大は、巨大化したプラットフォームの上で顧客の膨大なデータが収集しそれが更にプラットフォームの競争力を強化するという形で、サービス生産力の要素として、労働、資本に加え、蓄積情報が大きな影響を及ぼすようになりつつある。

このようなB2Cの顧客、サービスプロバイダ、プラットフォームプロバイダの三者間の非市場的相互連関に起因するロックインの構造は、従来B2B市場にはそれほど大きな形では生じていなかった。しかしそれでも、パソコン市場と、携帯電話市場を比べたとき、オープンなオペレーティングシステムがプラットフォームとして存在するパソコン市場と、アンドロイドとiOSの二つのOSに事実上寡占化されているスマートフォン市場ではその競争環境は大きく異なっている。

現在、ビジネスソフトウェア市場は、多くのマイクロサービスを提供するマイクロサービスプロバイダー

(SaaSの提供企業)と、そのマイクロサービスのプラットフォームプロバイダ(PaaSの提供企業)がオープン利用可能なクラウドやインターネット上での競争を開始したばかりである。そこでは一般の顧客にかわり、マイクロサービスとその組合せによるビジネスソフトウェアの利用者(以下B顧客)とSaaSの提供企業とPaaSの提供企業の間で、B2Cプラットフォームのロックイン過程で生じた様な相互連関が生じ始めている。結果としてfig. 1.3に示す様に、今後B2Bの業務ソフト市場でロックインが進むことが予想される。

fig. 1.3 B2Bプラットフォームのロックインリスク



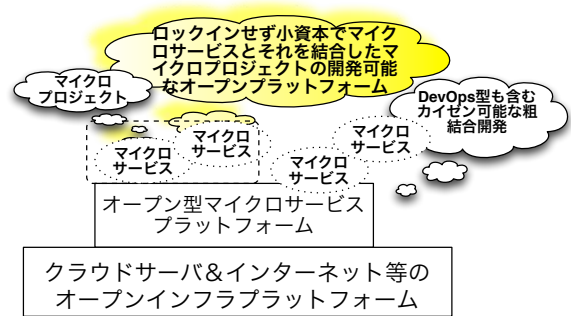
しかしこれだけであれば、ビジネス領域の一部に影響を及ぼすに過ぎず、物理的実態を持った多くのビジネスは、そのマネジメント業務の一部のシステムがプラットフォーム構造でロックインされるだけでその影響は限定されているという見方もあり得ないではない。

しかしIoT革命が状況を一変させる。IoT或はIoEと呼ばれる、ひと・もの・ソフトモジュールが自律的にネットワーク環境の中で相互作用する世界では、様々なビジネスプロセスそれ自体が、ひと・もの・ソフトモジュール等の多様なIoTのノード間の相互作用としてデザインされ、遂行され、管理されるようになる。このような新たなインターネット革命の時代には、多様なマイクロサービスが様々な形で結びついたビジネスプロジェクトが次々にビジネスプロジェクトイノベーションとして生み出される社会を構想することが求められる。また近年のクラウドファンディングによるイノベーションの広がりや、分散台帳技術、分散組織、共有経済に対する関心の高まりは、ボトムアップにIoTノードの組合せで様々なビジネスプロセスにおけるマイクロサービスとビジネスプロセスの共進化がボトムアップに可能となる新たな社会の到来を予想させる[Rifkin, 2015;Mele, 2014]。

そこに立ちをはだかるのがB2B市場に於ける新たなロックインのリスクである。ロックインされた世界では、IoTベースの実世界のマイクロサービスとそれらを組み合わせたマイクロプロジェクトとしてのビジネスユニットが組織の壁を越え沸き立つ様に出現し、共進化して行くビジネス生態系は想定し難い。

それゆえに我々は、実世界のマイクロサービスを組立てたマイクロプロジェクトを支援するオープンプラットフォームを設計・普及することで、この新たなロックインを未然に防ぎ、超多様な実世界のマイクロサービスとそれを組織間を超えて結びつけたマイクロプロジェクトの絶えざる創成と共進化を可能とする社会を構築することを目指したい。

fig. 1.4 オープン型のマイクロサービスプラットフォーム



本稿では、マイクロプロジェクトという単位で、様々なビジネスシステムを把握した上で、そこに新たな開発のモデリングサイクルとしてDSEMDCATSというマイクロプロジェクト開発のための枠組みを提案する。このDSEMDCATSは、より一般的なビジネスプロセス等の開発のモデリングサイクルであるPDS或はPDCAサイクルを、マイクロプロジェクト型のシステム開発に対して拡張したものである。

このマイクロプロジェクト型のシステムは、IoTを利活用した様々な業務システムの実現のために基盤となるシステム概念となっている。そこでは要素となる業務タスクの粗結合によりマイクロプロジェクト型のシステムは表現される。これはタスクを単位とする結合システムとなっており、その結合が半順序型の構造を持つシステムである。工学の世界ではシステムの結合は線形システム論の中で広く扱われてきた。また論理回路や電気回路、関数型の計算でもサブシステム間の結合の問題は広く論じられてきた。しかしここでは、動的プロセスが微分方程式であったり、回路と行った限定されたサブシステム記述が前提となっている。

ここでは、より広い実世界の業務タスクに関して、人工物として、業務タスクが半順序型に結合したマイクロプロジェクト型のシステムを設計する方法論が新たに課題となる。そのためのモデリングサイクルの枠組みとして我々が提起するのが、DSEMDCATS (Design, Scheduling, Execution, Monitoring, Data Scanning, Control, Analysis, Traceability, Security) というモデル設計からシステム開発・運用・カイゼンのためのフレームワークである。

この DSEMDCATS というフレームワークは、IoTを前提として、サイバーフィジカル空間 (CPS: Cyber Physical Space) 上での実システム開発のためのフレームワークであり、情報システムのみを対象としたものではない。DSEMDCATSのフレームワークでのマイクロプロジェクト型のシステム開発は、粗結合型のシステム開発であり、タスク間の情報伝達は、Pub/Sub型のメッセージングを基盤とするシステム設計となっている。

ここで我々が目指す粗結合型の実世界システムの開発と実装・運用は、特定のプラットフォームにロックされることのない、自律分散協調型のシステム設計・実装・運用のフレームワークを目指している。さらにそこでは、現場のケーパビリティデベロップメント(能力開拓)を行いつつ、ボトムアップに現場知を

取り込み、絶えざるシステムのタスク単位でのプロダクトイノベーションやプロセスイノベーション（カイゼン）を現場が関与する形で可能とするオープンイノベーションを可能とするフレームワークの構築が目指される。この実世界のタスクの開発・実装・運用・カイゼンのための支援環境もまた必要となる。これが我々が実世界OS（リアルワールドOSオペレーティングシステム）として提起している枠組みである。DSEMD-CATSはリアルワールドOSにより支援される、マイクロプロジェクト開発のためのフレームワークである。

リアルワールドOSでは、人やものやソフトウェア等様々な実施形態を持つ個々のタスクに対し、ロールコンテナという枠組みの中に収納することで、個々のタスクが実行する役割の実行管理を行い、他のタスクと連結する為の機能を提供する。その上で、ロールコンテナに収納された様々なタスクをダッシュボード上からタスクやマイクロプロジェクトの諸状態を収集し、見える化し、分析し、管理することができる。

またプロジェクトプログラミングという、実世界のノード間の連結のマイクロプロジェクトとしての自動実行管理を行う為の、プログラミング言語OWLIEを提供している。このようなマイクロプロジェクトに関するタスクの実行管理やタスクそのものの管理機能を含む、サイバーフィジカル空間上でのノード管理のためのフレームワークを我々は実世界OSと呼んでいる。

これは現在のクラウドプラットフォーム上での粗結合の情報システム設計、或はIoTシステム設計とはその設計思想が大きく異なる。我々はオープンイノベーションを可能とし、特定のプラットフォームにロックインしない、実世界ビジネスプロセスの開発と実装・運用のフレームワークを構築することを課題とする。ここでは、クラウドプラットフォーム上でのマイクロサービスの記述やその間の連結技術とは異なった枠組みでの技術が求められる。

本稿では、理論・技術・ビジネスモデルのそれぞれの側面からこの課題に対する取組みと課題解決のための枠組みの提示を行う。またこの枠組みに従った理論・技術・ビジネスモデルに関する応用事例についても述べたい。そのために、我々は人工物としての組織或は組織間関係で用いられるビジネスプロセスの記述分析の方法に関しての評価を行う。その上で、ビジネスプロセスに関する情報システムの開発と、実世界の様々なビジネスプロセスそのものの開発との差異を明確にする。

実世界でのビジネスプロセスを設計実装運用する手法として、プロジェクトマネジメント（PM）が広く行われてきた。ただし PM は主に一回限りの何らかの開発型のプロジェクトという形態のビジネスプロセスを中心に、その設計・実装・運用の方法論は発展してきた。他方で、プロジェクトマネジメントの中で用いられる開発プロセスの記述のためのビジネスプロセスのPART（Program Evaluation and Review Technique）による記述やクリティカルパス分析によるタスク間関係の分析や資源配分計画、スケジューリングの技術はそれ自体、実世界の様々なビジネスプロセス自体の記述や分析に用いることのできる技術である。実際、医療に於けるクリティカルパス分析では、様々な医療プロセスの中でタスクの切り分けとその結合の標

準化とスケジューリングや資源管理のためにクリティカルパス分析を用いており、それを特にクリティカルパス分析と呼んでいる。ここで注意したいのは、個々の検査や手術等の中身をここで詳細に決めている訳ではなく、それらのある範囲で業務モジュール（タスク）として認識しそれらの遂行とその連結、そこへの資源配分等を管理していることである。

本稿ではこのプロジェクトマネジメントに用いられるビジネスプロセスの記述と分析の方式を拡張する形で様々な業務プロセスを設計・実装・運用・カイゼンの管理を行うためのフレームワークをまず DSEMD-CATS（Design, Scheduling, Execution, Monitoring, Data Scanning, Control, Analysis, Traceability, Security）という形で導入する。そこで導入されるビジネスプロセスのことを、一般のプロジェクトマネジメントにおけるタスクの連結としてのプロジェクトと区別して、特にマイクロプロジェクトと呼ぶ事にする。その要素となるタスクについてもマイクロサービスと呼ぶことにする。

このマイクロプロジェクトとしての業務システムの分析・設計・実装・運用・カイゼンの一連のサイクルの中で、我々は「情報システム」ではなく、ひと・もの・ソフトウェアのモジュールが構築するマイクロサービスとしての様々なタスクが連結した実世界システムを構築することを目標としている。この実世界システムを構築するということはそのままビジネスプロセスをその付加価値形成や資源割当等の管理システムを含めて構築するということでもある。従って、ここでは各タスクの付加価値計算に関する会計的な仕組みもまた必要とされる。またマイクロプロジェクト型の業務システムの記述では、個々のマイクロシステムは全体業務システムの一部に過ぎず、それらが更に複雑に連携した全体システム管理が必要となる。マイクロプロジェクトは全体業務システムのサブシステムとして、あるシステムの管理境界で業務システムを切り出したものとなる。更に、一企業の壁を越えてマイクロプロジェクトが連携するプロセスを分析する必要がある場合もある。例えば、部品製造企業の部品が、製品製造企業で用いられるサプライチェーンでは、組織間でのマイクロプロジェクト間の連結が市場を経由して構築されている。しかしそこでもトレーサビリティの様な形でのマイクロプロジェクトのネットワーク化は必要であり、また外注の様な形で、マイクロプロジェクトの一部のタスクが外部組織のマイクロプロジェクトとして実施されるケースも多々ある。これらのマイクロプロジェクトの市場や外注を経由しての連結は、そこでのアプトットである財やサービスの連結関係で見た時に、サプライチェーンを形成する。このような組織の壁を越えた連結データは、経済システムとしてみたときには産業連関表やコモディティーフロー表を構築する為の基礎資料とも成り得る。また様々なものやサービスを産出する多様なマイクロプロジェクトが相互に連結した巨大な回路網として経済システムを把握し、それを制御するマネジメント手法が経済政策であると見なすことも可能になる。

マイクロサービスの要素としての個々のタスクに関しては、そのタスクが実世界の物理プロセスや人間による作業によって実現される場合もあれば、情報処理プロセスによって実現される場合もある。

そこでのタスクのプロセスの詳細な実現を論じる際には、タスクの機能要件、役割要件を如何に実現・実施するかという階層化された詳細設計の課題が生じる。例えば切削加工のタスクではNC工作機械のプログラムや段取り換え工程の詳細プロセスの設計が必要とされる。個々のタスクの詳細実現設計という課題は、マイクロプロジェクトのDSEMDCATS分析の課題とは課題を異にし、タスクの実現設計問題として別途扱われる必要がある。

本稿ではさらに、マイクロプロジェクトによる様々な業務システムの設計・実装・運用・カイゼンの一連のプロセスが新しい社会の組織原理となる可能性を提案したい。現在の業務システムの構築では、情報システム開発が大きな比率を占めている。その過程で業務プロセスの標準化が進み、ERPパッケージが業務を標準化するという逆転した発想がある。これが一体型のシステムからマイクロサービスの粗結合型のサービスに変化する中でも、情報システムを中心とする業務システム設計では、人間に関する人的資本サービスとしての把握や、ケーパビリティデベロップメントなどの「働きかた」に関する視座が希薄なままとなっている。その上で、ビジネスプロセスの構築と運用のための巨大なクラウドプラットフォームが成長する世界は、実ビジネスのプラットフォームに対してロックインが生じる社会となるリスクがある。既に我々は1990年代に始まった第一次のインターネット革命で、コミュニケーションや情報検索や情報活用型サービスを中心に巨大なプラットフォームが成長した歴史を体験してきた。

一体型の業務システムを開発していた時代には、業務パッケージの形での競争はあるが、業務システムの巨大なプラットフォームへのロックインは無かった。

我々が設計する必要があるのは、業務の為の情報システムではなく、人や機械を含む実世界の業務システムである。そこでは付加価値の計測や、資源割当、スケジューリング等、業務の見える化や、タスク設計に於ける人的資本サービスと物的資本サービスの実物簿記による計測や原価計算的課題に答える等、プロジェクトマネジメント的課題と情報システムの課題、人間活動システムの課題が全て含まれる必要がある。

今日インターネット革命の次のステップとして始まりつつある、IoT (Internet of Things) の革命は、従来の情報システムの枠を超えて、ひと・もの(機械)・ソフトウェアなどからなるタスクをネットワーク上で連結しながら業務システムを構築・運用することが可能となりつつある。このIoT技術の進化は、必然的に我々の社会の組織の構造や働きかたに強い影響を与えざるを得ない。特にクラウド上の巨大プラットフォーム上で構築・運用されるビジネスシステムは、マイクロサービスの粗結合ではあっても、そのプラットフォームが強くロックインされることで、ビジネスプラットフォームの寡占化を招き、結果的にそのプラットフォーム上で提供されるマイクロサービスの範囲でしか自由度を持たないシステムに収束するリスクがある。これは現場でのケーパビリティデベロップメントや不断に行われ、或はマイクロサービス単位での超多様なオープンイノベーションがそれぞれの現場から頻出する社会ビジョンとは共存しない。我々は新しい時

代の働きかたや組織の設計に関するプログラムを持つIoT時代の実世界のシステム開発のアーキテクチャを模索したい。

我々が提唱する実世界の業務システムの開発・実装・運用・カイゼンのためのシステム開発では、マイクロサービスの単位となる諸タスクを組み合わせたマイクロプロジェクトの設計・実装・運用・カイゼンのためのモデリングサイクルとして、DSEMDCATSという枠組みを提起し、その中でIoTを利活用したシステム構築を目指す。そこではまず、Publishing & Subscribe型のプルーカを経由したメッセージングのプロトコル(Pub/Subプロトコル)を共通のデータ交換の基盤インフラ(データバス)として、業務タスクをインターネット(イントラネット)上のノードとして扱う事ができる様にする。これにより実世界上のひと・もの・ソフト等で実現される業務ノードからPub/Subプロトコルを用い様々なデータを取得したり、ノードの進捗管理をしたり管理業務が可能となる。またノード間でも様々な業務メッセージ交換がPub/Subプロトコルを用いできるようになる。

このマイクロプロジェクトとしての実世界の業務プロセスの設計・実装・運営・カイゼンのフレームワークの構築では、単独で小規模なマイクロプロジェクトのシステム構築は、比較的簡単なPub/SubプロトコルだけのIoTの枠組みを援用することで構築・運用が可能となる。しかしより多様で大規模な業務に関するマイクロプロジェクトの構築のための設計・実装・運営・カイゼンのフレームワークの適用では、ネットワーク上の様々なタスクをPub/Subプロトコルを用い共通のAPIを用いて管理する必要が生じる。その為には実世界の業務タスクのノードを管理し、さらにマイクロプロジェクトの自動実行を支援することのできる専用の支援システムが求められる。個々のタスクを共通のAPIを持つノードとして扱える様にするために、ひと・もの(機械)・ソフトウェアなどからなるタスクを共通のコンテナに収納することで、共通のAPIを通じてタスクをネットワークのノードとして管理し、或はマイクロプロジェクトとしてのタスクの遂行制御の枠組みを提供するなどの支援環境の構築が求められる。これはより複雑な業務タスクのオーケストレーションをIoT環境でネットワーク上で行うためには必須の機能である。クラウド上では例えば、ドッカーのような仮想化コンテナ上にマイクロサービスを実現する情報処理システムを置き、それらをNode Redのようなエンタープライズサービスバスで接合するシステム構築が行われている。

我々はこれをオープン化され、特定の巨大クラウドサービスに収斂しない実世界ビジネス構築の枠組みとして構築する。そのための枠組みとして我々が提起するのが、後述するDSEMDCATSという業務プロセスを設計・実装・運用・カイゼンの管理を行うための開発フレームワークであり、リアルワールドOSと呼ぶIoTを前提とするサイバーフィジカル空間でのノード管理とノード間でのマイクロサービス型のオーケストレーションのための支援枠組みである。

## 2. 組織の業務プロセスと情報処理システム

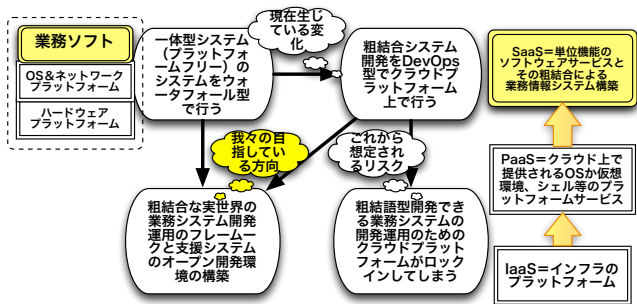
組織のマネジメントと情報システムの関係は、

今日大きく変化しようとしている。従来主流であったモノリシック（一枚岩）型の情報処理システムの開発は、開発とオペレーションの相互作用を強調するDevOpsなどの新しい開発手法とともに、モジュール間がルースカップリング（粗結合）したシステムへの移行が言われるようになりつつある。

だが現状その粗結合技術は、例えばマイクロソフトのAzureに見られるように、クラウド上でのマイクロサービス提供とそのサービス連結による業務システムの開発・実装・運用のプロセスのプラットフォーム化が進んでいる。

他方で、今日別の視点からもモノリシックな情報システムの在り方に対する問い掛けが求められる。第一に、組織の活動に対する情報システムの位置づけを問い直し、組織の活動を管理するためのシステムと、組織の活動を構成するタスクそのものの中の情報システムをきちんと切り分けて論じる必要がある。例えば工場での生産プロセスでは、個々の加工や組立のタスクそのものは機械或は人の手によるものであり、情報システムではない。実世界の業務システムをIoT環境下で粗結合なシステムとして扱う為には情報システムの開発に於けるビジネスプロセスのモデリング方法を超えた、モデリングの枠組みが必要となる。それがマイクロプロジェクトとしての業務システムのモデリング手法である。

fig. 2.1 一体型開発から粗結合型のシステム開発に向かう二つの方向性



組織の活動を遂行する基本単位としての様々なタスクを切り出し、その系統だった結合をマイクロプロジェクトとして設計する必要がある。そこではタスクの遂行主体或は主要資源が機械や人である場合も含めて、タスク単位での資源の投入と財やサービスの産出構造が明確にされ、そこでの原価計算も適切に行うことが求められる。更に実際にマイクロプロジェクトとしてひとまとまりの業務システムを実施する場合には、資源の割当のスケジューリングが行われねばならないし、業務の遂行の過程ではタスク単位での遂行状況のモニタリングやその見える化なども求められる。このような業務分析は従来プロジェクトマネジメントの領域で広く行われてきたものである。

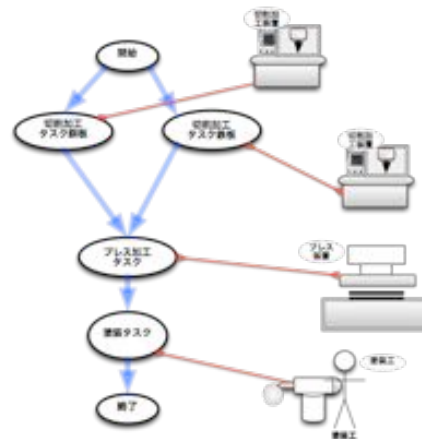
ビジネスプロセスに関する理論分析は、組織理論の歴史でもあり、それはH. A. SimonとJ. G. Marchによる組織理論[March, 2014]における組織の情報処理パラダイムを一つの契機としつつ、オブジェクト指向のシステム設計論等を経て、UMLへと結実し、

組織の情報処理プロセスの設計論へと変貌を遂げる。そこでは業務プロセスの設計の為の分析枠組みが課題となり、DFD (Data Flow Diagram) やBPMN(Business Process Model and Notation)或はWFA( Work Flow Architecture)のような業務プロセスの詳細分析の手法が様々に試みられる一方で、その業務そのものを目的仕様から分析し実現して行く為の、Zachmanフレームワークのような、要求仕様分析の方法も発展してきた。

他方でプロジェクト型のタスクの遂行のためのスケジュールや資源管理のためのPART(Program Evaluation and Review Technique) を経て、プロジェクトマネジメントの枠組みが発達してきた。ここでは、情報システムの開発管理を中心に発展してきた、PMBOK (Project Management Body of Knowledge) と、プラント等の広範囲な実世界の業務開発のための枠組みとしてプログラムマネジメントを付加する形で発展してきたP2M (Project & Program Management) のように幾つかの流れはあるが、総じて何らかの業務或は業務の支援システムとしての情報システムの開発プロセスそのものを課題としている。プロジェクトのアウトプットは、開発目的となるプラントやビジネス情報システム等の業務システムであり、それを解析する為の手法となるのが、Zachmanフレームワークのような要求仕様分析、DFD、BPMN、WFAのような詳細なワークフロー記述、あるいはUMLのような情報システム分析となる。これにE-R図(ERD: Entity Relationship Diagram)のようなデータベースと言う特定のアーキテクチャに依存したシステム設計の方法が加わっている。

その開発プロジェクト自体は、プロジェクトマネジメント或はプロジェクト&プログラムマネジメントの手法で管理される。このとき、プロジェクトのアウトプットは一体型の情報システムの時代が長く続いてきた。これに対して我々は、開発プロセスのアウトプットを、マイクロプロジェクト型の業務システムとして把握する。またマイクロプロジェクトの構成要素である個々のタスクをマイクロサービスとして捉える。

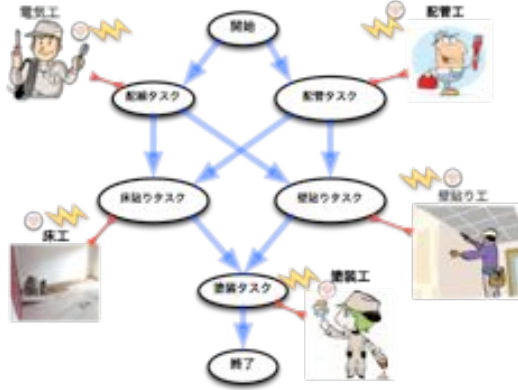
fig. 2.2 マイクロプロジェクトとしての業務プロセスの例1（切削加工での部品の加工業務）



実際、先に例示した医療のクリニカルパスのみならず多くの業務システムは、管理単位となるタス

クが半順序型で接合した構造で表現され、管理することができる。建築での内装工事や、ものづくりに於ける部品の加工プロセスもまた個々の加工のタスクをマイクロサービスとしたマイクロプロジェクトと見なせる。

fig. 2.3 マイクロプロジェクトとしての業務プロセスの例2 (内装工事業務)



これらのタスクが相互に結合したビジネスプロセスの表現では、個々のタスクに関してその実現プロセスを詳細設計することと、それを結びつけて一連の業務を実現するマイクロプロジェクトそのものの設計と管理をすることは別の課題となる。マイクロプロジェクトとしての業務システム設計は、業務のあるまとまった単位をマイクロサービス（タスク）として切り出すことから始まり、タスクの遂行に必要な資材・原料と各々のタスク遂行のために必要な人的サービス、物的サービスを確定し、その原価計算をタスク単位で行い、マイクロプロジェクトとしての原料、人的・物的サービスの資源割当のスケジューリングを行い、更にタスクの開始と終了や状態を管理するモニタリングによりマイクロプロジェクトの見える化と遂行管理を行う必要がある。またタスクの遂行に関して得られたデータを収集し分析し、タスクとマイクロプロジェクト自体のカイゼンに役立てるなどの一連の管理サイクルが求められる。

これらはプロジェクトマネジメントの課題でもある。例えばマンションの内装工事の業務プロセスは、内装の完成したマンションの部屋を業務のアウトプットとするプロジェクトであり、部品作りの加工プロセスは部品を業務のアウトプットとするプロジェクトと見なせる。医療に於ける特定領域の治療の標準作業手続きを定めるクリニカルパスは、その領域での患者を対象とした医療サービスの実施をアウトプットとするプロジェクトである。ここでは業務プロセスの型＝プロジェクトと見なされる。我々が業務プロセスをマイクロプロジェクト把握する場合、このプロジェクトとしての業務を一つの型（タイプ）として、実際の業務は、その型に基づきそのつど生成される業務実態（ビジネスプロセスのインスタンス）であると考えられる。医療のクリニカルパスは実際、領域毎に標準的な治療プロセスの型をデザインし、実際の治療はその型に基づいたインスタンスであると思える。部品の加工プロセスも加

工手配書がマイクロプロジェクトの型を定義し、製番単位での実際の加工がそのインスタンスとなる。このインスタンスを生成するにあたって、資源の割当やスケジューリングが必要となる。

このマイクロプロジェクトを型として、そのインスタンスを生成し業務遂行するという繰り返しのある業務プロセスに我々は着目し、マイクロプロジェクトの型を生成し、そのインスタンスを生成し、インスタンスの遂行を管理する一連のプロセスを、モデリングサイクルとして捉える。ビジネスに於けるモデリングサイクルには、PDS (Plan-Do-See) サイクルやPDAC (Plan-Do-Check-Action) サイクルがコンサルティングなどに用いられ良く知られている。我々は何らかの財やサービスの生産プロセスの型としてのマイクロサービスの設計と、それに基づく業務実態（インスタンス）の生成、運用、カイゼンなどの一連のプロセスを対象とする、マイクロサービスとしての業務システムを開発するためのフレームワークとして、次節でDSEMDCATSというモデリングサイクルを提案する。その上で、多様な実世界の業務をマイクロサービスとしてのタスクとそのマイクロプロジェクトとしての連結として開発・実装・運用・カイゼンする方法論と事例について論じる。

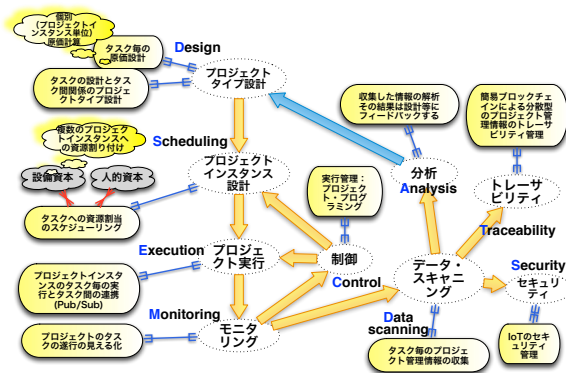
現在のDFD (Data Flow Diagram) やBPNM或はWFA (Work Flow Architecture) などの取組みやUML, E-R図 (ERD: Entity Relationship Diagram) などの設計手法は、結果的には既存の情報処理アーキテクチャを前提としたものになっている。マイクロサービス型のアーキテクチャをプラットフォームに非依存の形で実現し、ボトムアップに常にカイゼン可能はビジネスプロセスを実世界システムとして設計・実装・運用・カイゼンするという課題は、従来型のアーキテクチャとは一線を画す形で設計することが必要となる。

むしろこれは、WFAなどによる業務分析が不要と主張するわけではない。個々のタスクの実現方法は、マイクロプロジェクトの視点からはタスクの詳細なプロセス設計問題である。例えば内装工事の配管工の配管タスクに関しては、その詳細な業務プロトコルの設計も必要とされる。だが同時にタスクの結果は何で、それはどのような投入産出関係で記述され、タスク遂行に必要な原料と人や加工設備のサービス資源投入が明らかにされ、限定された資源の下でのスケジューリングを立てられる必要がある。このような視点は、DFDやBPNM、WFAにはない。人がタスクを実行する場合には詳細なプロセスは人の役割的な技能の標準作業手続きや、ケーパビリティの中に埋め込まれており、業務プロセスの詳細設計とは独立の課題と見なせる場合も多い。これは機械加工でのNC工作機械のプログラムについても同様である。更にマイクロプロジェクトとしての業務プロセスは、そのタスクも、タスク間の結合も頻繁にカイゼンされ変化することを前提に置いたシステム設計と運用が必須となる。我々が課題とするのは、人・もの・ソフトウェアが混在して、ネットワークを利用して結びついた実世界での人間活動システムとしての業務システムの開発のフレームワーク構築である。

### 3. DSEMDCATS分析によるマイクロプロジェクトの設計・実施・運用・カイゼン

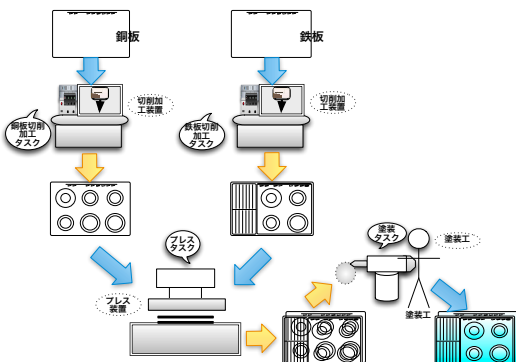
我々は、実世界の生産やサービスのマイクロプロジェクト開発・運用管理のためのモデリングサイクルとしてfig. 3. 1で示されるようなDSEMDCATS (Design, Scheduling, Execution, Monitoring, Data Scanning, Control, Analysis, Traceability, Security) というモデリングサイクルを提唱する。DSEMDCATSは、一回限りのプロジェクトの開発のためのプロジェクトの開発ではなく、繰り返し遂行されるマイクロプロジェクトの型に従った業務について、その型(タイプ)の開発から業務実態の生成(インスタンス生成)・運用、更にカイゼンという一連のプロセスを継続してマネジメントする為のモデリングサイクルである。マイクロプロジェクトの型の設計から、プロジェクトインスタンスの生成、資源割当、計画、実装、運用、見える化、データ収集、制御などの開発・実装・運用・カイゼンを支援するための一連のモデリングサイクルがDSEMDCATSである。

fig. 3.1 DSEMDCATSフレームワーク



DSEMDCATSによる、IoT時代の、ものづくりやサービス生産のマイクロプロジェクト開発と管理を説明する為、fig. 3.2で示される「鉄板、銅板を切削加工した後、これをプレスし、塗装した簡単な部品(筐体)の製造工程」を事例とする。

fig. 3.2 簡単な部品の製造工程のマイクロプロジェクト



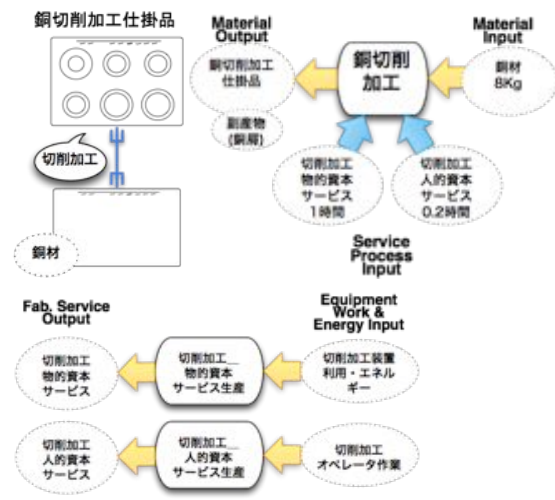
#### (1) プロジェクトタイプのデザイン

プロジェクトタイプの設計では、サービスや製

品の生産、部品の加工などのプロセスを、何らかのまとまった仕事を遂行する単位であるタスクに切り分け、その遂行を行う順序付けられた集まりとして定義する。ある範囲の業務をタスクとして認識することで、その遂行に必要な機械や人の資源や原料を明確にすることができ原価計算がタスク単位で可能となる。またタスクはマイクロプロジェクトのインスタンスに対するスケジューリングと進捗管理、様々なデータ取得の基本単位ともなる。

個々のタスクがどのようにその役割を遂行するかについては、タスクの詳細設計と実現の課題になるが、それらとは別にタスクに対する資源割り付けのタスク単位での計算が必要となる。一つのタスクでは何らかのサービスが遂行され、その結果としてサービス対象の状態が変化する。fig. 3.2の事例の加工のマイクロプロジェクトで、銅板を切削加工するタスクに注目する。銅板の切削加工のタスクでは、原料の銅板と切削加工サービスを投入することで、銅の切削加工仕掛品と副産物の銅屑が生成される。この加工を実際にどのようにNC工作機を動かして実施するかは、設計に基づく機械加工の技能の課題とし別扱いとする。その上でその加工の設計に基づく入出力関係をfig. 3.3のように記述することができる。銅板の切削加工のタスクでの何を元に、何が作られるかの投入産出関係とそこで用いられる物的資本サービス、人的資本サービスの投入産出関係は実物簿記で記述される。

fig. 3.3 銅の切削加工タスクの記述



産出		投入	
借り方	数値 単位	貸し方	数値 単位
銅切削加工仕掛品	1 個	銅材	8 Kg
銅くず	2 Kg		
		切削加工物的資本サービス	1 時間
		切削加工人的資本サービス	0.2 時間

#### (2) プロジェクトインスタンスの生成と資源割当のスケジューリング

プロジェクトタイプは、マイクロプロジェクトの設計図となるが、その設計に基づいて、具体的に製品やサービスを生産する実態(インスタンス)を定

めて遂行するには、製品やサービスをどれだけというスケジュールで生産するかを決める必要がある。その生産のスケジュールを立てる為には、原料の調達計画（MRP）と設備や人員（物的資本サービスと人的資本サービス）の割当計画を立てる必要がある。BOM（部品展開表）は、実物簿記で記述されたタスクから計算できる。ここから原料の調達計画（MRP）が可能となる。しかし無限生産能力がない以上、部品展開表に基づく原料調達には限界がある。無駄のない資材手配計画には、生産に必要な機械や人の生産スケジュールリングが不可避となる。各タスクに原料・物的資源・人的資源が割り当てられることで個々の製番（プロジェクトインスタンス）の実行スケジュールリングが立てられる。

部品の切削加工の事例で、部品を3つ（製番P1、P2、P3）製造すると想定しよう。これは部品1つの単品生産のマイクロプロジェクトのインスタンスが3つ生成されてそれをスケジュールして生産工程を実行することになる。むしろこれは、単品ではなくロット単位でのタスク設計とマイクロプロジェクト設計を行っているが良い。

この事例ではこの3つの加工業務のインスタンスを遂行するための、物的資本サービスの資源として切削加工装置、プレス装置、塗装装置が1台ずつ利用可能と仮定する。マイクロプロジェクトのスケジュールリングを実際に行うためには、さらにタスク遂行の優先順位を決めるためのディスパッチルールが必要となる。fig. 3.4では工数の長いタスクから優先して遂行する長さ優先、短いタスクを優先する短さ優先、さらにマイクロプロジェクトのインスタンス毎にまとめて実行することを優先するワークフロー優先の割当ルールで計算した例を示す。

fig. 3.4 3つのロットに対するスケジュールリング事例



このようなスケジュールリングは、参照スケジュールリングとして、工程計画を立てる上で有用に用いられることができる。更に実際のマイクロプロジェクトのインスタンスの遂行過程では、様々な理由で遅延が入ったり、特急品の割り込み等が生じる。その都度再スケジュールリングが必要となり、スケジュールリングアルゴリズムには、再スケジュールリングの機能も必要とされる。このスケジュールリングのアルゴリズムとそれを遂行するプログラムもリアルワールドOSの構成する機能部品となる。

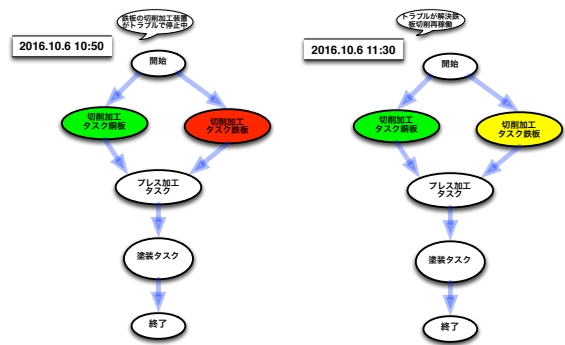
### (3) タスクの遂行のモニタリング

単品やロット（製番）単位での製造プロセスが並列に複数常時稼働している工場や、多くのサービスプロセスの並列的な遂行を行っている現場では、どのマイクロプロジェクトのどのインスタンス（プロセス）のどのタスクがいま着手中なのか、終了したのか、未着手なのか、或はトラブルが生じているか等を工場等の現場や管理部門が把握して情報共有することが必要となる。従来の多くの工場や現場ではこれがうまくいっていない。このようなタスクの遂行状態は、一部の全自動のタスクを除き、自動取得するというよりは、そのタスクの遂行や管理にあ

たっている現場の人間が、タスクの着手状況については認識し、適切に発信する必要がある。機械のトラブルなど一部自動的に認識すべきところもあるがタスクの着手は、例えば工作機械を作動させる前の段取り換えも含めて着手を認識し、或は加工後の何らかの検査を含めて終了を認識する必要がある。

プロジェクトインスタンスの遂行では、様々なトラブルで計画通りに行かないことが多々ある。稼働中の複数のプロジェクトインスタンスに対しその着手状況を見える化し、遠隔で把握することはIoTの利活用で簡単に実現でき、また複数の場所、国内と国外での進捗共有も容易となる。fig. 3.5は先の部品の加工事例での進捗状況管理を見える化する事例である。

fig. 3.5 タスク遂行のモニタリング例



### (4) タスク単位でのデータ収集

ものやサービスの生産と言うプロジェクトに関する様々なデータは、それを構成するタスク単位に認識され収集されなければ、ただの乱雑なデータの固まりにすぎない。そのためにはまず、一連の仕事の中でタスクをどう認識し切り出すかが鍵となる。

その上でタスクの単位でデータをブロック化して収集することが必要となる。データの収集は、センサー等で自動収集されるものもあれば、人が入力を必要とするものもある。工場の現場は既に様々な機械や計測装置が導入されている現場であり、それを前提としてのマイクロプロジェクトとしてのシステム設計、システムカイゼンが必要となる。このような既存のシステムが既に存在している業務現場はブラウンフィールドと呼ばれ、まっさらなシステムを導入する新規の現場（ブルーフィールド）と区別して扱う必要がある。実際の業務現場の大部分はブラウンフィールドであり、それゆえに工作機械や計測機械の状態把握も現状の装置を前提として工夫するところから出発する必要がある。いずれそれらの機械や装置が自動化されるとしても今度はプロトコルの問題が生じる。現在多くの工作機械メーカーが、他社に公開していない関わらず多くの小作機械メーカーは独自のデータ取得の規格を用いており、それらを更にブリッジすることは非現実的で無駄となる。他方で業界団体の主唱する規格も、各メーカーがそれを導入しない限り机上の空論となる。更に、業務現場に複雑なハンドシェイクを利活用する人材を新たに導入する費用対効果も課題とされねばならない。そもそもデータの取得は、多くの場合複雑なハンドシェイクのプロトコルや標準化は不要である。物理的なデータはその値そのものが物理的な規格であり、データ記述の規格はレコード形式かせいぜいJSON (JavaScript Object Notation) 程度の規格で十分となる。またデータ取得のためのハンドシェイクの規格は、データリクエストに対してシリアルなデータ出力をすることができれば十分である。この



ような規格にはバーコードリーダで用いられているHIDや、Modbusのようなインタフェース規格がある。これらの規格は工場の現場で幾つかの装置で既に用いられており取り回しもよい。この規格で取得したデータをパブリッシュすることで遠隔で容易に計測データを取得し共有することができる。

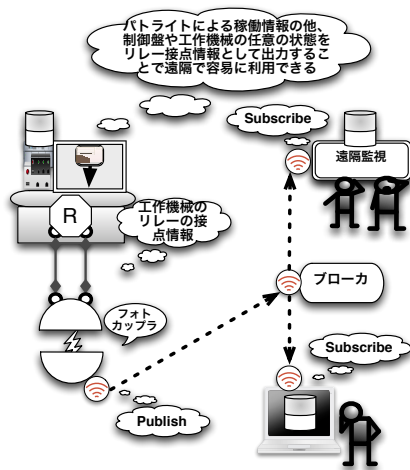
ここではデータ収集の事例を2種類示す。その具体的な利活用例は次節で述べる。

#### 1) 工作機械の状態把握

日本の工作機械では工作機械の状態把握のためにパトライトがほぼ標準で装備されており稼働状態や停止などの状態をランプで表示している。このパトライトの表示は、工作機械を制御するシーケンサの末端のリレーボードにより制御される。このリレーボードに工作機械の内部状態を表示することは、リレーボードの装着された工作機械や制御装置ではシーケンサのプログラムの設定で可能となる。むしろこのような工作機械の内部状態の把握は工作機械メーカー独自の制御システムを導入すれば、そのメーカーに限っては可能となるが、多くのメーカーの新旧取り混ざった工作機械の混在する工場のブラウンフィールドとしての現場で、機種を問わず、しかも工場内外、国内外で工作機械の稼働状態や内部状態を見える化共有し、さらにそのデータを収集しようとするとは容易ではない。

しかしリレーボードに転写された内部状態を取得しパブリッシュすることでこれが容易に実現される。このような目的のためには、通常24VのPLCの末端から、パブリッシュするための装置へと逆電流が流れない様な電気的な接続のないフォトカップラ接続の装置を用いれば良い。fig. 3.6はその装置とその利活用例を示している。

fig. 3.6 工作機械の稼働状態の取得装置とその利活用例



#### (5) 測定装置のデータ取得

様々な検査工程での検査データは、従来の工場ではそれぞれの工程で人が検査してその多くは紙の記述として管理されてきた。現在はパソコンへの取り込みも行われているが、プロジェクトのインスタンスとしての製番単位でのデータの管理はなかなか実現されていない。また依然として紙での記録も多く見受けられる。ブラウンフィールドとしての工程カイゼンの中で、計測データのタスク単位の取得とそれを製番単位でまとめて管理する仕組みを徐々に確立して行くことは工程カイゼンのためのデータ分析やトレーサビリティのためにも重要となる。

計測装置のデータをデジタルデータとして外部利活用するための規格は様々なものがあり、データをパブリッシュするためにはそれら様々な規格に基づいてデータを取得し、パブリッシュする必要がある。その中でも前述のHIDは、極めて単純だが有益な規格である。HIDはバーコードリーダで広く用いられているが、最近では大手の計測器メーカーがデジタルノギスなど多くの測定装置でHID規格のケーブルを用意する様になった。HIDはデータを送出する為の複雑なハンドシェイクは不要で、バーコードリーダの様に装置の側での読み取りの指令から自動的に読み取り結果のシリアルデータが送出される。この送出データをそのままパブリッシュする、マイクロサービスのモジュールを用意すれば、接続する装置に依存せずに簡単に計測データをパブリッシュすることができる。次節ではその具体的な事例を示す。

#### (6) データ解析

収集したデータを様々な角度から解析することは、タスクやマイクロプロジェクトのカイゼンためには必須となる。その為には、タスク単位、マイクロプロジェクトのインスタンス単位で、データを収集蓄積しそれを分析する必要がある。

同じプロジェクトタイプの複数の製番の同一タスクを串刺しして分析することで、タスク毎の品質に関するデータ分析ができる。また資源（例えば加工機や担当者）単位でタスクを串刺し分析する、加工機や担当者に依存した品質のばらつきに関する分析も可能となる。さらにプロジェクト単位のデータをアーカイブして、次項に述べる様にブロックチェーンでそれらの製造時データを企業の壁を越えてサプライチェーンを通じて管理することができる。これにより、製品に問題が出たときブロックチェーンを遡り、部品やサービスのデータをトレースバックすることが可能となる。

#### (7) トレーサビリティ

IoT時代には製品に問題があったとき、原因究明のためにサプライチェーンを遡り部品やサービス生産時の当該のプロジェクトインスタンスを構成するタスクの製造時の諸データにアクセスしそれを解析することが求められる。これは製品から部品に向けての遡りであり、トレーサビリティと呼ばれる。このトレーサビリティの確保は近年のものやサービスの生産では次第に重視されるようになってきている。その為には、サプライチェーンに従って、製造時のデータを管理し、必要になったときにそのデータを開示し分析することが求められる。その際に、データが製造時のデータそのまま中途で改竄されていない保証や、個々の部品企業等の製造時のデータは、事故等でトレーサビリティに従ったデータ開示と分析が部品製造企業と製品製造企業間の契約によって求められる状況以外には、開示されず秘密が守られることも求められる。

これらの条件を満たすトレーサビリティの確保をサプライチェーンに渡って可能とする一つの方法は、工程全般に渡り共通の規格を作り、システム化されたデータ収集の仕組みの中で、クラウド上でサプライチェーン上のデータを全て扱えるプラットフォームを構築し、製品企業と部品企業がサプライチェーン単位でクラウドに参画し製造時データの共有を行うことである。巨大なプラットフォーム型のシステムを介在する製造時のデータ収集とトレーサビリティ確保のシステムは、製造業の生産管理システムのクラウドプラットフォーム上で共通のトレーサビリティパッケージの利用で可能となるだろう。だがそれは規格化競争と巨大なプラットフォームへのロックインを招くリスクが高い。インダストリー4などの中で構想されている、クラウドベース

の生産管理からサプライチェーンを貫くトレーサビリティの確保のシステム化は、このようなクラウドベースの巨大なシステムとなる可能性が高い。

他方で、製造時の計測データ等の取得はブラウンフィールドとしての工場の現場では様々であり、それらが物理データとしてデジタル化されタスク単位で梱包されており、トレーサビリティが必要となった時点で、それが改竄されていないという保証のもとで利用可能であれば、トレーサビリティに必要なデータの種類は親企業との関係の中で次第にカイゼンされ高度化されればよい。

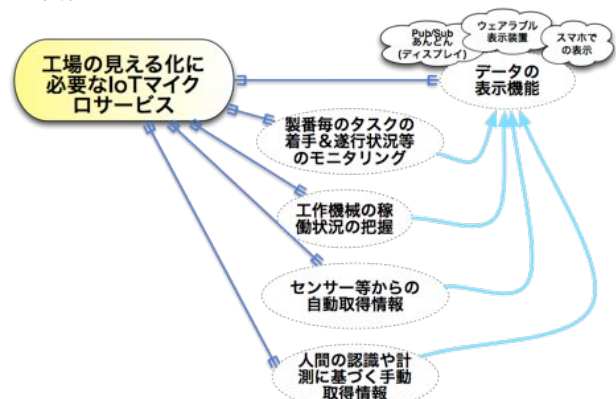
#### 4. IoTカイゼン会の活動ーボトムアップなマイクロプロジェクト型のシステム開発の実践とオープンイノベーションー

DESMDCATSによるマイクロプロジェクト設計・実装・運用・カイゼンのモデリングサイクルを、具体的なものづくりの現場で利活用しその成果を水平展開するために、「IoTカイゼン会」という名称での活動を行っている。このIoTカイゼン会活動はリDSEMDCATSモデリングフレームワークをサポートし、ノード管理やマイクロプロジェクトの遂行支援、データ収集、モニタリング、トレーサビリティ等の各種機能をサポートする、リアルワールドOSと我々が呼ぶ支援システムの普及とオープンイノベーションによる発展のためのアルワールドOSコンソーシアムというコンソーシアム活動の一環として行われている。

第一回のIoTカイゼン会は、2016年12月2日に下諏訪で開催され、ものづくり企業、コンサルタント、サービス企業、情報企業等25社以上、40人以上の参加があった。IoTカイゼン会では、DSEMDCATSのモデリングサイクルの中で、企業の工場でのIoT化に関する興味関心が高い領域から、粗結合のモジュールでサービス単位で容易にIoT化が実現できることを実践的な講習を通じて示して行くことを目指している。

従来の情報システムの開発実装を行うエスアイアー（SIer）ではなく、マイクロサービスの粗結合を設計・実装・運営する為の環境を設定する、IoTコンフィギュレータそのものを地域から発展させることもIoTカイゼン会の目的となる。

fig. 4.1 工場の見える化のためのマイクロサービスの諸機能



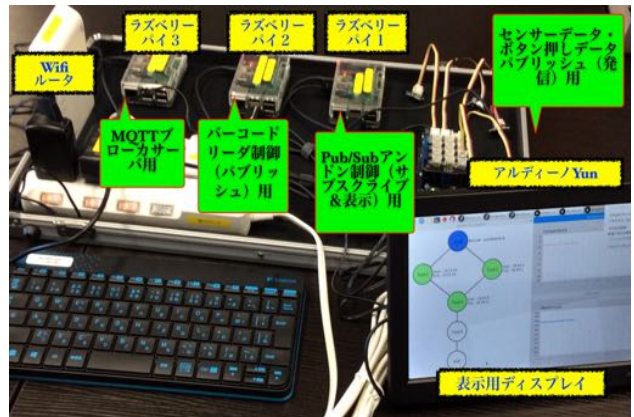
そのためにまず工場での物作りをマイクロプロジェクトとして把握し、それを設計・実装・運用・カイゼンの管理するためのマイクロサービスの中で、タスク単位の着手状況と工作機械の稼働状況のデータの収集と見える化の仕組みfig. 4.1のように設計し実現している。それらの機能を独立のモジュールとして実現しその粗結合で見える化やデータ収集の仕組みを、低コストで実現できるIoT開発キットを

用意しその講習会を行っている。以下では実際のシステムの事例を示す。

#### (1) IoT開発キットを構成するマイクロサービスモジュール

工程の見える化の為の基本的な組合せとして、ロット（製番）情報や担当者、タスク名などの情報をバーコードリーダーで入力し、タスクの開始と、タスクの終了、異常などの情報を入力することで、タスクの着手状況を任意の場所から見る事のできるシステムを想定する。このようなシステムをエッジ領域でモジュールの粗結合によって構築するためには、機能の切り分けが必要となる。本事例では、1) ブローカモジュール、2) HIDインタフェースによるバーコードリーダー入力&パブリッシュモジュール、3) 表示装置（以下Pub/Subアンドン）へのデータのサブスクライブと表示モジュール、4) センサーデータと手入力によるタスクの着手状況のパブリッシュモジュールという4つのモジュールとLAN環境からシステムは構成されている。モジュール1)～3)はラズベリーパイで実装され、4)はアルディーノで実装されている。講習会で用いた実際のキットの写真が、fig. 4.2である。

fig. 4.2 IoTカイゼン会の講習で用いられたIoT開発キット



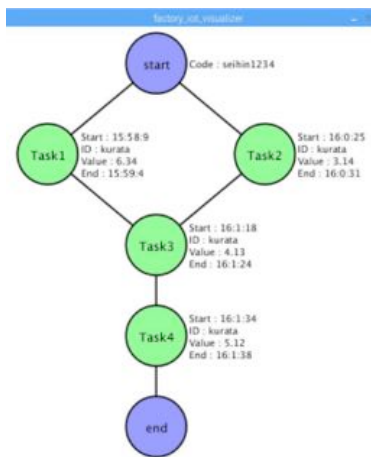
マイクロプロジェクトのインスタンス（製番）単位での製造過程での、各タスクの着手、終了、異常状況は、必要な場所でサブスクライブされ表示されることで、関係者間で情報の共有ができる。また同時に各タスクでの何らかの品質管理データや担当者名、着手時間、終了時間等の情報を取得することで製造時の基本的な品質管理のためのデータが得られる。fig. 4.3 はマイクロプロジェクトのインスタンスの遂行の見える化の事例と、タスクに紐づけられた管理情報の例を表している。

これは前節で論じた様に、工程カイゼンのための基本情報となると同時に、製造のアウトプットとしての部品や製品のトレーサビリティのための基本情報となる。

工作機械の稼働状況に関するデータ取得は機械メーカーがそれぞれ独自の規格を提唱しており、それらの規格はたとえオープン規格を謳っていても、他社が採用するとは限らない。また工場には様々な種類の工作機械が入っており、ブラウンフィールドとしての工場の中で、複数の異なる年代、異なるメーカーの機械からの、オープン稼働データなどの取得方法が必要とされる。工作機械メーカー主導のIoTは、加工機械という工場の製造工程の一部に関しての、IoT化であり我々が、主唱するマイクロプロジェクト全般に渡るIoT化とそれによる、製造工程の設計・実装・運用・カイゼンのサイクルを支援するものではない。またメーカー主導のデータ取得は、

データを機械メーカーに吸い上げられ、それを管理する一体型のクラウドシステムへの加入を要請される可能性さえある。

fig. 4.3 プロジェクト遂行状況の見える化とタスク単位の管理データ

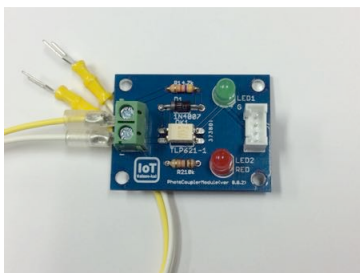


## (2) 工作機械の稼働状況の取得

我々は既に前節で、fig. 3.6で、様々な工作機器や制御板のPLCの末端からリレー制御盤を経由しての工作機械の稼働状況を取得しパブリッシュする枠組みを示した。この設計に従って、データを取得する為には、フォトカップラモジュールと、取得されたデータをパブリッシュするモジュールが必要となる。

我々は、IoTカイゼン会で独自のフォトカップラモジュールを開発し (fig. 4.4)、これにより取得されたデータをパブリッシュするボックスを制作した。

fig. 4.4 IoTカイゼン会ブランドのフォトカップラモジュール



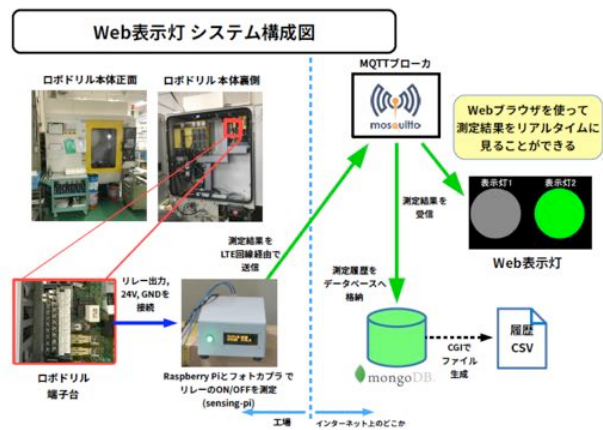
これにより、ブラウンフィールドとしての工場内の工作機械等、PLCによって制御される制御盤を持つ様々な危機の内部状態を取得する事が可能となる。

fig. 4.5はその工作機械の稼働状態をWeb上で遠隔表示する装置の装置 (Web表示灯) のシステム構成を示している。

このように、工作機械のPLCからの情報をリレー末端を介して取得する方法は、現在のPLC制御の全ての工作機械に後付け的に設置可能な方法で、コスト的にも数万の追加出費で簡単な稼働状態の取得なら可能となる。またそれに必要な技術も、従来の工場が持つPLC技術の延長にあり、簡単なIoTに関する技術を追加するだけで、複雑で独自性の強いプロトコルやプログラミングのスキルは不要である。むしろ取り出したい情報が4ビットであればリレー4

つが必要であるとか、当該の工作機械が標準でリレー出力可能な状態情報以外の内部状態をリレーに転写する為には、専用のPLCプログラムを書く必要があるなど制約はあるが、他方で組合せ自由で工場の実情に併せて、異機種を混合する形で何台でも、どこからでも工作機械の稼働状態等の情報を取得し、またそのログを取る事で保全のための情報も得る事ができる。

fig. 4.5 機械の稼働状況の遠隔表示装置のシステム構成



## 5. リアルワールドOSによるマイクロプロジェクト管理

フォグ領域の現場からのIoTを支援するという事は、ネットワーク上の様々な人・もの・ソフトウェアなどのノードが行うタスクとその結びつきを支援することである。そのためには、ネットワーク上のノードの設定や協調分散的な制御を支援する実世界のオペレーティングシステムが必要とされる。我々はこれを実世界OS (リアルワールドOS : RWOS) と呼んでいる。

IoTカイゼン会では、第一段階で、単純な管理モジュールの粗結合によるIoT化を進める。そおでは個々のタスクを遂行するノードからの情報は、必要とされるノード間でPublish/Subscribeのプロトコルによりブローカを経由して相互に結びつく。

他方でノードが正常に機能しているか等の状態把握やノードのブローカ接続等の設定支援、ノード間のタスクの自動実行などの管理や制御を行う仕組みはこの時点では導入されていない。DSEMDCATSで提供される様々な機能を、一つのフレームワークの上から、モジュール単位で容易に利活用でき、マイクロプロジェクトのインスタンス生成から、ノードの状態管理、プロジェクトの遂行管理等を一括して行う実世界のノードの管理機能が求められる。そのための仕組みをまとめたものが我々がリアルワールドOSと呼んでいるものとなる。

様々なひと・もの・ソフト等からなるマイクロサービスをタスクとして、そのタスクをPub/Subを仲介するブローカに接続したり、状態を把握する等を管理するためには、タスクを共通のAPIを持つコ

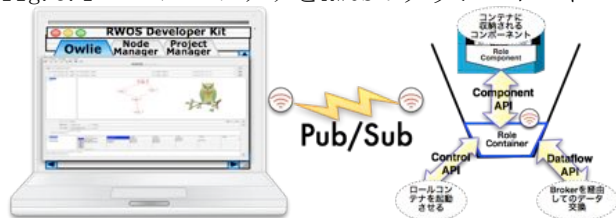
ンテナソフトウェアに収納し、それをダッシュボードから管理する仕組みが必要となる。そのために提供する枠組みがリアルワールドOSのロールコンテナとダッシュボード機能となる。さらにロールコンテナに収納されたタスクをPub/Subを用いて分散協調的に制御することで、マイクロプロジェクトのインスタンスを自動実行することも求められる。このようなマイクロプロジェクトとしてのタスクの実行管理のためのプログラミングを、我々はプロジェクトプログラミングと呼んでおり、その開発と実行のためのOWLIE(Onsite Workflow Language for Internet of Everything)というプログラム言語も提供している。

fig. 5.1はロールコンテナとダッシュボードの関係を示している。ダッシュボードからロールコンテナ経由で、共通の枠組みでタスク管理を行うためにはタスクを収納するコンテナが共通の3つのインターフェイスモデルに従う必要がある

- 1 管理コマンドインターフェイス
- 2 タスク間連結インターフェイス
- 3 ロール遂行インターフェイス

ロールコンテナは、ダッシュボードとやりとりするControl API（管理コマンドインターフェイス）と、ノード間でデータをやりとりするためのData Flow API（タスク間連結インターフェイス）と、実際にタスクの機能を遂行するモジュールがロールコンテナの機能を利用するためのComponent API（ロール遂行インターフェイス）を持つコンテナとなる。

fig. 5.1 ロールコンテナとRWOSのダッシュボード



工場のIoT化では、絶えざるカイゼンと後付的生産システムの変更が求められている。そのためには様々なタスクをロールコンテナ対応のマイクロサービスとして開拓することが求められる。コミュニティや建築現場等あらゆる領域でこのようなマイクロサービスが開拓されることで、様々な領域で超多様なサービスプロジェクトが作られて行くだろう。fig. 5.1はロールコンテナ対応の様々なマイクロサービスの事例を示したものである。マイクロサービスは必要に応じて、現場の工夫で、多様な中小の企業や個人が自己資金や小規模の補助金、クラウドファンディング等の小規模投資で開発することができ付け加えることができ、カイゼン、改良され様々な用途に多様化し進化することができる。

## 6. 結語

本稿ではIoT時代の実世界のマイクロサービスとその組合せとしてのマイクロプロジェクトがオープンなプラットフォーム環境で共進化するシナリオを描いてきた。IoT時代には、多様で分散型の新たな

ビジネスが、クラウドファンディング等の新たな金融スキームのもとで、沸き立つ様に生じてくる可能性がある。その可能性を現実のものとするためには、オープンプラットフォームのもとでロックインしない世界を構築する必要がある。我々のDSEMD-CATSはそのためのモデリングスキームの手案であり、実世界OSはオープンプラットフォームの提案となる。

## 参考文献

[CISCO, 2013]シスコ IoT インキュベーションラボ, 『Internet of Everythingの衝撃』、インプレスR&D、2013

[Deguchi, 2000] 出口弘, 『複雑系としての経済学：自律的エージェント集団の科学としての経済学を目指して』 日科技連, 2000

[Deguchi, 2004] Hiroshi Deguchi, Economics as an agent-based complex system: toward agent-based social systems sciences, Springer, 2004

[Deguchi, 2011] 出口弘, 市川学, 石塚康成, 志手一哉, 染谷俊介, 湯浅洋一, 並列プロジェクト・タスク処理への多能工割付けの動的スケジューリング, 国際P2M学会誌, 国際P2M学会, 2011 Oct, Vol.6, No.1, pp.179-189

[Deguchi, 2014A] 出口弘, POE(Point of Event)データとその利活用—IOE時代に向けての個人・企業・政府のデータの利活用のための三つの原則、システム／制御／情報, VOL.58, NO.7, PP.274-281, 2014

[Deguchi, 2014B] 出口弘, サービスチェーンと仕組みビジネス, 情報処理, pp.140-147, Vol.55, No.2, Feb, 2014

[Deguchi, 2015A] 出口弘, IOE時代のP2M支援環境としての実世界OS - Real World OS as an Infrastructure of P2M for IOE Era -, 国際P2M学会誌, Vol.9 No.2, pp.99-121, 2015

[Deguchi, 2015C] 出口弘, IoT時代のもの・サービスの生産支援システム- 代数的多元簿記に基づく自律分散協調型システムとして -, 経営情報学会 2015年度秋季大会予稿集

[Familiar, 2016] ボブ・ファミリア著, 株式会社クイープ翻訳, Microservices on Azure, 翔泳社, 2016

[Mattessich, 2007] Richard Mattessich, Two Hundred Years of Accounting Research- Routledge New Works in Accounting History, Routledge, 2007

[Mele, 2014] ニコメレ著, Nicco Mele (原著), 遠藤真美 (翻訳), 『ビッグの終焉: ラディカル・コネクティビティがもたらす未来社会』、2014、東洋経済新報社

[Newman, 2016] Sam Newman 著, 佐藤直生 監修, 木下哲也 翻訳, マイクロサービスアーキテクチャ, オライリージャパン, 2016

[pwc, 2014] technology forecast, Issue 1, 2014, プライスウォーターハウスクーパース株式会社

[Rifkin, 2015] ジェレミー・リフキン著, 柴田裕之訳, 『限界費用ゼロ社会—モノのインターネット—and 共有型経済の台頭』 NHK出版, 2015

謝辞：本研究はJSPS科研費課題番号「15H01719」の成果及び、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務の成果を含んでおります。ここに謝意を表します。

またIoTカイゼン会のコーディネータの（株）ロジカルワークス、（株）パイケーキ、（株）協和精工の皆様感謝致します。