



株式会社Empress Software Japan

**生野 雷人**

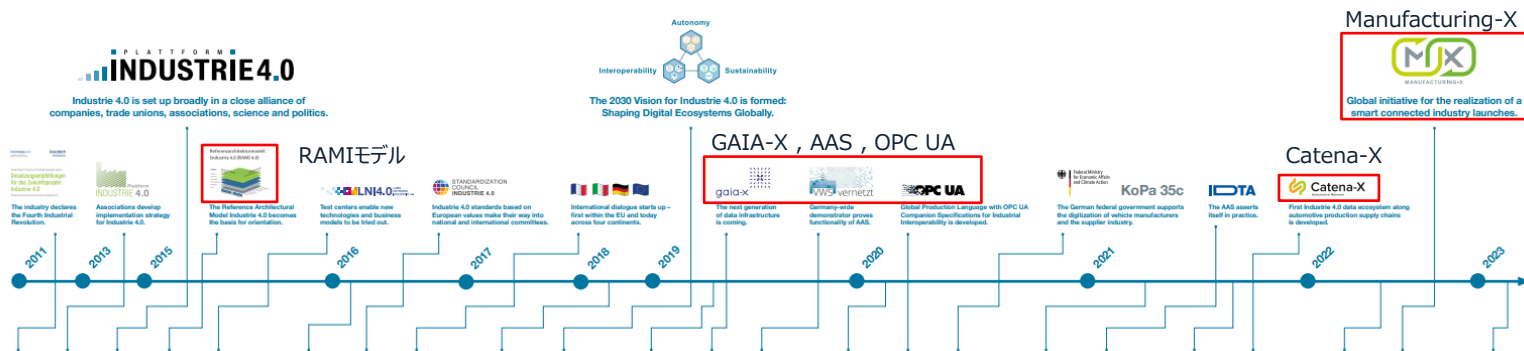
## ● Industry 4.0とは

- 製造業のコンピュータ化を促進するドイツ連邦政府のハイテク戦略のプロジェクト名に由来。
- ドイツ工学アカデミーと連邦教育学科省が2011年に「Industry 4.0」を発表。

## ● Industry 4.0には3つの指針（ミッションステートメント）が存在

- データ主権：設計と自由の自己決定により、デジタルビジネスモデルにおける競争力の保証
- 相互運用性：協力とオープンなエコシステムにより、多様性と柔軟性を両立
- 持続可能性：現代の産業の付加価値が高い生活水準の保証

## ● Platform Industry 4.0において、OPC UAやCatena-Xは以下のような時系列関係となっています。



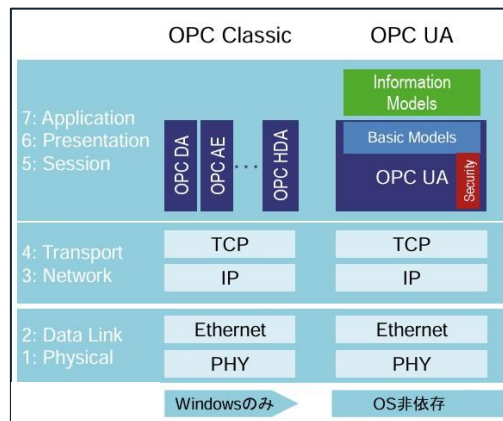
※出展：Zeitstrahl der Plattform Industrie 4.0

## ● OPC Classic の課題

- 前身のOPC Classicは、MicrosoftのCOM/DCOM技術に依存していました。これにより、Windows環境が前提となり、プラットフォームの制約や、増大するセキュリティ要件への対応が課題となりました。

## ● OPC UA (Unified Architecture) の誕生

- OPC Classicのコンセプトを継承しつつ、根本的な課題を解決。プラットフォーム独立性、堅牢な情報セキュリティ対策、拡張可能な情報モデリングの枠組みを設計段階から考慮した、新しいOPC仕様として策定されました。



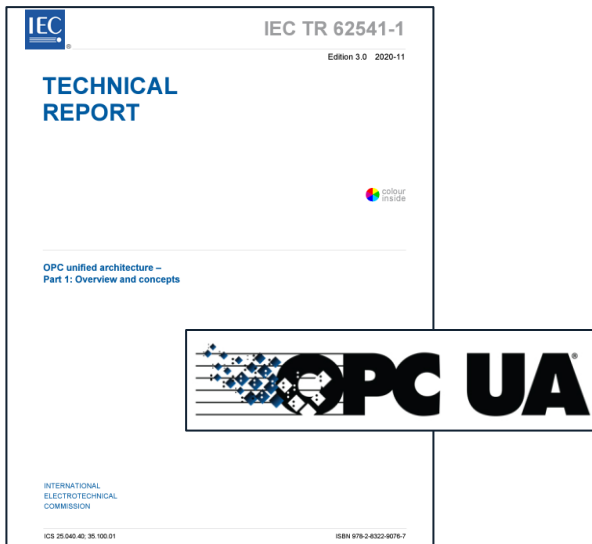
※出展：日本OPC協議会 OPC Day2020より

観点	OPC DA (Data Access)	OPC UA (Unified Architecture)
登場時期／位置付け	COM/DCOM ベースの“クラシック”OPC リアルタイムデータ転送に特化	DA/AE/HDA 等を統合した次世代仕様。
通信基盤	COM/DCOM (Windows 依存)	独自バイナリ/TCP、HTTPS(WebSocket) 等。OS 非依存。
データ表現	フラットなタグ＋値型は基本データ型中心	階層構造の情報モデルをネイティブサポート (Objects, Variables, Methods)
Type 定義／情報モデル	概念的に未定義。タグ名や外部ドキュメントで意味を補完するのみ基本的に再利用・継承不可	<b>ObjectType</b> 、 <b>VariableType</b> 、 <b>DataType</b> 、 <b>ReferenceType</b> などを持ち、継承・カスタム拡張が可能 インスタンス化によりモデルと実体の一貫性を保証
セキュリティ	COM/DCOM 依存 ファイアウォール越えが煩雑	証明書ベースの認証・署名・暗号化を標準装備
拡張性／相互運用	ベンダ固有拡張はCOMインターフェース追加が必要	Companion Information Modelにより業界共通モデルを標準化
運用環境	基本Windowsローカルネットワーク向け	組み込み〜クラウドまで同スタックで対応
通信機能	クライアントサーバのみ	クライアントサーバとPubSub (UDP/MQTT)

- OPC UA (Open Platform Communications Unified Architecture) は、工場や生産ラインにあるさまざまな機器やシステムを“つなげる”ための共通ルール・規格です。

## ポイント①

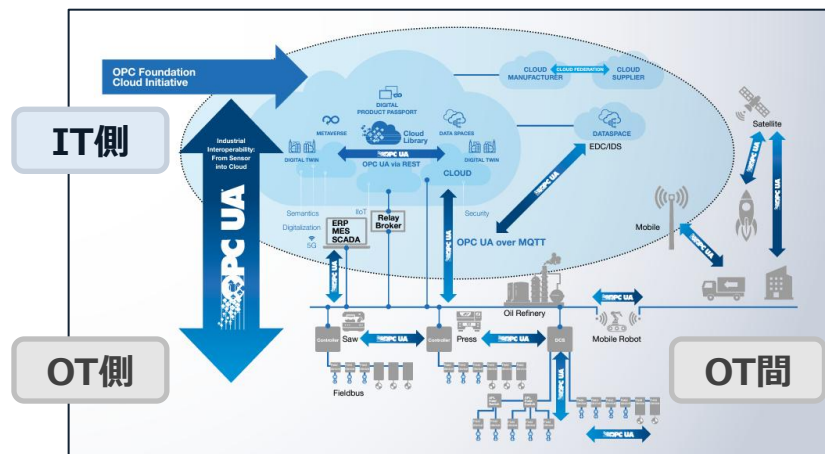
国際標準規格である (IEC62541)



※出展：IEC TR 62541-1 OPC unified architecture - Part 1: Overview and concepts

## ポイント②

機器同士間やクラウドへの共通言語として機能する



※出展：OPC Foundationより

## ● OPC UAは他の産業用ネットワークと共存

- 様々な産業用ネットワークはIT側へつながる手段を有していない。
- そのため、OPC Foundationに参画する動きが加速しています。
- OPC Foundationには、産業ネットワークだけではなく各業界団体も参画しています。
  - ✓ 例) EUROMAP(射出成型機)、VDMA(ロボットや工作機械)、OMAC(包装、パッケージ機械)などの業界団体。

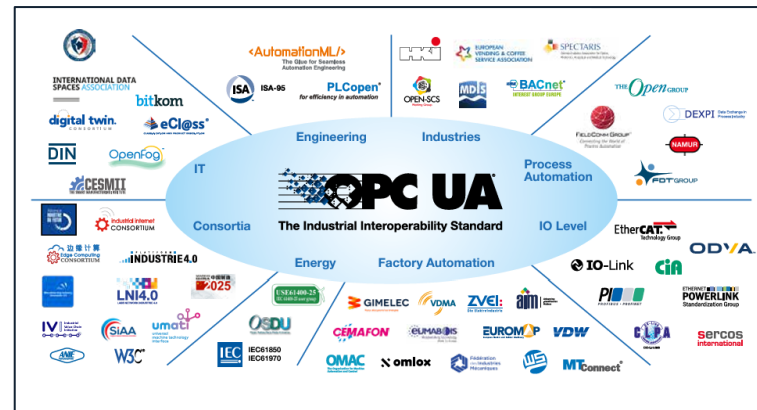
### ポイント③

#### 他の産業ネットワークとも共存

物理層	プロトコル	OPC UAの情報モデル（コンパニオンスペック）としての有無
産業用イーサネット	EtherCAT®	取り組み中
	PROFINET	あり
	EtherNet/IP™	取り組み中
	CC-Link IE	なし
	POWERLINK	あり
	SercosIII	あり
	Modbus/TCP	なし
	OPC UA	一
フィールドバス	DeviceNet™	なし
(シリアル通信)	Modbus/RTU	なし
	PROFIBUS	なし
	IO-Link	あり

### ポイント④

#### 機械団体も世界中からJOIN



※出展：OPC Foundationより

- OPC UAの基本コンセプトは「つなげる・伝える・安全に」です。

➤ つなげる「相互接続性」

- ✓ データを確実（堅牢性、信頼性）に伝達する

国際標準規格  
プラットフォーム非依存

➤ 伝える「相互運用性」

- ✓ 情報を正確（表現、解釈）に伝達する

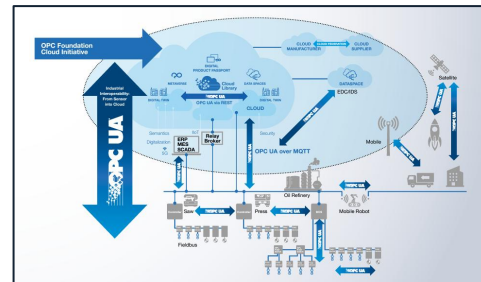
情報モデル

➤ 安全に「セキュリティ」

- ✓ データの情報の完全性及び、機密性を保証する

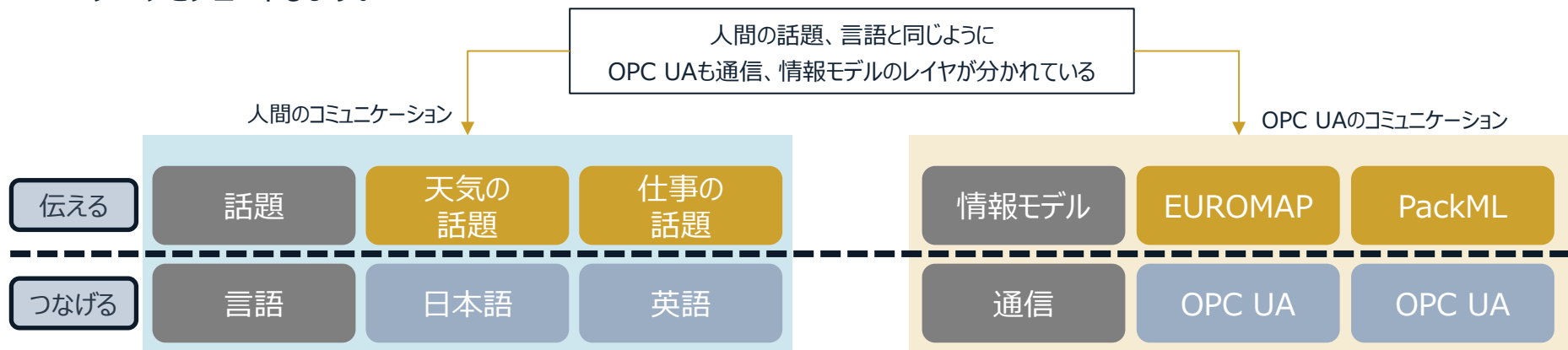
暗号化&証明書

データ活用



## ● OPC UAの基本的なデザイン

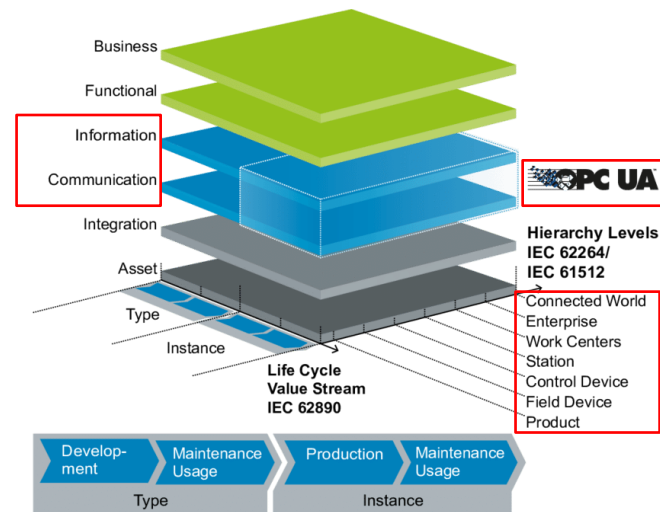
- OPC UAは基盤となるコンピューティングテクノロジーおよびネットワークトランスポートから分離するために階層化され、さまざまな階層でデータを公開するように設計されています。(=> 通信プロトコルと情報モデルは分離されています)
- OPC UAは統合セキュリティモデルを提供します。
- OPC UA Severは包括的な機能セットが用意されていますが、ユーザが独自のサブセットも実装可能です。
- OPC UA Serverはデータ、Alarm、EventをAddressSpaceで統合できます。
- 取得したデータのデータ形式について事前に知識を持たないで、実行時にSeverから構造体やデータ形式情報を取得しデータをデコードします。





## ● OPC UAの提供する範囲とRAMIモデル

- OPC UAはドイツが押し進めているIndustrie 4.0が推奨する産業コミュニケーションプロトコルです。
- OPC Unified Architectureの仕様は国際標準規格であるIEC 62541となっています。
- 情報モデルを構築できます。
- 高度なセキュリティを備えています。
  - ✓ SSL通信を使用します。
  - ✓ ServerとClient間の通信には証明書が必要です。
  - ✓ 暗号化はBasic256Sha256まで可能です。
- 右記はIndustry 4.0のRAMIモデルです。  
(Reference Architectural Model Industrie)
- OPC UAはCommunicationレイヤとInformationレイヤを跨ぎ、  
プロダクトから上位側のアプリケーションまでの利用が想定されています。



※図はOPC UA Foundationの資料より抜粋



## ● 他団体とのコラボレーション

- 標準化、つまり国際的な規格であるISO、IECなどを用いることでいつ、誰が行っても、同じ手順で無駄なく行えるかを示し、属人化をなくす。
- 製品やサービスをこれらの国際標準に合わせることは、取引の機会を増やし安全を保障する上でも重要な意味がある。



+

PLCopen



<AutomationML/>

Weihenstephan



EtherCAT

CC-Link IE



IO-Link

MTConnect

OPC UA上で扱える情報モデルとして順次策定が始まっている。  
OPC UA for PackML, OPC UA for IO-Link, OPC UA for CASなど

## ● OPC UAのセキュリティ仕組み

- アプリケーションで使用するSessionはSecure Channelに依存することで交換されるすべてのメッセージのセキュリティを保証します。
- アプリケーションはSessionが確立するときに1回だけ認証を受けます。
- X.509証明書はClientとServerを識別するために使用されます。
- Serverはユーザを認証しServer内のObjectにアクセスするためのリクエストを承認します。

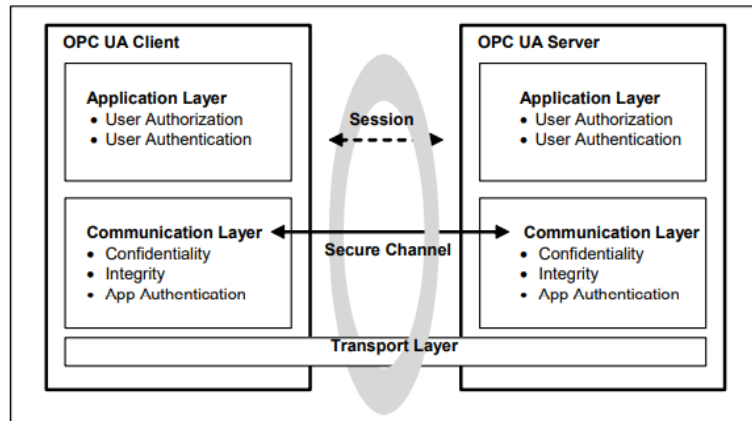


Figure 2 – OPC UA security architecture – Client / Server

※図はOPC UA Foundationの資料より抜粋

## ● OPC UAのセキュリティモード、セキュリティポリシー、ユーザ認証、アプリケーション認証

**セキュリティモード：メッセージの暗号化の指定です。**

	None	セキュリティなし
	Sign	メッセージに署名は付けるが、暗号化はしない
	SignAndEncrypt	メッセージに署名を付け、暗号化もする

**セキュリティポリシー：暗号化アルゴリズムを定義する以下のProfileを指定します。**

	None	
	Basic128Rsa15	
	Basic256	
	Basic256Sha256	
	Aes128_Sha256_RsaOaep	
	Aes256_Sha256_RsaPss	

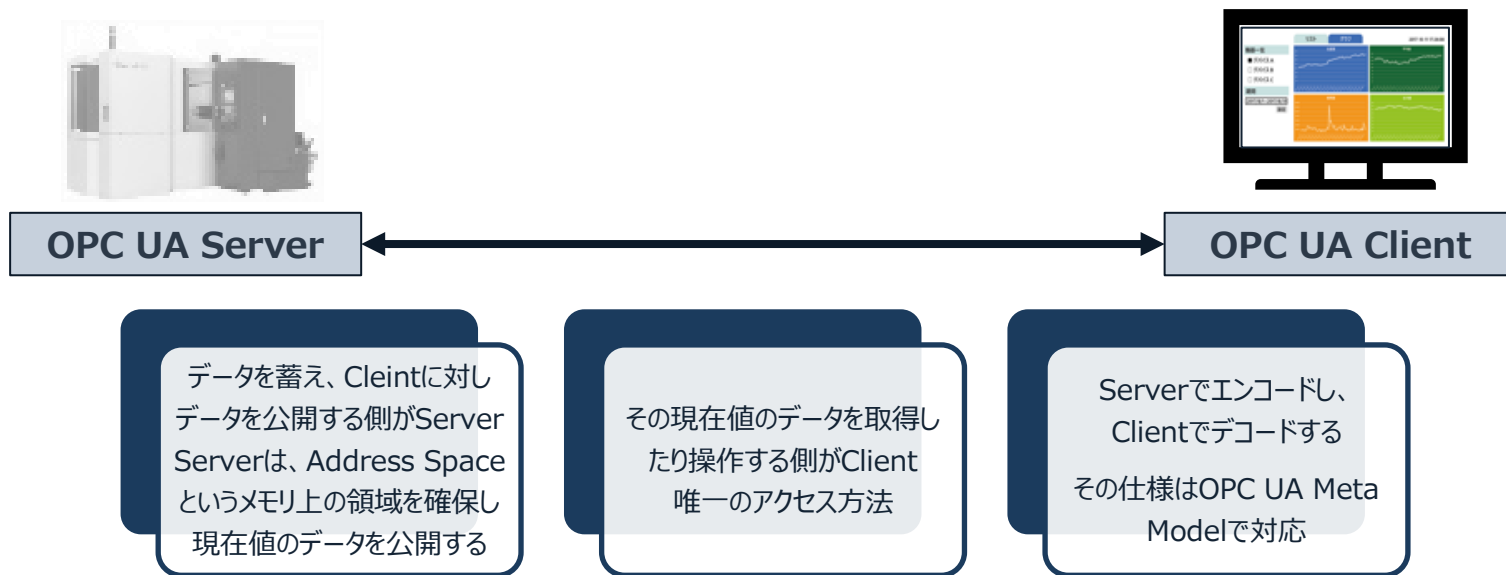
**ユーザ認証：ユーザーの資格情報をクライアントがセッションサービスを用いサーバーに渡します。サーバーはこの資格情報を使用しユーザーを認証します。  
サーバーはどのユーザーがどのデータにアクセスでき、どの操作を許可するかを決定します。**

	Anonymous	
	UserName/Password	

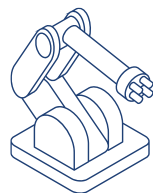
**アプリケーション認証：アプリケーション認証は、SSLが使用するX.509証明書を使用します。**

## ● OPC UAのServer・Client通信

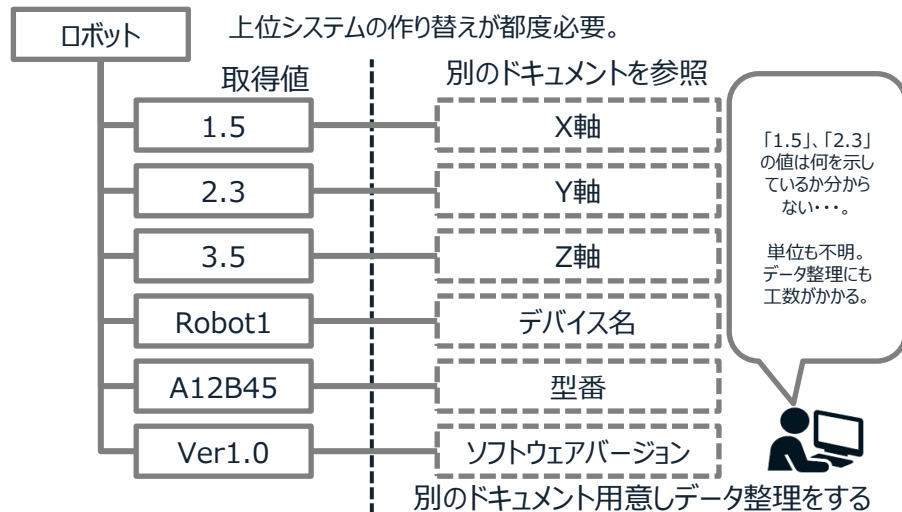
- OPC UAはデータを保有している機器・機械側がOPC UA Serverとなります。
- データを取得・操作する上位側がOPC UA Clientとなります。
- OPC UAは基本的にカレントデータ(現在値)をメモリ上に展開する仕様となっています（履歴データは後述します）。



## メタ情報や構造体なし

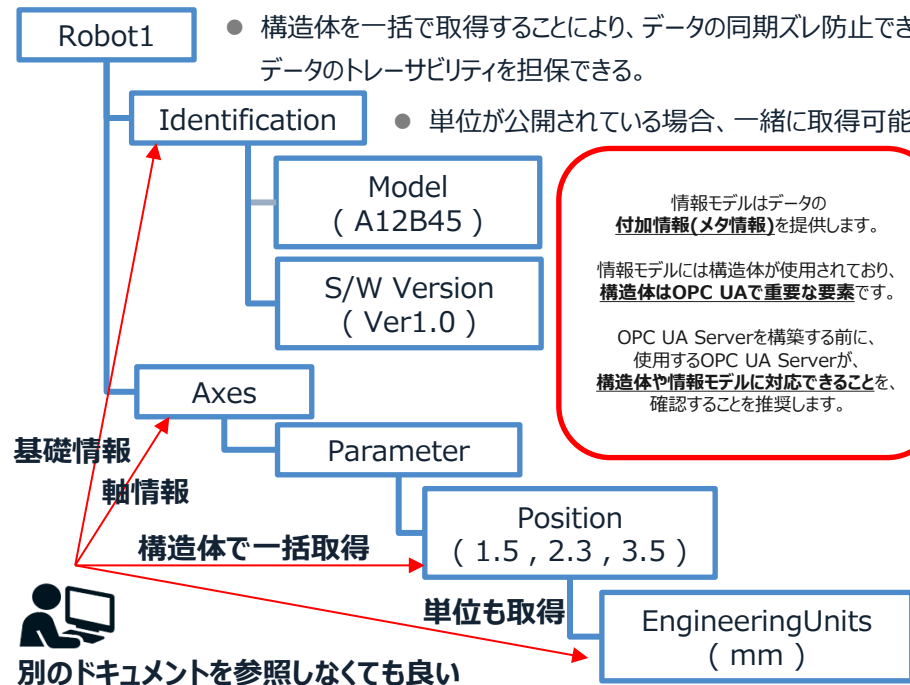


- 取得したいデータのアドレスや意味を別ドキュメントで調べる必要がある。
- ロボット側が稼働しているとき、X軸、Y軸、Z軸のデータを都度取得していると、ズレが発生してしまうため考慮が必要。
- ロボットメカにより、データの保管場所アドレスが異なるため、上位システムの作り替えが都度必要。



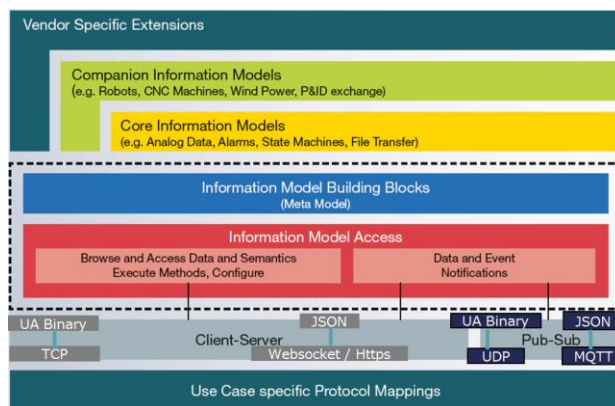
## メタ情報や構造体あり

- OPC UA Serverに接続すると、公開されているデータが見れる。
- 構造体を一括で取得することにより、データの同期ズレ防止でき、データのトレーサビリティを担保できる。
- 単位が公開されている場合、一緒に取得可能。



## ● 情報モデルの仕組み

- XMLフォーマットとして表現されます。
- 基本的に4つのレイヤーで構築されています。それ以上になる場合もあります。
- 3層目は一番重要なレイヤーで業界標準として使用し**Companion Information Model**と呼ばれます。
- ドイツ機械工業連盟、ドイツ自動車工業会など権威ある団体が作成します。
- 4層目はベンダが独自に作成するレイヤーで**Vendor Specific Extensions**と呼ばれます。



- 4層目 ユーザー特有の機能を記述する箇所、**競争領域**  
インスタンス化する箇所
- 3層目 Type定義しかない業界毎に決められた、**協調領域**
- 2層目 } OPC Foundationが作成し、インターネット上で公開  
どのOPC UAのベンダーも、ユーザーも同じものを使用
- 1層目 }

: Meta Modelと言われる箇所

・情報を記述する**規則**

・意味・構造、状態、ふるまい、メッセージ、情報と情報の関係を定義

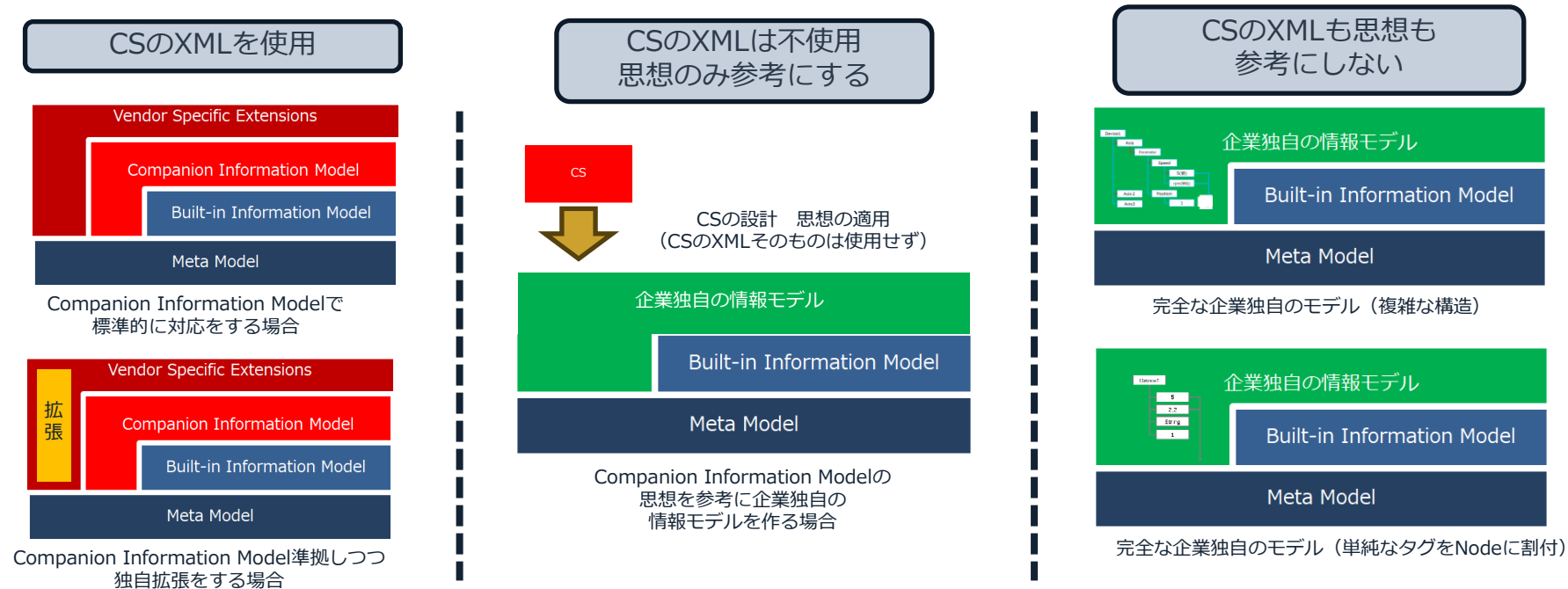


情報を提供する側も利用する側も、情報の中身を解釈できる

出典：日本OPC協議会

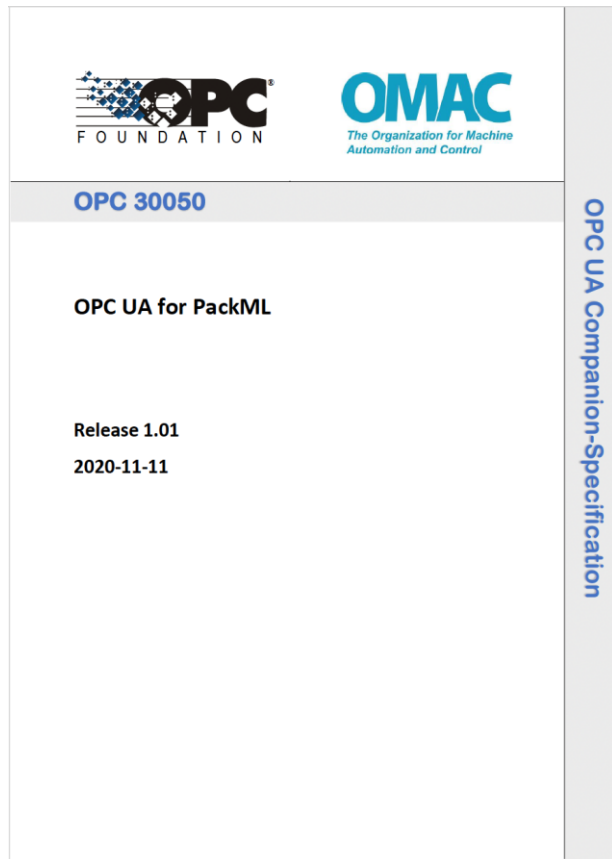
- 業界標準のCompanion Information Modelが無い場合（独自情報モデル）

- 業界標準となるCompanion Information Modelがない場合やCompanion Information Modelが製品と合わない場合などに独自に情報モデルを作成する必要があります。





# OPC UA for PackMLの概要



- PackMLの正式名称は「OPC UA for PackML」です。
- 規格番号は「OPC 30050」が割り当てられています。
- 最新版は Release 1.01（2020-11-11）です。
- 本ドキュメントはこのバージョンに準拠して作成しています。

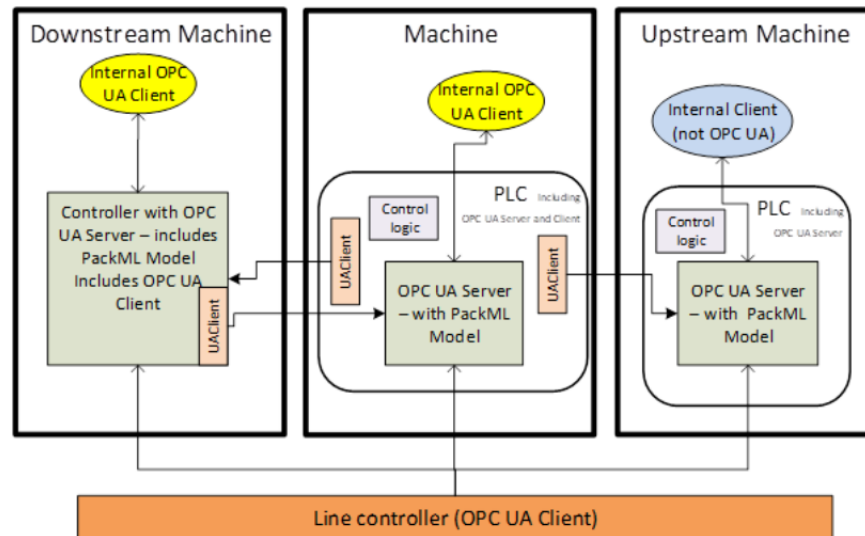
- 元々のPackMLの目的

- PackMLは「Packaging Machine Language」の略称です。
- もともとは、包装業界のバッチ製造（1つの品種をまとめて生産する方式）で用いられていたインターフェース標準です。
- PackMLの主な目的は、生産ラインを構成するすべての機械操作に一貫性を持たせることです。
- その目的は以下のとおりです。

1	標準で定義されたマシン状態と運用フローの提供
2	Overall Equipment Effectiveness (設備総合効率) データの提供
3	Root Cause Analysis (根本原因解析) データの提供
4	柔軟なレシピスキームと一般的なSCADAまたはMES入力の提供

## ● PackMLのスコープ

- 典型的な環境におけるPackMLの適用範囲を示しています。各ユニットはOPC UAサーバとして動作し、ラインコントローラはOPC UAクライアントアプリケーションとして動作します。PackMLは、ユニット／機械との入出力のための標準インターフェース群を定義し、ユニットを生産ライン全体の一要素として制御できるようにします。ユニット内部の状態は標準の状態モデルに、内部コマンドは標準コマンド群にマッピングされ、ユニット固有の実装詳細は隠蔽されます。
- また、ライン全体の状態やモードを調整するために、ユニット同士がピアツーピア通信を行う典型例も示しています。この場合、各ユニットは自ユニットの状態を公開するOPC UAサーバであると同時に、上流・下流のユニットと通信するOPC UAクライアントとしても振る舞うことができます。



- MachineとアップストリームMachineはPackMLのモデルが搭載されます。
- ダウンストリームMachineは独自モデルにPackMLモデルが追加されます。
- RemoteCommandを実装するため、Machine、アップストリームMachine、ダウンストリームMachineにはOPC UA Clientが搭載されます。

## ● AddressSpaceのインスタンス

- OPC UA ServerはOPC UA ServerのOPC定義オブジェクトフォルダーの下にPackMLObjectsフォルダーを持っている必要があります。
- このフォルダはPackMLBaseObjectTypeの1つ以上のインスタンスが含まれている必要があります。
- 単一のOPC UA Serverは、PackMLシステムの単一のインスタンスが含まれる場合もあれば、複数のPackMLシステムが含まれる場合もあります。

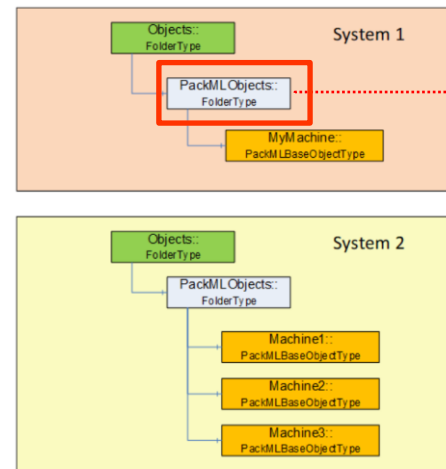
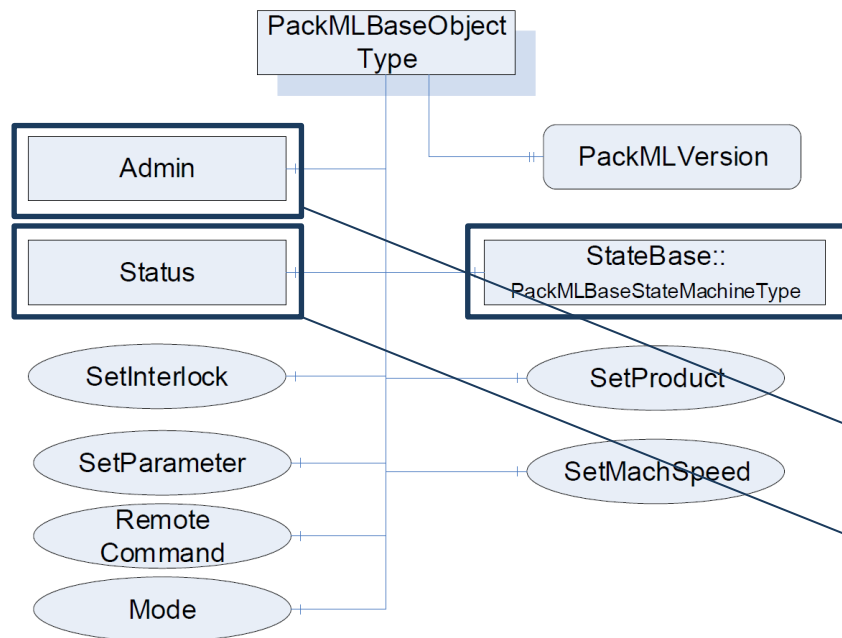


Figure 10 - PackML Object Instance Overview

Table 2 – PackMLObjects definition

Attribute	Value		
BrowseName	PackMLObjects		
References	NodeClass	BrowseName	TypeDefinition
OrganizedBy by the Objects Folder defined in OPC 10000-5			
HasTypeDefinition	ObjectType	FolderType	



- PackMLBaseObjectTypeは任意のマシンで使用できる基本タイプを定義します。
- この基本タイプは動作するPackMLシステムに必要なすべての情報を提供します。

機械の状態を示す  
どの状態に居られるか/次に行けるかを定義

機械の管理情報の定義  
履歴/集計/停止理由の台帳

運転中の健全度・可用性の要約  
何を要求中か/どのモードか/何を扱っているか

**Table 3 – PackMLBaseObjectType Definition**

Attribute	Value					
BrowseName	PackMLBaseObjectType					
IsAbstract	False					
Reference	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule	RW
Subtype of the <i>BaseObjectType</i> from OPC 10000-5.						
HasProperty	Variable	TagID	String	PropertyType	Optional	
HasProperty	Variable	PackMLVersion	String	PropertyType	Optional	
HasComponent	Object	Admin		PackMLAdminObjectType	Mandatory	
HasComponent	Object	Status		PackMLStatusObjectType	Mandatory	
HasComponent	Object	BaseStateMachine		PackMLBaseStateMachine Type	Mandatory	
HasComponent	Method	SetUnitMode	Defined in section 6.7.2		Mandatory	
HasComponent	Method	SetMachSpeed	Defined in section 6.7.3		Mandatory	
HasComponent	Method	SetProduct	Defined in section 6.7.4		Mandatory	
HasComponent	Method	SetParameter	Defined in section 6.7.17		Mandatory	
HasComponent	Method	RemoteCommand	Defined in section 6.7.15		Optional	
HasComponent	Method	SetInterlock	Defined in section 6.7.16		Optional	



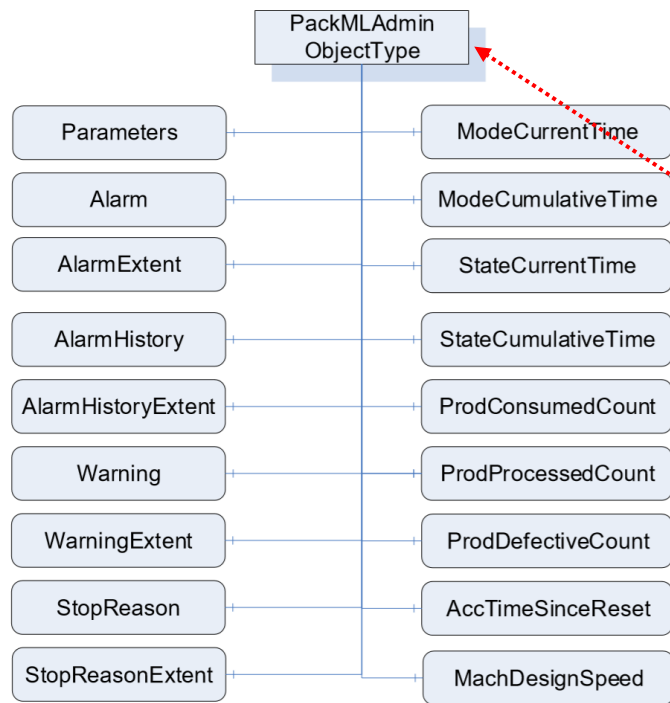


Figure 13 - PackMLAdminObjectType Overview

- PackMLAdminObjectTypeは PackMLBaseObjectTypeの構成要素です。
- PackMLAdminObjectTypeは、PackML情報モデルのマシンのすべての管理情報をグループ化するために使用されるObjectTypeを定義します。

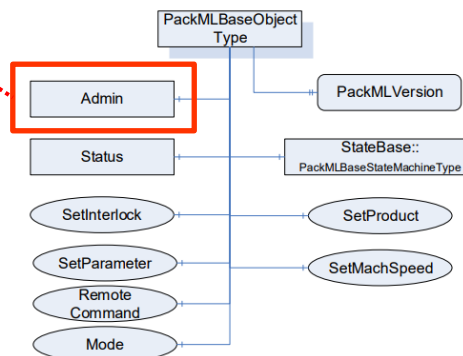


Figure 11 - PackMLBaseObjectType Overview

Table 5 – PackMLAdminObjectType Definition

Attribute	Value					
BrowseName	PackMLAdminObjectType					
IsAbstract	False					
Reference	Node Class	BrowseName	Data Type	Type Definition	Modeling Rule	R W
Subtype of the <i>BaseObjectType</i> from OPC 10000-5.						
HasComponent	Variable	Parameters	PackMLDescriptorDataType[]	BaseDataVariableType	Optional	R
HasAlarm	Variable	<Alarm>	PackMLAlarmDataType	PackMLAlarmVariableType	Optional Placeholder	R
HasComponent	Variable	AlarmExtent	Int32	BaseDataVariableType	Optional	R
HasAlarmHistory	Variable	<AlarmHistory>	PackMLAlarmDataType	PackMLAlarmVariableType	Optional Placeholder	R
HasComponent	Variable	AlarmHistoryExtent	Int32	BaseDataVariableType	Optional	R
HasWarning	Variable	<Warning>	PackMLAlarmDataType	PackMLAlarmVariableType	Optional Placeholder	R
HasComponent	Variable	WarningExtent	Int32	BaseDataVariableType	Optional	R
HasStopReason	Variable	<StopReason>	PackMLAlarmDataType	PackMLAlarmVariableType	Optional Placeholder	R
HasComponent	Variable	StopReasonExtent	Int32	BaseDataVariableType	Optional	R
HasComponent	Variable	ModeCurrentTime	Int32[]	BaseDataVariableType	Optional	R
HasComponent	Variable	ModeCumulativeTime	Int32[]	BaseDataVariableType	Optional	R
HasComponent	Variable	StateCurrentTime	Int32[]	BaseDataVariableType	Optional	R
HasComponent	Variable	StateCumulativeTime	Int32[]	BaseDataVariableType	Optional	R
HasComponent	Variable	ProdConsumedCount	PackMLCountDataType	BaseDataVariableType	Optional	R
HasComponent	Variable	ProdProcessedCount	PackMLCountDataType	BaseDataVariableType	Optional	R
HasComponent	Variable	ProdDefectiveCount	PackMLCountDataType	BaseDataVariableType	Optional	R
HasComponent	Variable	AccTimeSinceReset	Int32	BaseDataVariableType	Optional	R
HasComponent	Variable	MachDesignSpeed	Float	BaseDataVariableType	Optional	R

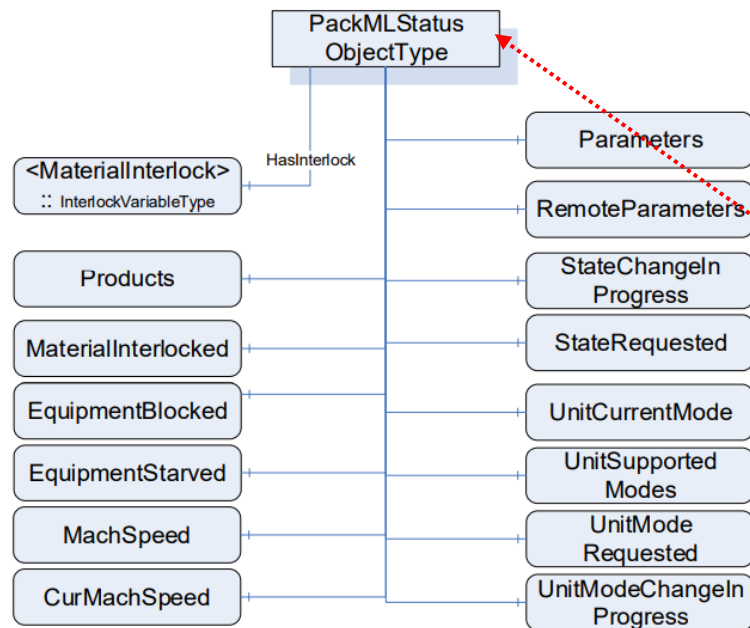


Figure 12 - PackMLStatusObjectType Overview

- PackMLStatusObjectTypeは PackMLBaseObjectTypeの構成要素です。
- PackMLStatusObjectTypeは、PackMLのすべてのステータス情報をグループ化するために使用されるObjectTypeを定義します。

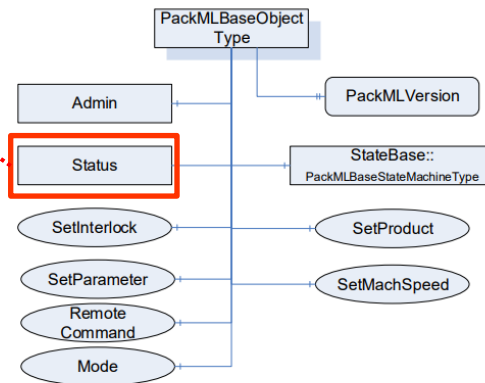


Figure 11 - PackMLBaseObjectType Overview

**Table 4 – PackMLStatusObjectType Definition**

Attribute	Value					
BrowseName	PackMLStatusObjectType					
IsAbstract	False					
Reference	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule	R W
Subtype of the <i>BaseObjectType</i> from OPC 10000-5.						
HasComponent	Variable	UnitModeRequested	Boolean	BaseDataVariableType	Optional	R
HasProperty	Variable	UnitSupportedModes	NodeId	PropertyType	Mandatory	R
HasComponent	Variable	UnitCurrentMode	Enumeration	PropertyType	Mandatory	R
HasComponent	Variable	UnitModeChangeInProgress	Boolean	BaseDataVariableType	Optional	R
HasComponent	Variable	StateRequested	Int32	BaseDataVariableType	Optional	R
HasComponent	Variable	StateChangeInProgress	Boolean	BaseDataVariableType	Optional	R
HasComponent	Variable	MachSpeed	Float	AnalogItem	Mandatory	R
HasComponent	Variable	CurMachSpeed	Float	AnalogItem	Mandatory	R
HasComponent	Variable	EquipmentBlocked	Boolean	BaseDataVariableType	Mandatory	R
HasComponent	Variable	EquipmentStarved	Boolean	BaseDataVariableType	Mandatory	R
HasComponent	Variable	MaterialInterlocked	Boolean	BaseDataVariableType	Optional	R
HasInterlock	Variable	<MaterialInterlock>	Boolean	InterlockVariableType	OptionalPlaceholder	R
HasComponent	Variable	Parameters	PackMLDescriptorDataType[]	BasedDataVariableType	Optional	R
HasComponent	Variable	RemoteParameters	PackMLDescriptorDataType[]	BasedDataVariableType	Optional	R
HasComponent	Variable	Products	PackMLProductDataType[]	BaseDataVariableType	Optional	R

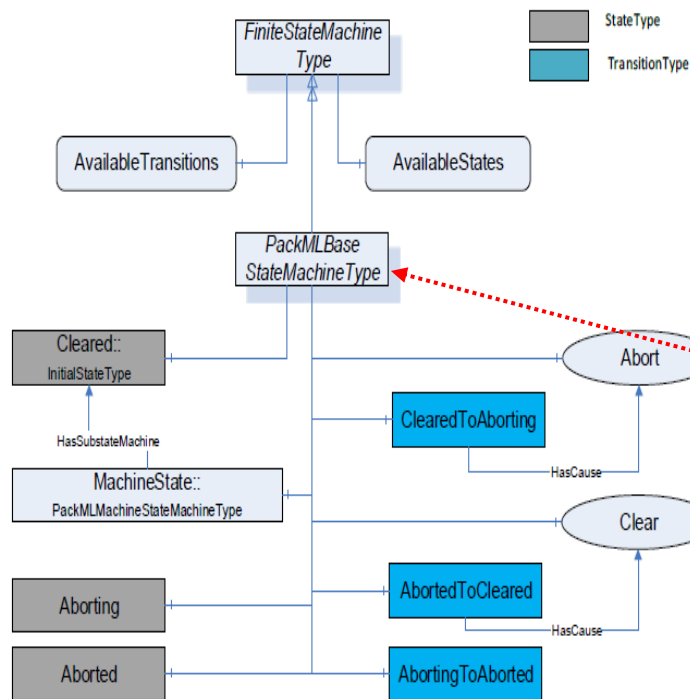


Figure 16 - PackMLBaseStateMachineType illustration

- PackMLBaseStateMachineTypeはPackMLの最上位のStateMachineです。
- 通常、StateMachineはAborted StateまたはCleared Stateを初期状態として使用します
- 初期状態をAborted StateにするかCleared Stateかは独自に定義します。

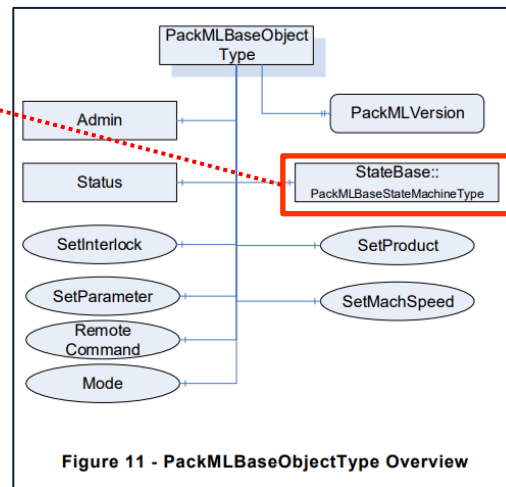


Figure 11 - PackMLBaseObjectType Overview

**Table 6 – PackMLBaseStateMachineType Definition**

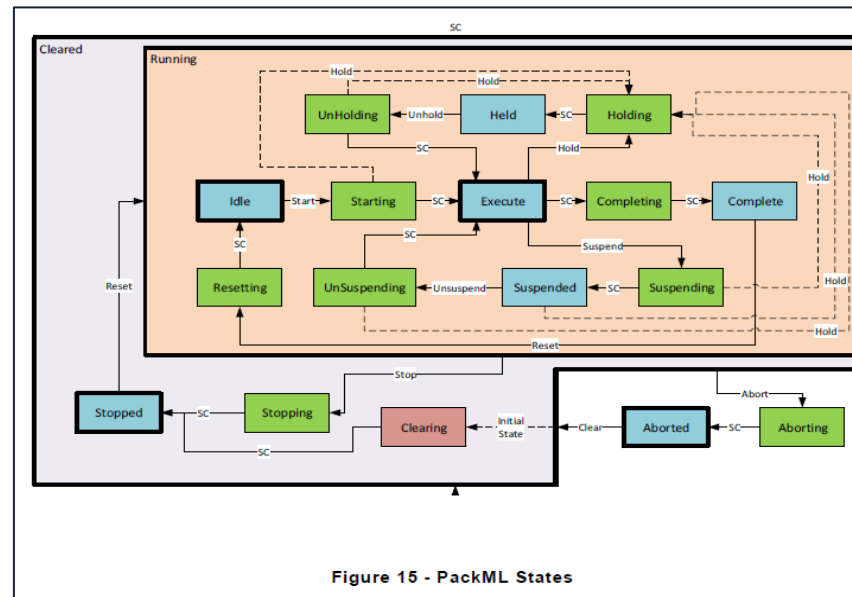
Attribute	Value				
BrowseName	PackMLBaseStateMachineType				
IsAbstract	False				
References	Node Class	BrowseName	Data Type	TypeDefinition	Modelling Rule
Subtype of the <i>FiniteStateMachineType</i> defined in OPC 10000-5					
HasComponent	Variable	0:AvailableTransitions	NodeId[]	BaseDataVariableType	Mandatory
HasComponent	Variable	0:AvailableStates	NodeId[]	BaseDataVariableType	Mandatory
HasComponent	Object	Aborting		StateType	
HasComponent	Object	Aborted		StateType	
HasComponent	Object	Cleared		InitialStateType	
HasComponent	Object	MachineState		PackMLMachineStateMachineType	Mandatory
HasComponent	Object	AbortedToCleared		TransitionType	
HasComponent	Object	AbortingToAborted		TransitionType	
HasComponent	Object	ClearedToAborting		TransitionType	
HasComponent	Method	Abort	Defined in 6.7.5		Optional
HasComponent	Method	Clear	Defined in 6.7.6		Optional

## ● PackMLのStateMachine要点

- すべてのStateMachineは、各インスタンスで AvailableTransitions / AvailableStates を必ず公開する。
  - ✓ クライアントは「今使える状態／遷移」を把握できる。
  - ✓ 利用可能な状態・遷移は、状況に応じて制限される。

## ● 初期状態の考え方

- Cleared の初期サブ状態は通常 Stopped。
- Running は多くの場合 Resetting から始まるが例外あり。
- 本モデルの推奨初期状態：Idle または Resetting。
- システム全体の初期状態は Aborted。
- 破線の Hold 遷移は任意の拡張。



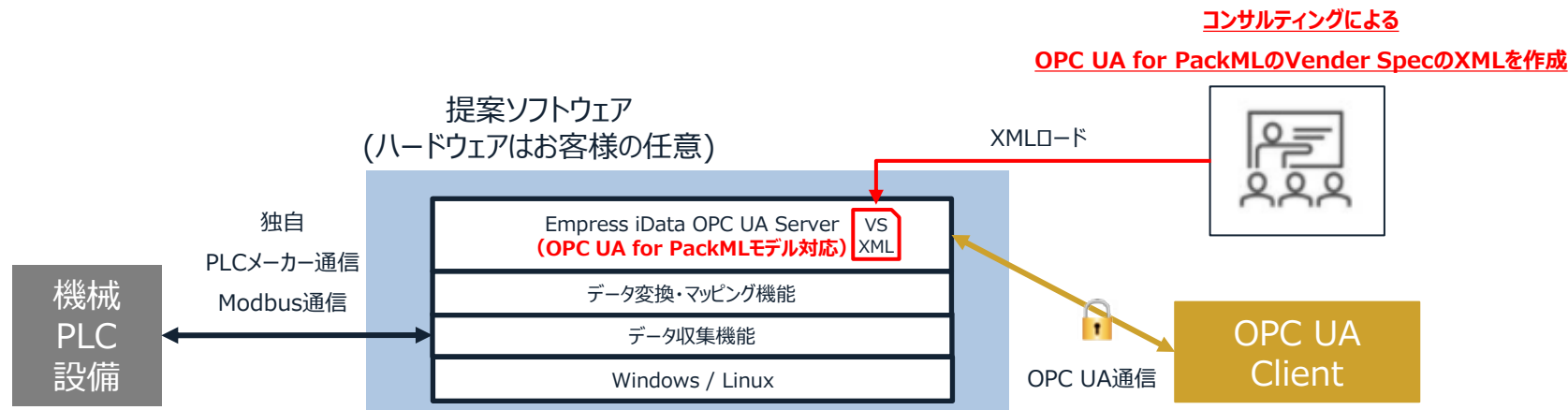


# OPC UA for PackMLの採用事例とアプローチ

## ● 機械メーカ様のOPC UA for PackML採用事例

### ➤ GWを設けたOPC UA for PackMLの対応

- ✓ 機械とOPC UA for PackMLのデータマッピングを行い、対応したVender SpecのXML作成を行います。
- ✓ Vender SpecのXMLをEmpress iData OPC UA Serverにロードさせることで、OPC UA for PackMLが表現されます。
- ✓ 機械からのデータ収集機能を開発し、収集したデータをEmpress iData OPC UA ServerのAPIでNodeに連携させます。
- ✓ これにより、モデル調査や検討の工数が大幅に削減でき、機械を容易にOPC UA for PackMLに対応させることができます。



## ● Empress iData OPC UA Server SDK

Linux, Windows, VxWorksなど様々なOSに対応  
パワフルで自由な開発が可能なSDK

あらゆる情報モデルに対応  
PackML, LADS, Euromap, MachineTools, Visionなど

HDA機能搭載対応  
(設定ファイルの一修正のみで設定完了)

マルチクラウド対応 OPC UA for PubSub機能対応

認証取得済み完全国内自社製  
日本スタッフによる技術サポートおよびコンサルティングも充実



OPC UAバージョン	OPC UA Version 1.04
OPC UAトランスポート	OPC UA TCP Transport, UA Binary Encoding, UA Secure Conversation
OPC UAセキュリティ	セキュリティポリシー：Basic128Rsa15, Basic256, Basic256Sha256, AES128_Sha256_RsaOaep, AES256_Sha256_RsaPss 認証方式：ユーザ名/パスワード認証方式もしくはユーザ名/ユーザ証明書方式
ロード済情報モデル	モデル定義基盤 (Meta Model) 基本モデル構成要素 (Built-in Information Model)
ヒストリカルデータ用DB	Empress暗号化組込みデータベース
対応Profile	Embedded 2017 UA Server Profile
対応Profile (Add)	Publisher UDP UADP Periodic Fixed Profile Profile PubSub MQTT UADP Profile PubSub MQTT JSON
対応Facet	EmbeddedUA2017 ComplexType 2017 Server Facet Method Server Facet Attribute WriteMask Server Facet A&C Base Condition Server Facet
対応Facet (Add)	Historical Raw Data Server Facet Historical Data AtTime Server Facet Historical Access Modified Data Server Facet Historical Access Structured Data Server Facet Historical Data Update Server Facet Historical Data Insert Server Facet Historical Data Replace Server Facet Historical Data Delete Server Facet Base Historical Event Server Facet

## ● Empress iData Modeler

- Empress iData Modelerは弊社で開発した国内初OPC UA専用 情報モデリングアプリケーションです。
- OPC UA Server開発の際、ベンダーが独自に作成するVendor Specific Extensionsのxmlの領域で、Companion Specで定義されたTypeをインスタンス化をするシステムを構築に絶対に必要なアプリケーションです。

表示される用語はOPC UA用語のみを使用。アプリケーション専用の用語を勉強/理解する必要はなし

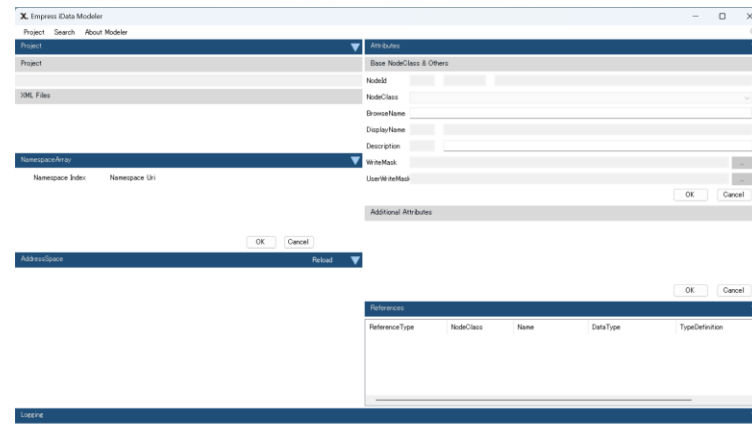
プログラムコードを自動生成するなどの付随機能がないため初心者の方でも簡単にProjectを作成することが可能

ナビゲーションや検索機能付き

AddressSpaceでNodeを選択し、右クリックすると使用可能なメニューのみが表示されますので、初心者の方でも失敗なく情報モデルの作成が可能

独自情報モデル対応は勿論、PackML、EUROMAP、Robotics、Machine Tools、Machine Vision、PLCOpenなどの公開されている情報モデルにも勿論対応

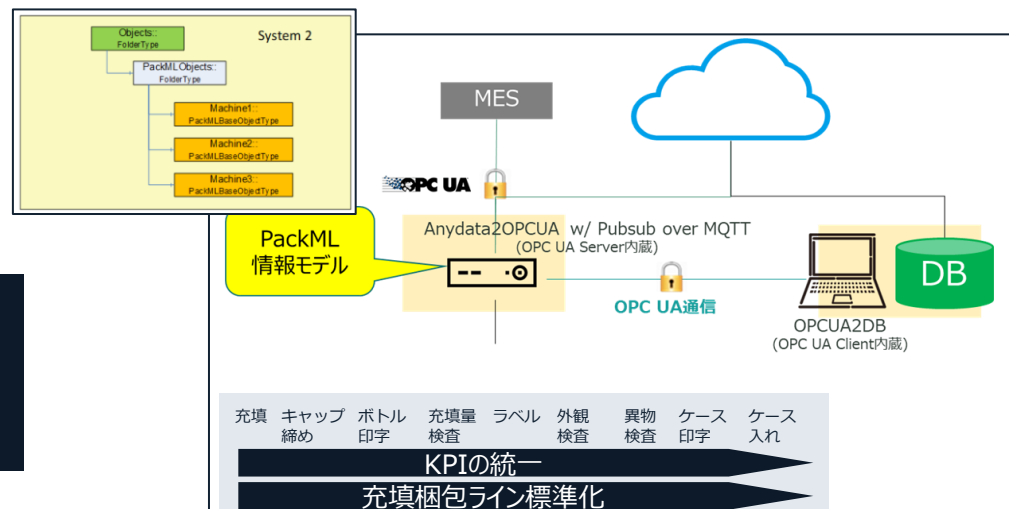
## 画面サンプル



## ● エンドユーザー様のOPC UA for PackML採用事例

- OPC UA for PackML情報モデルでラインデータを標準化&自社独自KPIを他のCSを参考にモデルを拡張
- ラインごとのKPI、OEE、チョコ停の状態と可視化の分析
- クラウドへのデータ連携
- 複数のPackMLシステムが含まれる想定での設計

ラインのロス可視化、工場間で  
取得しているデータの種類と粒度を統一させる



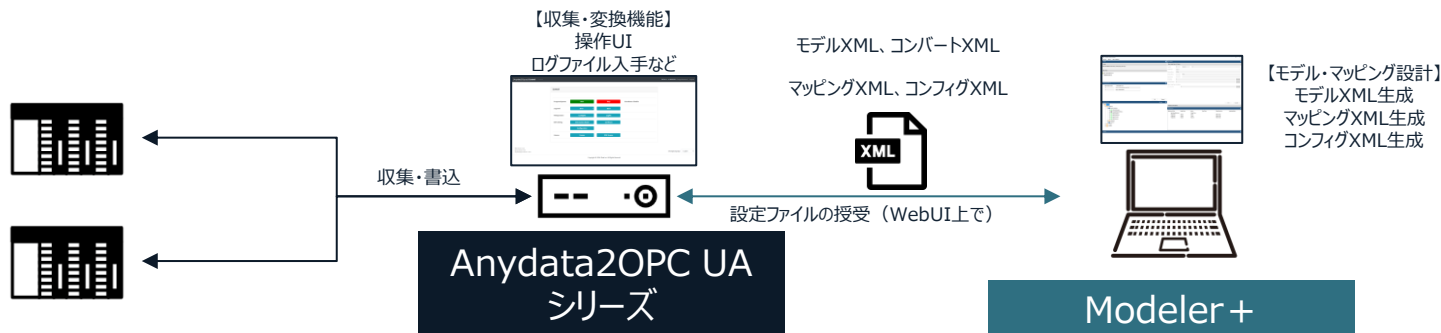
- 弊社の製品であるAnydata2OPC UAシリーズとModeler+について説明を致します。

## ➤ Anydata2OPC UAシリーズ

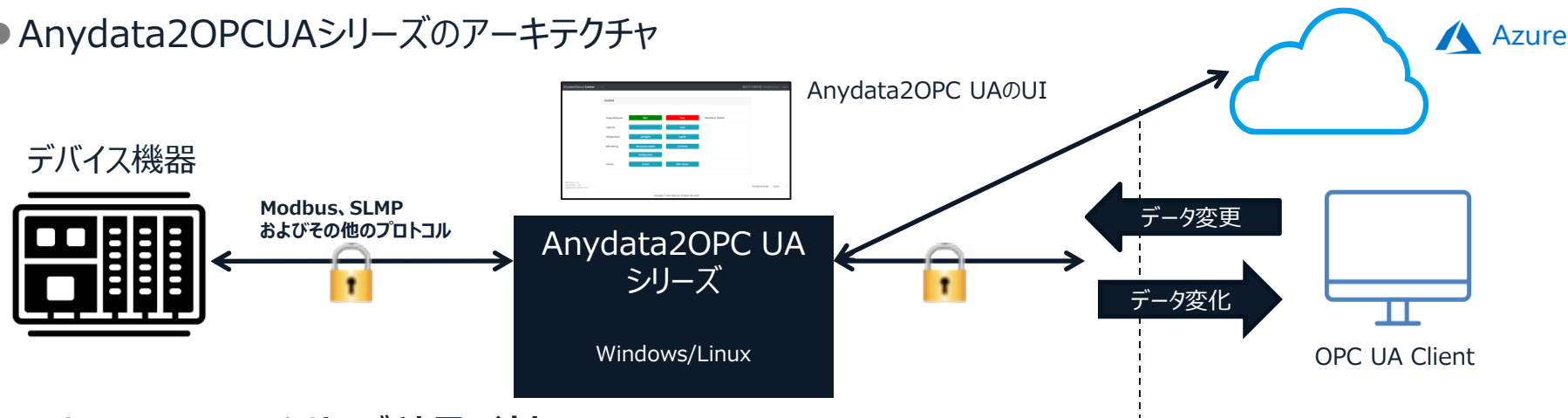
- ✓ 各データリソース(PLC、設備等)から、データを収集しOPC UA Serverに変換するソフトウェアです。
- ✓ 本ソフトウェアはWindows / Linux上で動作することができます。
- ✓ Anydata2OPC UAシリーズにはユーザのために、WebUI画面が付属しています(各設定ファイル授受 / Server ON・OFF操作など)。
- ✓ Anydata2OPC UAシリーズにはModeler+が付属しております。

## ➤ Modeler+

- ✓ Modeler+はOPC UA Serverのモデル設計ツール & 各データリソースとのマッピングを行うことができるWindowsアプリケーションです。
- ✓ 設計したモデルとコンバートファイルはAnydata2OPC UAシリーズで使用する“XML”ファイルとして生成されます。



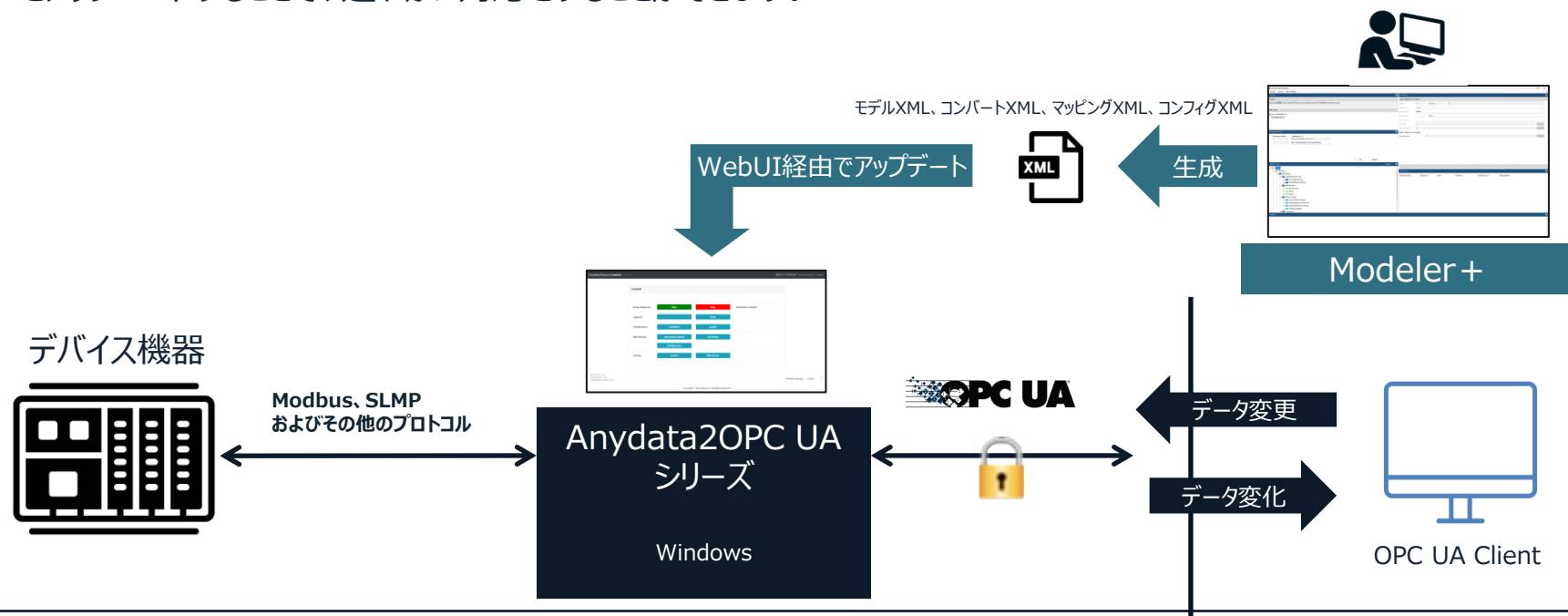
## ● Anydata2OPCUAシリーズのアーキテクチャ



## Anydata2OPCUAシリーズ 適用の流れ

1. Windows /Linux システムのゲートウェイにインストールします。
2. ModbusやSLMPなどの通信プロトコルを通じてPLC内のデータを収集します。
3. 収集したデータをOPC UA形式に変換します。
4. 標準のOPC UA Serverが付属しており、ユーザーは標準のOPC UA Clientを使用して変換されたデータを取得できます。
5. ユーザーがOPC UA Clientを通じてAnydata2OPCUA UAにデータを書き込むことで、PLC内のデータを変更することが可能です。

- Modeler+はAnydata2OPC UAシリーズの情報モデルの設計と、設備のアドレスをマッピングするのに役立ちます。
- Modeler+で設計したモデルと、コンバートファイルはAnydata2OPC UAシリーズのWebUIからアップロード可能です。
- また、モデルを変更する場合や、接続する機器のアドレスなどが変更になった際も、Modeler+で修正を行い再度ファイルをアップロードすることで、速やかに対応をすることができます。





- Anydata2OPC UAシリーズは以下のデータ収集プロトコルをサポートしています。

※対応通信については順次拡充中

データ収集プロトコル	⇔ Anydata2OPCUA			OS 対応		備考
	現状	今後予定	カスタムで対応可	Windows	Linux	
SLMP	◎			◎	◎	
SLMP/ASCII	◎			◎	◎	
Modbus/TCP	◎			◎	◎	
Modbus/RTU	◎			◎	◎	RS232C/485
FINS	◎			◎	◎	
S7		◎				
EtherNet/IP	◎			-		実装対応中
OPC UA	◎			◎	◎	
上記以外のプロトコル			○			

- OPC UAについて

- OPC UAは機器間で相互運用性を高めるための重要な技術です。
- 通信の標準化だけでなく、その情報の標準化もOPC Foundationは推進をしています。

- OPC UA for PackMLについて

- OPC UA for PackMLは、包装機械分野の標準であるPackML（Packaging Machine Language／ISA-TR88.00.02）をOPC UAの情報モデルとして定義したコンパニオン仕様です。
- 通信にOPC UA、モデルにOPC UA for PackMLを採用することで、通信と情報の両面で世界標準に準拠したシステムを構築できます。
- OPC UA for PackMLは、状態（State）、モード（Mode）、トランジション（Transition）、および生産関連の変数（タグ）を一貫した語彙で表現し、異機種間やライン全体での監視・制御・データ連携を容易にします。

- Empress Software Japanについて

- Empress Software JapanではOPC UAのSDK販売・サポートだけではなく、情報モデルのコンサルティングサービスを提供しております。
- 様々な業界で業界標準の情報モデルが用意されており、今までに数十社以上のOPC UAおよび情報モデル対応のお手伝いをさせていただきました。
- また、OPC Foundationが規定している、CTTテスト(自己認証)サポートやOPC Foundation認証のサポートなども実施しており、お客様の中で正式な認証を受けた企業様が多数おります。

- 本資料を複製、譲渡、貸与、頒布、二次配布、講習送信化するなどの著作権を侵害する行為は禁止します。
- また本資料の利用によって生じたいかなるトラブル・損害等について、当社は一切責任を負わないものとします。
- 本資料は予告なく内容の変更や削除を行う場合があります。
- Microsoft、MS、Windows、Windows Server、Windows Vista、Excel及び関連する名称並びにそれぞれのロゴは、米国Microsoft Corporationの米国およびその他の国における登録商標です。
- Linuxは、Linus Torvalds氏の米国およびその他の国における商標または登録商標です。
- その他本資料に記載されるすべての情報やコーポレートマーク、商標、映像や画像などの著作権は各社に帰属します。



# **Empress Software Japan, Inc.**