

TR-1076

IoTエリアネットワーク
情報モデルの概説

Overview of Information Model
for IoT Area Network

第2版

2021年12月7日制定

一般社団法人

情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、一般社団法人情報通信技術委員会が著作権を保有しています。
内容の一部又は全部を一般社団法人情報通信技術委員会の許諾を得ることなく複製、転載、改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

目次	
1 はじめに.....	6
2 各種情報モデルのまとめ.....	6
3. 各情報モデルの概要：アーキテクチャ/概論.....	7
3.1 ITU-T SG13/20.....	7
3.1.1 概要.....	7
3.1.2 詳細.....	8
3.2 IIC.....	10
3.2.1 概要.....	10
3.2.2 詳細.....	10
4. 各情報モデルの概要：インターネット/クラウド系.....	15
4.1 OMA (IPSO).....	15
4.1.1 概要.....	15
4.1.2 詳細.....	15
4.2 oneM2M.....	17
4.2.1 概要.....	17
4.2.2 情報モデルの一例.....	19
4.3 W3C.....	20
4.3.1 概要.....	20
4.3.2 詳細.....	21
4.4 FIWARE.....	22
4.4.1 概要.....	22
4.4.2 詳細.....	23
4.5 Microsoft Digital Twin Description Language (DTDL).....	24
4.5.1 概要.....	24
4.5.2 詳細.....	25
5. 各情報モデルの概要：デバイス系.....	26
5.1 Echonet.....	26
5.1.1 概要.....	26
5.1.2 詳細.....	27
5.2 OPC-UA.....	28
5.2.1 概要.....	28
5.2.2 詳細.....	29
5.3 OCF.....	34
5.3.1 概要.....	34
5.3.2 詳細.....	34
5.4 ZigBee.....	37
5.4.1 概要.....	37
5.4.2 詳細.....	37
5.5 BACnet.....	39
5.5.1 概要.....	39
5.5.2 詳細.....	40
5.6 LonWorks.....	41

5.6.1	概要	41
5.6.2	詳細	41
5.7	KNX	45
5.7.1	概要	45
5.7.2	詳細	45
6.	各情報モデルの概要：ネットワーク（NW）系	45
6.1	IETF	45
6.1.1	概要	45
6.1.2	詳細	46
6.2	Broadband Forum	47
6.2.1	概要	47
6.2.2	詳細	48
7.	まとめ	50

<参考>

1. 国際勧告等との関連

本技術レポートに関する国際勧告は本文中に記載している。

2. 改版の履歴

版数	制定日	改版内容
第1.0版	2019年6月4日	制定
第2.0版	2021年12月7日	検討項目追加、分類変更他

3. 参照文章

主に、本文内に記載されたドキュメントを参照したが代表的な文書を下記に示す。

3. 1 規準参照文書

- [1] oneM2M TS-0023-V2.0.0, Home Appliances Information Model and Mapping
- [2] TS-M2M-0023v2.0.2 oneM2M 技術仕様書 家電機器の共通デバイス管理モデル
- [3] Home Gateway Initiative Smart Device Template
<https://github.com/Homegateway/SmartDeviceTemplate/tree/7c890b69d9764e341ef1768c5a0e7d53a47cff5c>

3. 2 非規準参照文書

- [TTC標準 JJ-300.30] 「橋梁モニタリング用加速度センサの情報モデル及び低消費電力無線通信における動作概要」, 2017, 一般社団法人情報通信技術委員会
- [TR-1066] TTC技術レポートTR-1066 「橋梁モニタリングのための低消費電力無線通信方式のガイドライン」, 2017, 一般社団法人情報通信技術委員会
- [oneM2M] <http://www.onem2m.org/>
- [TR-1083] TTC技術レポートTR-1083 「NETCONFに関する技術報告書」, 2020, 一般社団法人情報通信技術委員会
- [TR-1087] TTC技術レポートTR-1087 「YANG 1.1 データモデリングに関する技術報告書」, 2021, 一般社団法人情報通信技術委員会

4. 技術レポート作成部門

- 第1.0版 : IoTエリアネットワーク専門委員会 (SWG3604)
- 第2.0版 : IoTエリアネットワーク専門委員会 (SWG3604)

1 はじめに

本概説は、IoTエリアネットワーク情報モデルの概要書であり、デバイス側とサービス側のデータを伝達する仕組みである、情報モデルについて各種の分野で検討されているモデルについて調査したものである。今後IoTに関する通信制御はますます増大するものと考えられ、対象となる分野は家庭内のみならず、街レベルや電力制御や社会インフラ等への制御として変遷・拡大を続けてゆくものと考えられる。そのため、これらに係る各種フォーラム等の情報モデルを調査し、関連情報・プロトコル・データ情報等を整理し今後の方向性も合わせて検討したものである。

具体的には、情報モデルとはデバイスの機能・情報を表現するためのモデルでデバイス管理モデルとも呼ばれている。図1-1 にoneM2M等で述べられているイメージ図を示す。

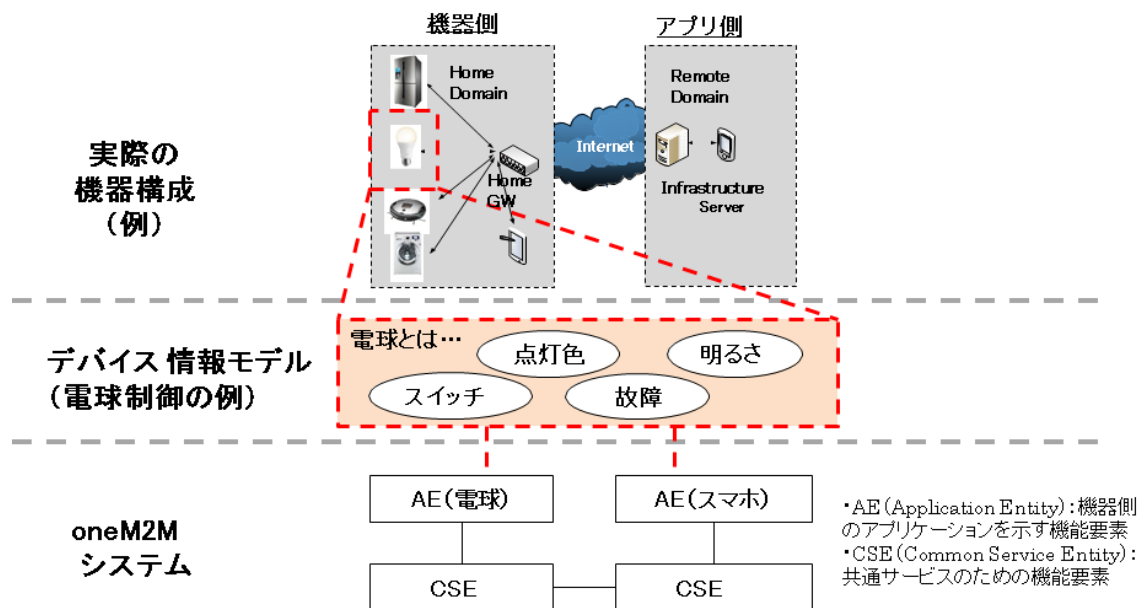


図 1-1 情報モデルイメージ図

2 各種情報モデルのまとめ

情報モデルは各所で検討されており、それぞれ名称や構造が異なっている。具体的な下記検討機関を調査・検討して、下記カテゴリ毎にとりまとめた。

- ① アーキテクチャ/概論: Y.2070/Y.4409、Industrial Internet Consortium (IIC)
- ② インターネット/クラウド系: LwM2M、oneM2M、Web of Things (WoT)、FIWARE、Microsoft Digital Twins Description Language (DTD)
- ③ デバイス系: ECHONET Lite、OPC-UA、OCF、ZigBee、BACnet、LonWorks、KNX
- ④ ネットワーク(NW)系: SNMP、NETCONF、TR-069

これらを、上記カテゴリ毎に分類した比較表を表2-1 に示す。

3章以降にこれらの詳細について記載する。

表 2-1 各種情報モデルのまとめ

No.		標準化団体	標準規定範囲		
			アーキテクチャ	プロトコル関係	情報モデル
①	概論	ITU-T SG13/20	Y.2070/Y.4409	(規定範囲外、もしくは策定中)	
		IIC	IIRA*		
②	インターネット/クラウド	OMA	LWM2M	LWM2M	IPSO
		oneM2M	TS-001	TS-0004	TS-0023(home appliance)
		W3C	Web of Things(WoT)	Binding Template	Thing Description
		FIWARE	OMA NGSI	OMA NGSI	ETSI CIM
		Microsoft	Azure Digital Twins	REST	DTD
③	デバイス	ECHONET	規定外	ECHONET Lite	Device Description
		OPC UA	Overview	Mappings	Information Model
		OCF	OIC Core	Cloud API/Binding	OIC Core Smart Home
		ZigBee	規定外	ZigBee IP/Pro	ZigBee Cluster Library
		BACnet	規定外	BACnet	FSGIM*
		LonWorks	規定外	LonTalk	Application Description
		KNX Association	規定外	KNX	Application Description
④	ネットワーク	IETF	NETCONF	NETCONF(RFC6241)	YANG (RFC6020)
			SNMP	SNMP	MIB/SMI
		Broadband Forum	TR-069	TR-069	TR-181

注：IIRA→IICF(Industrial Internet Connectivity Framework)、FSGIM (Facility Smart GridInformation Model)

3. 各情報モデルの概要：アーキテクチャ/概論

アーキテクチャや概論に関係する規定で、ITU-T SG13/20におけるY.2070/Y.4409、Industrial Internet Consortium (IIC) らがそれに該当する。以下に詳細を述べる。

3.1 ITU-T SG13/20

3.1.1 概要

ITU-T SG13 (Study Group 13) は、ITU-Tの標準化グループの1つである。将来ネットワーク(Future Networks)の要求条件、アーキテクチャの策定を担当する。SDN(Software Defined Networking)やスライスなどの仮想化技術から、最近では機械学習を利用したネットワーク管理、量子鍵配送などをカバーしている。Y.2070は、ホームエネルギーマネジメントシステム(HEMS)とホームネットワーク(HN)サービスの機能要件とアーキテクチャを規定する規格であり、日本が主導して策定し2015年に勧告化された。現在は、この勧告がSG20に移管されたため、Y.4409と番号が変更されている。なお、Y.4409の実装ガイドライン文書として補足文書(Y.sup57)が2019年12月に承認されており、Y.4409を実装するうえで利用可能な規格等について例が示されている。

Y.4409では以下のような内容が規定されている。

- ・HEMSや他のHNサービスのためのHNサービスアーキテクチャの概要

- ・HNサービスアーキテクチャにおけるデバイス、ホームゲートウェイ(HGW)、管理プラットフォーム(PF)の機能要件、アーキテクチャに要求されるセキュリティ要件
- ・デバイスのタイプに合わせて、4通りのデバイスとHGWとの接続方法を持つ参照アーキテクチャ。デバイスのタイプは、IPまたは非IPで接続されるベーシックデバイスと、HGWと直接またはアダプタ経由で接続される非ベーシックデバイスからなる
- ・デバイス、HGW、管理PF、アプリケーションの4つのエンティティを持つ機能アーキテクチャ
- ・機能アーキテクチャの3種類の機能カテゴリ(デバイス操作、アプリケーション実行、システム管理)に関するエンティティ間の関係
- ・主として、HEMSに関するHNサービスに対するセキュリティモデルと機能

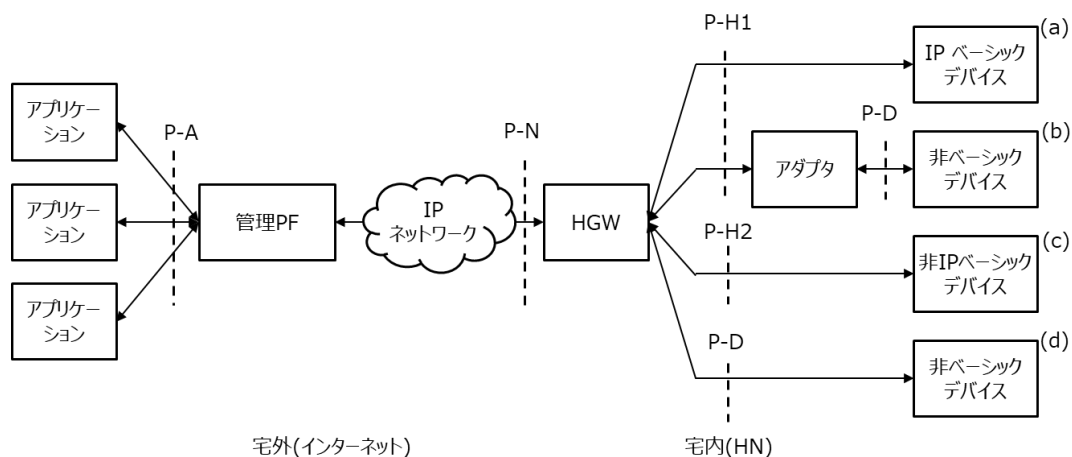


図 3-1 リファレンスアーキテクチャ (JT-Y2070)

Y.4409は、HEMSをユースケースとしたアーキテクチャとなっているが、他のIoTユースケースにも適用できるように汎用性を意識して作成された。検討段階では、IoT規格としてECHONET Liteのような家電の規格だけでなく、BACnetやZigBeeアライアンスのSEPを参照しているだけでなく、ネットワーク機器の運用管理規格であるBroadband ForumのTR-069やTR-181を参照している。また、補助文書では、W3C Web of Things(WoT)、OMA NGSI、OneM2m、OMA LwM2M及びIPSO、IEC 61970、KNX等についても言及しており、Y.4409に記載されるアーキテクチャの中で、これらの規格が適合することを示している。

3.1.2 詳細

Y.4409のアーキテクチャは、HGW(Home GateWay)の右側がHNであり、左側はWAN(インターネット)である。HNの接続はIPを基本とし、WANの接続はHTTPである。HGWはこれらの通信を中継する。この参照アーキテクチャには、ベーシックデバイスと非ベーシックデバイスに分類されたデバイスが示されている。ベーシックデバイスは、デバイスの機能を表現する抽象的な情報モデルを持つ。ベーシックデバイスのインターフェースは、HGWに標準で備わっており、ここでは情報モデルが利用される。また、この情報モデルは管理PFにおいても表現されており、アプリケーションが内部状態を参照し、制御する仮想デバイスとして扱われる。

ベーシックデバイスのインターフェースには、いくつかの標準仕様がある。そのうちのいくつかはIPベースの通信プロトコルをサポート(図3-1(a))し、他のものは非IPベースのプロトコルをサポート(図3-10(c))している。非ベーシックデバイスは、内部に情報モデルを持たない。非ベーシックデバイスは独自通信インターフェースで接続するため、このHGWに接続するためには情報モデルを代わりに備えるアダプタを接続する(図3-1(b))が必要となる。デバイスとHGWがLAN等を経由して直接接続可

能な場合には、HGWにアダプタ機能を持つことも可能であり、この場合には非ベーシックデバイスはHGWに直接接続される（図3-1(d)）。

こうしてHGWに接続されるデバイスは、HGWで共通の情報モデルにより記述され、WANから接続可能な仮想デバイスとして参照できる。WANを経由したクラウド等からデバイスへの接続については、参照点P-Nを通じてHGWの仮想デバイスにアクセスするか、APIゲートウェイとして動作する管理PFを通じてHGW経由で接続する。

図3-2は機能アーキテクチャを示す。ここではデバイスの持つ情報モデルをクラウドから利用可能となるための仕組みについて述べる。デバイスはそれぞれ固有の機能を持つ。これらの機能はプロファイルとして定義され、HGWや管理PFを通じて、アプリケーションに通信プロトコルによって送信される。この結果、本アーキテクチャでは、アプリケーションがデバイスの内部状態を参照し、制御することが可能である。

図3-2は、IPベースのベーシックデバイスに対するIPベースの通信プロトコルを示しているが、図3-1で説明したように、HGWとデバイス間ではIPベース、非IPベース、独自の通信プロトコルが利用される。HGWとデバイス間でやりとりされるコマンドは、HGWがデバイスに対して、内部状態を取得するための”GET”（参照）、プロパティを指定して情報を設定する”SET”（更新）、内部状態やイベントが更新したときに通知するように要求する”INFORM”（通知）のような制御メソッドを提供する。HGWは、通信プロトコルをHTTPに変換し、WANを介して、管理PFと通信する。したがって、デバイスの情報モデルをクラウドから透過的に利用することができる。

また、図3-2の最下層に記されるデバイス管理(リソース管理)についても、標準的な通信規格が存在する。IETF NETCONF/YANGやBBF TR-069/TR-181がこれに当たる。これらの規格についても、デバイスと同様に、管理情報がプロパティとして定義された情報モデルとして表現される。この情報モデルに対して、各規格では、”GET”、”SET”、”INFORM”のようなコマンドにより操作されており、同じ構造を持っている。

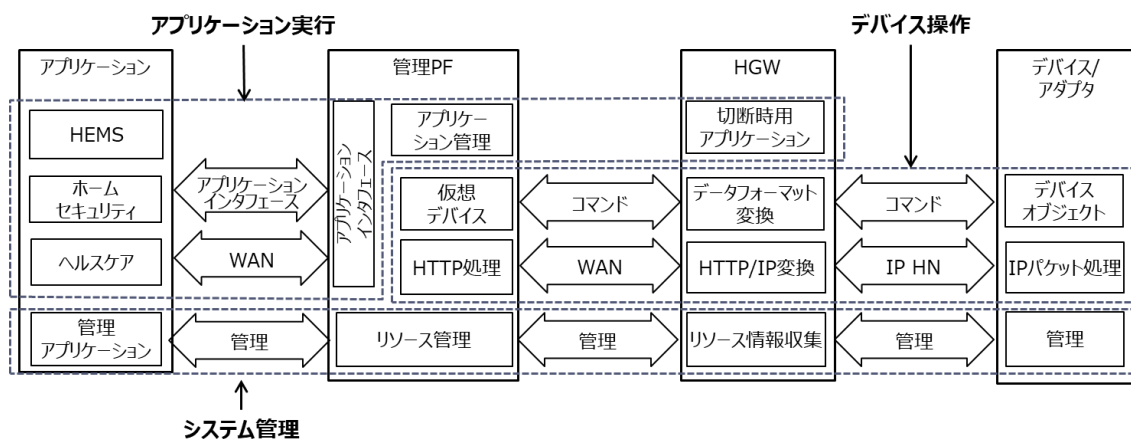


図 3-2 機能アーキテクチャ

Y.4409における具体的な通信規格との対応関係については、補助文書Y.sup57に記載されている。

<参考文献>

ITU-T Recommendation Y.4409/2070: <https://www.itu.int/rec/T-REC-Y.2070-201501-I/en>

ITU-T Recommendation Y.sup57: <https://www.itu.int/rec/T-REC-Y.Sup57-201912-I>

3.2 IIC

3.2.1 概要

IIC (Industrial Internet Consortium) は 2014 年 3 月に米国企業 5 社 (AT&T、Cisco、General Electric、IBM、Intel) により Industrial IoT (IIoT) の実現のために設立されたコンソーシアムである。

IIC が定めた共通のアーキテクチャやフレームワークをガイドラインとして使用しており、一例として IIC Connectivity Framework がありシステム間のデータ交換が可能になる。

3.2.2 詳細

コネクティビティは、機能ドメイン内の参加者間、システム内の機能ドメイン間、システム間のデータ交換を可能にする。交換されるデータには、センサの更新、イベント、アラーム、ステータスの変更、コマンド、および構成の更新が含まれている。接続性は、図 3-12 に示すように、Industrial Internet Reference Architecture¹ で定義されている機能ドメイン間のクロスカット機能で、接続性は、インダストリアルインターネット参照アーキテクチャにおける分野横断的な機能である。これは、機能ドメイン内 (管理、運用、情報、アプリケーション、ビジネス) の参加者間でデータを交換する機能を提供する。

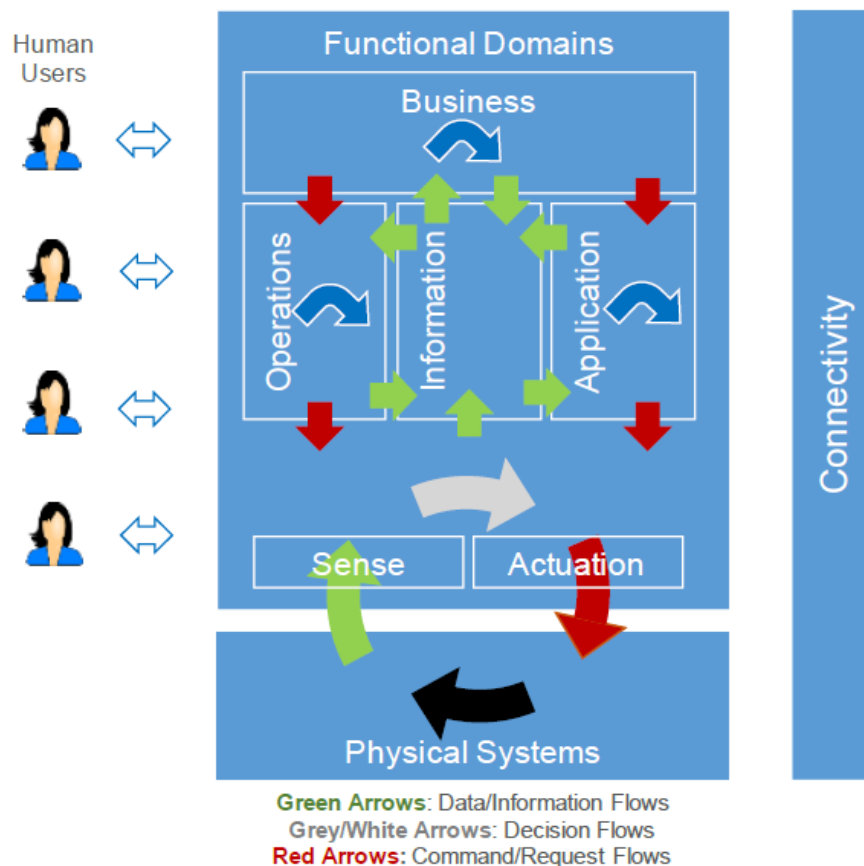


図 3-3 Industrial Internet Reference Architecture

出典: The Industrial Internet of Things Volume G5: Connectivity Framework

Figure1-1 Industrial Internet Reference Architecture より

IIoT (Industrial Internet of Things) 接続スタックモデルを図 3-4 に示す。IIoT 接続スタックモデルと、Industrial Internet Reference Architecture (IIRA) 3 のクロスカット機能としての接続範囲を示している。接続機能は、IIoT システム内の機能ドメイン内の機能ドメイン間のデータ共有メカニズムを提供している。

図 3-4 は、IIoT 接続スタックモデル、および Industrial Internet Reference Architecture (IIRA) 内の分野横断的機能としての接続の範囲を示している。接続機能は、機能ドメイン内の参加者間および IIoT システム内の機能ドメイン間でのデータ共有メカニズムを提供する。The data services framework in the distributed data interoperability and management function は、コネクティビティフレームワークによって提供される foundation for semantic interoperability により、インダストリアルインターネット参照アーキテクチャ (IIRA) 1 の動的な構成およびコーディネーション機能の semantic interoperability の基礎を提供する。

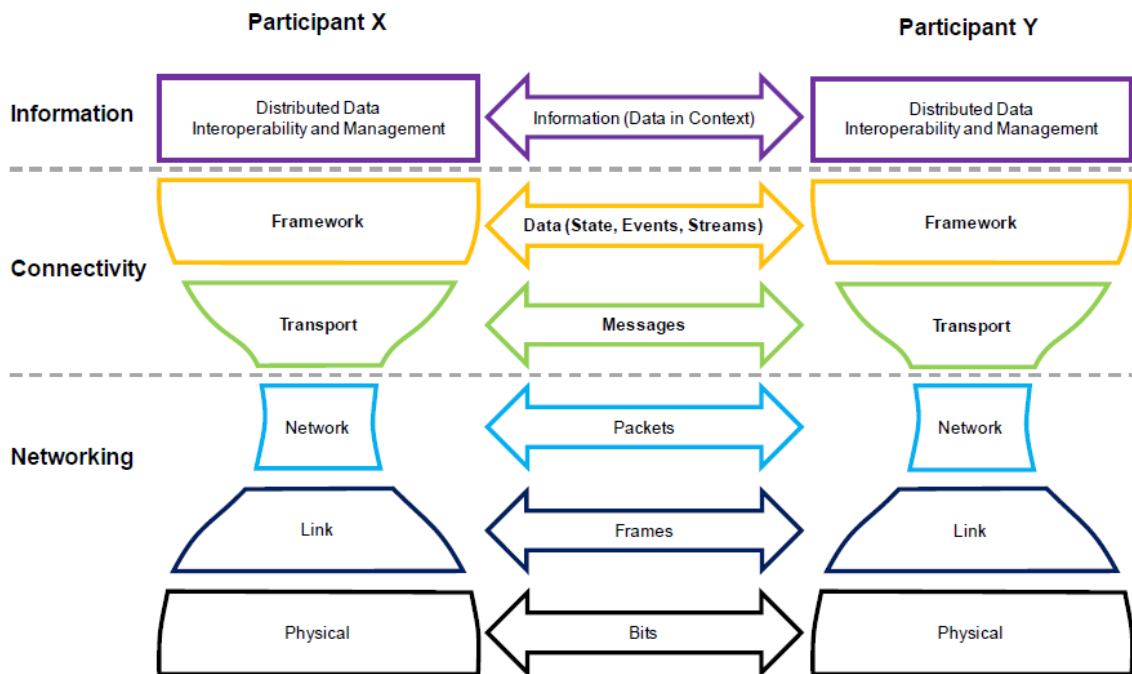


図 3-4 IIoT (Industrial Internet of Things) 接続スタックモデル

出典： The Industrial Internet of Things Volume G5: Connectivity Framework

Figure 2-1: Industrial Internet Connectivity Stack Model. より

(https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.01_PB_20180228.pdf)

この IIoT 接続機能レイヤの役割と範囲を表 3-1 に示す。

表 3-1 Role and scope of the Connectivity functional layers

IIoT Connectivity Stack Model	Correspondence to OSI Model (ISO/IEC 7498)	Correspondence to Internet Model (RFC 1122)	Correspondence to Levels of Conceptual Interoperability
Framework Layer	7. Application	Application Layer	Syntactic Interoperability: Structured data types shared between endpoints. Introduces a common

	6. Presentation		structure to share data; i.e., a common data structure is shared. On this level, a common protocol is used to exchange data; the structure of the data exchanged is unambiguously defined.
	5. Session		
Transport Layer	4. Transport	Transport Layer	Technical Interoperability: Bits and Bytes shared between endpoints, using an unambiguously defined communication protocol.
Network	3. Network	Internet Layer	Packets shared between endpoints that may not be on the same physical link. Packets are routed between physical links by a “network router”.
Link	2. Data Link	Link Layer	Digital Frames shared between endpoints on a shared substratum (link).
Physical	1. Physical		Analog signal modulation between endpoints on a shared substratum.

出典： The Industrial Internet of Things Volume G5: Connectivity Framework
Table 2-1: Role and scope of the Connectivity functional layers.より
(https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf) .

次に機能ドメイン間のやり取りの例を図 3-5 に示す。異なる機能ドメイン間で横方向での相互運用性によるコミュニケーションの目標を実現するには、図に示すように、各業界・標準間で、標準化されたコアゲートウェイで、各コアを接続することにより実現する。

各コア接続規格には、他のすべてのコア規格に対する標準化されたゲートウェイが必要で、コア規格が追加されるたびに、複雑さと相互運用性の課題が高まる。設計をいくつかのコア接続規格に限定することで、機能ドメイン全体で IIoT システムのニーズをカバーし、業界間での水平方向の相互運用性の目標を達成している。

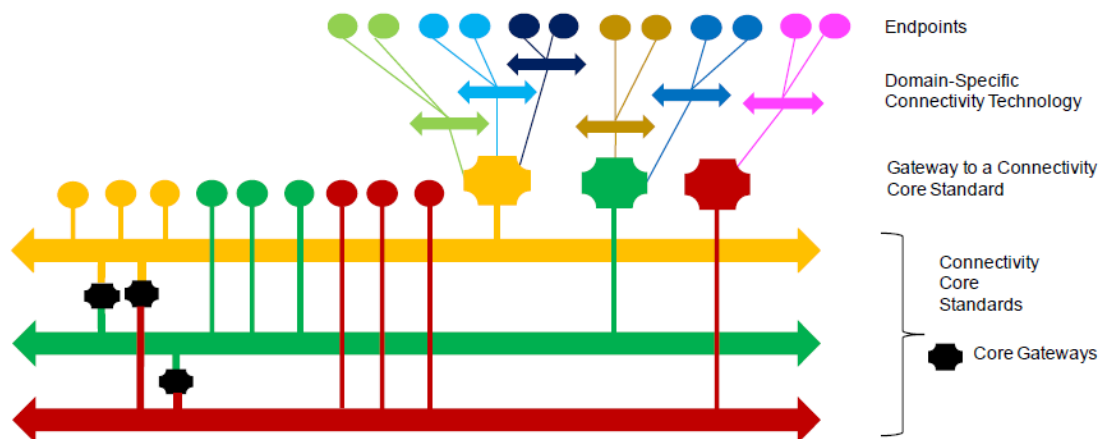


図 3-5 standardized Core Gateway

出典： The Industrial Internet of Things Volume G5: Connectivity Framework
Figure 3-4: Each core connectivity standard より
(https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.01_PB_20180228.pdf)

図 3-6 は、IIoT 接続フレームワークとトランスポートをまとめたもので、汎用的な使用のためにとらわれない方法で作成された接続フレームワークを示している。点線で囲まれたボックスは、複数の業界・

標準にまたがって適用可能な特定のアプリケーションに焦点を当てて、特定業界の業界で起きた接続基準を示している。いくつかの接続フレームワークは、独自のトランスポートプロトコル（例えば、DDS、OPC-UA：次項目参照）を定義しており、これらのフレームワークは、フレームワークとトランスポート層ボックスとの間に隙間なく示されている。その他のもの（Web サービス、oneM2M など）は、汎用転送プロトコルに依存している。網（IP）および下位層（有線および無線）も参考として示している。

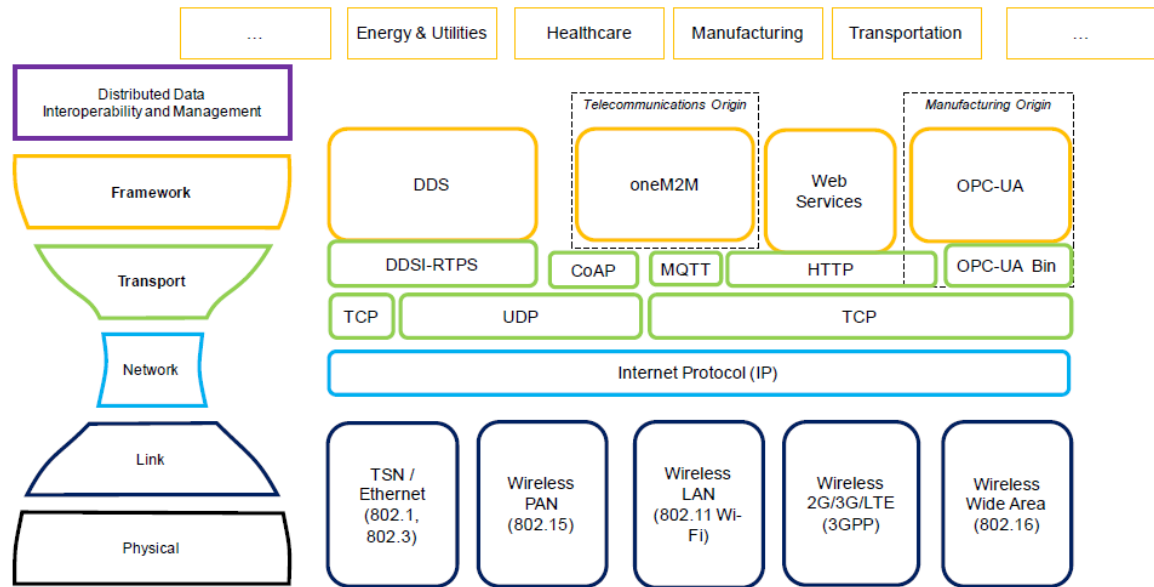


図 3-6 The prominent IIoT connectivity frameworks and transports

出典： The Industrial Internet of Things Volume G5: Connectivity Framework

Figure 7-1: IIoT connectivity standards. Dotted より

(https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf)

主要な Connectivity framework 機能には、データリソースモデル、パブリッシュ - サブスクライブおよび要求 - 応答データ交換パターン、データのサービス品質、データセキュリティ、およびプログラミング API が含まれる。 これらを図 3-7 に示すと同時に、主要機能について以下に記載する。

◆ DATA RESOURCE MODEL

Connectivity framework は、時間とともに状態を変えることができるデータオブジェクト 1 を表す方法を提供する。データオブジェクトは、プログラミング言語のように、構造化されたフィールドの集まりで、それは階層的であり得、そして静的または動的に型付けされ得る。

Connectivity framework は、参加者の間でデータオブジェクトへの変更を配布する。さまざまなアプリケーション分野や業界のデータモデルは、通常、接続フレームワークによって提供される抽象データオブジェクトにマッピングされる。

◆ ID AND ADDRESSING

Connectivity framework は、各データオブジェクトを識別してアドレス指定する手段を提供する。ID は、データオブジェクトをアドレス指定し、データオブジェクト表現内のフィールドを読み書きするために使用される。

・データオブジェクト表現内の明示的な ID フィールド

- ・データオブジェクト表現で特別にマークされたフィールドに基づく暗黙の ID
- ・デバイス、アプリケーション、またはネットワークエンドポイントの名前空間内の統一リソース識別子 (URI)。

◆ DATA TYPE SYSTEM

構文上の相互運用性を保証するために、コネクティビティスタックはデータの構文を記述する方法を提供する。データ型は、データの解釈に課せられる構文上の制約で、暗黙のうちに（例えばコード内で）または明示的にデータ型を共有またはマッピングすることなくシステムを接続することは不可能である。

Connectivity framework は、プログラミング環境においてデータオブジェクトを構造として表現するため、およびワイヤ上で通信されるデータをフォーマットするためのデータ型システムを提供する。データ型システムは、静的型プログラミング言語（例えば C、C++、C# または Java）に見られるデータ型のようにオブジェクト指向型でもよいし、または動的型プログラミング言語に見られる動的データ型（例えば C、JavaScript、Python、Lua）でも良い。

データ型システムは、データ型の進化を管理する手段を提供する必要がある。これには、バージョン間のバージョニングおよび割り当て可能性ルールが含まれるため、新しいバージョンのデータ型を使用しているアプリケーションは、古いバージョンのデータ型を使用しているアプリケーションと可能な限り最大限に通信できる。

データ型システムはまた、通信中（移動中）および格納中（静止中）の直列化データフォーマット、ならびにプログラミング言語表現から直列化フォーマットに直列化する動作、およびプログラミング言語表現に逆シリアル化する動作を定義する。

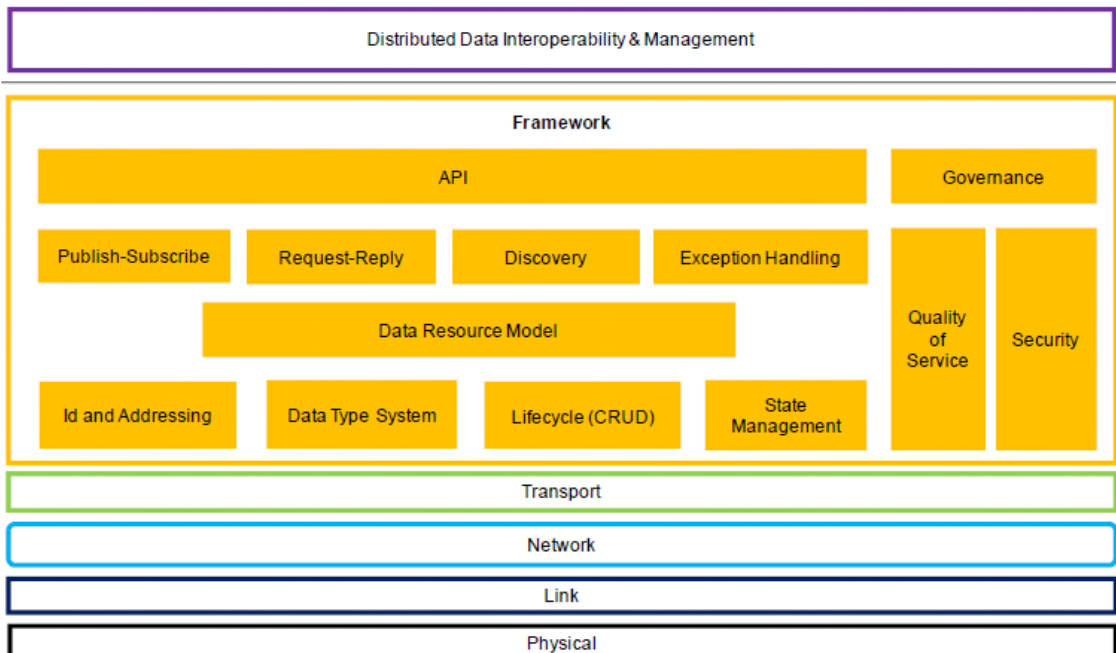


図 3-7 Connectivity framework layer functions

出典： The Industrial Internet of Things Volume G5: Connectivity Framework

Figure 4-1: Connectivity framework layer functions. より

(https://www.iiconsortium.org/pdf/IIC_PUB_G5_V1.0_PB_20170228.pdf)

4. 各情報モデルの概要：インターネット/クラウド系

インターネット/クラウド系に関する規定で、OMAの LwM2M、oneM2M、Web of Things (WoT)、FIWARE、Microsoft Digital Twins Description Language (DTDL) らがそれに該当する。以下に詳細を述べる。

4.1 OMA (IPSO)

4.1.1 概要

IPSOは2008年9月に設立されたアライアンス (IP for Smart Objects Alliance) で、目的は“Smart Object”をIP (Internet Protocol) ベースで構築。設立はArch Rock、Atmel、Cimetrics、Cisco、Duke Energy、Dust Networksなど、テクノロジー、通信、エネルギーの各企業25社で行われ、エネルギー、消費者、ヘルスケア、および産業用アプリケーションにおけるIPネットワークデバイスを提唱している。

IPSOスマートオブジェクトは

- スマートオブジェクトアライアンス (IPSO) 用 IP によって開発
- IoT デバイスおよびアプリケーション間のセマンティック相互運用性
- LWM2M オブジェクトモデルに基づく、再利用可能なオブジェクト ID とリソース ID
- LWM2M アドレス指定、データタイプ、およびコンテンツフォーマットをサポートできる、さまざまなトランスポートプロトコル (CoAP、HTTP、MQTT) で使用可能
- 単純なオブジェクト用の Basic Starter Pack が 2014 年に公開
- Basic Objects は単純なセンサとアクチュエータから構成

4.1.2 詳細

IPSO Smart Objectsの構成を図4-1に示す。

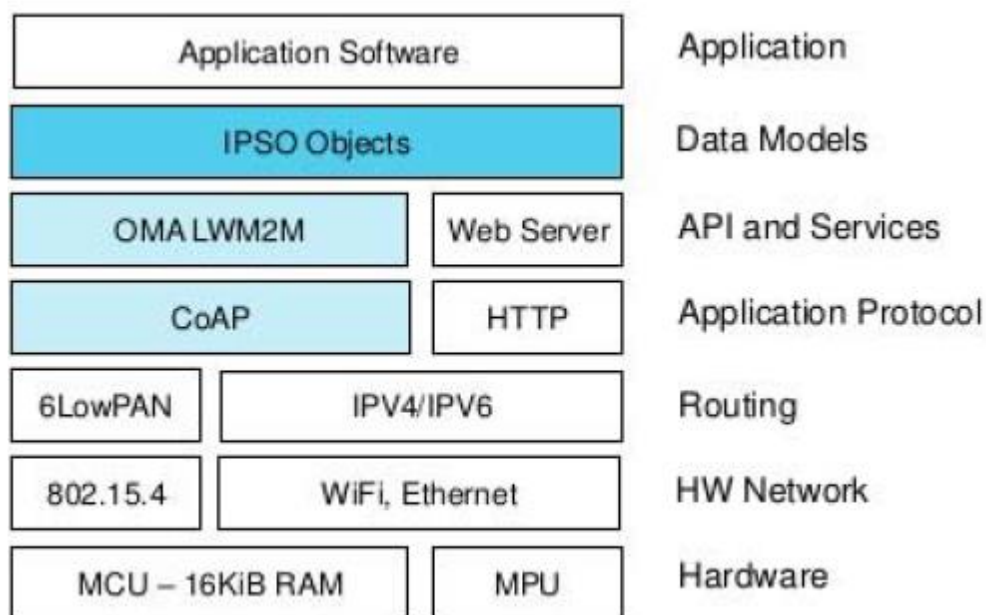


図 4-1 IPSO の 7 層モデル

出典： IPSO Smart Object for IoT、P3より

(<https://www.slideshare.net/michaeljohnkoster/ipso-smart-objects-for-iot>)

IPSOには以下のような特徴がある。

- ◆ 組み込みデバイスへの RESTful Web サービス
 - エンタープライズ SOA 統合を容易にする
 - アプリケーション開発を容易にする Web プログラミング
- ◆ HTTP または CoAP が可能
 - Observe など、制約のある環境での CoAP (および EXI)
 - HTTP-CoAP インターワーキング
- ◆ リソースビューをサポート
 - Web リソース - Web リンク
 - セマンティックアノテーションとシンプルなプロファイルが望ましい
 - XML - EXI, JSON
- ◆ 「オープン」とは発見を示す
 - 出版と検索
- ◆ レガシーデバイスのラッピングが可能

図4-2に、温度管理をモデルとした制御概念図を示す。

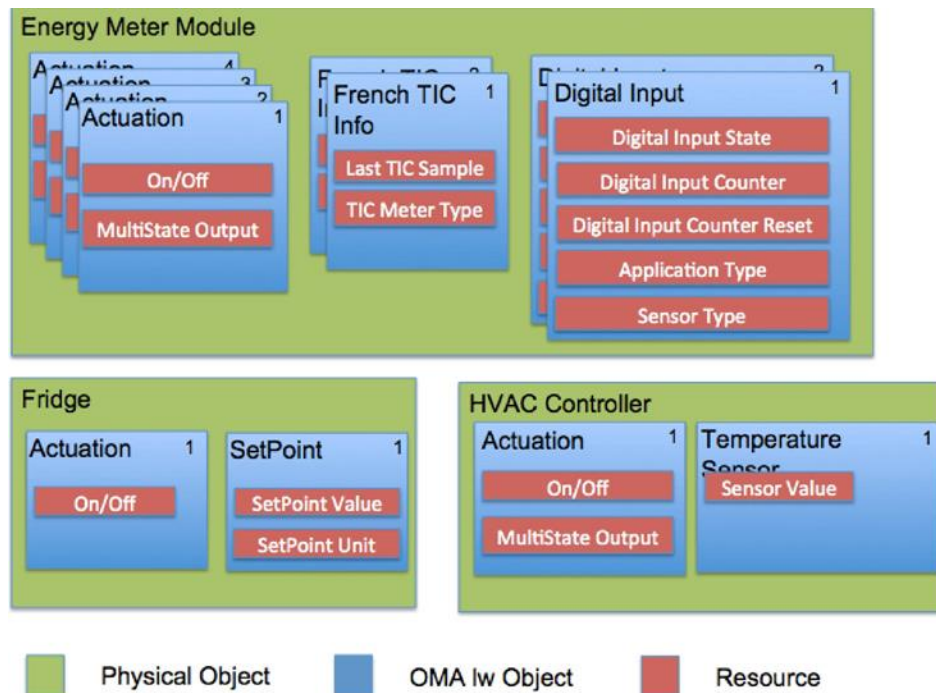


図 4-2 IPSO スマートオブジェクト概念図

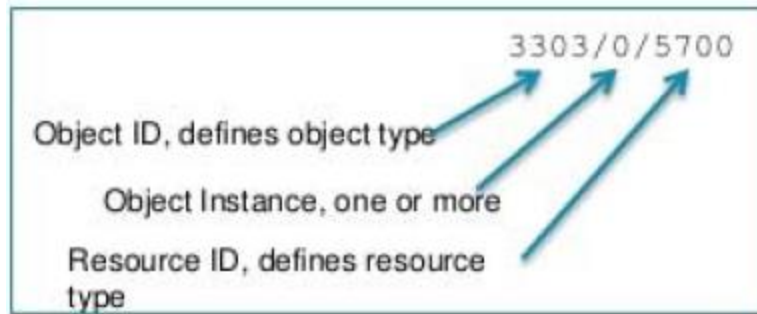
出典： IPSO Smart Object for IoT、P11より

(<https://www.slideshare.net/michaeljohnkoster/ipso-smart-objects-for-iot>)

また、具体的な制御のデータモデルとしての IPSO BAROMETER OBJECT の制御例を図4-3に示す。

- ◆ URI テンプレートを使った REST API
 - オブジェクト
 - オブジェクトインスタンス
 - リソース
 - (リソースインスタンス)

- ◆ 再利用可能なリソースとオブジェクト ID
 - 概念の共通定義
 - セマンティック用語へのマッピング 温度、現在の価値
 - ID は OMNA に登録されています
- ◆ サーバー上のパス階層に埋め込むことができる
 - 一例 : /home/weather/3303/0/5700



Resources:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float			Last or Current Measured Value from the Sensor
Units	5701	R	No	Optional	String			Measurement Units Definition e.g. "Cel" for Temperature in Celsius.
Min Measured Value	5601	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The minimum value measured by the sensor since power ON or reset
Max Measured Value	5602	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The maximum value measured by the sensor since power ON or reset

図 4-3 IPSO BAROMETER OBJECT 制御の一例

出典 : IPSO Smart Object for IoT, P4.5より

(<https://www.slideshare.net/michaeljohnkoster/ipso-smart-objects-for-iot>)

4.2 oneM2M

4.2.1 概要

oneM2M は、欧米日中韓の標準化団体（7 団体 : ETSI、ATIS、TIA、ARIB、TTC、CCSA、TTA）が協力し、2012 年に結成した M2M に関するグローバル標準仕様策定プロジェクトである。情報モデルに関しては oneM2M がホームアプライアンスの抽象情報モデル策定に適用した、HGI（Home Gateway Initiative）で規定された記述様式 SDT（Smart Device Template）3.0 を利用した TS-0023 「Home Appliances Information Model and Mapping」がある。図 4-4 にその概要を示す。

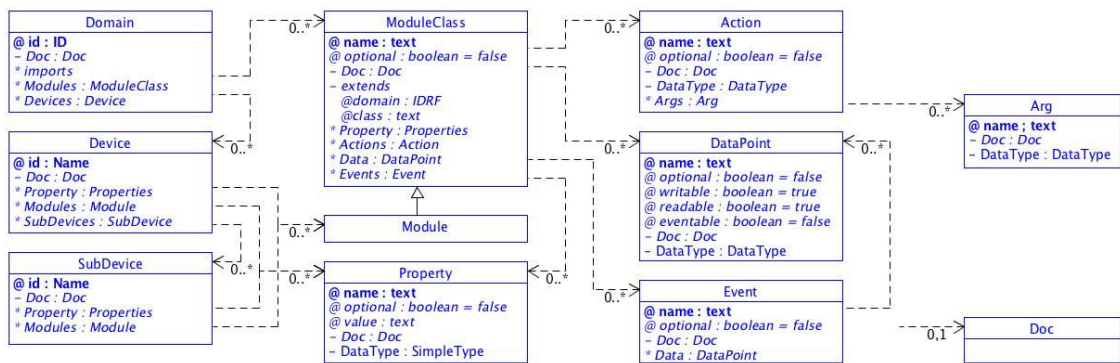


図 4-4 oneM2M の SDT

(出典: oneM2M TS-0023 出典: TS-M2M-0023v2.0.2

「oneM2M 技術仕様書家電機器の共通デバイス管理モデル」Figure 5.2.1-1 より引用

(http://www.ttc.or.jp/jp/document_list/pdf/j/TS/TS-M2M-0023v2.0.2.pdf)

図にあるように SDT の構成要素としては、以下の要素で構成するものとする。

(1) ドメイン (Domain)

- ネームスペースを示し、規定するモジュールやデバイスを一意に参照できる
- テクノロジーやセグメントの異なる SDT テンプレートへのラベル付けが出来る
- これにより、異なる詳細構造を持つ複数のテンプレートが利用できる

(2) デバイス (Device)

- モデル化するセンサ、アクチュエータなどの基本的な実体を示す
- Sub Device を定義して階層化することが出来る
- 0 または複数の Sub Device および Module と Property で構成される

(3) サブデバイス (Sub Device)

- Device に組み込まれた、個別の機能を持った実体を示す
- Sub Device を追加することで Device の機能を変化させる場合などに用いる
- Sub Device は 0 または複数の Module と Property で構成される

(4) モジュールクラス (Module Class)

- 一つのサービスを示す
- 複数の Device 間で再利用ができる
- 0 または複数の Action、DataPoint、Event、Property から構成される

(5) モジュール (Module)

- Module Class から生成するサブクラスである
- Device または Sub Device に特有のサービスを表す
- Device または Sub Device に包含される

(6) アクション (Action)

- Device のオペレーションを表す
- Module に包含される

(7) データポイント (DataPoint)

- Device が持つ機能の 1 つの態様を表す

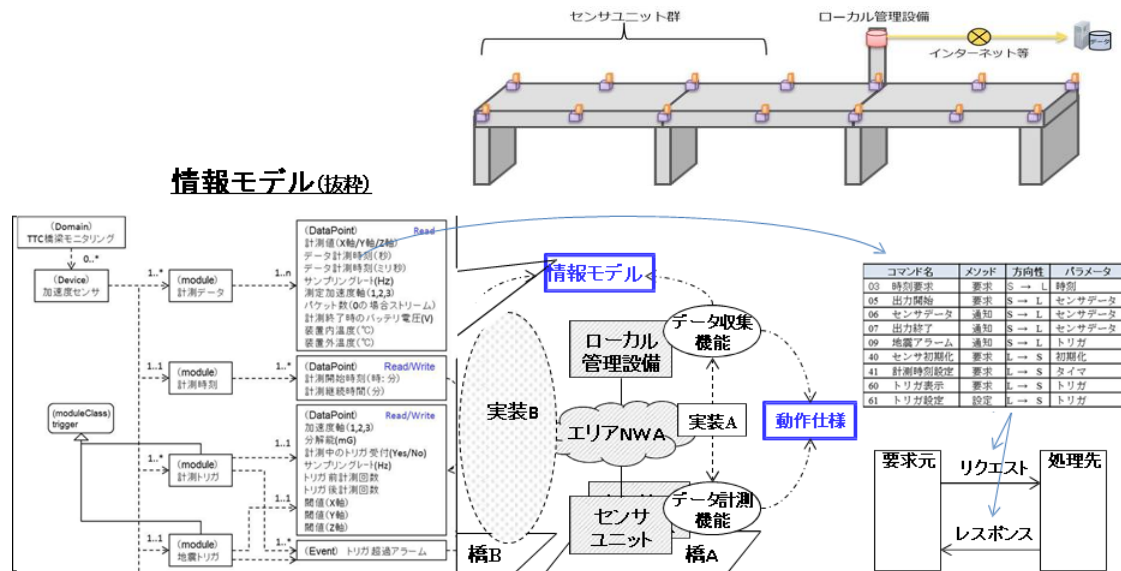
- ステートレスなオペレーション（読み、書き、または両方）が可能である
 - Module に包含される
- (8) イベント (Events)
- Device に特定の事象が発生した時にそれを公開する
 - Module に包含される
- (9) プロパティ (Property)
- Device または Module に任意の追加情報を付加するために用いる
 - Device または Module の機能に直接関係しない情報を示す（ベンダー名、シリアル番号等）
 - Device または Module に包含される
- (10) ドキュメンテーション (Doc)
- 補足説明を記載する
 - Device や Module の機能には影響しない
 - 全ての構成要素に含まれる

4.2.2 情報モデルの一例

OneM2MのSDTを参考にした情報モデルの一例として、TTC技術レポートTR-1066 「橋梁モニタリングのための低消費電力無線通信方式ガイドライン」とTTC標準 JJ-300.30 「橋梁モニタリング用加速度センサの情報モデル及び低消費電力無線通信における動作概要」を、以下に記載する。

加速度センサの情報モデルは、TR-1066 に記載の橋梁モニタリングシステムにおける「センサユニット」をアプリケーションで取り扱う際に必要となる情報の要件、関係性、制約、規則、操作などをSDTに従って記述したものである。

本標準では、情報モデルの対象とするドメインを「橋梁モニタリング」とし、そこで定義するデバイスを「加速度センサ」とし、これに関わるモジュールクラスを規定したもので、図4-5 にその概要を示す。



図出典：TTC標準 JJ-300.30「橋梁モニタリング用加速度センサの情報モデル及び低消費電力無線通信における動作」

その他参考資料：TR-1066「橋梁モニタリングのための低消費電力無線通信方式ガイドライン」

図 4-5 橋梁モニタリング用加速度センサの 情報モデル例

4.3 W3C

4.3.1 概要

W3Cとは「World Wide Web Consortium」の略称で、Web技術の標準化を行う非営利団体の名称である。W3Cは1994年に創設され、Webで使用される技術を標準化し、よりスムーズな開発や品質向上を目標に活動が続けられている。W3Cが勧告しているWWW関連の規格は「Web標準」と呼ばれ、Web制作の現場でもHTMLやCSS、DOMに関する仕様としてサービスの品質を決める大きな役割を果たしている。インターネット通信では互換性が重要視されており、Web標準規格に準拠することはこの「互換性を保証する」ことにつながるとされている。

その中で情報モデルに係るものとしてWoT Architectureがあり、物のデータモデルとBinding Templateを用いて連携するWoT Thing Description（以下TDと略す）と言う情報モデルがある。TDは、Linked DataポキャブラリとIoTシステムの機能APIとの間のギャップを埋める構造化データで、それは“物のためのHTML”と見ることができる。TDは、Thingの一般的なメタデータと、Thingの相互作用、データモデル、通信、およびセキュリティメカニズムに関するメタデータを提供している。

通常、TDは、WoTが明示的な拡張ポイントを提供するドメイン固有のメタデータを使用している。ただし、ドメイン固有の語彙は、W3Cの標準化活動の範囲外である。図4-6にW3C WoT Thing Descriptionのデータモデルを示す。

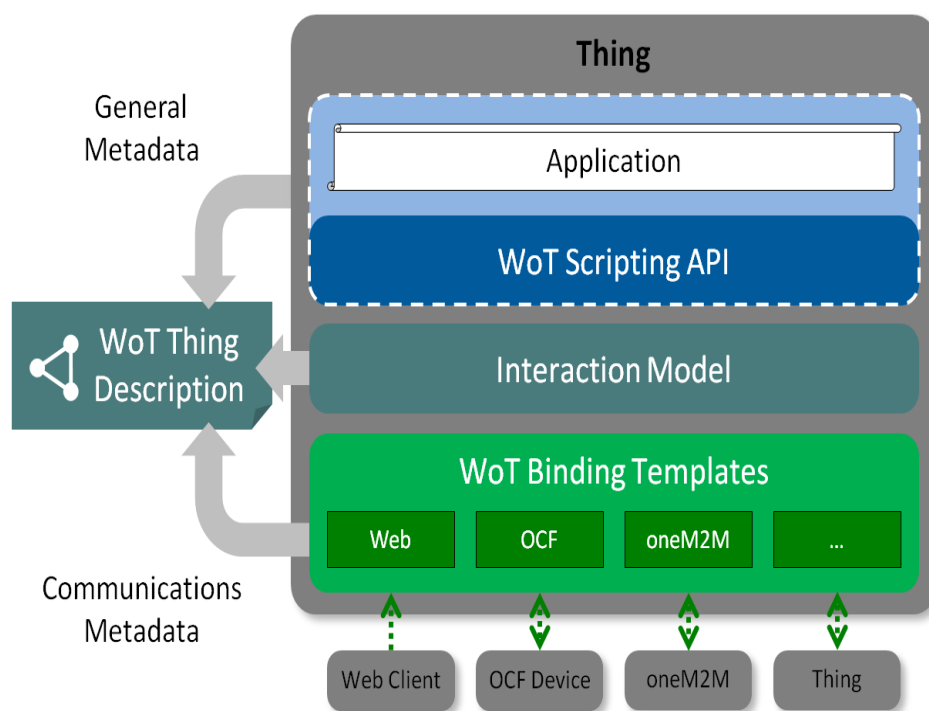


図 4-6 W3C WoT Thing Description のデータモデル

出典： W3Cサイトより引用：The W3C Web of Things (WoT)、
「Figure 10 Conceptual Architecture of the WoT Building Blocks」より
(<http://www.w3.org/TR/wot-architecture/>)

さらにTDはユーザデータ、運用データについては、多くのプロトコルでEntityに対するプロパティとその値として管理される。TDは、Thing自体のセマンティックメタデータ、WoTのプロパティ、アクション、およびイベントを含む narrow-waistの相互作用モデル、データモデルを機械で理解可能にするセマンティックスキーマ、およびWebリンクのフィーチャ間の関係を表現するフィーチャで構成される。

4.3.2 詳細

情報モデルの特徴としては、プロトコル、セキュリティ等の情報を含み、インタラクションモデルとして、property、action、event を定義している。具体的には TD はユーザデータ、運用データについては、多くのプロトコルで Entity に対するプロパティとその値として管理される。 TD は、Thing 自体のセマンティックメタデータ、WoT のプロパティ、アクション、およびイベントを含む narrow-waist の相互作用モデル、データモデルを機械で理解可能にするセマンティックスキーマ、および Web リンクのフィーチャ間の関係を表現するフィーチャで構成される。

図 4-7 にその概要を示す。

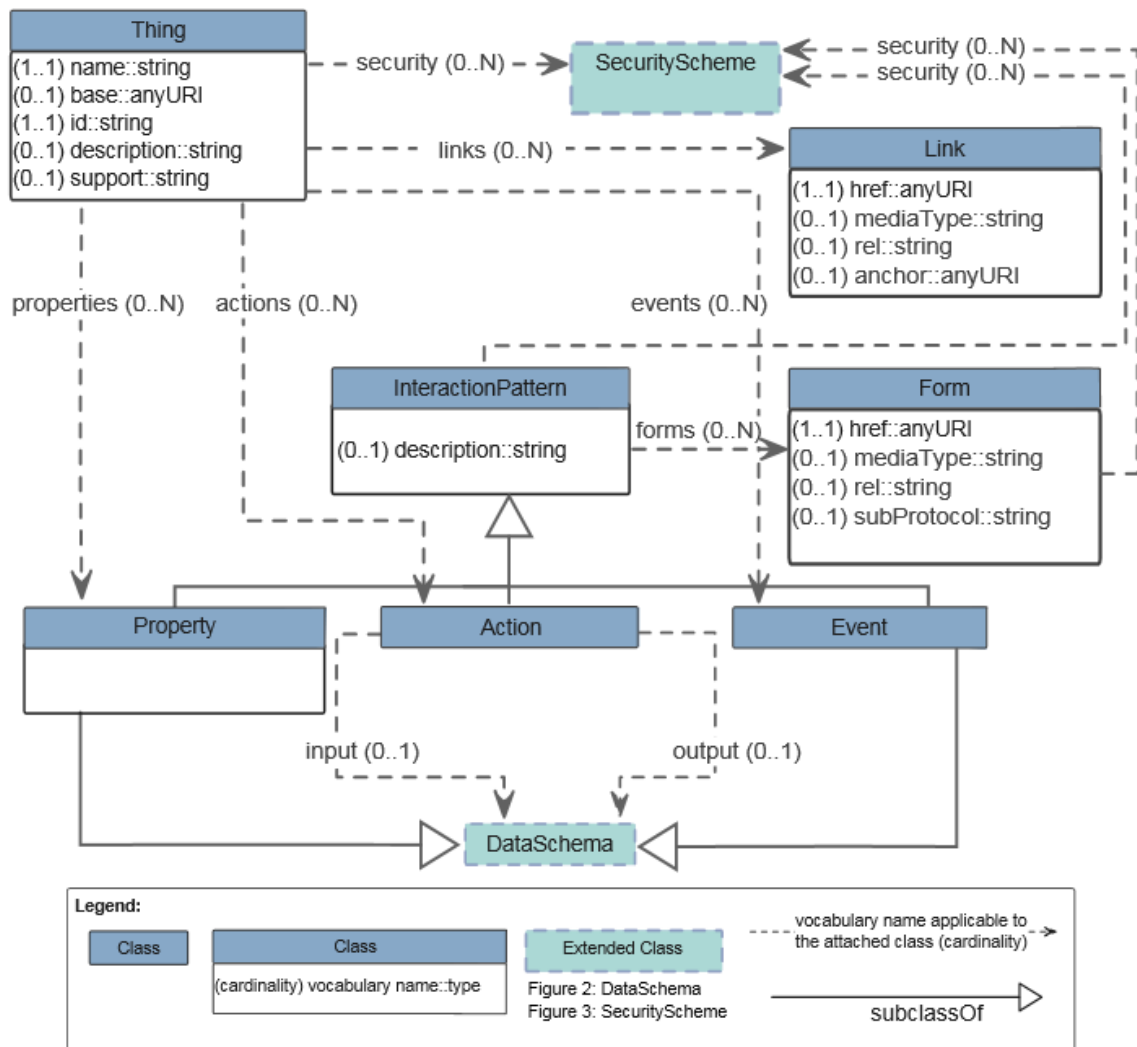


図 4-7 W3C WoT Thing Description のデータモデル

出典： W3Cサイトより引用、Web of Things (WoT) Thing Description、Figure 1 TD core modelより
<http://www.w3.org/TR/2018/WD-wot-thing-description-20181021/#changes>

4.4 FIWARE

4.4.1 概要

FIWAREは次世代インターネットのサービス基盤で、2007年から始まった第7次計画（FP7）におけるICTプロジェクトに「次世代インターネット官民連携プログラム（正式名：the Future Internet Public-Private Partnership（FI-PPP））が元となっている。FI-PPPは、2011年からの5カ年計画で、総額3億ユーロ（約390億円）の予算で遂行された。そしてFI-PPPの多数のプロジェクトの一つに「FI-WAREプロジェクト」がある。

FI-WAREプロジェクトの目標は、EU経済のグローバル競争力の向上と、その核となる次世代インターネットの多岐にわたる分野で活用できる基盤の開発で、この基盤が本連載のテーマであるFIWAREである。

FI-PPPプロジェクトの取り組みは、下表に示すように、多岐に渡った。これらの活動を展開することでFIWAREの普及拡大を目指した。

表 4-1 FI-PPP プロジェクトでの取り組み

プロジェクト名	概要
FIWARE Lab	FIWAREの開発者が利用できるクラウド環境の提供
FIWARE Accelerate	FIWAREを普及促進させるために、FIWAREを活用したアプリケーションを開発したい中小企業やスタートアップ等に対する資金援助
FIWARE Mundus	FIWAREの世界での普及促進
FIWARE iHubs	FIWARE活用コミュニティの立ち上げ支援

こう説明すると「FIWAREはスマートシティのための専用基盤ソフトウェアではないのか」と思われている方もいるかと思うが、実際は、そうではない。FI-PPPでは、表4-2に挙げた8つのカテゴリセクターでFIWAREの実証実験に取り組みながら、スタートアップの育成や、社会・公共分野のアプリケーションを開発し、EU圏の産業の活性化を図ろうとした。

表 4-2 FI-PPP における 8 つのカテゴリセクター

プロジェクト名	カテゴリ	実施場所
FINEST	物流	蘭アムステルダム、トルコ・イスタンブールなど3カ所
OUTSMART	スマートシティ	独ベルリン、英バーミンガム、スペイン・サンタンデルなど5カ所
Instant Mobilty	交通・輸送	伊ローマ、仏ニース、スペイン・トレドなど6カ所
Smart Agrifood	農業	ギリシャ・アテネ、スペイン・マドリッド、独カイザースラウテルンなど8カ所
SAFECITY	街の安全	スウェーデン・ストックホルム、ルーマニア・ブカレスト、フィンランド・ヘルシンキなど6カ所
ENVIROFI	環境	オーストリア・ウィーン、アイルランド・ダブリン、ノルウェー・オスロなど8カ所
FI-CONTENT	メディア・コンテンツ	仏マルヌラヴァレ、英ロンドン、独デュッセルドルフなど6カ所
FINESENY	エネルギー	仏グルノーブル、フィンランド・ヘルシンキ、スウェーデン・ストックホルムなど10カ所

4.4.2 詳細

FIWAREはオープンなソフトウェア部品の集まりで FI-WARE/FI-COREプロジェクトで開発されたFIWAREは、「Generic Enabler」と呼ぶソフトウェアコンポーネント群で構成されています。Generic Enablerは、オープンアーキテクチャーと、新しいサービスインフラストラクチャによるオープンソースのリファレンス実装です。汎用的で再利用可能なビルディングブロック思想に基づいており、プロジェクト期間中に整備されたGeneric Enablerは52種類（当時）に上る（図4-8）。

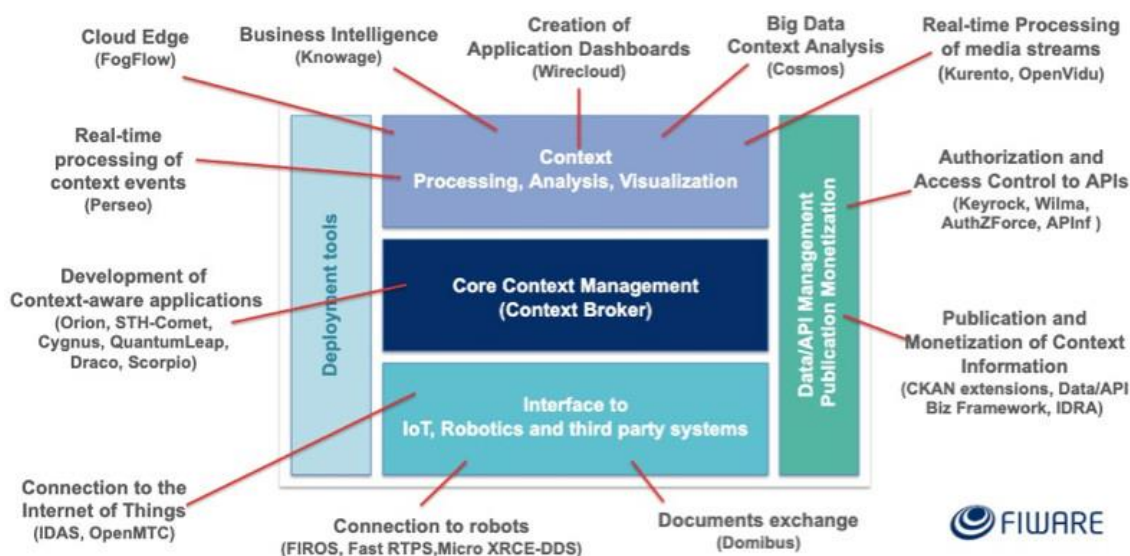


図 4-8 FIWARE の構成図。2020 年 1 月時点の内容

（出典：Github。 <https://github.com/FIWARE/catalogue>）

FIWAREの特長は、様々な既存システムを接続し、データ連携と相互利用を可能とすることである。各GEはOMAが策定したNGSI（Next Generation Service Interface）を採用しており、オープンなインターフェースにより相互接続を行う。

また、FIWAREでは標準の情報モデルを採用している。エンティティ（モノ、データなど）が持つ属性情報とそのメタデータを定義し、分野やサービスにかかわらず共通の情報モデル（コンテキスト）で表現することができる。

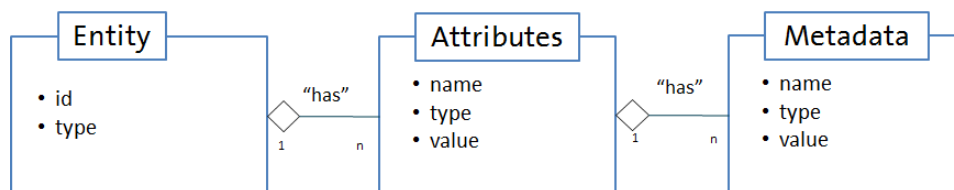


図 4-9 FIWARE の情報モデル

下記に情報モデルの解説に行いて記載する。

(1) Entity :

FIWARE NGSI情報モデルの中心。エンティティはモノ、すなわち、物理的または論理的オブジェクトである。例えば、センサ、人、部屋、システムがエンティティである。エンティティはentity IDを持つ。

NGSIでは型システムにより、エンティティはエンティティ型 (entity type) を持つ。型はエンティティによって表されるモノの種類を記述することを意図しており、idと型の組み合わせにより各エンティティは一意に識別可能である。

(2) Attribute :

エンティティのプロパティである。例えば、現在の自動車の速度をcurrent speedという属性で表現することができ、Attributeは、属性名、属性型、属性値を持つ。

属性名：その属性値がどのような種類のプロパティを表すかを示す

属性型：属性値のNGSI値型を表す。FIWARE NGSIには、属性値を表現する独自の型システムがある。

属性値：実際のデータ。オプションで、メタデータをもち、属性値のプロパティを記述する。

(3) Metadata :

属性値のオプションを示す

4.5 Microsoft Digital Twin Description Language (DTDL)

4.5.1 概要

Microsoft Azure IoTは、マイクロソフト社が提供するIoTソリューションのためのプラットフォームである。Azure IoTでは、IoTデバイスを接続するAzure IoT Hubとクラウド(Azure)からなっており、IoT Hubを通じて得られたIoTデバイスのデータをクラウドで利用できる。

図3-15に示すように、既にクラウドサービスとして、データの分析 (INSIGHTS) や分析結果にもとづいて実行されるビジネスプロセス (ACTIONS) が提供されており、ここにIoT HubがIoTデバイス (THINGS) を仲介することで、デバイス・エッジからクラウドまでのプラットフォームを実現している。

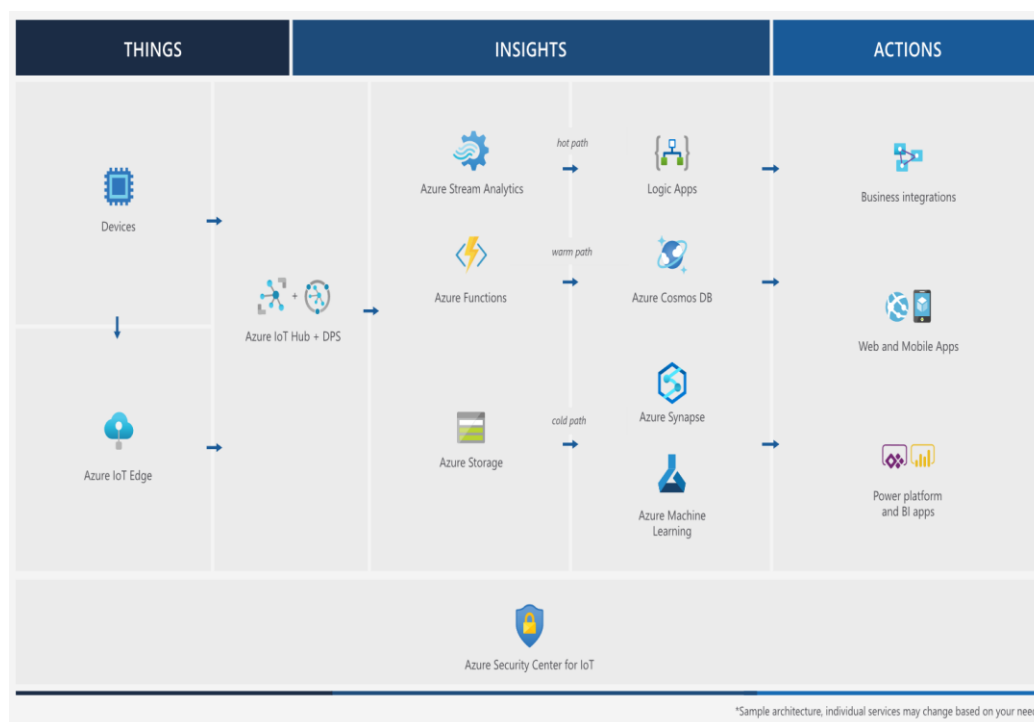


図 4-10 Azure IoT

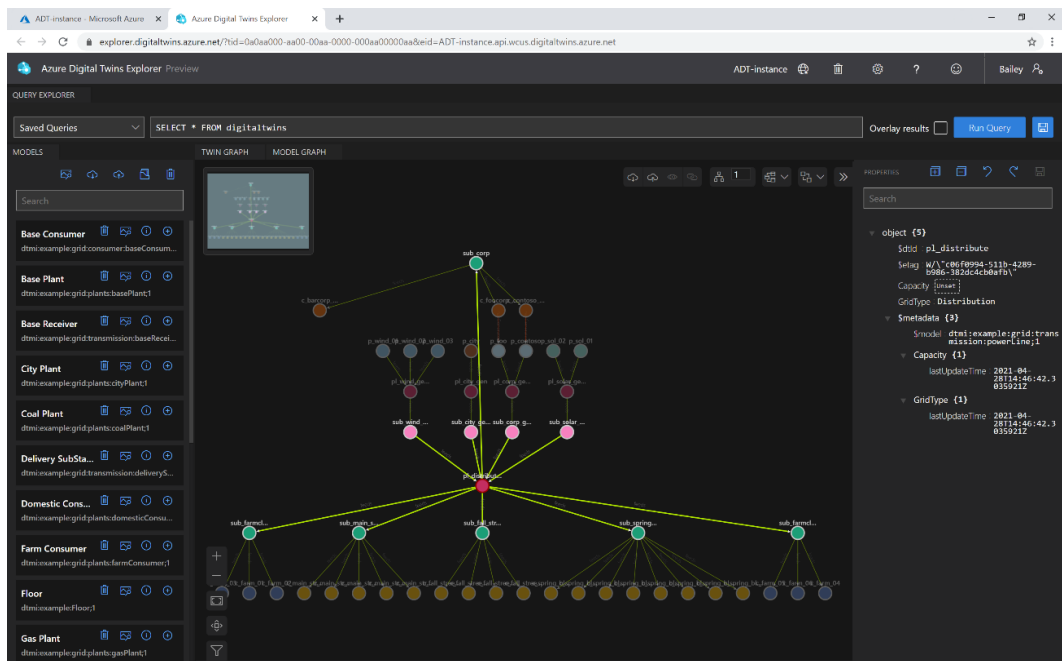


図 4-11 Twin Graph の例

<https://docs.microsoft.com/en-us/azure/digital-twins/overview>

Digital Twin Description Language (DTDL) は、Azureで実現されるDigital Twinsのモデルを定義するための言語である。Azure IoTではIoT HubでDigital Twinsが実現されており、接続されるIoTデバイスがDTDLで表現されて存在する。また、DTDLでは、IoTデバイスなどのリアル空間におけるエンティティをモデルとして表現するほか、その関係をTwin Graphとして表現し、リアル空間全体を定義することができる。

4.5.2 詳細

モデルはインターフェースで定義され、インターフェースには、以下のフィールドがある。

◆ Property

エンティティの状態を表すフィールド。Azureのプロパティは、バックストレージを備えていて、いつでも読み出すことが可能。

◆ Telemetry

テレメトリフィールドはイベントを表す。多くの場合、デバイスのセンサの読み取りに使用される。プロパティと異なり、データはデジタルツインに格納されない。

◆ Component

コンポーネントは、モデルインターフェースを他のインターフェースと組み合わせることができる。例として、Phoneを定義するときに、frontCameraインターフェースを組み合わせることができる。

◆ Relationship

リレーションシップは、あるデジタルツインと他のデジタルツインの関係性を表すことができる。リレーションシップによって、様々なセマンティックな意味を表すことができる。

◆ Commands

デジタルツインに対して実行できるメソッド。リセットコマンドやファンをONまたはOFFに切り替えるコマンドなどが定義される。

語彙については、オントロジを利用する。特定の分野（建造物、IoTシステム、スマートシティ、エネルギーネットワーク、Webコンテンツなど）のモデルのセットがオントロジとして定義されており、それを利用することができる。

5. 各情報モデルの概要：デバイス系

デバイス系に関する規定で、ECHONET Lite、OPC-UA、OCF、ZigBee、BACnet、LonWorks、KNXらがそれに該当する。以下に詳細を述べる。

5.1 Echonet

5.1.1 概要

ECHONET コンソーシアムは、白物家電、住宅設備機器を中心としたホームネットワークシステム実現に向けて1997年から活動を行っている団体である。2018年10月現在、幹事会員7社、幹事準会員42社、一般会員165社、一般準会員36社から構成されている。

ECHONETとECHONET Liteがあり後者が新しい規格である。ECHONET Liteは、センサ類、白物家電、設備系機器など省リソースの機器をIoT化し、エネルギーマネジメントやリモートメンテナンスなどのサービスなどを実現するための通信仕様で、通信仕様や各機器の制御コマンドを共通仕様とすることで、マルチベンダー環境でのシステム構築を実現しており、下記特徴がある。

- ・各種既存の標準伝送メディアの利用が可能
- ・家庭・中小ビル・店舗向け等、広範囲な機器に対応
- ・重点機器のさらなる相互接続性の向上
- ・クラウド上のサービスとの連携が可能

ECHONET Liteの規格類の構成を図5-1に示す。

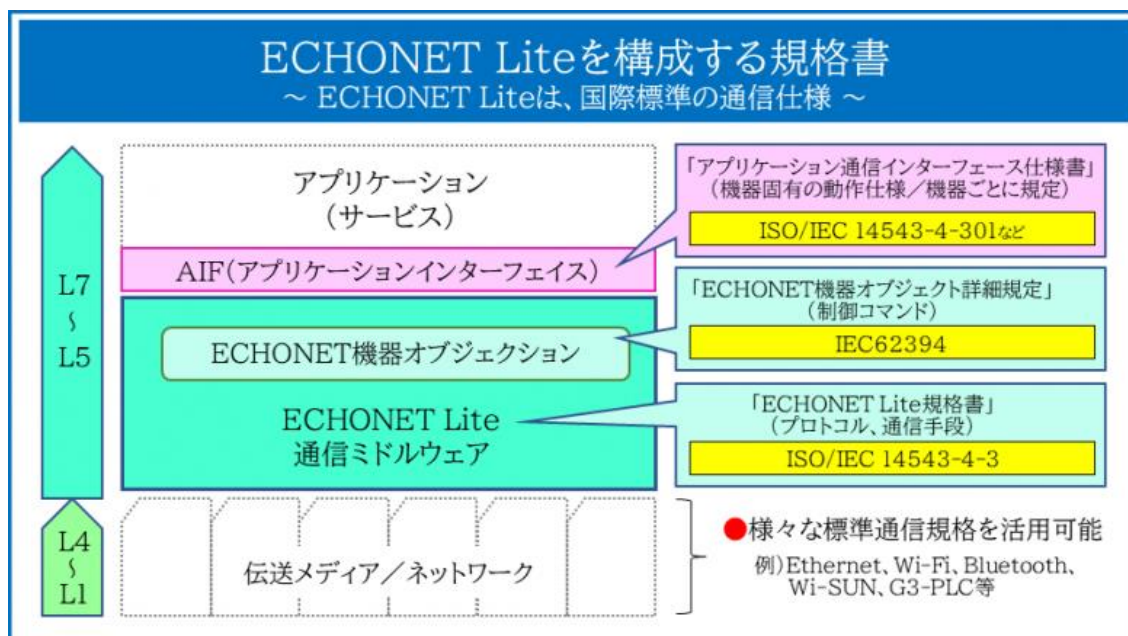


図5-1 ECHONET Liteの規格類の構成

(出典：ECHONETホームページ、「ECHONET Lite規格の特長と概要」より
<https://echonet.jp/about/features/>)

5.1.2 詳細

ECHONET Lite は、センサ類、白物家電、設備系機器など省リソースの機器を IoT 化し、エネルギーマネジメントやリモートメンテナンスなどのサービスを実現するための通信仕様である。

通信仕様や各機器の制御コマンドを共通化することで、マルチベンダ環境でのシステム構築を実現している。下記に特徴を示す。

① 各種既存の標準伝送メディアが利用可能

ECHONET Lite で規定している範囲は、OSI 参照モデルにおける第 5 層以上で、下位の通信仕様には依存しない。通信アドレスは、IP アドレスや伝送メディアの MAC アドレスなどを利用して送信先を指定する。したがって、サービスやアプリケーションなどの様々な要件に基づいて、市場からリーズナブルな伝送メディアを選択することが可能である。

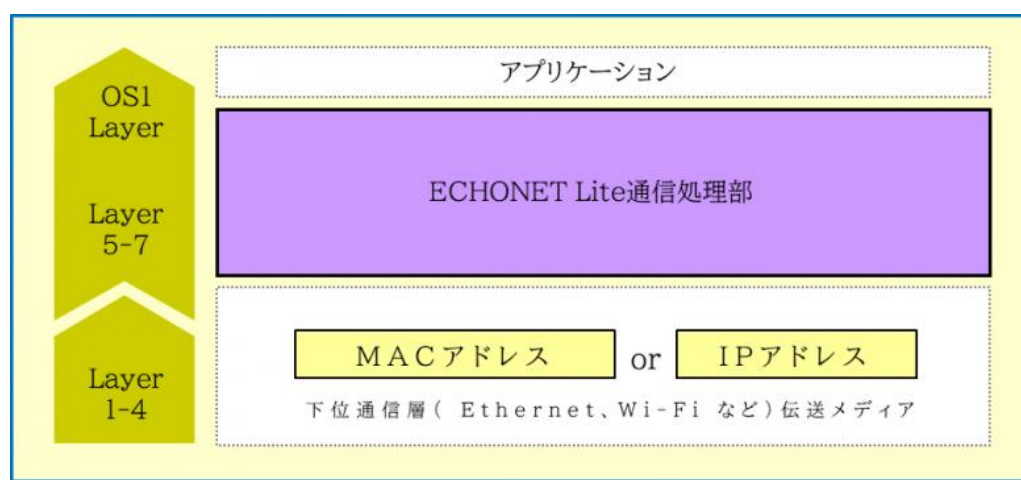


図 5-2 OSI 参照モデル

(出典：ECHONETホームページ、<https://echonet.jp/about/features/>)

② 家庭・中小ビル・店舗向けなど広範囲な機器に対応

1997年のエコーネットコンソーシアム設立以来、継続して機器の制御コマンドの定義を拡充している。エコーネットコンソーシアムでは、様々な機器が持つ多様なプログラム群やデータ群をモデル化することで、機器オブジェクトとして制御コマンドを定義している。

現在では、センサ類、エアコン、照明などの省エネ機器、蓄電池、HP給湯機などの蓄エネ機器、太陽光発電、燃料電池などの創エネ機器、各種スマートメータなどの計測機器、業務用パッケージエアコン、ショーケースなどの業務用機器の定義を進めており、既に110種類以上の機器の詳細な機器オブジェクト（制御コマンド）を定義している。なお、機器オブジェクトは、下記に示すように用途別に7種類のグループ（クラスグループ）に分類されている。

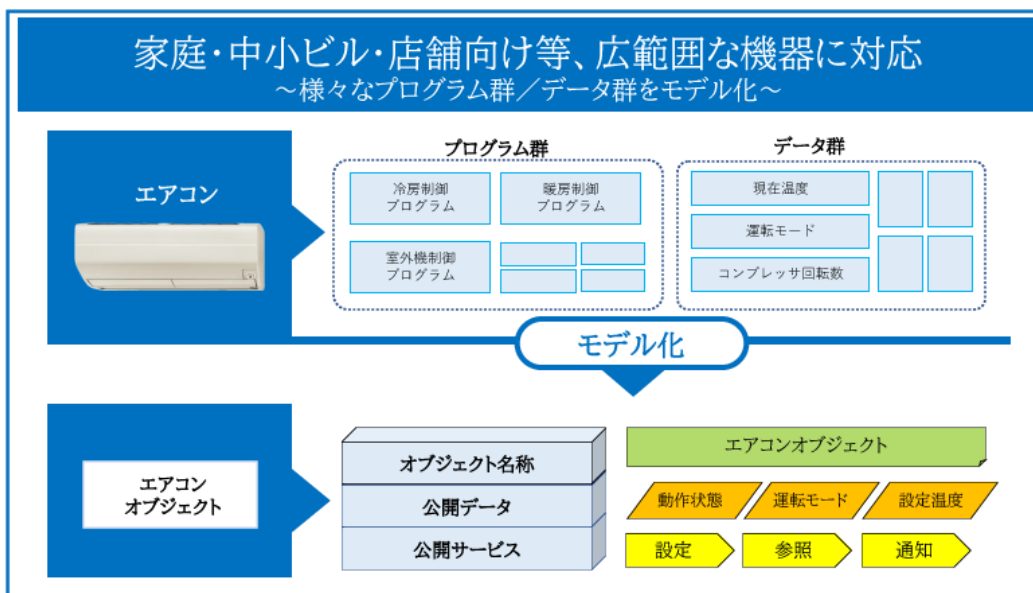


図 5-3 機器オブジェクトとして制御コマンド

(出典：ECHONETホームページ、<https://echonet.jp/about/features/>)

表 5-1 データモデル例

クラスグループコード	クラスグループ	機器例
0×00	センサ関連機器	火災センサ、人体検知センサ、温度センサ、CO2センサ、電流量センサ、etc.
0×01	空調関連機器	エアコン、扇風機、換気扇、空気清浄機、ホットカーペット、石油ファンヒーター、業務用パッケージエアコン、etc
0×02	住宅・設備関連機器	電動ブラインド、電動カーテン、温水器、電気錠、スマートメータ、太陽光発電、蓄電池、燃料電池、一般照明、単機能照明、非常灯、照明システム、拡張照明システム、etc.
0×03	調理・家事・冷設関連機器	電子レンジ、食器洗い機、食器乾燥機、洗濯機、衣類乾燥機、業務用ショーケース、etc.
0×04	健康管理関連機器	体重計、体脂肪計、体温計、血圧計、血糖値計、etc.
0×05	管理・操作関連機器	コントローラ、スイッチ (HA機器)、etc.
0×06	AV関連機器	TV、ディスプレイ、etc.

(出典：ECHONETホームページ、<https://echonet.jp/about/features/>)

5.2 OPC-UA

5.2.1 概要

OPC ホームページ (<https://jp.opcfoundation.org/about/what-is-opc/>) によると、OPC は、産業オートメーション分野やその他業界における、安全で信頼性あるデータ交換を目的とした相互運用を行うための標準規格である。

OPCはプラットフォームから独立し、多くの製造ベンダーのデバイス間で、シームレスな情報の流れを確保し、OPC Foundationは、この標準規格の開発と維持に責任を負っている。また、OPC標準規格は、業界の製造ベンダー、エンドユーザー、そしてソフトウェア開発者が開発した一連の仕様であり、これら仕様は、リアルタイムデータのアクセス、アラームとイベントの監視、履歴データやその他アプリケーションのアクセスを含み、クライアントとサーバー間に加え、サーバーとサーバー間のインターフェースを定義している。

この標準規格は、1996年に初めて発表され、当初、OPC標準規格は、Microsoft® Windows®オペレーティング・システムに限られていたが、現在、OPC Classic として知られるそれら仕様は、製造、建設、

オートメーション、石油・ガス、再生可能エネルギー、ユーティリティ等、数々の業界で広く適応されている。

その後、次世代のOPC 技術として、OPC UA (統合アーキテクチャ) は安全で信頼性あるプラットフォーム非依存の相互運用性に向け大きく進化し、工場やプロセス制御装置から大規模な上位情報システムへ情報転送をするために設計された。

5.2.2 詳細

OPC-UA (IEC 62541、OPC Unified Architecture:統合アーキテクチャ) は次世代の OPC 技術として、安全で信頼性あるプラットフォーム非依存の相互運用性に向け大きく進化したものである。

以下、OPC ホームページからの抜粋を記載する。

OPC UA は、工場やプロセス制御装置から大規模な上位情報システムへ情報転送をするために設計されており、Industrie 4.0 の通信の情報交換として重要な役割を持っている。図 5-4 に OPC UA 階層モデルを示す。

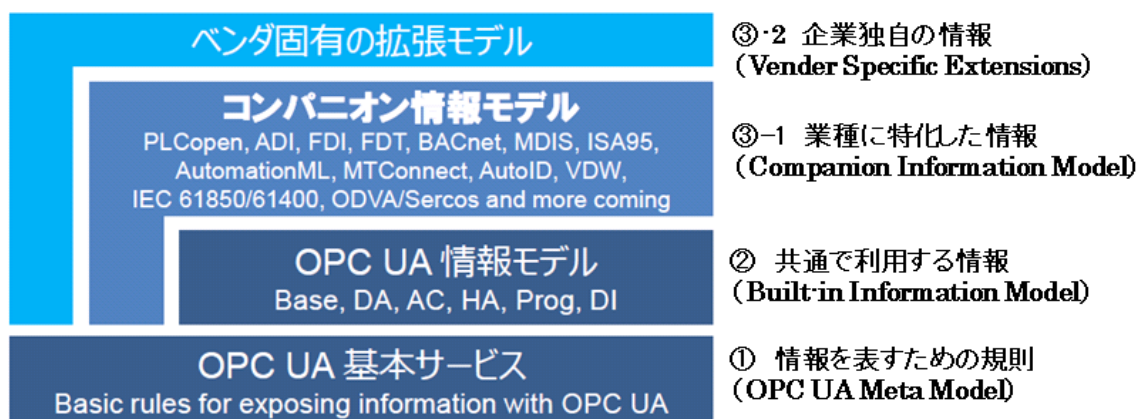


図 5-4 OPC UA 階層モデル

出典： OPC UA コラボレーション取組とそのユーザー価値
第一のカギ：伝えるための枠組み=情報モデルより

(https://jp.opcfoundation.org/wp-content/uploads/sites/2/2017/12/OPCDay2017_07_collaboration.pdf)

図中の①~③は具体的には、以下の意味を示す。

- ① モデル定義基盤 (Meta Model) の確立
- ② 基本モデル構成要素 (Built-in Information Model) の定義
- ③ 拡張モデルの拡充
 - 1 協調領域 (Companion Information Model)
 - 2 競争領域 (Vender Specific Extensions)

OPC UA 階層モデル別の見方をすると、図 5-5、5-6 に示す構成となる。

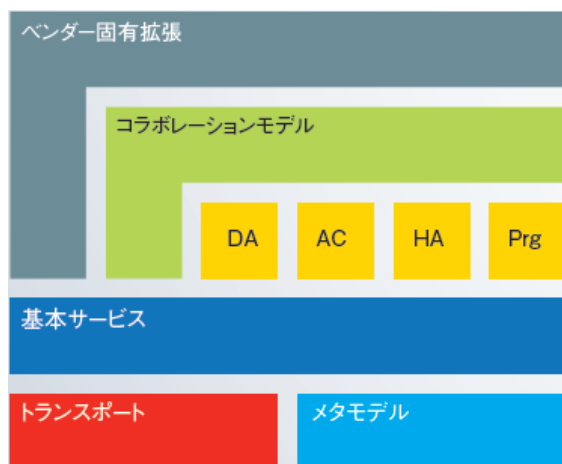


図 5-5 OPC UA 階層モデル

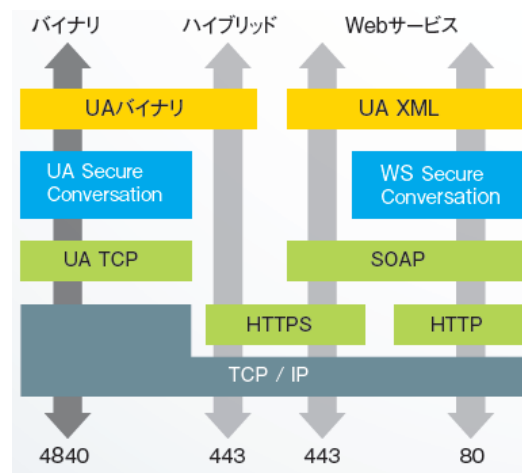


図 5-6 OPC UA トランスポート・プロファイル

出典： OPC ホームページ、 OPC UA 技術の詳細より

<https://opcfoundation.org/wp-content/uploads/2015/12/OPC-UA-Interoperability-For-Industry4-and-IoT-JP-v3.pdf>

具体的には、以下の構成となる。

- ◆ トランスポート – OPC UA アプリケーション間のデータ交換機構に対応：異なる要求事項に対応する様々なトランスポート・プロトコル（速度とスループットへの最適化 = UA TCP と UA バイナリ、ファイアーウォール親和性 = HTTP+ SOAP）。
- ◆ メタモデル – OPC UA で情報モデルを発行するためのルールと基本コンポーネントを指定：様々な基本ノードと基本タイプも含む。
- ◆ サービス – 情報を提供するサーバーと情報を使用するクライアント間のインターフェースを構成：情報モデルは階層化アプローチに従っている。

以下、図中の具体的な記載について記載する。

<汎用 OPC UA 情報モデル>

OPC UA 階層モデルに示した情報モデルは、一般的な有効情報（例えば、アラームやオートメーションデータ）のモデルとして、すでに OPC UA によって定義されている。一般定義をさらに特化したその他の情報モデルはここから派生する。そのため、一般モデルに対してプログラムされたクライアントは、ある程度特化したモデルも処理することができる。

① データアクセス (DA)

データアクセス、略して DA は、リアルタイムデータ（すなわち、基本的な産業用または業務用プロセスデータの現在の状態と挙動を表すデータ）のモデリングを記述する。これにはアナログ・ディジタル変数、工学単位、品質コードの定義が含まれる。データの出所は、センサ、コントローラ、ポジションエンコーダーなどで、これらをデバイスの I/O に直接接続することも、リモートデバイスの直列接続とフィールドバスで接続することも可能である。

② アラーム&コンディション (AC)

この情報モデルは、状態（ダイアログ、アラーム）の処理方法を定義する。状態の変化は、イベントを起こし、クライアントはこうしたイベントを登録し、利用可能な関連値の中から、イベントレポート（例えば、メッセージテキスト、ACK 挙動）に含めて受け取りたいものを選択することができる。

③ ヒストリアアクセス(HA)

HAにより、パラメータとイベントの履歴データにアクセスでき、これらのデータの読み取り、書き込み、変更が可能である。データはデータベース、アーカイブ、その他のストレージシステムに配置することができ、幅広い集計処理機能により、サーバーで前処理を行うことができる。

④ プログラム (PRG)

「プログラム」は、バッチ処理の演算や操作のような複雑なタスクを表す。各プログラムはステートマシンで示され、状態遷移は、クライアントに対してメッセージを起こす。

<統合アドレス空間モデル>

次にオブジェクトの記載方法の例について記載する。オブジェクトモデルは、生産データ、アラーム、イベント、履歴データをひとつの OPC UA サーバーに統合可能にしている。これにより、例えば、温度測定機器を温度値、アラーム・パラメーター、対応するアラーム限度と共にオブジェクトとして示すことができる。図 5-7 にその例を記載する。

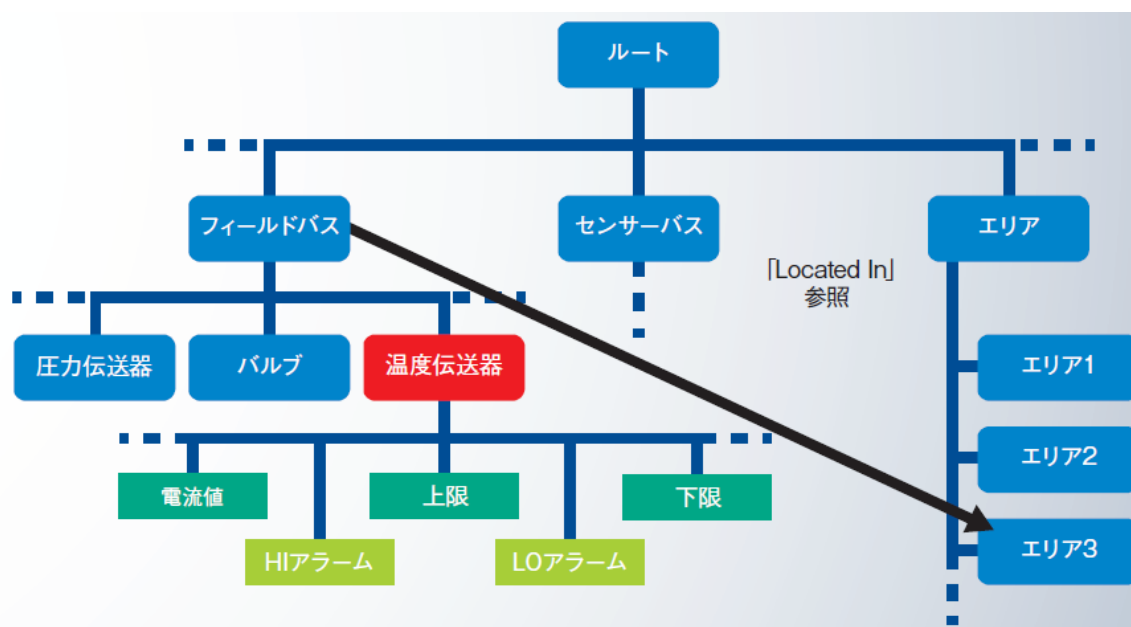


図 5-7 OPC UA 一貫性のあるアドレス空間

出典： OPC ホームページ、統合アドレス空間モデルより、

<https://opcfoundation.org/wp-content/uploads/2015/12/OPC-UA-Interoperability-For-Industrie4-and-IoT-JP-v3.pdf>

OPC UA は異なるアドレス空間とサービスを統合および標準化し、OPC UA アプリケーションはひとつのインターフェースだけでナビゲーションできるようになる。

OPC UA のアドレス空間は、階層的に構成され、クライアントとサーバーの相互運用性を促進する。最上位レベルはサーバー用に標準化され、アドレス空間のすべてのノードには階層経由で到達することができ、ノード間に相互参照を付けられ、アドレス空間が結束したノードのネットワークを形成する。

OPC UA のアドレス空間には、インスタンス（インスタンス空間）だけでなく、インスタンスタイプ（タイプ空間）も含まれている。

OPC UA は、Industrie 4.0 の情報レベルの相互運用性に関する要求事項を満たす、成熟した規格で

ある。OPC UA は、包括的な情報モデル（What）を公開し、個別に開発されたアプリケーション間で複雑なデータを交換するために、プロトコルとサービス（How）を提供する。

<テクノロジー固有の情報モデル>

制御／オートメーションを扱う標準化委員会は、テクノロジー固有の情報モデルを作成している。例えば、IEC61804 (EDDL)、ISA SP 103（フィールドデバイスツール）、ISA-S88、ISA-S95、IEC-TC57-CIM などがある。これらは、特定の知識分野の単位、関係、ワークフローの記述を標準化しているため、重要な仕様である。

OPC Foundation は、新しい規格をゼロから開発するため、他の組織と意欲的に協力しており、これらの組織の情報モデルを OPC UA（コンパニオン規格）にマッピングするルールは、共同作業部会で規定されている。図 5-8 に IEC62541: OPC UA 仕様を示す。

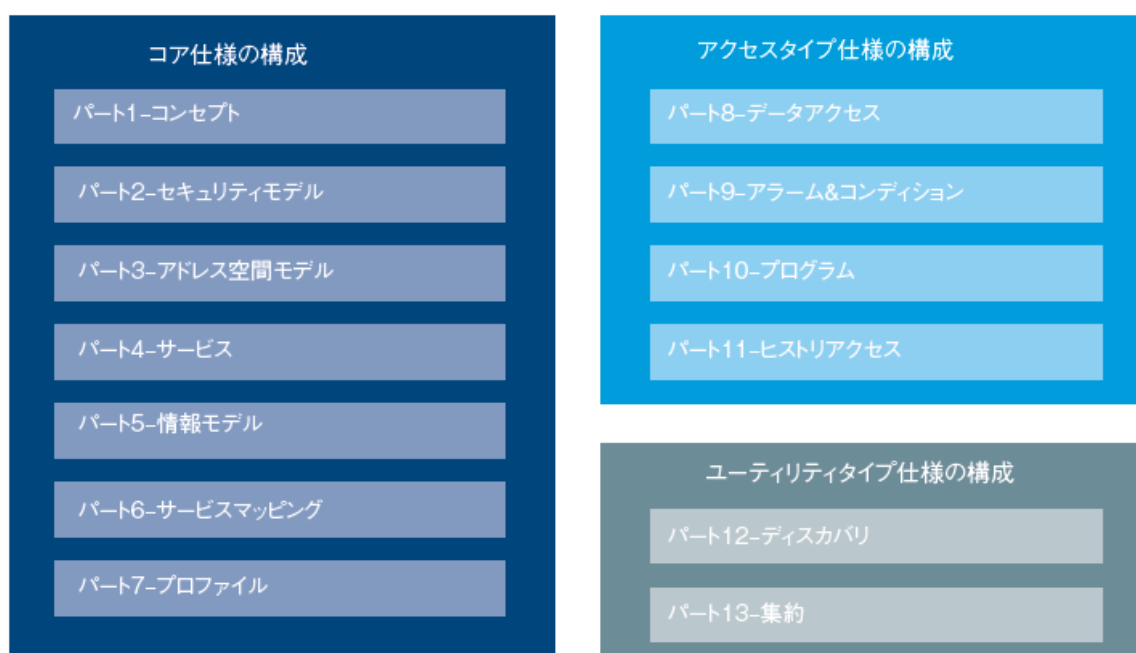


図 5-8 IEC62541: OPC UA 仕様

出典： OPC ホームページより、

<https://opcfoundation.org/wp-content/uploads/2015/12/OPC-UA-Interoperability-For-Industry4-and-IoT-JP-v3.pdf>

現在作成中のコンパニオン規格 は以下の通りである。

- OPC UA for Devices (IEC 62541-100)
- OPC UA for Analyser Devices
- OPC UA for Field Device Integration
- OPC UA for Programmable Controllers based on IEC61131-3
- OPC UA for Enterprise and Control Systems based on ISA 95
- OPC UA for Machine Tool Connectivity (MTConnect)
- OPC UA for AutoID (AIM)

- OPC UA for BACnet (Building Automation)

<テクノロジー固有の情報モデル>

その他の取り組みと具体的なデータモデルの例について、以下に記載する。

- コラボレーションの取り組み
 - OPC が適用 される 市場・業界団体と協力し適用するための仕様（コンパニオン仕様）
 - 適用事例（ユースケース）や適用のためガイドラインを創出
- 工場自動化分野:
 - PLCopen (PLC)
 - FDT-Group (ソフトウェアインターフェース)
 - PROFIBUS, EtherCAT, CC-Link, IO-Link など(産業用ネットワーク)
 - OMAC (包装機)
 - EUROMAP (成形機)
- ビルオートメーション:
 - BACnet (ビルオートメーション)
- オイル・ガス:
 - MDIS (MCS-DCS 間通信)
- IoT、セキュリティ技術:
 - OpenFog (IoT)
 - M2M Alliance (セキュリティ)

さらに、5 件のコンパニオン仕様・ガイドラインが会員に公開中。(2017/12/10 現在)

<具体的な情報モデル>

OPC UA は、図 3-24 にあるように、関連する型システムを含む一般的なオブジェクトモデルを定義している。このデータモデルに加えて、すべての物理システムを OPC UA に準拠するモデルに変換して OPC UA サーバーで表現する方法を記述するための規則が定義されている。このメタモデルを使用して、あらゆる種類のデバイス、機能、およびシステム情報を記述できる。基本型システムは、オブジェクト間の関係、いわゆる参照、および多重継承をサポートしている。したがって、それは現代のオブジェクト指向プログラミング言語と比較することができる。基本モデルは、参照型とデータ型だけでなく、オブジェクト型と変数型も提供する。このモデルに基づいて、OPC UA はメタデータとセマンティクスを含むあらゆる種類のデータを表現できる。

◆ UA モデル

OPC UA データモデルは、UA 情報モデルの基盤を形成する。特殊なモデルであるため、データアクセス、アラームと状態、履歴アクセス、プログラムなどの特定の機能を追加することで基盤を拡張します。

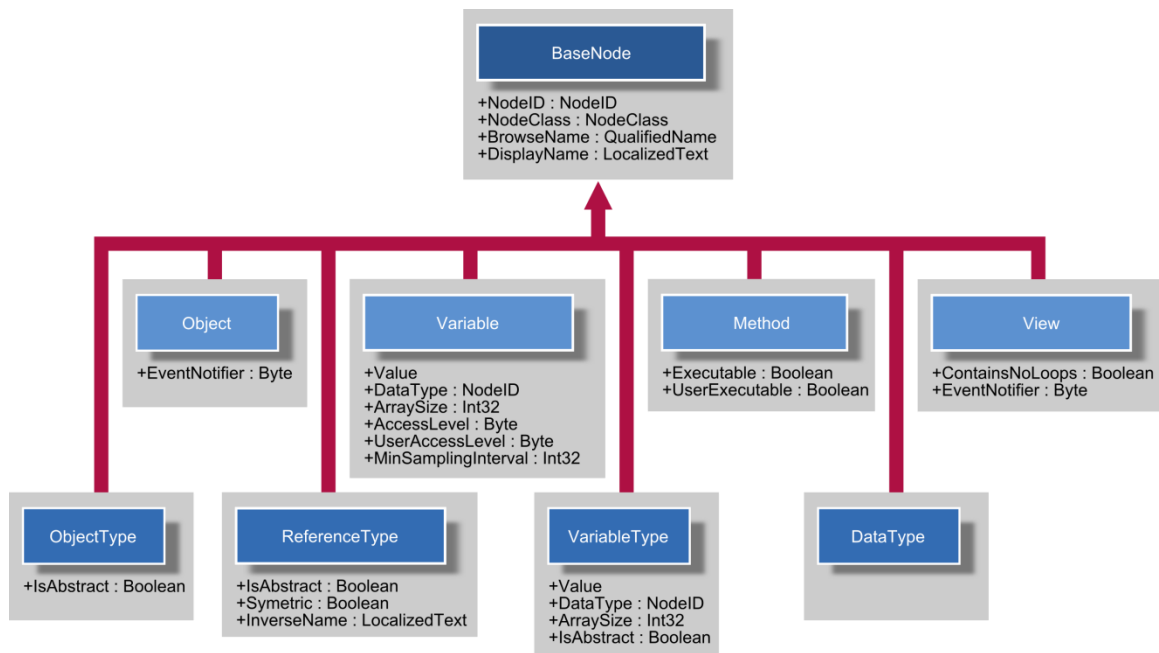


図 5-9 IEC62541: OPC UA 情報モデル

出典：ascolab 社資料より、

(<http://www.ascolab.com/en/technology-unified-architecture/meta-model.html>)

5.3 OCF

5.3.1 概要

OCF (Open Connectivity Foundation) は2016年2月にできた新しい組織である。OCFはインテルやサムスン、デルなどが中心となって2014年7月に設立したOpen Interconnect Consortium (OIC) と、2013年2月にQualcommが中心となって設立したAllseen Allianceの主力メンバーであるマイクロソフト、クアルコム、エレクトロラックスの3社がボードメンバーとして参加して組織された。OCFの2016年12月のプレスリリースによれば、新たに68社が加入して会員数が300社以上になったとのことである。図5-10 に統合の流れを記載する。

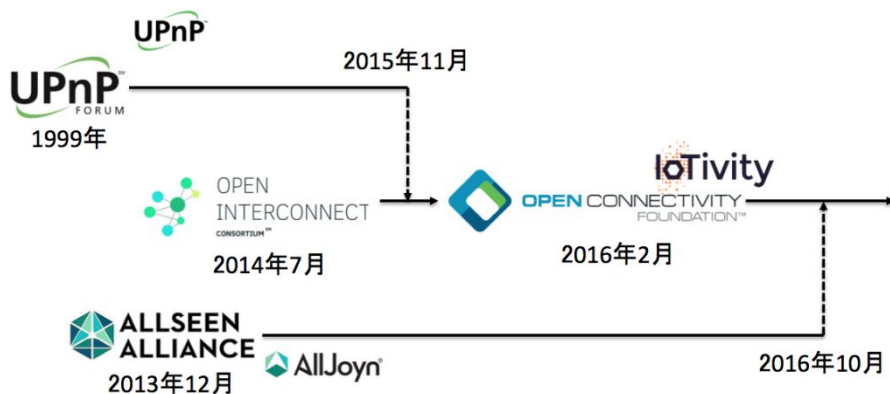


図 5-10 OCF の変遷

出典：Thech Blog (フューチャーアーキテクト開発者ブログ) 図1より引用

(<https://future-architect.github.io/articles/20161209/>)

5.3.2 詳細

OIC Core Smart Home は Samsung、Intel などを中心となり設立した Open Connectivity Foundation (OCF)

のデータモデルで詳細は以下の通りである。図 5-11 に詳細を示す。

- OIC は OS、トランスポート、ベンダーに依存せず全てのモノをつなぐための IoT ソフトウェアフレームワーク
 - ✓ 現在は OCF (Open Connectivity Foundation) で OCF = OIC + UPnP Forum
- データモデルとデータプロトコル (メッセージング) を明確に区別し
 - ✓ データモデルは RESTful リソースモデルで、Create, Read, Update, Delete, Notify (CRUDN) で通信
 - ✓ データプロトコルは、CoAP/HTTP/MQTT 等で CRUDN を実現
 - ✓ リソースモデルは、物理デバイスの抽象表現

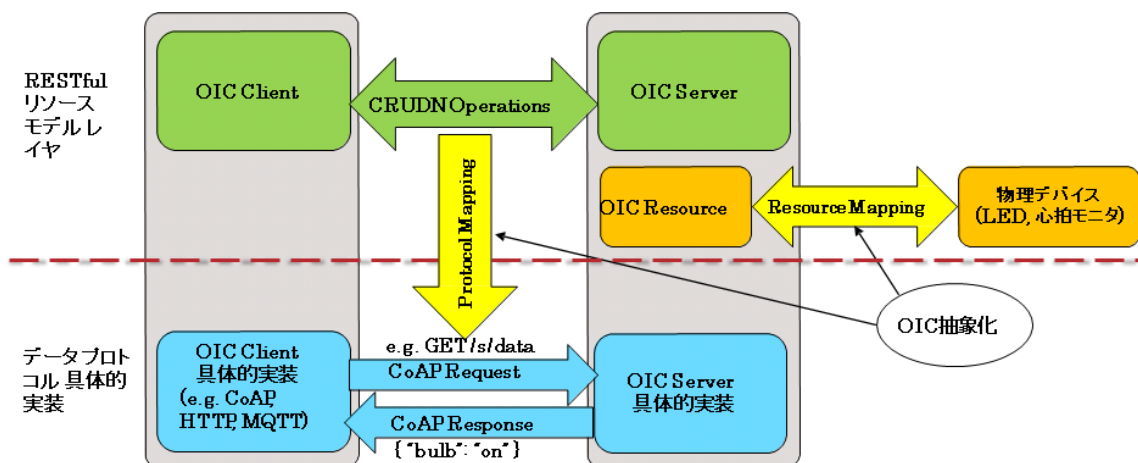


図 5-11 OCF のデータモデル

出典 : OCF_Core_Specification_v2.0.0、Figure 1: Architecture - concepts を和訳化

(https://openconnectivity.org/specs/OCF_Core_Specification_v2.0.0.pdf)

最新版は V2.0.1)

(I) プロトコル構成

下図にOCFのプロトコル構成を示す。OCFでは、機器間で通信するための機能ブロックとして図5-12のようなプロトコル構成を取っている。CBORは「The Concise Binary Object Representation」の略でIETF RFC 7049で定義されており、設計目標が、非常に小さいコードサイズ、かなり小さいメッセージサイズ、およびバージョンネゴシエーションを必要としない拡張性の可能性を含むデータフォーマットで、JSONデータモデルに基づく簡潔なバイナリオブジェクト表現である。

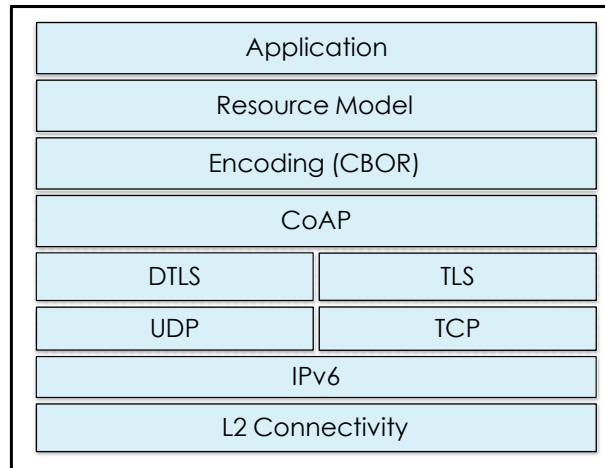


図 5-12 OCF Protocol Stack

出典：OCF Specification Introduction and Overview

https://openconnectivity.org/specs/OCF_2.0_Specification_Overview.pdf

(2) 具体的なデバイス制御

具体的には機器タイプ毎に共通する操作をResourceとして下記の様に定義している。

- ・一般的な操作について、色々な機器に対して定義を用意
- ・対応機器はこれを元に動作・応答を実装
- ・メーカー独自操作等はオプションで可能
- ・Resourceとして値の型、範囲、Read-only、collectionなどのプロパティを用意

下記に具体的なデバイスタイプの一例を表5-2に示す。

表 5-2 必須リソースを含む、デバイスタイプの一例

Device Name (informative)	Device Type ("rt") (Normative)	Required Resource name	Required Resource Type
Active Speaker	oic.d.speaker	Binary Switch	oic.r.switch.binary
		Audio Controls	oic.r.audio
Air Conditioner	oic.d.airconditioner	Binary Switch	oic.r.switch.binary
		Temperature	oic.r.temperature
Air Purifier	oic.d.airpurifier	Binary Switch	oic.r.switch.binary
Air Quality Monitor	oic.d.airqualitymonitor	Air Quality Collection	oic.r.airqualitycollection

出典：(株)ピクセラ社資料より、

<https://www.slideshare.net/PIXELAcorporation/what-is-ocf>

さらに詳細なデバイスの制御例として電球の制御について図5-13に記載する。

- ・サンプル概要：バイナリスイッチ、明るさリソースを備えたスマートライトデバイス
- ・機種[ドメイン定義]：ライトデバイス (oic.d.light)
- ・関連資料：必須コアリソース：oic/res、oic/p、oic/d

- ・ 必須セキュリティリソース (図示せず)
- ・ 機器固有リソース：バイナリスイッチ (oic.r.switch.binary) 、
- ・ この例では、他のオプションのリソースを公開している。
Brightnessリソース (oic.r.light.brightness)

Device Title	Device Type	Associated Resource Type	M/O
Light	oic.d.light	oic/res (oic.wk.res)	M
		oic/p (oic.wk.p)	M
		oic/d (oic.d.light)	M
		Binary switch (oic.r.switch.binary)	M
		Brightness (oic.r.light.brightness)	O

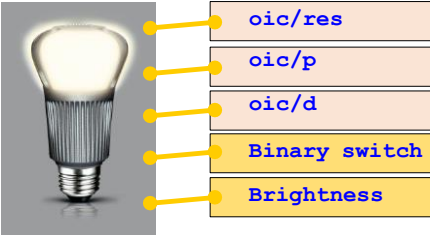


図 5-13 Example: Smart light device

出典：OCF Specification Introduction and Overview

https://openconnectivity.org/specs/OCF_2.0_Specification_Overview.pdf

5.4 ZigBee

5.4.1 概要

ZigBee は、近距離無線ネットワークの世界標準規格の一つであり、2017年9月で37ヶ国、400社以上がメンバー、2,200以上の認定商品がある。

技術的には、信頼性のある、低消費電力・低コストの無線通信として2001年からZigBee Allianceにて研究が進められてきた。サービスプロバイダー、インストーラ、小売業者に適した技術で、電池駆動可能な超小型機器への実装に向いている。

基礎部分の（電氣的な）仕様は IEEE 802.15.4 として規格化されている。論理層以上の機器間の通信プロトコルについては「ZigBee Alliance」が仕様の策定を行っている。なお、ZigBee の名称はミツバチ (Bee) がジグザグに飛び回る行動にちなんで名付けられたとされている。

5.4.2 詳細

ZigBeeは幅広いスマートホームデバイスのための言語であり、企業はホームモニタリング、エネルギー管理、冷暖房、セキュリティ、およびコンビニエンスデバイスの統合エコシステムを提供できる。

ZigBeeがカバーする範囲は、OSI参照モデルのネットワーク層以上の部分で、物理層/MAC層についてはIEEE802.15.4を採用している。ZigBeeはPAN (Personal Area Network) に分類されるが、ネットワークトポロジーとして、スター、ツリー (木構造)、メッシュをサポートすることで市場の様々な要求に応えることができる。

具体的なデータモデルとしては、ZigBee Cluster Libraryがある。ZigBeeでは「プロファイル」という形で交換するデータの形式と解釈を細かく定義して運用している。パケット毎にそのパケットがどのプロファイル用なのかということを明示して送信する。(図5-14)

具体的なプロファイルは各種の仕様を持っており、それらの例の一部を以下に記載する。



図 5-14 ZigBee の 7 層モデル

出典： ZigBee入門「データ送受信」より引用

(<http://www.skyley.com/wiki/index.php?ZigBee%E5%85%A5%E9%96%80>)

具体的な制御対象例は以下のとおりである。

- Home Automation 1.2
- Smart Energy 1.1b
- Smart Energy 1.2
- Smart Energy 1.2a
- Smart Energy 1.2b
- Telecommunication Services 1.0
- Health Care 1.0
- RF4CE – Remote Control 1.0
- RF4CE – Input Device 1.0
- Remote Control 2.0
- Light Link 1.0
- IP 1.0
- Commercial Building Automation 1.0
- Gateway 1.0
- Green Power 1.0 (Optional battery-less remote control feature of Zigbee 2012)

一例としてのON/OFFの制御モデルを図5-15に示す。

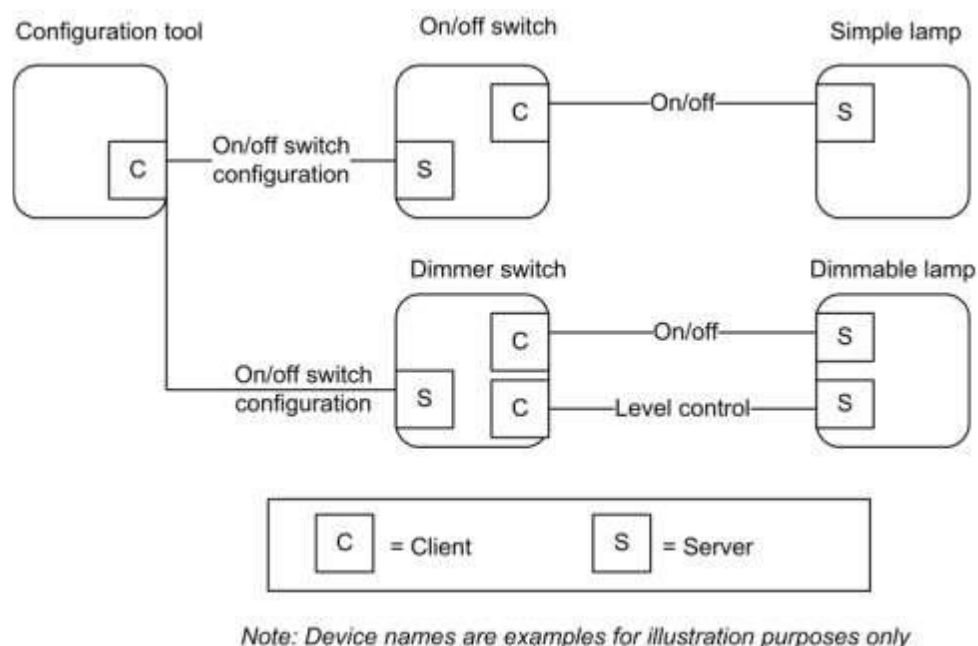


図 5-15 Typical Usage of On/Off and Level Control Clusters

出典： ZigBee Cluster Library Specification Revision 6

Figure 3-2. Typical Usage of On/Off and Level Control Clustersより引用

(<http://www.zigbee.org/~zigbeeor/wp-content/uploads/2014/10/07-5123-06-zigbee-cluster-library-specification.pdf>)

表 5-3 制御コマンド例

ID	Name	Description
0x0006	On/Off	Attributes and commands for switching devices between 'On' and 'Off' states.
0x0007	On/Off Switch Configuration	Attributes and commands for configuring on/off switching devices
0x0008	Level Control	Attributes and commands for controlling a characteristic of devices that can be set to a level between fully 'On' and fully 'Off'.

出典： ZigBee Cluster Library Specification Revision 6,

Table 3-3. On/Off and Level Control Clustersより引用

(<http://www.zigbee.org/~zigbeeor/wp-content/uploads/2014/10/07-5123-06-zigbee-cluster-library-specification.pdf>)

5.5 BACnet

5.5.1 概要

BACnet とは、ビルディングネットワークのための通信プロトコル規格で、正式名称は「Building Automation and Control Networking Protocol」である。1995年にASHRAE（米国冷暖房空調工業会）によって制定された「ANSI/ASHRAE Standard 135-1995」をもとに、2003年にISO:16484-5として「BACS (Building Automation and Control System)」および「BACnet」が国際標準規格に登録された。

BACnet は空調設備業界では即座に影響があり、1996年にはシーメンスが採用した。それ以前にもいくつかの企業が BACnet 対応機器を開発していたが、同じ1996年、Alerton が空調制御に関する総合的な BACnet 製品群を発表した。BACnet 標準は2回改版され（2001年と2004年）、新たな機能が追加されていった。2001年版では、特に火災などの安全関係のシステムへの拡張が行われた。2006年1月、BACnet Manufacturers Association と BACnet Interest Group of North America が合併し、BACnet International が結成された。

BACnet International には関連する様々な組織が加盟し、普及推進を行っており、ANSI/ASHRAE 135-1995以降、ANSI/ASHRAE 135-2001, 135-2004, 135-2008, 135-2010, 135-2012 が発行され、2019年10月時点の最新は 135-2016 である。

日本では、電気設備学会（IEIEJ）が BACnet の ASHRAE/ANSI Standard 135-1995 に、積算、電力デマンド等の独自の拡張を加えた「BAS標準インターフェース」（IEIEJ-P-0003:2000）を発行した（通称、IEIEJp）⁴。しかし、BACnet との相互接続ができない仕様であったため、2002年にアデンダムA（IEIEJ-P-003:2000-a）を策定した（通称、IEIEJp-A）。しかし、IEIEJp-A を使っても完全な相互運用性は達成できておらず、問題を残している。

2006年9月に、電気設備学会は、ASHRAE/ANSI Standard 135-2004 をベースにした、「BACnetシステムインターオペラビリティガイドライン」（IEIEJ-G-0006:2006）を発行している。2017年3月に ASHRAE 135-2012 に対応する改定を行った（IEIEJ-G-0006:2017）。2019年5月に、ANSI/ASHRAE Standard 135-2016 の対応を追加する、「BACnetシステムインターオペラビリティガイドライン(追補）」（IEIEJ-G-0008:2019）を発行している。

最大の特長は、照明や空調などのビル設備のそれぞれがメーカー独自の仕様であっても、BACnet プロトコルを介してすべて接続できることにある。このため、ビル内のさまざまなインフラ設備を、統合的に監視・制御・保安・保守するための通信規格として世界的に普及した。大規模なビルシステムはもちろん、中小規模のビルや商業施設、ホテルや空港システムなどでも採用され、現在のビル制御プロトコルのデファクトスタンダード（業界標準規格）となっている。

5.5.2 詳細

国内では、BACnet 2004の仕様を日本仕様に拡張した「BACnetシステムインターオペラビリティガイドライン（IEIEJ-G-0006）」が電気設備学会によって発行され、この仕様に則った運用が主流になっている。

BACnetには、イーサネット上でBACnetサービスを通信するための「BACnet IP」プロトコルと、データ伝送の物理層にEIA-485（RS-485）を使用する「BACnet MS/TP」プロトコルがある。下図にその概要を占めす。

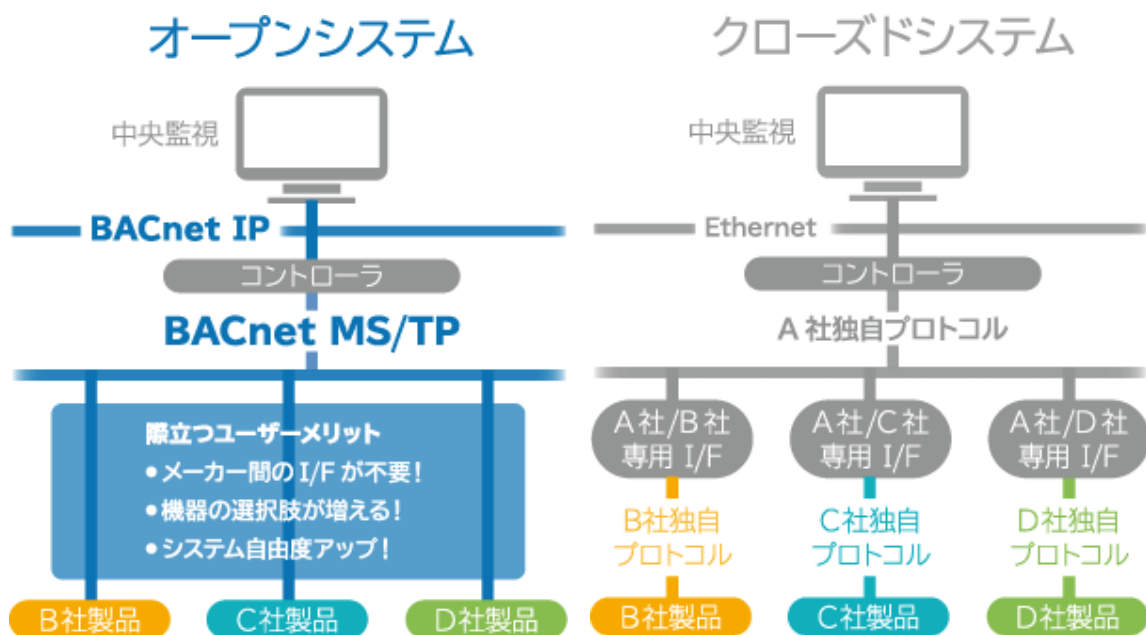


図 5-16 従来システム（左）と BACnet（右）

出典：渡辺電機工業ホームページより (<https://www.watanabe-electric.co.jp/bacnet/about.html>)

5.6 LonWorks

5.6.1 概要

LonWorks は1990 年初めに登場以来、業界最初のオープンなフィールドバスとして、主にビルディングオートメーションをはじめ多様な分野で多くの実績(累計1億5千万台以上)がある。LonWorks 以外のオープンプロトコル(BACnet、KNXなど)に対してもマルチプロトコル対応と垂直統合を可能にするIAP 統合アーキテクチャを進めている。

5.6.2 詳細

(1) LonWorks の特長

- ツイストペア線の配線トポロジー制約が無く、ノイズに強く、無極性のため配線容易
- SNVT（標準ネットワーク変数型）、SFPT（標準機能プロファイル型）による相互運用性
- ネットワーク管理仕様の明確な規定、ネットワーク経由のファームウェア更新が可能
- ネットワーク管理データベースの可搬性
- インストール時に有効なサービスピン、Wink 機能の存在
- ツイストペア線以外に電力線、IP、無線など各種メディアに対応し配線コストを軽減

(2) SNVT 標準ネットワーク変数

さまざまなメーカーが提供するアプリケーションプログラムがネットワーク変数を使って容易に相互運用可能にするためには、ネットワーク変数内のデータを同じ方法で解釈する必要がある。たとえば、ネットワーク上でやり取りする温度の値はすべて共通のフォーマットであることが必要である。フォーマットは固定小数点/整数/浮動小数点/文字列、上下限の範囲、有効数字の桁数、サイズ(バイト数)などがあり、真の相互運用性を確保するためには、これらを統一する必要がある。これらの変数は標準的なネットワーク変数のタイプ (SNVT : Standard Network Variable Types) と呼ばれ、200 を超える標準のタイプが定義されている。表 5-4 に SNVT の例を示す。また図 5-17 に装置間通信データと SNVT の関係を示す。

表 5-4 SNVT の例

変数	単位	サイズ	範囲
SNVT_temp_f	温度 °C	4	-3.40282E38～3.40282E38
SNVT_temp_p	温度 °C	2	-273.17～327.66
SNVT_vol_f	電力 V	4	-3.40282E38～3.40282E38
SNVT_res	抵抗 Ω	4	-3.40282E38～3.40282E38
SNVT_mass_	質量 g	2	0～6553.5
SNVT_mass_f	質量 g	4	-3.40282E38～3.40282E38
SNVT_mass_kilo	質量 kg	2	0～6553.5

出典：ロンワークス導入ハンドブック 2015年7月第一版 P.85

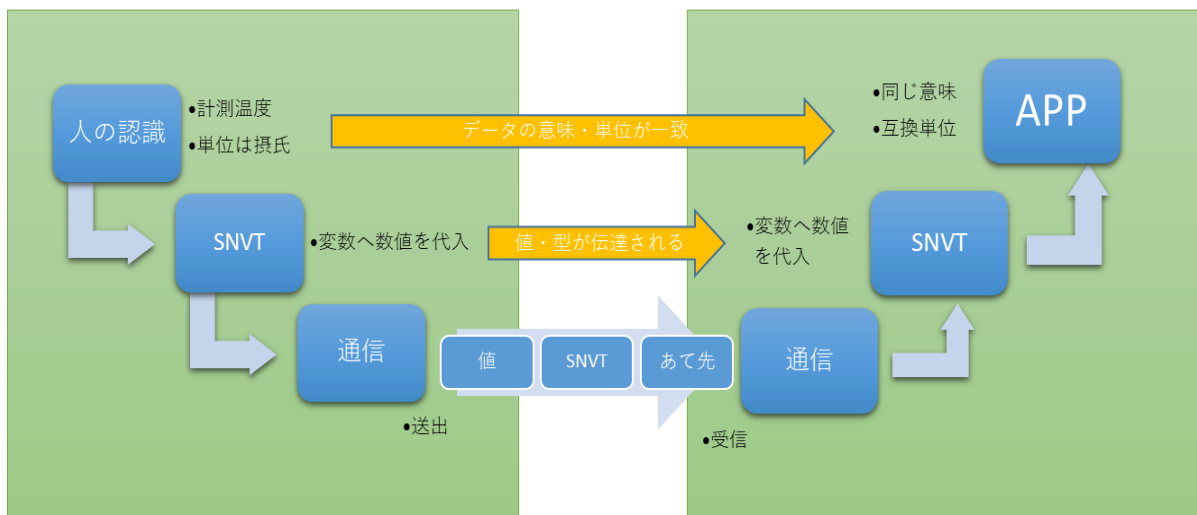


図 5-17 装置間通信データと SNVT の関係

出典：ロンマークジャパントレーニング資料「アプリケーション間データと相互運用性」P. 2

(3) LonWorks に関連する国際標準規格

表 5-5 に LonWorks がサポートする通信媒体と規格の一覧を示す。

表 5-5 LonWorks がサポートする通信媒体と規格

規格	国際標準	欧州標準	米国標準
制御ネットワーク仕様	ISO/IEC 14908-1	EN14908-1	ANSI/CTA 709.1
TP/FTフリートポロジー	ISO/IEC 14908-2	EN14908-2	ANSI/CTA 709.3
PL-20 電力線通信	ISO/IEC 14908-3	EN14908-3	ANSI/CTA 709.2
IPトンネリング	ISO/IEC 14908-4	EN14908-4	ANSI/CTA 852
LONMARKアプリケーション	not proposed	EN14908-5	ANSI/CTA 709.5
アプリケーション要素SNVT他	not proposed	EN14908-6	ANSI/CTA 709.6
LON-IP	ISO/IEC 14908-7	EN14908-7	ANSI/CTA 709.7
HD-PLCチャネル通信	ISO/IEC 14908-8	EN14908-8	ANSI/CTA 709.8
ISM帯無線通信	not proposed	EN14908-9	ANSI/CTA 709.9
IAP(IoT Access Protocol)	not proposed	EN14908-10	ANSI/CTA 709.10

Red strings: Standardization in progress

出典：電気設備学会誌Vol.40, 2020 No.7 p406

LonWorksのルータはLonWorksのパケットの中の宛先アドレスによるルーティングを行うことにより、ネットワークに機器を増設してもルータの設定の変更は不要であり、また台数の拡張を可能とする。さらにツイストペア線とIP（イーサネット等）など異なる通信媒体の接続を可能にする。また異なるフィールドバスとのゲートウェイを経由することで、マルチプロトコルのシステム構成が可能である。図5-18にマルチプロトコルのシステム構成を示す。

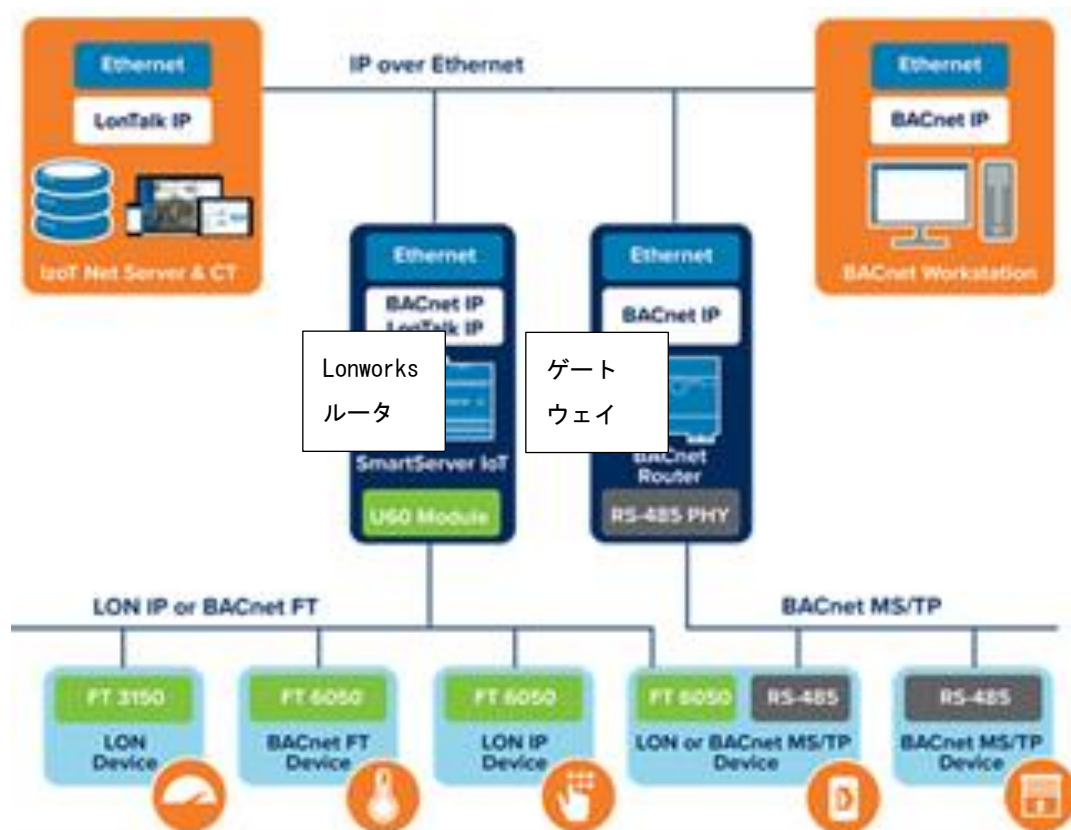


図 5-18 マルチプロトコルのシステム構成

出典：電気設備学会誌Vol.40, 2020 No.7 p406

(4) IAP のアーキテクチャ

IAP(インターネットアクセスプロトコル)は、インターネットやIT技術と、ビルディングオートメーションなどでのフィールドバスを接続するためのプロトコルである。IAPは、LonWorks、BACnet、およびその他のフィールドバスを統一して扱うことを可能とし、フィールドの既存の設備/装置へのアクセスをIoTに拡張するものである。

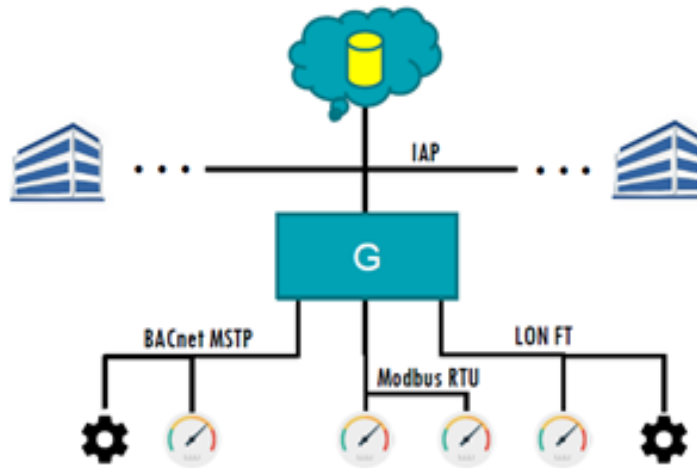


図 5-19 各種フィールドバスを統合する IAP

出典：電気設備学会誌Vol.40, 2020 No.7 p406

表5-6にIAPの特徴を示す。

表 5-6 IAP の特徴

項目	解説
オープン	API仕様公開、各種オープンフィールドバスサポート
軽量	小さなデータ量、通信の遅延や非同期を許容。HTTPセッション不要。
簡単	MQTTのパブリッシュ/サブスクライブか、REST APIのクライアント/サーバでの実装
高速	小さなデータ量、低いプロトコルオーバーヘッド。HTTPセッション不要。
イベント駆動	フィールドバス側の機器の状態変化などのイベント発生時の通知を予約(サブスクライブ)
柔軟性	エッジデバイスとクラウドサービスに最適な機能配置が実現可能
IoTの主流	MQTT/REST APIはIoTの主流
セキュリティ	TLSセキュリティ設定に対応
互換性	主なプログラミングおよびスクリプト言語対応

出典：電気設備学会誌Vol.40, 2020 No.7 p407

IAPは次のような機能を持ち、MQTTまたはREST APIを使ってアクセス可能である。

- スケジュール：日時指定もしくは 日出・日没に対する相対時間が指定可能
- データポイント読書：各種プロトコルに対してグループアドレス指定が可能。
- ファームウェア更新：対応するデバイスのファームウェア更新が可能。
- データロギング：データ更新、イベント発生を記録。記録周期、保存期間の設定。
- アラーム：アラームの発生・確認可能。Email 通知、アラーム発生時動作登録。
- フィールドバス間連携：異なるフィールドバス間での連携動作が可能

5.7 KNX

5.7.1 概要

KNX とは、インテリジェントビルのためのOSIベースのネットワーク通信プロトコルであり、EN 50090 および ISO/IEC 14543 として標準化されている。既存の3つの標準、European Home Systems Protocol (EHS)、BatiBUS、European Installation Bus (EIB) の後継として策定された。

KNX は KNX Association が管理している。日本ではまだ商業物件への採用実績は少ないが、2014年に日本KNX協会が日本におけるKNXを代表するナショナルグループ(国別支部)として設立され、KNXの啓蒙及び普及活動を行っている。

5.7.2 詳細

KNX 機器は二種類に分類される。

- ・ E-mode (Easy mode、簡易モード)：取り付けに基本的な訓練が必要。ユーザーの環境に合わせたパラメータ設定が必要である。
- ・ S-mode (System mode、システムモード)：インテリジェントビルとして建築時に実装される機器。個々のビルの環境に合わせてプログラミングする必要がある。

KNXとは

1. KNXはISO/IEC14543-3として採択された住宅、ビル用オープンスタンダードである。
2. 発祥の欧州ではすでに支配的な技術ですが、世界各地に広まりつつある。
3. ツィストペア、IP、無線やPLCなどの伝送メディアをサポートし、各種ニーズに対応。
4. 安心、効率的で、経済的な通信技術と分散処理に基づく制御方式を採用。

KNXのメリット

1. 市場にはすでに7,000を越える認定デバイスが提供されており、HVACや照明制御、エネルギー管理、セキュリティから情報の可視化用途まであらゆるソリューションを用意。
2. KNXのロゴマークが製品の相互接続性を保証する。ロゴマークは認定テストラボで認定された機器のみに表示が許可される。
3. トレーニングセンターでは初級者、中級者向けにトレーニングが用意されている。
4. 唯一のツールであるETSが製品の開発から物件のコミッショニングや保守まで幅広くサポート。
5. KNXは機器製造者のみならず、世界各地の支部団体、認定技術者、トレーニングセンターや認定テストラボなどで、サポート体制が確立している。

6. 各情報モデルの概要：ネットワーク (NW) 系

ネットワーク (NW) 系に関する規定で、IETFのSNMP、NETCONFやBroadband Forum (BBF) のTR-069らがそれに該当する。以下に詳細を述べる。

6.1 IETF

6.1.1 概要

IETF (Internet Engineering Task Force)は、インターネット技術の国際標準化を推進する団体である。TCP/IPやHTTPなどのほか、インターネットと接続する機器を管理するためのプロトコルの標準化が行われている。ネットワーク管理と情報モデルという点では、主にネットワーク機器の監視を行うためのSNMP (Simple Network Management Protocol) とネットワーク機器設定・操作などを行うNETCONF Protocolが策定されている。

SNMPにはその管理情報であるMIB (Management Information Base) とMIB情報の構造を定義するSMI (Structure of Management Information) が定義されており、広く利用されているが、2014年に発表された声明文により、現在はNETCONFのデータモデルであるYANGへの移行が推奨されている。

なお、NETCONFとYANGに関するRFCの翻訳については、TTC企業ネットワーク専門委員会から発行された下記の文書があるので、詳細はそちらを参照されたい。

- ・TR-1083 「NETCONFに関する技術報告書」 (企業ネットワーク専門委員会)

<https://www.ttc.or.jp/topics/20200310>

- ・TR-1087 「YANG 1.1 データモデリング言語に関する技術報告書」 (企業ネットワーク専門委員会)

<https://www.ttc.or.jp/topics/20210409-1>

6.1.2 詳細

SNMPは、ルータ、スイッチなどIPネットワーク上のネットワーク機器を監視・制御するための通信プロトコルである。管理するSNMPマネージャと管理されるSNMPエージェントとで構成され、それぞれがネットワーク管理に関係するMIBを維持する。SMIはMIBの構造と使用可能なデータ型を定義したものであり、ツリー構造となっている。MIBのリソース表現としては、ISOとITU-Tが策定したASN.1 (Abstract Syntax Notation One)が利用されている。

NETCONF (Network Configuration Protocol) は、ネットワーク機器の設定を設定、操作更新、削除するための通信プロトコルである。設定データをXMLで記述し、遠隔手続き呼び出し(RPC)で通信される。2006年に初版1.0がRFC 4741として公開され、その後、2011年に改定版1.1がRFC 6241として発行されている。

従来は、ネットワーク機器の設定等を行うときには、機器ごと、サポートするOSSごとに異なるコマンドを使って設定する必要があった。これは、マルチベンダ構成で、特に大規模なネットワークでは管理を難しくする。そのため、共通のデータモデルに基づいて統一したオペレーションにより操作できるように規格が必要となった。

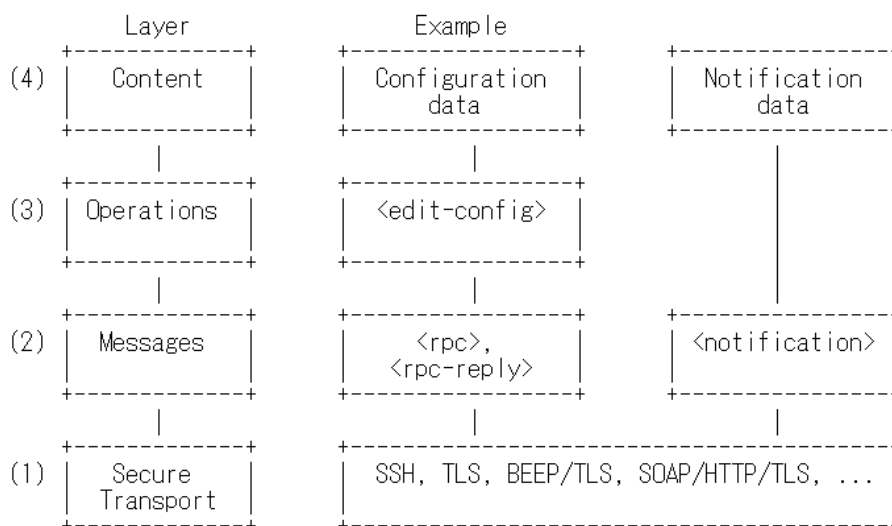


Figure 1: NETCONF Protocol Layers

図 6-1 NETCONF のプロトコルレイヤ

NETCONFでは、ネットワーク機器を管理するソフトウェアオーケストレータ (Client) が、個々のネットワークデバイス (Server) を制御する仕組みになっており、プロトコルは大まかに4レイヤに分かれている。

(1) Secure Transport

Secure Transportレイヤは、ClientとServerとの間の通信パスを提供する。NETCONFは様々なトランスポートプロトコルの上に実現される。SSHやTLSが良く使われる。

(2) Messages

Messagesレイヤは、単純でトランスポートに依存しない、RPCと通知(Notification)をエンコードするためのメカニズムを実現する。

(3) Operations

Operationsレイヤは、XMLでエンコードされたパラメータをRPCで通信する方式を提供する。デバイスの操作方法やデータストアへのアクセス方法、そのときのパラメータを規定する。

(4) Content

Contentレイヤは、NETCONFのスコープ外である。このレイヤはNETCONFで使われるデータモデルを定義するレイヤである。このレイヤは、YANGによって規定される。

YANG (Yet Another Next Generation) は、データモデル言語の一種であり、NETCONFのためのデータモデル言語として策定が始められたが、RESTCONFなどXML以外のエンコーディングも可能なように拡張されてきている。NETCONFやRESTCONFなどのネットワーク管理用プロトコルでアクセスされる、ネットワーク機器の設定や状態、RPCや通知をモデル化する。YANGの仕様は、RFC 6020及びRFC 7950により規定されている。YANGは、データの階層構造をツリーとしてモデル化し、データモデルはモジュールおよびサブモジュールに構造化される。データノードとしては、leaf ノード、leaf-listノード、containerノード、listノードの4つが定義されている。

6.2 Broadband Forum

6.2.1 概要

1991年Frame Relay Forum、ATM Forumとしてそれぞれスタートし、1994年にADSL Forumとして合体の後DSL Forumと名称変更し、2008年にBroadband Forum (以下BBFと略す) へ名称変更。2009年、IP/MPLS Forumと統合された。

活動目的は、ホームネットワーク (以下HNと略す) とビジネスIPネットワークインフラ、グローバルネットワークのベストプラクティスを定義し、新しいサービスとコンテンツ配信を可能にし、テクノロジー移行戦略を策定し、重要なデバイス、サービス&開発管理ツールを設計することである。ブロードバンド市場において、アーキテクチャ、デバイスおよびサービス管理、ソフトウェアデータモデル相互運用性および認証に対応するマルチサービスブロードバンドパケットネットワークング仕様を開発する。

HNの世界では、これに関係する組織として、HGI (Home Gateway Initiative) とOSGIがある。HGIではスマートホーム分野における他標準化団体であるBBF (Broad Band Forum) との協業を2013年5月から開始しており、本協業を通じて、本文書RD039は、BBFメンバーからも参照可能な要求仕様となっている。

また、OSGIでは、TR-069 を利用してHGW上のOSGiフレームワークやバンドルを管理するための仕様を提案しており、既存のOSGi仕様とのマッピングに関するガイドラインの策を行っている。

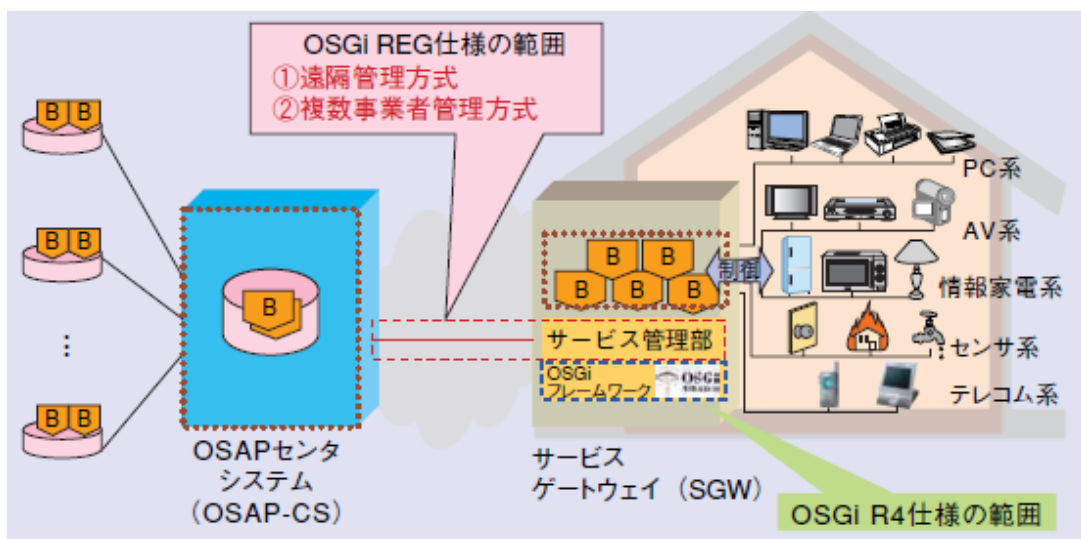


図 6-3 ホーム ICT 基盤における OSGi 標準化の範囲

出典：電子情報通信学会 研究会 ICT関連の標準化の取り組みについて○山崎毅文（NTT）

6.2.2 詳細

BBFにおいて策定しているCWMP (CPE Wan Management Protocol、仕様番号TR-069)は、ホームゲートウェイをセンターより遠隔管理するためのプロトコルであり、実質的なデファクトスタンダードであると言える。その他ホームICTに関連のある物として、TR-181 issue 2では、ホームゲートウェイを含む宅内機器用のデータモデルが抜本的に作り替えられた。新しく策定されたTR-069 amendment 3では、CWMPを用いてソフトウェアモジュールの管理する方法が追加されている。PD-174ではホームゲートウェイ上に配置したプロキシによりプロトコル変換を行い、ホームゲートウェイ配下のデバイスを管理する方式を定めている。

図6-4にTransportとしてのプロトコルであるCWMPを示す。CWMPはテキストベースのプロトコルであり、デバイス (CPE) と自動構成サーバー (ACS) の間で送信される注文は、HTTP (またはより頻繁にはHTTPS) を介して転送される。デバイス制御の例を、表6-1、図6-5にTR-181 issue の、ホームゲートウェイを含む宅内機器用のデータモデル示す。

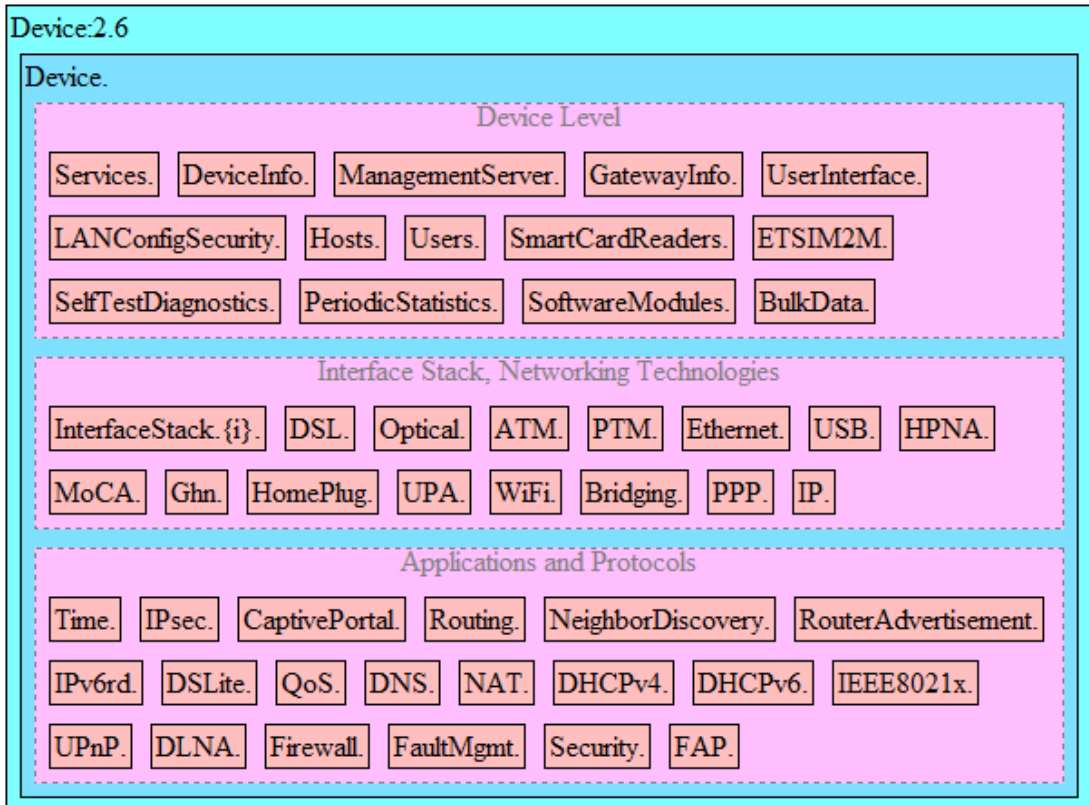


図 6-4 CWMP 概要

出典：TR-181より

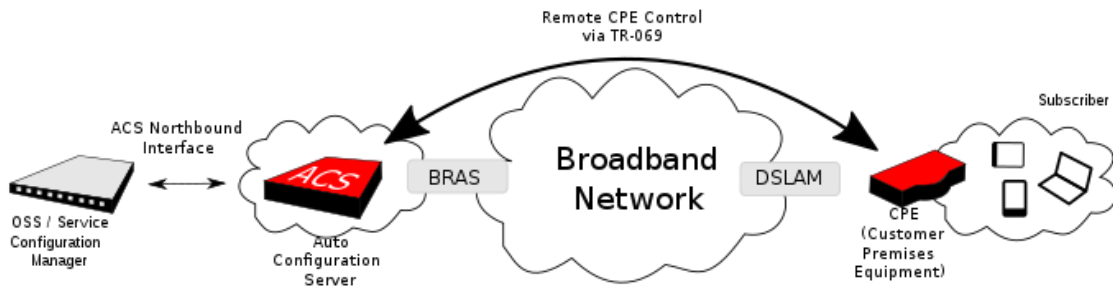


図 6-5 宅内機器用のデータモデル

出典：TR-181より

表 6-1 デバイス制御の一例

メッセージ Message	説明 Description
GetParameterNames	デバイスからサポートされているパラメータのリストを取得します。
GetParameterValues	キーで識別されるパラメータの現在の値を取得します。この呼び出しのバリエーションは、オブジェクトをキーとして受け取ります。オブジェクトのすべてのパラメータを取得します
SetParameterValues	1つ以上のパラメータの値を設定します
GetParameterAttributes	1つ以上のパラメータの属性を取得する
SetParameterAttributes	1つ以上のパラメータの属性を設定する
ダウンロード	URLで指定されたファイルをダウンロードして使用するようにCPEに注文します。ファイルタイプには、ファームウェアイメージ、構成ファイル、呼び出し音ファイルなどがあります。
アップロード	指定された宛先にファイルをアップロードするようにCPEに注文します。ファイルタイプには、現在の構成ファイル、ログファイルなどが含まれます。
AddObject	オブジェクトに新しいインスタンスを追加する
DeleteObject	オブジェクトからインスタンスを削除します

出典：TR-181を加工

7. まとめ

以上をまとめると、下記の5つの分類となると考えられる。

- ◆ 方向性1：情報モデルの表現方法（階層化、使用語彙など）
- ◆ 方向性2：通信方式を、情報モデルとプロトコルに分けて整理（REST ベース/CRUD+N ベース）
- ◆ 方向性3：通信方式を、アクション、イベントで実現（RPC ベース）
 - ネットワーク系では方向性2中心でやってきたが、IoTでは方向性3が必要。
 - 一般のインターネット/クラウドアプリケーションでは方向性3（RPC）が中心。
- ◆ 方向性4：分野間の接続。各情報モデルを接続するゲートウェイ。IICFに整理されている内容。
- ◆ 方向性5：抽象表現から具体表現へのマッピング（Protocol Binding）
 - 方向性4の一部かもしれない

いづれにしても、今後台頭してくる様々な情報モデルに対して調査を継続して、機会を狙って整理して行くなどの見直しが必要と考える。

以上