



装置用Control & Communication システムプロファイル仕様書 Part 5: OPC UA 情報モデル



改定履歴

副番	改定内容	発行年月
	初版作成	2017 年 4 月

目次

1. 序文	Part5-4
2. 適用範囲	Part5-5
3. 引用規格	Part5-5
4. 用語、定義、略称	Part5-6
4.1. 用語	Part5-6
4.1.1. 装置用 CSP+	Part5-6
4.1.2. CSPP セクション	Part5-6
4.1.3. CSPP パート	Part5-6
4.1.4. CSPP 要素	Part5-6
4.1.5. CSPP 項目	Part5-6
4.2. 略語とシンボル	Part5-6
5. 装置用 CSP+と OPC UA	Part5-7
5.1. 装置用 CSP+の紹介	Part5-7
5.1.1. 概要	Part5-7
5.1.2. 装置用 CSP+の基本構造	Part5-7
5.1.2.1. CSPP セクション	Part5-7
5.1.2.2. CSPP パート	Part5-8
5.1.3. CSPP 要素と CSPP 項目	Part5-9
5.1.4. 装置情報と装置データ	Part5-10
5.2. Introduction to OPC Unified Architecture	Part5-11
5.2.1. General	Part5-11
5.2.2. Graphical Notation	Part5-12
5.3. ユースケース	Part5-14
6. 装置モデル	Part5-15
6.1. 概要	Part5-15
6.1.1. 基本的な考え方	Part5-15
6.1.2. モデルの位置付け	Part5-15
6.2. タイプ定義	Part5-16
6.2.1. CspMachineType ObjectType	Part5-16
6.2.1.1. CspMachineType ObjectType	Part5-16
6.2.1.2. ParameterSet Object	Part5-17
6.2.1.3. COMM_IF セクションに対応する Object	Part5-17
6.2.1.4. COMM_IF_VARIABLE パートに対応する Object	Part5-18
6.2.1.5. COMM_IF_CONFIGURATION パートに対応する Object	Part5-18
6.2.2. CspAnalogItemType VariableType	Part5-18
6.2.3. CSP+装置 ObjectType	Part5-18
6.3. 装置 Object とエントリーポイント	Part5-19

7. マッピング	Part5-20
7.1. 概要	Part5-20
7.2. FILE セクション	Part5-20
7.3. DEVICE セクション	Part5-20
7.3.1. ヘッダ情報	Part5-20
7.3.2. DEVICE_INFO パート	Part5-20
7.3.2.1. ヘッダ情報	Part5-20
7.3.2.2. COMMON 情報の CSPP 要素	Part5-20
7.3.2.3. COMMON 情報以外の CSPP 要素	Part5-21
7.3.3. DEVICE_IF パート	Part5-21
7.4. COMM_IF セクションと BLOCK セクション	Part5-22
7.4.1. 基本的な考え方	Part5-22
7.4.2. COMM_IF セクションのヘッダ情報	Part5-23
7.4.3. BLOCK セクションのヘッダ情報	Part5-23
7.4.4. COMM_IF_INFO パート	Part5-23
7.4.5. BLOCK_INFO パート	Part5-23
7.4.6. COMM_IF_VARIABLE パート、COMM_IF_CONFIGURATION パート	Part5-23
7.4.6.1. ヘッダ情報	Part5-23
7.4.6.2. CSPP 要素	Part5-23
7.4.7. BLOCK_MEMORY パート、BLOCK_PARAM パート	Part5-25
7.4.7.1. ヘッダ情報	Part5-25
7.4.7.2. 既定ラベルの CSPP 要素	Part5-25
7.4.7.3. 既定ラベル以外の CSPP 要素	Part5-25
7.4.8. ENUM パート	Part5-26
7.4.8.1. ヘッダ情報	Part5-26
7.4.8.2. CSPP 要素	Part5-26
8. データタイプのマッピング	Part5-27
9. Profile とネームスペース	Part5-28
9.1. ネームスペースメタデータ	Part5-28
9.2. OPC UA コンフォーマンスユニットと Profile	Part5-28
9.3. OPC UA ネームスペースのハンドリング	Part5-29
付属書 A: ネームスペースとマッピング	Part5-30
A.1 装置用 CSP+情報モデルのネームスペースと識別子	Part5-30
A.2 装置用 CSP+情報モデルの Profile URI	Part5-30
関連仕様書	Part5-30

図 2-1	装置用 CSP+の OPC UA サーバへの適用イメージ	Part5-5
図 5-1	装置用 CSP+ファイルの構造イメージ	Part5-7
図 5-2	CSPP セクションの構造イメージ	Part5-8
図 5-3	CSPP パートの構造イメージ	Part5-9
図 5-4	装置用 CSP+の構成イメージ (装置情報と装置データの関連付けイメージ)	Part5-10
図 5-8	装置用 CSP+対応 OPC UA サーバの動作イメージ	Part5-14
図 6-1	装置に関する OPC UA 情報モデルの位置付け	Part5-15
図 6-2	CsppMachineType ObjectType の全体像	Part5-16
図 6-3	エントリポイントと装置 Object、CSP+装置 ObjectType の関係のイメージ	Part5-19
図 7-1	COMM_IF セクション・BLOCK セクションと CSP+装置 ObjectType の関係イメージ	Part5-22
表 5-1	装置用 CSP+の各セクションの概要	Part5-7
表 5-2	FILE セクションに含まれる CSPP パート	Part5-8
表 5-3	DEVICE セクションに含まれる CSPP パート	Part5-8
表 5-4	COMM_IF セクションに含まれる CSPP パート	Part5-8
表 5-5	BLOCK セクションに含まれる CSPP パート	Part5-9
表 6-1	CsppMachineType ObjectType の定義	Part5-16
表 6-2	ParameterSet Object の定義	Part5-17
表 6-3	COMM_IF セクションに対応する Object の定義	Part5-17
表 6-4	COMM_IF_VARIABLE パートに対応する Object の定義	Part5-18
表 6-5	COMM_IF_CONFIGURAIION パートに対応する Object の定義	Part5-18
表 6-6	CsppAnalogItemType VariableType の定義	Part5-18
表 6-7	CSP+装置 ObjectType の定義	Part5-18
表 7-1	DEVICE_INFO パートの COMMON 情報の CSPP 要素のマッピング仕様	Part5-21
表 7-2	COMM_IF_VARIABLE パート・COMM_IF_CONFIGURATION パートの CSPP 要素の マッピング仕様	Part5-24
表 7-3	BLOCK_MEMORY パートおよび BLOCK_PARAM パートの CSPP 要素のマッピング仕様	Part5-25
表 8-1	装置用 CSP+のデータ型と OPC UA データタイプのマッピング規則	Part5-27
表 9-1	NamespaceMetadata Object for this Specification	Part5-28
表 9-2	CSP+ for machine Server Facet Definition	Part5-28
表 9-3	CSP+ for machine Client Facet Definition	Part5-28
表 9-4	Namespace used in CSPPlusForMachine Server	Part5-29
表 9-5	Namespaces used in this specification	Part5-29

1. 序文

本書は、装置用 Control & Communication システムプロファイル仕様書の OPC UA 情報モデルです。

装置向け Control & Communication システムプロファイル（以降「装置用 CSP+」と記す）は、装置の管理や監視、制御等を行うアプリケーションソフトウェアのアプリケーションベンダによる開発や装置利用者による設定を容易化することを目的に、装置の情報を見える化したデータの集合です。装置用 CSP+には、表現対象とする装置に関する以下の情報が含まれます。

- ・ 装置のスペックに関する情報
- ・ アプリケーションソフトウェア向けに公開する装置の情報（装置情報）
- ・ 装置から取得するデータとその取得方法に関する情報（装置データ）
- ・ 装置情報と装置データの紐付け情報

装置用 CSP+は一般的に、XML 形式ファイルで表現した装置用 CSP+ファイルとして扱います。

OPC UA 情報モデルでは、装置用 CSP+に記述されている情報を OPC UA 情報モデルにマッピングする仕様を規定します。ここで規定する仕様は、装置用 CSP+を利用するアプリケーションソフトウェアが OPC UA サーバである場合に適用します。

本書に記載されている装置用 Control & Communication システムプロファイル仕様のバージョン（以後「装置用 CSP+仕様バージョン」と記す）は、1.0 です。

2. 適用範囲

本書は、装置用 CSP+ の OPC UA 情報モデルへのマッピング仕様を規定します。本書で規定する仕様は、装置用 CSP+ を利用するアプリケーションソフトウェアが OPC UA サーバである場合に適用することを推奨します。適用イメージを図 2-1 に示します。

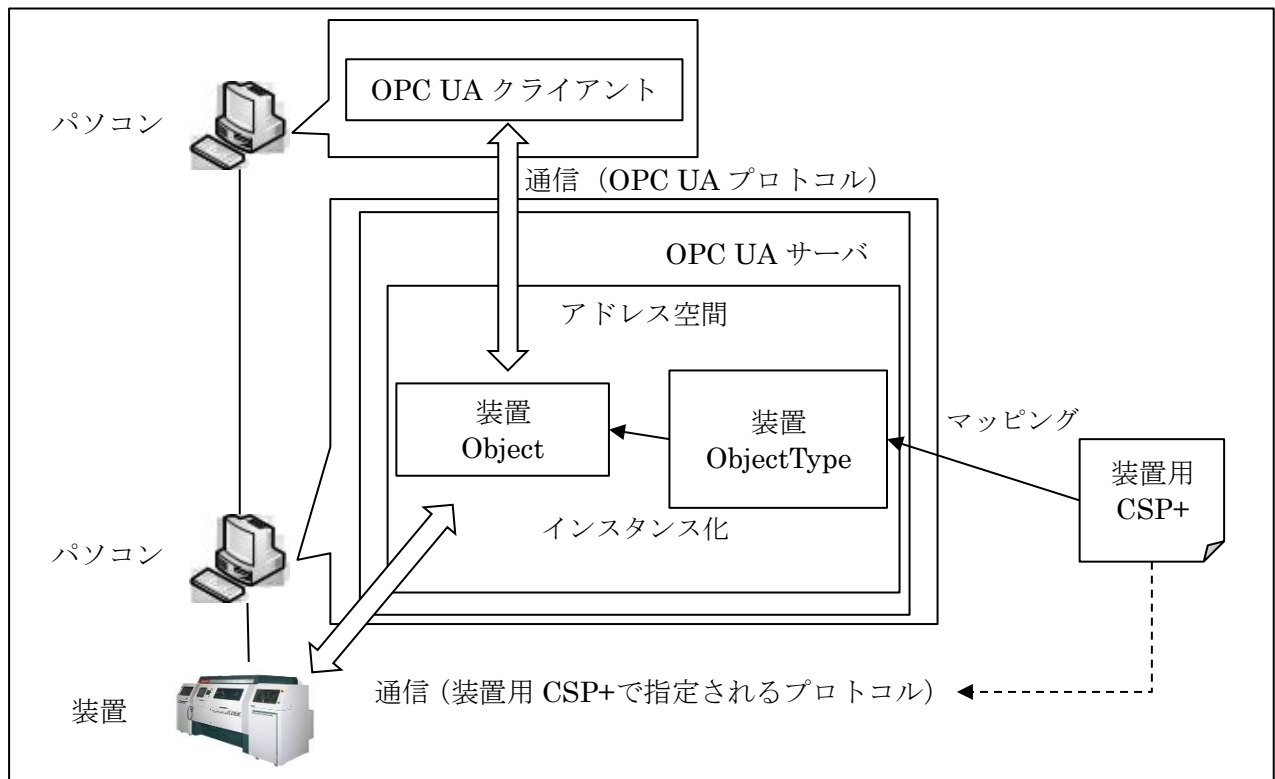


図 2-1 装置用CSP+のOPC UAサーバへの適用イメージ

3. 引用規格

- IEC 62541-3:2015 OPC Unified Architecture – Part 3: Address Space Model
- IEC 62541-4:2015 OPC Unified Architecture – Part 4: Services
- IEC 62541-5:2015 OPC Unified Architecture – Part 5: Information Model
- IEC 62541-6:2015 OPC Unified Architecture – Part 6: Mappings
- IEC 62541-8:2015 OPC Unified Architecture – Part 8: Data Access
- IEC 62541-100:2015 OPC Unified Architecture – Part 100: Device Interface

4. 用語、定義、略称

4.1. 用語

4.1.1. 装置用 CSP+

装置に関する以下情報を表現するためのデータ集合。

- ・ 装置のスペックに関する情報
- ・ アプリケーションソフトウェア向けに公開する装置の情報
- ・ 装置から取得するデータとその取得方法
- ・ 装置情報と装置データの紐付け情報

4.1.2. CSPP セクション

装置用 CSP+の構成要素。

4.1.3. CSPP パート

セクションの構成要素。

4.1.4. CSPP 要素

パートの構成要素。

4.1.5. CSPP 項目

要素に関する詳細情報。例：データ型、工学単位。

4.2. 略語とシンボル

CNC	Computer Numerical Control
CSP+	Control & Communication System Profile
PLC	Programmable Logic Controller

5. 装置用 CSP+ と OPC UA

5.1. 装置用 CSP+ の紹介

5.1.1. 概要

装置用 CSP+ は、装置の管理や監視、制御等を行うアプリケーションソフトウェアのアプリケーションベンダによる開発や装置利用者による設定を容易化することを目的に、装置の情報を見える化したデータの集合である。装置用 CSP+ を XML 形式ファイルで表現したものが装置用 CSP+ ファイルとなる。装置用 CSP+ には装置に関する以下情報が含まれる。

- ・ 装置のスペックに関する情報
- ・ アプリケーションソフトウェア向けに公開する装置の情報（装置情報）
- ・ 装置から取得するデータとその取得方法に関する情報（装置データ）
- ・ 装置情報と装置データの紐付け情報

5.1.2. 装置用 CSP+ の基本構造

5.1.2.1. CSPP セクション

装置用 CSP+ は、FILE セクション、DEVICE セクション、COMM_IF セクション、BLOCK セクションの 4 種類の CSPP セクションで構成する。各 CSPP セクションの概要を表 5-1 に示す。

表 5-1 装置用 CSP+ の各セクションの概要

CSPP セクション名	概要	個数
FILE	装置用 CSP+ ファイルの管理情報を記述する。	1 個
DEVICE	装置の名称や識別情報、装置仕様等の情報を記述する。	1 個
COMM_IF	装置情報の定義情報を記述する。	1 個以上
BLOCK	装置データの定義情報を記述する。	1 個以上

図 5-1 に装置用 CSP+ の構造イメージを示す。

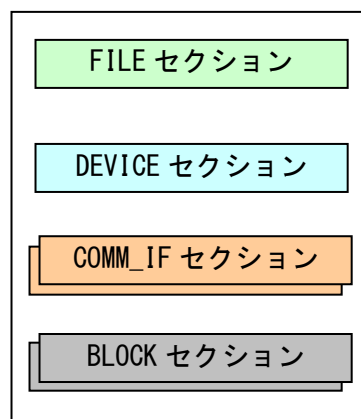


図 5-1 装置用 CSP+ ファイルの構造イメージ

5.1.2.2. CSPP パート

各 CSPP セクションは 1 個以上の CSPP パートによって構成する。図 5-2 に CSPP セクション内の構造イメージを示す。

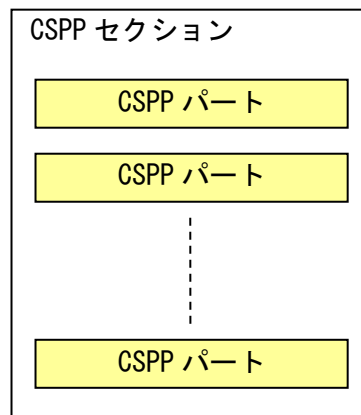


図 5-2 CSPPセクションの構造イメージ

各 CSPP セクションに含まれる CSPP パートの種別は、CSPP パートごとに異なる。以下、FILE セクションに含まれる CSPP パート種別を表 5-2、DEVICE セクションに含まれる CSPP パート種別を表 5-3、COMM_IF セクションに含まれる CSPP パート種別を表 5-4、BLOCK セクションに含まれる CSPP パート種別を表 5-5 に示す。

表 5-2 FILEセクションに含まれるCSPPパート

CSPP パート種別	記述する情報	個数
FILE_INFO	・ 装置用 CSP+ファイルの管理情報 (例：ファイル生成日、言語情報、ファイルバージョン)	1 個

表 5-3 DEVICEセクションに含まれるCSPPパート

CSPP パート種別	記述する情報	個数
DEVICE_INFO	・ 装置の識別情報 (例：ペンダ名、型名) ・ 装置の製品情報 (例：仕様情報、画像ファイル名)	1 個
DEVICE_IF	・ 装置との通信手段に関する情報 (例：通信プロトコル種別)	1 個以上

表 5-4 COMM_IFセクションに含まれるCSPPパート

CSPP パート種別	記述する情報	個数
COMM_IF_INFO	・ 装置情報の識別情報	1 個
COMM_IF_VARIABLE	・ リアルタイムモニタ用の装置情報 (例：電流値)	0 個以上
COMM_IF_CONFIGURATION	・ 汎用の装置情報 (例：30 分間で消費した電力量)	0 個以上
ENUM	・ 設定範囲の選択肢	0 個以上

表 5-5 BLOCKセクションに含まれるCSPPパート

CSPP パート種別	記述する情報	個数
BLOCK_INFO	・ 装置データの識別情報	1 個
BLOCK_MEMORY	・ 装置から取得する変数値の装置データ (例：現在値、測定時刻)	0 個以上
BLOCK_PARAM	・ 装置から取得しない装置固有の装置データ (例：精度、収集周期)	0 個以上
ENUM	・ 設定範囲の選択肢	0 個以上

装置情報を管理する CSPP パートとして COMM_IF_VARIABLE パートと COMM_IF_CONFIGURATION パートが存在する。COMM_IF_VARIABLE パートにはリアルタイムモニタ用の装置情報を記載し、COMM_IF_CONFIGURATION パートには汎用の装置情報を記載する。COMM_IF_VARIABLE パートに記載できる情報はリアルタイムモニタ用に限定されているため、記述は容易だが用途が限定される。一方で、COMM_IF_CONFIGURATION パートの記載に対する制約は少なく、汎用の用途に対して装置情報を記述できる。画面表示レイアウトの例では、COMM_IF_VARIABLE パートの装置情報はモニタリング画面として表示し、COMM_IF_CONFIGURATION パートの装置情報は装置パラメータ読書き画面として表示することが考えられる。

5.1.3. CSPP 要素と CSPP 項目

各 CSPP パートは 1 個以上の CSPP 要素によって構成する。各 CSPP 要素は 1 個以上の CSPP 項目によって構成する。図 5-3 にパートの構成イメージを示す。

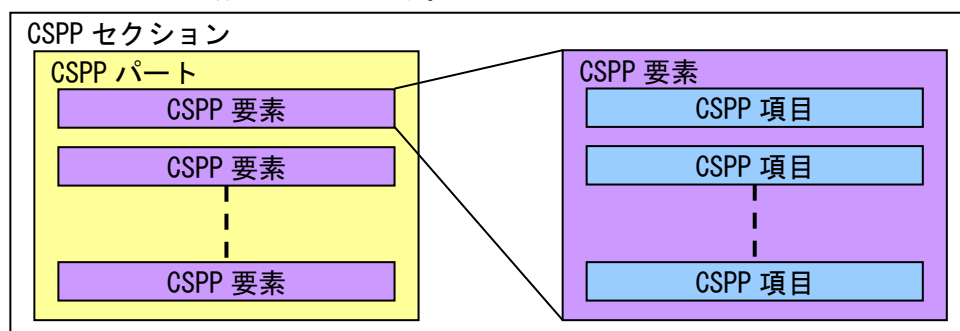


図 5-3 CSPPパートの構造イメージ

各 CSPP パートに含まれる CSPP 要素には、装置用 CSP+仕様として規定される CSPP 要素と、装置ベンダが自由に規定できる CSPP 要素とがある。一方で、各 CSPP 要素に含まれる CSPP 項目についてはすべて装置用 CSP+仕様で規定される。

5.1.4. 装置情報と装置データ

装置情報を管理する COMM_IF_CONFIGURATION パート内の一つの CSPP 要素から、BLOCK_MEMORY パート、もしくは BLOCK_PARAM パート内の 1 個以上の装置データに対して関連付け情報が設定される。これは言い換えると、複数の装置データを集約する形で、1 個の装置情報が形成されることを意味する。このような構成になっている理由は、アプリケーションソフトウェアが必要とする情報は、単なる測定値（現在値）だけではなく、その値が測定された時刻や、値の精度等、付随する情報が必要なためである。

一方で、COMM_IF_VARIABLE パート内の CSPP 要素、すなわち装置情報からは、BLOCK_MEMORY パート、もしくは BLOCK_PARAM パート内の装置データに対する関連付け情報は設定できない。これは、用途がリアルタイムモニタであり、装置情報内に必要となるすべての情報を記載可能であるためである。装置情報と装置データの関連付けイメージを図 5-4 に示す。

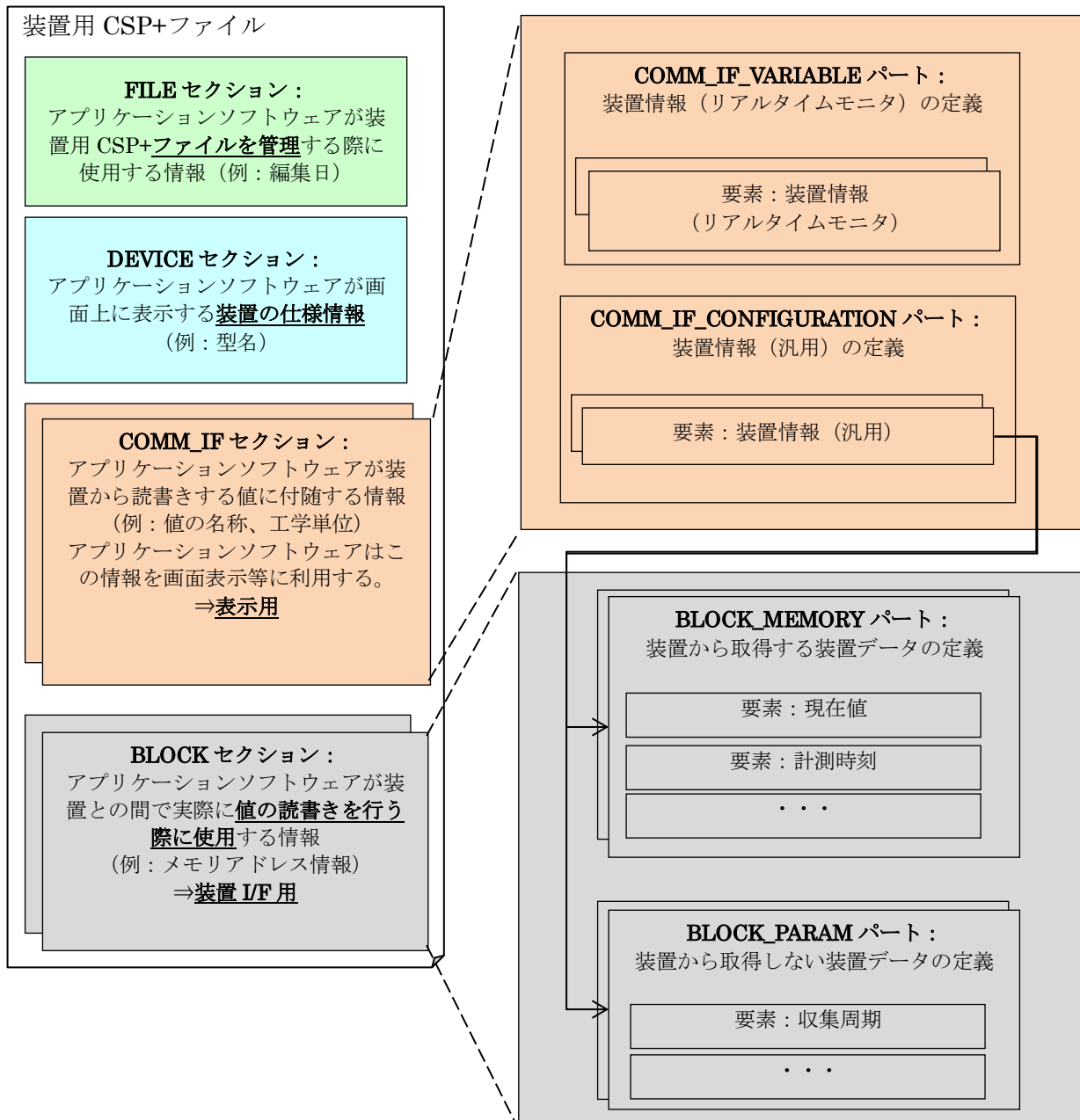


図 5-4 装置用CSP+の構成イメージ（装置情報と装置データの関連付けイメージ）

5.2. Introduction to OPC Unified Architecture

5.2.1. General

The main use case for OPC classic specifications is the online data exchange between devices and HMI or SCADA systems using Data Access functionality. In this use case the device data is provided by an OPC server and is consumed by an OPC client integrated into the HMI or SCADA system. OPC DA provides functionality to browse through a hierarchical namespaces containing data items and to read, write and to monitor these items for data changes. The classic OPC standards are based on Microsoft COM/DCOM technology for the communication between software components from different vendors. Therefore classic OPC server and clients are restricted to Windows PC based automation systems.

OPC UA incorporates all features of classic OPC standards like OPC DA, A&E and HDA but defines platform independent and secure communication mechanisms and generic, extensible and object-oriented modelling capabilities for the information a system wants to expose. OPC UA is directly integrated into devices and is used for configuration, diagnostic and maintenance use cases in addition to online data exchange. OPC UA is an integrated communication interface used from sensor level devices up to enterprise applications.

IEC 62541-6:2015 defines different mechanisms optimized for different use cases. The first version of OPC UA is defining an optimized binary TCP protocol for high performance intranet communication as well as a mapping to accepted internet standards like Web Services. The abstract communication model does not depend on a specific protocol mapping and allows adding new protocols in the future. Features like security and reliability are directly built into the transport mechanisms. Based on the platform independence of the protocols, OPC UA servers and clients can be directly integrated into devices and controllers.

The OPC UA *Information Model* provides a standard way for *Servers* to expose *Objects* to *Clients*. *Objects* in OPC UA terms are composed of other *Objects*, *Variables* and *Methods*. OPC UA also allows relationships to other *Objects* to be expressed.

The set of *Objects* and related information that an OPC UA *Server* makes available to *Clients* is referred to as its *AddressSpace*. The elements of the OPC UA *Object Model* are represented in the *AddressSpace* as a set of *Nodes* described by *Attributes* and interconnected by *References*. OPC UA defines eight classes of *Nodes* to represent *AddressSpace* components. The classes are *Object*, *Variable*, *Method*, *ObjectType*, *DataType*, *ReferenceType* and *View*. Each *NodeClass* has a defined set of *Attributes*.

This specification defines *Nodes* of the OPC UA *NodeClasses* *Object*, *Method*, *Variable*, *ObjectType* and *DataType*.

Objects are used to represent components of a system. An *Object* is associated to a corresponding *ObjectType* that provides definitions for that *Object*.

Methods are used to represent commands or services of a system.

Variables are used to represent values. Two categories of *Variables* are defined, *Properties* and *DataVariables*.

Properties are *Server*-defined characteristics of *Objects*, *DataVariables* and other *Nodes*. *Properties* are not allowed to have *Properties* defined for them.

DataVariables represent the data contents of an *Object*.

5.2.2. Graphical Notation

OPC UA defines a graphical notation for an OPC UA *AddressSpace*. It defines graphical symbols for all *NodeClasses* and how different types of *References* between *Nodes* can be visualized. Figure 5-5 shows the symbols for the six *NodeClasses* used in this specification. *NodeClasses* representing types always have a shadow.

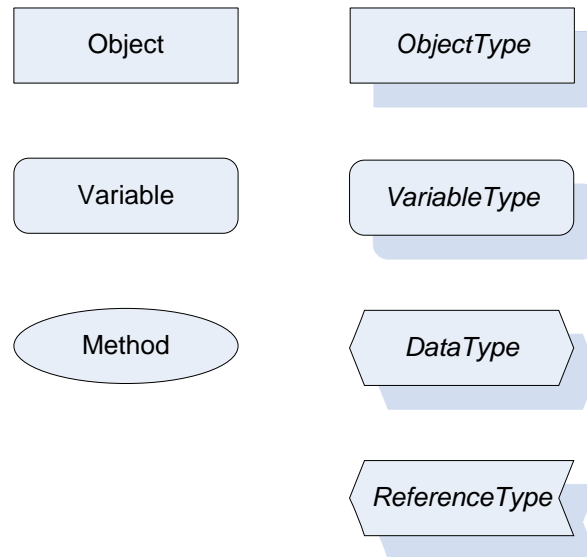


Figure 5-5 OPC UA Graphical Notation for NodeClasses

Figure 5-6 shows the symbols for the *ReferenceTypes* used in this specification. The *Reference* symbol is normally pointing from the source *Node* to the target *Node*. The only exception is the *HasSubType Reference*. The most important *References* like *HasComponent*, *HasProperty*, *HasTypeDefinition* and *HasSubType* have special symbols avoiding the name of the *Reference*. For other *ReferenceTypes* or derived *ReferenceTypes* the name of the *ReferenceType* is used together with the symbol.

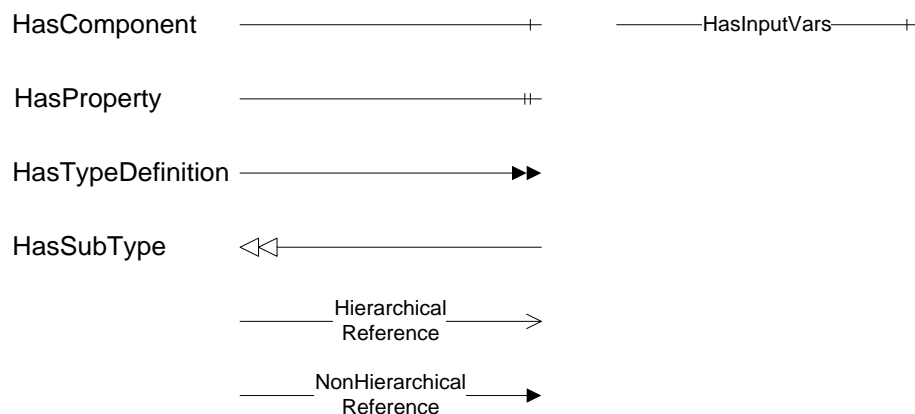


Figure 5-6 OPC UA Graphical Notation for References

Figure 5-7 shows a typical example for the use of the graphical notation. Object_A and Object_B are instances of the ObjectType_Y indicated by the *HasTypeDefinition References*. The ObjectType_Y is derived from ObjectType_X indicated by the *HasSubType Reference*. The Object_A has the components Variable_1, Variable_2 and Method_1.

To describe the components of an *Object* on the *ObjectType* the same *NodeClasses* and *References* are used on the *Object* and on the *ObjectType* like for *ObjectType_Y* in the example. The instance *Nodes* used to describe an *ObjectType* are instance declaration *Nodes*.

To provide more detailed information for a *Node*, a subset or all *Attributes* and their values can be added to a graphical symbol.

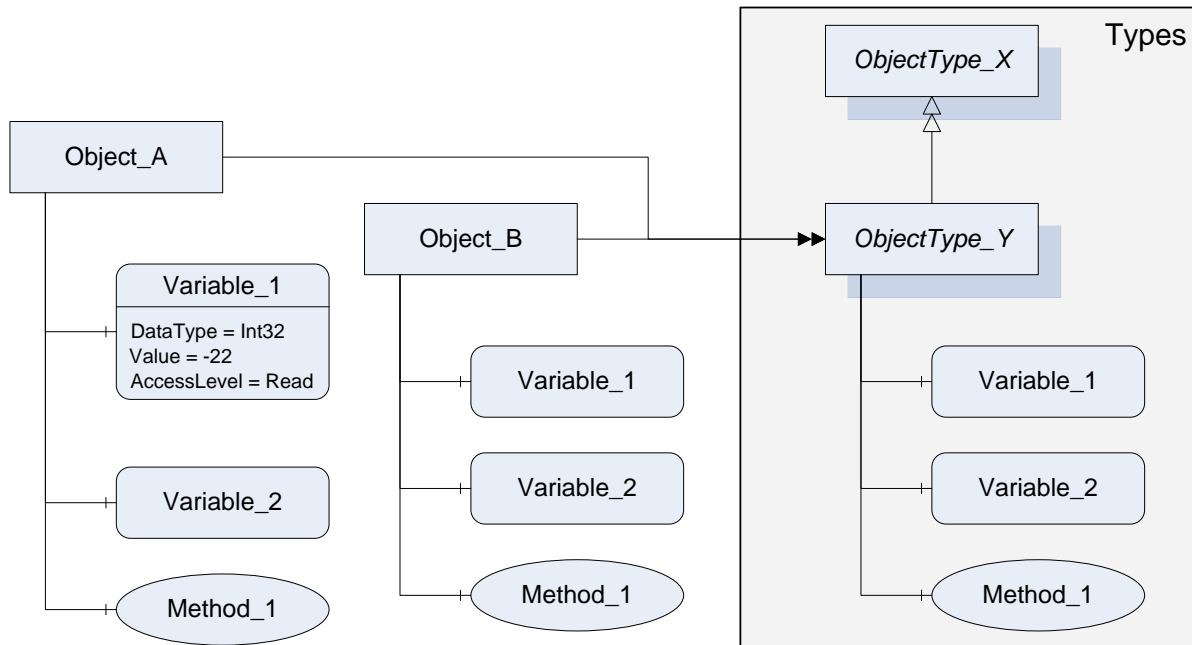


Figure 5-7 OPC UA Graphical Notation Example

5.3. ユースケース

本節では、装置用 CSP+を利用するアプリケーションソフトウェアの利用例として、OPC UA サーバの例を図 5-8 を用いて示す。

本例では装置用 CSP+を利用するアプリケーションソフトウェアである OPC UA サーバ自身は装置利用者に対して機能を提供せず、他のアプリケーションソフトウェア、例えば SCADA や MES が装置利用者に対して監視や管理等の機能を提供する。なお、装置用 CSP+を利用するアプリケーションソフトウェアが OPC UA サーバの場合、OPC UA サーバが OPC UA クライアントに対して公開するアドレススペースは、本書で規定する仕様に従って装置用 CSP+から生成される。

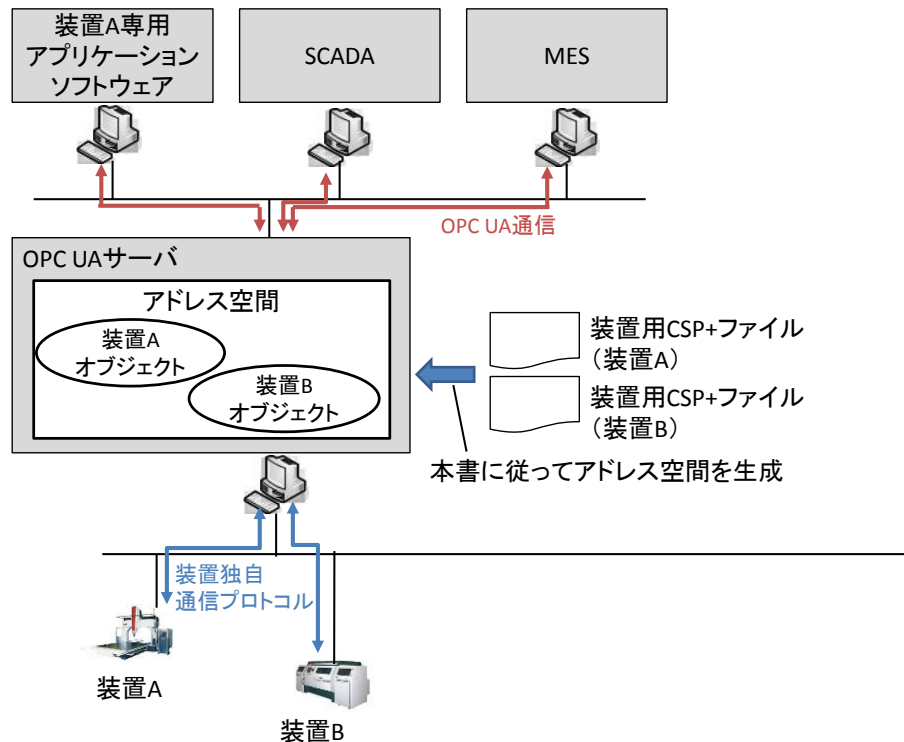


図 5-8 装置用CSP+対応OPC UAサーバの動作イメージ

このようなシステム構成を構築可能とすることで、SCADA や MES 等のアプリケーションベンダには以下に示すメリットがある。

- ・ 装置独自の通信プロトコルではなく、標準規格である OPC UA を装置との間の通信に使えるため、通信機能の開発量を抑えることが可能

OPC UA サーバのアプリケーションベンダには以下に示すメリットがある。

- ・ 装置用 CSP+に対応することで装置用 CSP+に対応する多くの装置を通信相手としてサポート可能
- また、装置ベンダには以下に示すメリットがある。
- ・ アプリケーションベンダに装置専用のアプリケーションソフトウェア開発を依頼する際に、装置独自プロトコルへの対応を依頼する必要がないため、開発費用を抑えることが可能
- ・ アプリケーションソフトウェアに公開する情報、例えば装置内のプロセス状況や装置の操作履歴など、を装置用 CSP+に記載されている範囲に限定することができるため、装置に関するノウハウに関する情報を秘匿することが可能

6. 装置モデル

6.1. 概要

本章では、装置用 CSP+によって表現される装置に関する OPC UA 情報モデルを規定する。

6.1.1. 基本的な考え方

装置用 CSP+によって表現されるすべての装置に対応する *CsppMachineType ObjectType* を定義し、さらに各装置用 CSP+によって表現される個別の装置に対応する専用の *ObjectType*（以下、CSP+装置 *ObjectType* と呼ぶ）を定義します。*CsppMachineType ObjectType* は、IEC 62541-100:2015 で規定される *DeviceType ObjectType* を継承した抽象 *ObjectType* (IsAbstract attribute が”true”) とします。CSP+装置 *ObjectType* は *CsppMachineType ObjectType* を継承する非抽象 *ObjectType* (IsAbstract attribute が”false”) とします。継承に際して、装置用 CSP+に記述されている情報に対応する *Node* 定義を追加します。

装置用 CSP+によって表現される装置の情報は、CSP+装置 *ObjectType* の Object として表現します。

6.1.2. モデルの位置付け

本書で規定する OPC UA 情報モデルの位置付けを図 6-1 に示します。装置に対応する *CsppMachineType ObjectType* の他に、*CsppMachineType ObjectType* 向けの *VariableType* である *CsppAnalogItemType* を規定します。

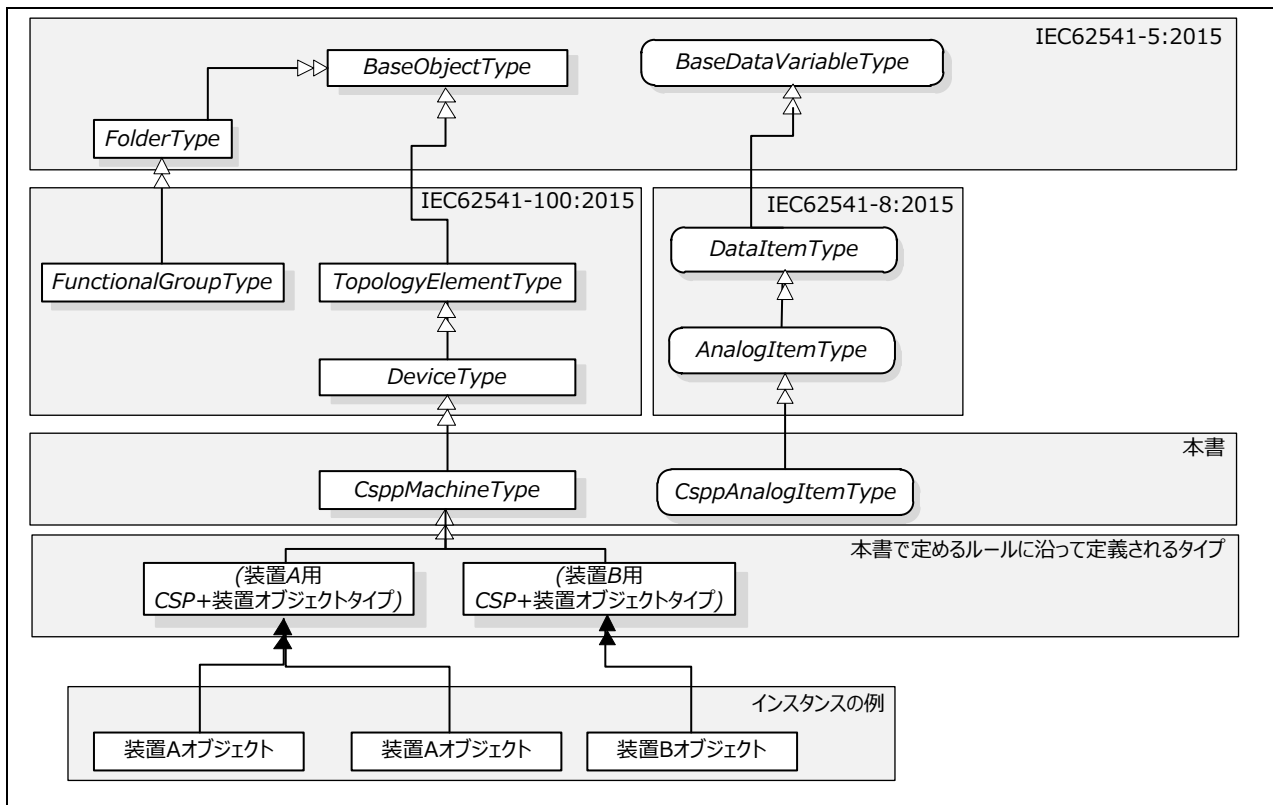


図 6-1 装置に関する OPC UA 情報モデルの位置付け

6.2. タイプ定義

6.2.1. CspMachineType ObjectType

本項では、*CspMachineType ObjectType*、および *CspMachineType ObjectType* から参照される *Node* の定義を示します。図 6-2 に *CspMachineType ObjectType* の全体像を示します。

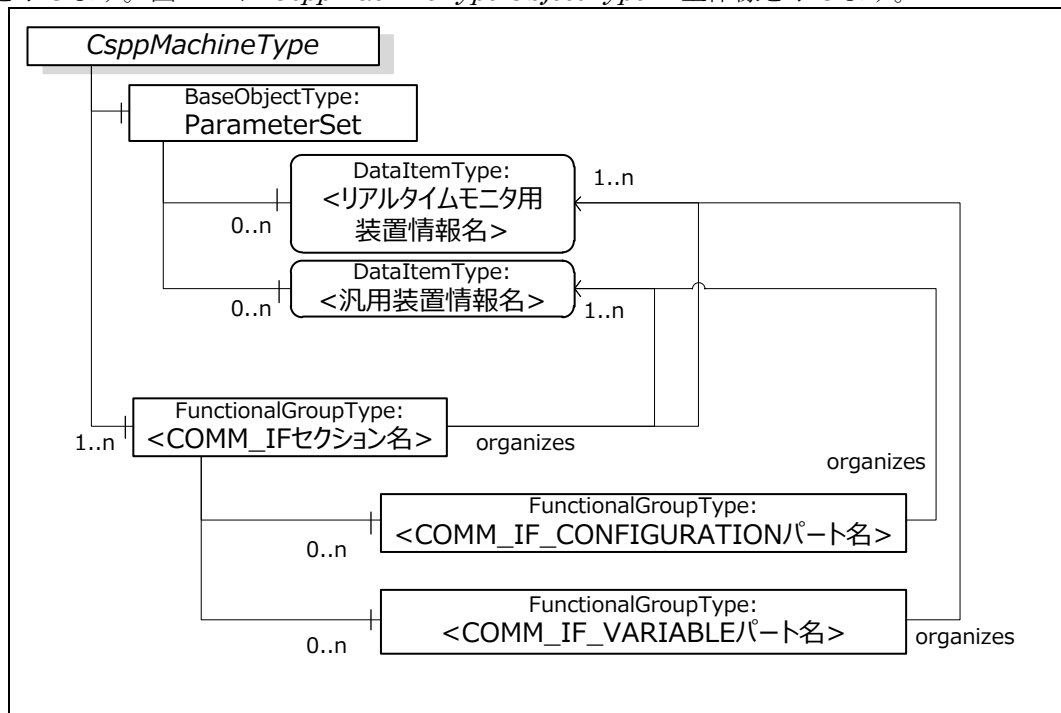


図 6-2 CspMachineType ObjectTypeの全体像

6.2.1.1. CspMachineType ObjectType

CspMachineType ObjectType の定義を表 6-1 に示します。

表 6-1 CspMachineType ObjectTypeの定義

Attribute	Value					
BrowseName	CspMachineType					
IsAbstract	True					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	詳細
Subtype of the DeviceType defined in IEC 62541-100:2015						
HasComponent	Object	1:ParameterSet	-	BaseObjectType	Mandatory	6.2.1.2
HasComponent	Object	<CommIfSection>	-	FunctionalGroup Type	MandatoryPlaceholder	6.2.1.3

ParameterSet Object の *ModellingRule* は継承元では Optional ですが、*CspMachineType ObjectType* では Mandatory とします。また、*ParameterSet Object* 自身の定義も override します。詳細は 6.2.1.2 を参照してください。

<CommIfSection>で示す *Object* は、装置用 CSP+ の COMM_IF セクションに対応し、装置の持つ装置情報をグループ化する役割を持ちます。

6.2.1.2. ParameterSet Object

ParameterSet Object は、装置 *Object* が持つ装置情報をまとめる役割を持ちます。表 6-2 に *ParameterSet Object* の定義を示します。

表 6-2 ParameterSet Objectの定義

Attribute	Value					
BrowseName	1:ParameterSet					
IsAbstract	False					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	詳細
HasTypeDefinition	ObjectType	0:BaseObjectType				
HasComponent	Variable	<VariableName>	(Any)	DataItem Type	OptionalPlaceholder	-
HasComponent	Variable	<ConfigurationName>	(Any)	DataItem Type	OptionalPlaceholder	-

<VariableName>で示される *DataVariable* は、装置用 CSP+ の COMM_IF_VARIABLE パート内の CSPP 要素に対応します。*VariableType* は IEC 62541-8:2015 で規定される DataItem Type *VariableType* かその派生タイプとなります。

<ConfigurationName>で示される *DataVariable* は、装置用 CSP+ の COMM_IF_CONFIGURATION パート内の CSPP 要素に対応します。*VariableType* は IEC 62541-8:2015 で規定される DataItem Type *VariableType* かその派生タイプとなります。

6.2.1.3. COMM_IF セクションに対応する Object

COMM_IF セクションに対応する *Object* は、装置 *Object* が装置情報を CSPP セクション単位でグループ化する役割を持ちます。また、同時に COMM_IF セクション内の COMM_IF_VARIABLE パートと COMM_IF_CONFIGURATION パートをまとめる役割を持ちます。表 6-3 に *Object* の定義を示します。

表 6-3 COMM_IFセクションに対応するObjectの定義

Attribute	Value					
BrowseName	(name of COMM_IF section)					
IsAbstract	False					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	詳細
HasTypeDefinition	ObjectType	1:FunctionalGroupType				
HasComponent	Object	<CommIfVariablePart>	-	FunctionalGroup Type	OptionalPlaceholder	6.2.1.4
HasComponent	Object	<CommIfConfigurationPart>	-	FunctionalGroup Type	OptionalPlaceholder	6.2.1.5
Organizes	Variable	<VariableOrConfigurationName>	(Any)	DataItem Type	MandatoryPlaceholder	-

<CommIfVariablePart>で示す *Object* は、装置用 CSP+ の COMM_IF_VARIABLE パートに対応し、装置の持つリアルタイムモニタ用装置情報をグループ化する役割を持ちます。

<CommIfConfigurationPart>で示す *Object* は、装置用 CSP+ の COMM_IF_CONFIGURATION パートに対応し、装置の持つ汎用装置情報をグループ化する役割を持ちます。

6.2.1.4. COMM_IF_VARIABLE パートに対応する Object

COMM_IF_VARIABLE パートに対応する *Object* は、装置 *Object* が持つリアルタイムモニタ用装置情報をパート単位でグループ化する役割を持ちます。表 6-4 に *Object* の定義を示します。

表 6-4 COMM_IF_VARIABLEパートに対応するObjectの定義

Attribute	Value					
BrowseName	(name of COMM_IF_VARIABLE part)					
IsAbstract	False					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	詳細
HasTypeDefinition	ObjectType	1:FunctionalGroupType				
Organizes	Variable	<VariableName>	(Any)	DataItem Type	MandatoryPlaceholder	-

6.2.1.5. COMM_IF_CONFIGURATION パートに対応する Object

COMM_IF_CONFIGURATION パートに対応する *Object* は、装置 *Object* が持つ汎用装置情報をパート単位でグループ化する役割を持ちます。表 6-5 に *Object* の定義を示します。

表 6-5 COMM_IF_CONFIGUTRAIONパートに対応するObjectの定義

Attribute	Value					
BrowseName	(name of COMM_IF_CONFIGURATION part)					
IsAbstract	False					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	詳細
HasTypeDefinition	ObjectType	1:FunctionalGroupType				
Organizes	Variable	<ConfigurationName>	(Any)	DataItem Type	MandatoryPlaceholder	-

6.2.2. Csp AnalogItemType VariableType

装置用 CSP+で表現する装置情報の内、測定期間を伴う装置情報を表現するための *Csp AnalogItemType VariableType* を定義します。表 6-6 に *VariableType* の定義を示します。

表 6-6 Csp AnalogItemType VariableTypeの定義

Attribute	Value					
BrowseName	Csp AnalogItemType					
IsAbstract	False					
ValueRank	-2 (-2='Any')					
DataType	Number					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	詳細
SubType of the AnalogItemType defined in IEC 62541-8:2015						
HasProperty	Variable	Duration	Number	PropertyType	Mandatory	-

Duration Property は、当該 variable が計算もしくは計測された期間をミリ秒単位で示します。

6.2.3. CSP+装置 ObjectType

CSP+装置 *ObjectType* の定義を表 6-7 に示します。

表 6-7 CSP+装置ObjectTypeの定義

Attribute	Value					
BrowseName	(DEVICE section LABEL of the corresponding CSP+)					
IsAbstract	False					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	詳細
Subtype of the <i>Csp MachineType</i> defined in 6.2.1						

CSP+装置 *ObjectType* から参照される *Node* の定義は、当該 CSP+装置 *ObjectType* が対応する装置用 CSP+の内容に従って決まります。装置用 CSP+の内容決定するマッピングのルールについては 7 章に示します。

6.3. 装置 Object とエントリーポイント

1 台の装置を示す装置 *Object* は、当該装置に対応する CSP+装置 *ObjectType* をインスタンス化したものとし、IEC 62541-100:2015 で規定される *DeviceSet* エントリーポイントから *HasComponent Reference* で参照されます。図 6-3 にエントリーポイントと装置 *Object*、CSP+装置 *ObjectType* の関係のイメージを示します。図 6-3 は装置が 3 台存在し、装置 A の装置 *Object* が 2 個、装置 B の装置 *Object* 1 個が存在することを示しています。

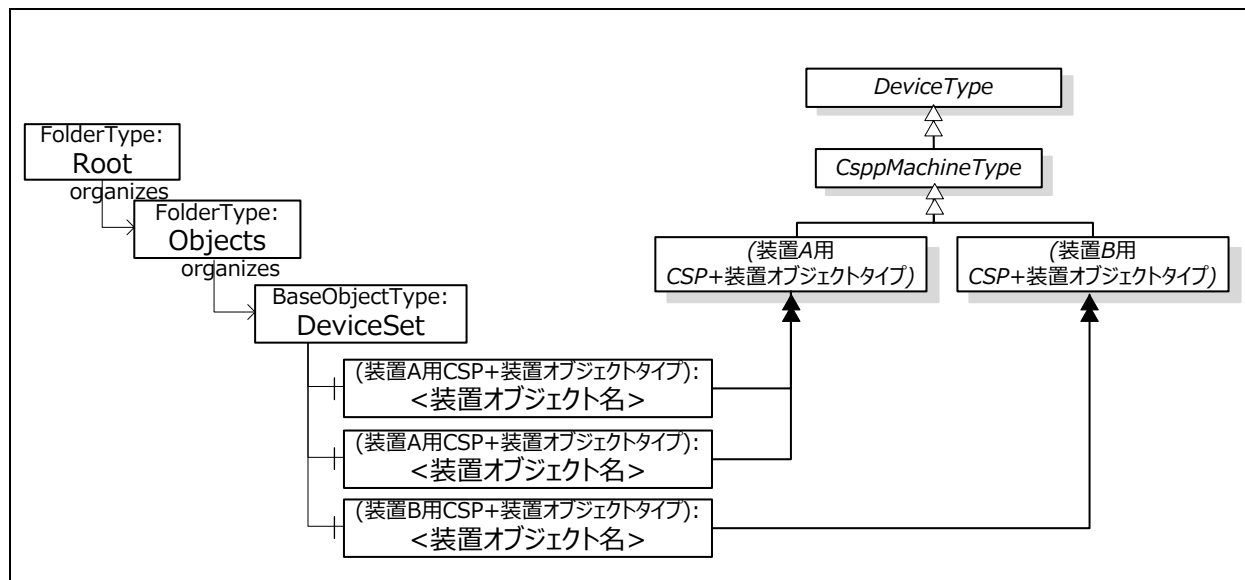


図 6-3 エントリーポイントと装置Object、CSP+装置ObjectTypeの関係のイメージ

7. マッピング

7.1. 概要

本章では装置用 CSP+の記述情報に基づいて *CsppMachineType ObjectType* から CSP+装置 *ObjectType* を生成する際のルールを規定します。

なお、*CsppMachineType ObjectType* で Mandatory の *Attribute* や *Property*、*DataVariable* の内、本書でルールを規定しないものの継承ルールについては、CSP+装置 *ObjectType* を生成するアプリケーション (OPC UA *Server*) の実装依存とします。同様に、CSP+装置 *ObjectType* から装置 *Object* をインスタンス化する際についても同様にアプリケーション (OPC UA *Server*) の実装依存とします。

7.2. FILE セクション

FILE セクションに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。ただし、COMMON 情報の一つである Language 要素の DATA 項目値を、文字列のロケールを示すために間接的に用いる場合があります。詳細は各ルールの規定箇所を参照してください。

7.3. DEVICE セクション

7.3.1. ヘッダ情報

DEVICE セクションのヘッダに記述される情報の内、ラベル名を CSP+装置 *ObjectType* の *BrowseName Attribute* の値の一部として用います。具体的には *Attribute* 値の後にテキスト”CsppDeviceType”を連結したものを、*BrowseName Attribute* の値とします。例えば、ラベル名が”ABC”の場合、*BrowseName Attribute* 値は”ABCCsppDeviceType”となります。

上記以外の情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.3.2. DEVICE_INFO パート

7.3.2.1. ヘッダ情報

DEVICE_INFO パートのヘッダに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.3.2.2. COMMON 情報の CSPP 要素

DEVICE_INFO パート内の COMMON 情報 CSPP 要素の一部を、CSP+装置 *ObjectType* の *Attribute* や *Property* にマッピングします。

表 7-1 にマッピング仕様を示します。

表 7-1 DEVICE_INFOパートのCOMMON情報のCSPP要素のマッピング仕様

No.	CSPP 要素（マッピング元）の LABEL 名	マッピング先	ルール
1.	VendorName	Manufacturer <i>Property</i>	CSPP 要素の DATA 項目値（Unicode 文字列）を、 <i>Property</i> の <i>Value Attribute</i> 値にマッピングする。
2.	DeviceModel	Model <i>Property</i>	CSPP 要素の DATA 項目値（ASCII 文字列）を、 <i>Property</i> の <i>Value Attribute</i> 値にマッピングする。
3.	ProductID	（マッピングしない）	－
4.	Version	DeviceRevision <i>Property</i>	CSPP 要素の DATA 項目値（*1）を文字列として <i>Property</i> の <i>Value Attribute</i> 値にマッピングする。
5.	ReferenceURL	DeviceManual <i>Property</i>	－
6.	URLInfo	（マッピングしない）	－
7.	Outline	Description <i>Attribute</i>	CSPP 要素の DATA 項目値（Unicode 文字列）を、 <i>Attribute</i> 値にマッピングする。
8.	SpecList	（マッピングしない）	－
9.	FileName	Icon <i>Property</i> (ImagePNG <i>DataType</i>)	CSPP 要素の DATA 項目で指定されたファイルを PNG ファイルに変換し、ファイルの内容をバイナリ列として <i>Property</i> の <i>Value Attribute</i> 値にマッピングする。
10.	GraphicsFileName	DeviceTypeImage <i>Object</i> から参照される <i>DataVariable</i> 。 <i>DataVariable</i> 名は DeviceModel 要素の DATA 項目値とする。	CSPP 要素の DATA 項目で指定されたファイルの内容をバイナリ列として、 <i>DataVariable</i> の <i>Value Attribute</i> 値にマッピングする。 <i>DataVariable</i> の <i>DataType</i> は、ファイル形式に応じた <i>DataType</i> (PNG⇒ImagePNG <i>DataType</i> 、BMP⇒ImageBMP <i>DataType</i> 、JPG⇒ImageJPG <i>DataType</i> 、GIF⇒ImageGIF <i>DataType</i>) とする。

*1: ビット列型、符号付整数型、符号なし整数型、および STRING(x) のいずれか

7.3.2.3. COMMON 情報以外の CSPP 要素

COMMON 情報以外の CSPP 要素に対し、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.3.3. DEVICE_IF パート

DEVICE_IF パートに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.4. COMM_IF セクションと BLOCK セクション

7.4.1. 基本的な考え方

以下に COMM_IF セクションと BLOCK セクション内の記述情報を CSP+装置 *ObjectType* の生成に適用する際の基本的な考え方を示します。

- COMM_IF_VARIABLE パートおよび COMM_IF_CONFIGURATION パート内で規定される各 CSPP 要素を、それぞれ CSP+装置 *ObjectType* の *ParameterSet* *Object* から *HasComponent Reference* で参照される *DataVariable* にマッピングします。
- 上記 *DataVariable* は、CSPP セクション単位と CSPP パート単位でそれぞれグループ化します。グループ化を目的に *FunctionalGroupType* の *Object* を CSPP セクション、パートの数だけ生成し、CSP+装置 *ObjectType* から CSP セクション相当の *Object* を *HasComponent Reference* で参照し、CSPP セクション相当の *Object* から CSPP パート相当の *Object* を *HasComponent Reference* で参照します。
- *DataVariable* の持つ *Attribute* や *Property* の情報 (例: 工学単位、最短収集周期) は、*DataVariable* に対応する CSPP 要素、および CSPP 要素もしくは CSPP 要素の所属する CSPP パートから参照される BLOCK_MEMORY パート・BLOCK_PARAM パート内の CSPP 要素に基づいて設定します。

基本的な考え方のイメージを図 7-1 に示します。

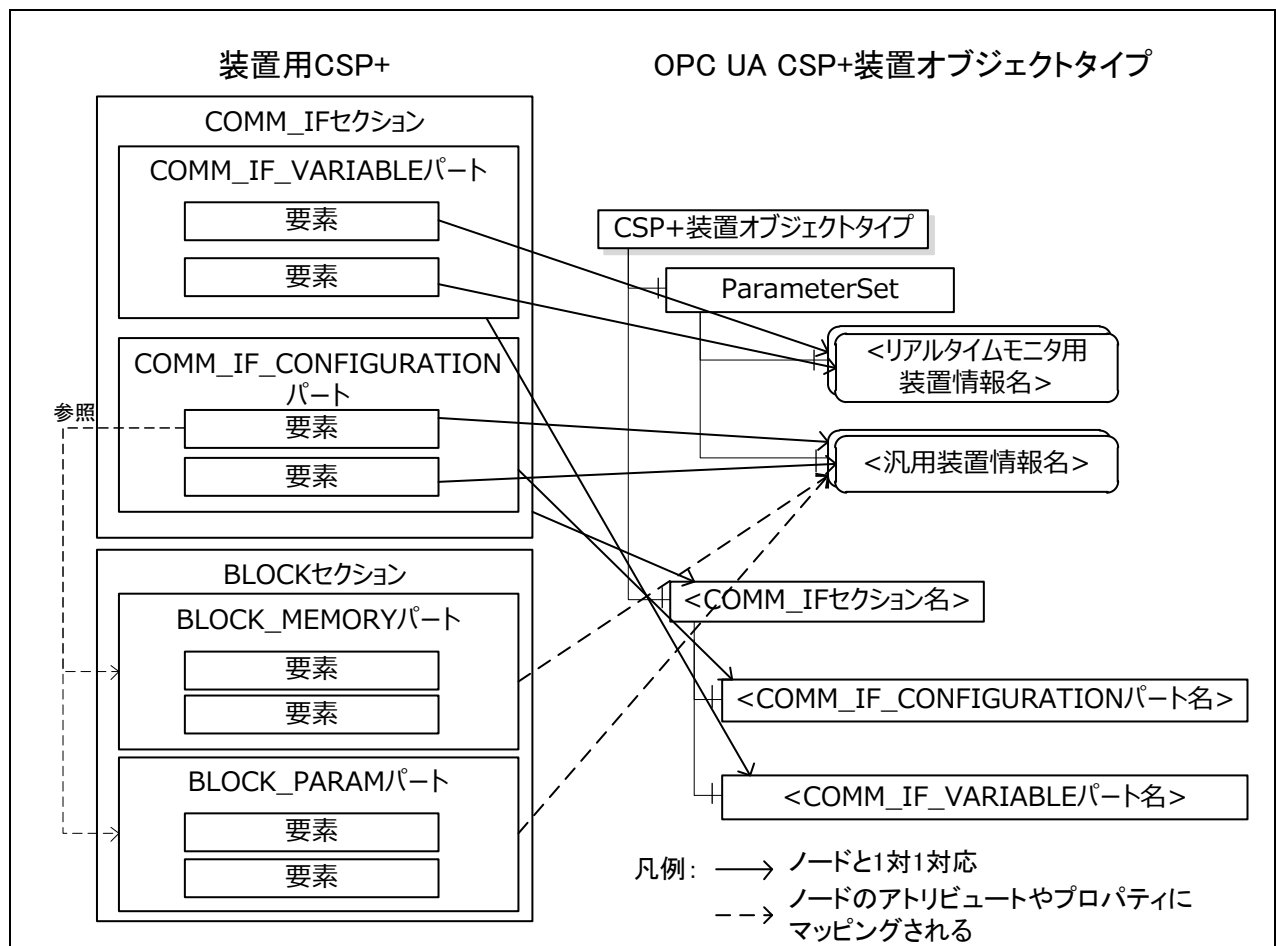


図 7-1 COMM_IF セクション・BLOCK セクションと CSP+装置 ObjectType の関係イメージ

7.4.2. COMM_IF セクションのヘッダ情報

COMM_IF セクションのヘッダに記述される情報の内、LABEL 名を *FunctionalGroupType Object* (6.2.1.3 参照) の *BrowseName Attribute* に、LABEL2 名を同 *Object* の *DisplayName Attribute* にマッピングします。

7.4.3. BLOCK セクションのヘッダ情報

BLOCK セクションのヘッダに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.4.4. COMM_IF_INFO パート

COMM_IF_INFO パートに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.4.5. BLOCK_INFO パート

BLOCK_INFO パートに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.4.6. COMM_IF_VARIABLE パート、COMM_IF_CONFIGURATION パート

7.4.6.1. ヘッダ情報

COMM_IF_VARIABLE パートおよび COMM_IF_CONFIGURATION パートのヘッダに記述される情報の内、LABEL 名を *FunctionalGroupType Object* (6.2.1.4、6.2.1.5 参照) の *BrowseName Attribute* に、LABEL2 名を同 *Object* の *DisplayName Attribute* にマッピングします。

上記以外の情報に対しては、CSP+装置 *ObjectType* の生成に用いるルールを規定しません。

7.4.6.2. CSPP 要素

COMM_IF_VARIABLE パートおよび COMM_IF_CONFIGURATION パート内で規定される各 CSPP 要素を、それぞれ CSP+装置 *ObjectType* の *ParameterSet Object* から *HasComponent Reference* で参照される *DataVariable* にマッピングします。*DataVariable* の *VariableType* は、各 CSPP 要素の項目値、および CSPP 要素から参照する BLOCK_MEMORY パート・BLOCK_PARAM パートの CSPP 要素の項目値に応じて以下に示すルールによって決定します。

- ・ RANGE 項目に ENUM を指定する要素は、*TwoStateDiscreteType*、*MultiStateDiscreteType*、*MultiStateValueDiscreteType* のいずれかの *VariableType* の *DataVariable* にマッピングします。
 - データ型がブール型の場合、*TwoStateDiscreteType VariableType*
 - ENUM パート内で、全要素の CODE 項目値が 0 から連番で割り当てられている場合、*MultiStateDiscreteType VariableType*
 - 上記以外の場合、*MultiStateValueDiscreteType VariableType*
- ・ 上記以外で、P_Period 要素を含む BLOCK_PARAM パートを参照している CSPP 要素は、*CsppAnalogItemType VariableType* の *DataVariable* にマッピングします。
- ・ 上記以外で、データ型がビット列型、符号付整数型、符号なし整数型、BCD 整数型、実数型の要素は、*AnalogItemType VariableType* の *DataVariable* にマッピングします。
- ・ 上記以外の CSPP 要素は、*DataItemType VariableType* の *DataVariable* にマッピングします。

また、各 CSPP 要素の項目の一部を、*DataVariable* の *Attribute* や *Property* にマッピングします。

表 7-2 にマッピング仕様を示します。なお、COMM_IF_VARIABLE パートおよび COMM_IF_CONFIGURATION パートで記述が省略される代わりに、BLOCK_MEMORY パートの P_Value 要素で記述される CSPP 項目についても、表 7-2 のマッピング仕様が適用されます。

表 7-2 COMM_IF_VARIABLE パート・COMM_IF_CONFIGURATION パートの CSPP 要素のマッピング仕様

No.	CSPP 要素（マッピング元）の項目名	マッピング先	ルール
1.	LABEL	<i>BrowseName Attribute</i>	項目値の末尾に、装置用 CSP+ファイル内で同名の LABEL 名を持つ項目の出現順序（初期値 1）を 10 進数の数値で付与した文字列を <i>Attribute</i> 値にマッピングする。 例：LABEL 名が” ABC” で出現順序が 12 の場合、 <i>Attribute</i> 値は” ABC12”
2.	LABEL2	<i>DisplayName Attribute</i>	項目値を <i>Attribute</i> 値にマッピングする。
3.	CATEGORY	（マッピングしない）	-
4.	NAME	（マッピングしない）	-
5.	DATATYPE	<i>DataType Attribute</i>	表 8-1 参照
6.	RANGE	<u>値や値の範囲を直接記述する場合</u> <i>EURange Property</i> <u>選択肢一覧 (ENUM) を用いる場合</u> 7. 4. 8 参照	値の範囲が 1 種類のみ規定されている場合のみ、マッピングする。値の種類が 2 種類以上規定されている場合や、値が指定されている場合はマッピングしない。 <i>EURange Property</i> の <i>DataType</i> である Range では閉区間、開区間は区別できないため、マッピング時にも区別せず、上限値を <i>Value Attribute</i> の high エlement に、下限値を low エlement にそれぞれマッピングする。
7.	MIN_INC	（間接的に <i>Value Attribute</i> に影響を与える）	表 7-3 No. 1 参照
8.	ENG_UNIT	<i>EngineeringUnits Property</i>	項目値を <i>Property</i> の <i>ValueAttribute</i> の <i>edisplayName Element</i> にマッピングする。
9.	ACCESS	<i>AccessLevel Attribute</i>	項目値を以下に示すルールで <i>Attribute</i> 値にマッピングする。なお、COMM_IF_VARIABLE パートの要素には本項目は存在しないが、項目値が R であるものとして扱う。 ・項目値が” R” : 0 ビット目を 1、1 ビット目を 0 ・項目値が” W” : 0 ビット目を 0、1 ビット目を 1 ・項目値が” RW” : 0 ビット目を 1、1 ビット目を 1 ・項目値が” NA” : 0 ビット目を 0、1 ビット目を 0 ・項目値が空白：ルールを規定しない
10.	REF_MEMORY (*1)	7. 4. 7 参照	-
11.	REF_PARAM (*1)	7. 4. 7 参照	-
12.	ASSIGN (*2)	（マッピングしない）	項目値はマッピングしないが、本項目値を用いて装置から取得した値は、 <i>DataVariable</i> の <i>Value Attribute</i> 値にマッピングする。
13.	COMMENT	<i>Description Attribute</i>	項目値を <i>Attribute</i> 値にマッピングする。

*1: COMM_IF_CONFIGURATION パートの CSPP 要素のみ対象

*2: COMM_IF_VARIABLE パートの CSPP 要素のみ対象

7.4.7. BLOCK_MEMORY パート、BLOCK_PARAM パート

7.4.7.1. ヘッダ情報

BLOCK_MEMORY パートおよび BLOCK_PARAM パートのヘッダに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.4.7.2. 既定ラベルの CSPP 要素

BLOCK_MEMORY パートおよび BLOCK_PARAM パートで既定のラベルを持つ要素は、CSP+装置 *ObjectType* に対してではなく、CSP+装置 *ObjectType* をインスタンス化した装置 *Object* に対してマッピングが行われます。

BLOCK_MEMORY パートおよび BLOCK_PARAM パートで既定のラベルを持つ要素は、その参照元である COMM_IF_CONFIGURATION パート内で規定される CSPP 要素のマッピング先である *DataVariable* の *Attribute* や *Property* にマッピングする場合と、*DataVariable* に対して実行する *Service* (*Read*, *Publish*, *Republish*) の戻り値にマッピングする場合があります。表 7-3 にマッピング仕様を示します。

表 7-3 BLOCK_MEMORY パートおよび BLOCK_PARAM パートの CSPP 要素のマッピング仕様

No.	CSPP 要素 (マッピング元) の LABEL 名	マッピング先	ルール
1.	P_Value	<i>DataVariable</i> の <i>Value Attribute</i>	装置から取得した値を <i>Attribute</i> 値にマッピングする。ただし、P_Value 要素もしくはその参照元の CSPP 要素で MIN_INC 項目が定義されている場合は、装置から取得した値を MIN_INC 項目値に応じて変換した値を <i>Attribute</i> 値にマッピングする。
2.	P_NA	<i>Service</i> 戻り値の内、 <i>DataValue DataType</i> である値*1 の <i>statusCode</i> エレメントの <i>Severity</i> フィールド (30, 31 ビット目)	装置から取得した値が 0 (欠測ではない) の場合、 <i>Severity</i> は "00" (Good Success) とする。取得した値が 1 (欠測) の場合、 <i>Severity</i> は "10" (Bad Failure) とする。
3.	P_Accuracy	<i>DataVariable</i> の <i>ValuePrecision Property</i>	装置から取得した値 (BLOCK_MEMORY パートの場合) もしくは CSPP 要素の DATA 項目値 (BLOCK_PARAM パートの場合) を、 <i>Property</i> の <i>Value Attribute</i> 値にマッピングする。
4.	P_ChangeDate	<i>Service</i> 戻り値の内、 <i>DataValue DataType</i> である値*1 の <i>sourceTimestamp</i> メンバ	装置から取得した値をメンバの値にマッピングする。
5.	P_MeasurementDate	<i>Service</i> 戻り値の内、 <i>DataValue DataType</i> である値*1 の <i>serverTimestamp</i> エレメント	装置から取得した値をメンバの値にマッピングする。
6.	P_Period	<i>DataVariable</i> の <i>Duration Property</i>	装置から取得した値 (単位は当該要素の ENG_UNIT 項目値で指定されている) をミリ秒単位に変換し、 <i>Property</i> の <i>Attribute</i> 値にマッピングする。
7.	P_Cycle	<i>DataVariable</i> の <i>MinimumSampleingInterval Attribute</i>	CSPP 要素の DATA 項目値を <i>Attribute</i> 値にマッピングする。

*1: *Read Service* の場合は *results[]* 値が *DataValue DataType* に該当する。*Publish*, *Republish Service* の場合は *notificationMessage* 値の *notificationData[]* エレメントの *Value* エレメントが *DataValue DataType* に該当する。

7.4.7.3. 既定ラベル以外の CSPP 要素

BLOCK_MEMORY パートおよび BLOCK_PARAM パートの既定ラベル以外の CSPP 要素の情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.4.8. ENUM パート

7.4.8.1. ヘッダ情報

ENUM パートのヘッダに記述される情報を、CSP+装置 *ObjectType* の生成に用いるルールは規定しません。

7.4.8.2. CSPP 要素

7.4.8.2.1. TwoStateDiscreteType VariableType

ENUM パートの参照元 CSPP 要素のデータ型がブール型の場合、参照元 CSPP 要素を *TwoStateDiscreteType VariableType* の *DataVariable* にマッピングします。ENUM パートの 2 つの CSPP 要素は、以下に示すルールで *DataVariable* の *Property* にマッピングします。

- CODE 項目値が 0 の CSPP 要素を *FalseState Property* に、CODE 項目値が 1 の CSPP 要素を *TrueState Property* に対応付けます。
- CSPP 要素の LABEL2 項目が規定されている場合、対応する *Property* の *Value Attribute* の text エレメントに CSPP 項目値を設定します。 *Locale* エレメントには、FILE_INFO パートの Language 要素の DATA 項目値を設定します。
- CSPP 要素の LABEL2 項目が規定されていない場合、LABEL 項目値を *Value Attribute* の text エレメントに設定します。 *Locale* エレメントには” en-US” を設定します。

7.4.8.2.2. MultiStateDiscreteType VariableType

ENUM パートの全 CSPP 要素の CODE 項目値が 0 から連番で割り当てられている場合、参照元 CSPP 要素を *MultiStateDiscreteType VariableType* の *DataVariable* にマッピングします。ENUM パートの CSPP 要素は、以下に示すルールで *DataVariable* の *Property* にマッピングします。

- CODE 項目値の昇順で CSPP 要素をソートし、 *EnumStrings Property* の各要素（注：前記 *Property* は配列型）に順番に対応付けます。
- CSPP 要素の LABEL2 項目が規定されている場合、 *EnumStrings Property* の対応要素の *Value Attribute* の text エレメントに項目値を設定します。 *Locale* エレメントには、FILE_INFO パートの Language 要素の DATA 項目値を設定します。
- CSPP 要素の LABEL2 項目が規定されていない場合、LABEL 項目値を text エレメントに設定します。 *Locale* エレメントには” en-US” を設定します。

7.4.8.2.3. MultiStateValueDiscreteType VariableType

ENUM パートの全 CSPP 要素の CODE 項目値が 0 から連番で割り当てられていない場合、参照元 CSPP 要素を *MultiStateValueDiscreteType VariableType* の *DataVariable* にマッピングします。ENUM パートの CSPP 要素は、以下に示すルールで *DataVariable* の *Property* にマッピングします。

- ENUM パート内での記述順序の通りに、 *EnumValues Property* の各要素（注：前記 *Property* は配列型）に順番に対応付けます。
- CSPP 要素の CODE 項目値を、 *EnumValues Property* の対応要素の *Value Attribute* の *Value* エレメントに設定します。
- CSPP 要素の LABEL2 項目が規定されている場合、 *EnumValues Property* の対応要素の *Value Attribute* の *Display* エレメントの text エレメントに項目値を設定します。 *Display* エレメントの locale エレメントには、FILE_INFO パートの Language 要素の DATA 項目値を設定します。
- CSPP 要素の LABEL2 項目が規定されていない場合、LABEL 項目値を text エレメントに設定します。 *Locale* エレメントには” en-US” を設定します。

8. データタイプのマッピング

装置用 CSP+ のデータ型から OPC UA の *DataType* へのマッピング規則を表 8-1 に示します。

表 8-1 装置用 CSP+ のデータ型と OPC UA データタイプのマッピング規則

No.	装置用 CSP+ のデータ型 (左列：分類、右列：データ型)		OPC UA の <i>DataType</i>	備考
1.	ブール型	BOOL	Boolean	装置用 CSP+ の 0 を OPC UA の FALSE に、1 を TRUE にマッピングする。
2.	バイナリ型	BIN8	Int16	バイナリ型が目的とする「ビット単位で値を表示する」ことは <i>ValueAsText Attribute</i> で表現する。 例：値 58 (0x3A) に対し <i>ValueAsText Attribute</i> は、BIN8 の場合 “00111010”、BIN16 の場合 “0000000000111010” と設定する。
3.		BIN16	Int16	
4.		BIN32	Int32	
5.		BINx (x=1~15)	Int16	
6.	ビット列型 (16 進表記)	BYTE	UInt16	16 進表記は <i>ValueAsText Attribute</i> で表現する。 例：値 58 (0x3A) に対し <i>ValueAsText Attribute</i> は、BYTE の場合 “0x3A”、WORD の場合 “0x003A” と設定する。
7.		WORD	UInt16	
8.		DWORD	UInt32	
9.		BIT_STRINGx (x=2~15)	UInt16	
10.	符号付整数型 (10 進表記)	INT8	Int16	-
11.		INT16	Int16	
12.		INT32	Int32	
13.		INTx (x=2~15)	Int16	
14.	符号なし整数型 (10 進表記)	UINT8	UInt16	-
15.		UINT16	UInt16	
16.		UINT32	UInt32	
17.		UINTx (x=2~15)	UInt16	
18.	BCD 整数型 (10 進表記)	BCD8	UInt16	4 ビットあたり 10 進数 1 桁という BCD による数値の表記は、 <i>ValueAsText Attribute</i> で表現する。 例：値 58 に対し <i>ValueAsText Attribute</i> は、BYTE の場合 “58”、WORD の場合 “0058” と設定する。
19.		BCD16	UInt16	
20.		BCD32	UInt32	
21.		BCDx (x=4、12)	UInt16	
22.	実数型 (10 進表記)	REAL	Float	-
23.		LREAL	Double	-
24.	文字列型	STRING(x) (x は 1 以上 2048 以下の整数)	String	-
25.		STRING_U(x) (x は 1 以上 2048 以下の整数)	String または LocalizedText	OPC UA の <i>String DataType</i> は Unicode であるため、装置用 CSP+ の STRING_U(x) データ型からのマッピングが可能。ロケール ID (例：“en-US”) を指定する必要がある場合は <i>LocalizedText DataType</i> にマッピングする。
26.	時間型	TIME	Int32	variable の <i>EngineeringUnits Property</i> で単位 “ms” を指定し、 <i>ValueAsText Attribute</i> で “T#3d11h45m15s123ms” のような既定の表記を表現する。
27.	日付型	DATE	DateTime	-
28.	精度型	ACCURACY	Double	-
29.	IP アドレス型	IP_V4	String	値を文字列で表現する。
30.		IP_V4_64	String	値を文字列で表現する。
31.	集合型	データタイプ+ “()” [例] INT8()、 STRING(10)()	-	このデータ型を持つ装置用 CSP+ の要素に対し、本書ではマッピングルールを規定しないことから、 <i>DataType</i> のマッピングルールも規定しない。
32.	配列型	データタイプ+ “[] 要素数+ “]”	-	OPC UA では、配列を variable の <i>ValueRank Attribute</i> (配列の次元数) と <i>ArrayDimensions Attribute</i> (各次元の要素数) で表現する。

9. Profile とネームスペース

9.1. ネームスペースメタデータ

本仕様に関するネームスペースメタデータを表 9-1 に示す。この *Object* は、ネームスペースのバージョン情報と静的 *Node* を提供するために使用される。静的 *Node* は、値 *Attribute* を含むすべての *Server* のすべての *Attribute* で同様である。詳細は IEC 62541-5:2015 を参照とする。

ネームスペースメタデータに関する情報は、*NamespaceMetadataType* 型の *Object* として提供される。この *Object* は、*Server Object* の一部である *Namespace* *Object* のコンポーネントである。*NamespaceMetadataType ObjectType* とその *Property* は IEC 62541-5:2015 で定義されている。

バージョン情報は *UANodeSet XML ファイル* の *ModelTableEntry* の一部としても提供される。*UANodeset XML スキーマ* は、IEC 62541-6:2015 で定義されている。

表 9-1 NamespaceMetadata Object for this Specification

Attribute		Value	
BrowseName		http://opcfoundation.org/UA/CSPPlusForMachine/	
References	BrowseName	Data Type	Value
HasProperty	NameSpaceUri	String	http://opcfoundation.org/UA/CSPPlusForMachine/
HasProperty	NameSpaceVersion	String	1.00
HasProperty	NameSpacePublicationData	DateTime	2017-11-28
HasProperty	IsNamespaceSubset	Boolean	False
HasProperty	StaticNodeIdTypes	IdType[]	{Numeric}
HasProperty	StaticNumericNodeIdrange	NumericRange	Null
HasProperty	StaticStringNodeIdPattern	String	Null

9.2. OPC UA コンフォーマンスユニットと Profile

本節は、装置用 CSP+に関する OPC UA 情報モデルの profile、コンフォーマンスユニットを定義している。*Profile* とはコンフォーマンスユニットをグルーピングして名前を付けたものである。

Facet となる *Profile* は、OPC UA *Server* またはクライアントの完全な機能を定義するために他の *Profile* と組合わされることが期待される。

表 9-2 に装置用 CSP+情報モデルのコンパニオン仕様を実装した *Server* が利用できる *Facet* を示す。

表 9-2 CSP+ for machine Server Facet Definition

Conformance Unit	Description	Optional/ Mandatory
CSP+ for machine Information Model	Support <i>Objects</i> that conform to the types defined by this specification.	M
CSP+ for machine DeviceSet	Support the full component hierarchy with <i>CsppMachineType</i> below the <i>DeviceSet Object</i> defined in IEC 62541-100:2015	M
Profile		
BaseDevice_Server_Facet (defined in IEC 62541-100:2015)		M

表 9-3 に装置用 CSP+情報モデルのコンパニオン仕様を実装した *Client* が利用できる *Facet* を示す。

表 9-3 CSP+ for machine Client Facet Definition

Conformance Unit	Description	Optional/ Mandatory
CSP+ for machine Information Model	Support <i>Objects</i> that conform to the types defined by this specification.	M
CSP+ for machine DeviceSet	Support the full component hierarchy with <i>CsppMachineType</i> below the <i>DeviceSet Object</i> defined in IEC 62541-100:2015	M
Profile		
BaseDevice_Client_Facet (defined in IEC 62541-100:2015)		M

9.3. OPC UA ネームスペースのハンドリング

ネームスペースは、様々な命名機関の間で唯一の識別子を生成するために利用される。*Attribute* の *NodeId* と *BrowseName* は識別子である。UA の *Address Space* の *Node* は、*NodeId* を使って確実に識別される。*NodeId* とは異なり、*BrowseName* は、確実に *Node* を識別するために使用することはできない。異なる *Node* 同士がおなじ *BrowseName* を有するかもしれない。*BrowseName* は 2 つの *Node* 間のブラウズパスを構築したりあるいは標準の *Property* を定義するために使用される。

server は、*NodeId* と *BrowseName* に対して、同じネームスペースの利用を選択するかもしれない。しかしながら、例えばローカル server が標準の *Property* を与えたい場合、その *BrowseName* は、*NodeId* のネームスペースが何か他のものを反映しているとしても、標準化団体のネームスペースを持たなければならない。この仕様で定義されていない *Node* の *NodeId* はすべて、標準のネームスペースを使用してはならない。

表 9-4 は server で利用される必須、オプションのネームスペースリストである。

表 9-4 Namespace used in CSPPlusForMachine Server

NameSpace	Description	Optional/ Mandatory
http://opcfoundation.org/UA/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in the OPC UA specification. This namespace shall have namespace index 0.	M
Local Server URI	Namespace for <i>Nodes</i> defined in the local server. This may include types and instances used in a <i>Cspp Machine</i> represented by the server. This namespace shall have namespace index 1.	M
http://opcfoundation.org/UA/DI/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in [IEC 62541-100:2015]. The namespace index is server specific.	M
http://opcfoundation.org/UA/CSPPlusForMachine/	Namespace for <i>NodeIds</i> and <i>BrowseNames</i> defined in this specification. The namespace index is server specific.	M
Vendor specific types and instances	A server may provide vendor specific types like types derived from <i>CsppMachineType</i> or vendor specific instances of devices in a vendor specific namespace.	O

Table 9-5 provides a list of namespaces and their index used for *BrowseNames* in this specification. The default namespace of this specification is not listed since all *BrowseNames* without prefix use this default namespace.

表 9-5 Namespaces used in this specification

Namespace	Namespace Index	Example
http://opcfoundation.org/UA/	0	0:BaseObjectType
http://opcfoundation.org/UA/DI/	1	1:ParameterSet

付属書 A: ネームスペースとマッピング

A.1 装置用 CSP+情報モデルのネームスペースと識別子

この付録では本仕様で定義されるすべての数値型 *NodeId* の数値識別子を定義する。識別子は CSV ファイル形式で次の構文で規定されている。

〈 SymbolName 〉, 〈 Identifier 〉, 〈 NodeClass 〉

上記の‘SymbolName’とは、本仕様に登場する *Instance Node* でその *NodeId* の識別子が数値であるところの、タイプノードの BrowseName か *Instance Node* の BrowsePath を指す。

Instance Node のブラウズパスは、それを保持するインスタンスまたはタイプの *BrowseName* に対象 *Instance Node* の *BrowseName* を繋げることにより構成している。下線文字(‘_’)がパスの各 *BrowseName* の区切り文字として使われている。

全ての *NodeId* のネームスペース URI は以下で定義されている。

<http://www.opcfoundation.org/UA/CSPPlusForMachine/>

本バージョンの仕様に関する CSV は下記から入手できる。

<http://www.opcfoundation.org/UA/CSPPlusForMachine/1/NodeIds.csv>

最新の *NodeId* は下記から入手できる。

<http://www.opcfoundation.org/UA/CSPPlusForMachine/NodeIds.csv>

本仕様で定義される完全な情報モデルの電子化バージョンも提供されている。それは、IEC 62541-6:2015 で定義された XML 形式の情報モデルのスキーマにしたがう。本バージョンに関する情報モデルスキーマは下記から入手できる。

<http://www.opcfoundation.org/UA/CSPPlusForMachine/1/Opc.Ua.csplusformachine.Nodeset2.xml>

最新の情報モデルスキーマは下記から入手できる。

<http://www.opcfoundation.org/UA/CSPPlusForMachine/Opc.Ua.csplusformachine.Nodeset2.xml>

A.2 装置用 CSP+情報モデルの Profile URI

表 A-1 に装置用 CSP+情報モデルにおける profile の URI を定義する。

表 A-1 Profile URI

Profile	Profile URI
CSP+ for machine Server Facet	http://opcfoundation.org/UA-Profile/External/CSPPlusForMachine/CSP+formachineServer
CSP+ for machine Client Facet	http://opcfoundation.org/UA-Profile/External/CSPPlusForMachine/CSP+formachineClient

関連仕様書

なし。

