

# PACKML UNIT/MACHINE IMPLEMENTATION GUIDE

---

## *Part 2: Standard Representation of PackTags in an OPC UA Server*

### **摘要**

本文档“Standard Representation of PackTag in an OPC UA server”是 PackML unit/machine Implementation Guide 的第二部分。PackML Unit/Machine Implementation Guide 是在 ANSI/ISA TR88.00.02-2015 的基础上编写的最佳做法建议。该指南的宗旨是帮助知识水平各异的企业在 Units/Machines 上实施 PackML interface。本文档根据众多 OMAC 成员企业及其他用户使用 OPC UA 实施 PackML interface 的 Implementation 实例，介绍最佳的做法建议。

本文档是构建 PackML Unit/Machines 相关系列文档的第二部分。后续的文档将陆续推出，并侧重于 PackML Machine software code structure、PackML and Safety、PackML and Line Integration、PackML OEE and PackML HMI。

※此文翻译成中文时，尽可能地保留了英文原文的专用词汇。此中文翻译仅供参考。

Author: C. BOSTROM, L. SMITH, C. NØKLEBY, M. SCHLÜTER

Translated: Mitsubishi Electric Corporation

本页最初留空。

OMAC, The Organization for Machine Automation and Control  
11911 Freedom Drive  
Suite 600  
Reston, VA 20190 US

## 目录

1.	执行摘要 .....	4
1.1.	简介 .....	4
1.2.	目标 .....	4
1.3.	背景 .....	5
1.4.	编写中的相关 PackML 文档 .....	5
2.	术语与定义 .....	6
3.	变量、属性与特性 .....	9
3.1.	共同节点属性 .....	9
3.2.	MAPPING OF ELEMENTARY DATA TYPES .....	11
3.3.	ENUMERATED DATA TYPES .....	11
3.4.	变量节点属性 .....	12
3.5.	MAPPING OF STRUCTURED DATA TYPES .....	13
3.6.	MAPPING OF ARRAY DATA TYPES .....	18
4.	PackML BASE STRUCTURE DATA TYPES .....	20
4.1.	ALARM .....	20
4.2.	COUNT .....	20
4.3.	DESCRIPTOR .....	21
4.4.	EQUIPMENTINTERLOCK .....	21
4.5.	INGREDIENT .....	22
4.6.	INTERFACE .....	22
4.7.	PRODUCT .....	23
5.	TOP-LEVEL ORGANISATION .....	24
6.	参考信息 .....	29
6.1.	定义和缩写 .....	29
6.2.	参考资料 .....	30
6.3.	参与编写本指南的人员 .....	31
7.	附录 .....	32
7.1.	PackTag: 作为变量节点映射 COMPLEX DATA STRUCTURE (即 <i>NodeClass=Variable</i> ) ...	32
7.2.	PackTag: 作为对象节点映射 Complex data structure (即 <i>NodeClass=Object</i> ) .....	39
8.	版本记录 .....	46

# 1. 执行摘要

---

## 1.1. 简介

PackML OPC UA Working Group 目前正在为 PackML 制定 OPC UA 配套规范。此规范将于 2018 年完成，将规范化如何在 OPC UA 服务器中以完全定义的信息模型显示 PackML 信息。此模型将描述 PackML 状态机的转换规则，并为服务器显示的所有 PackML 数据提供额外的语义信息。

遗憾的是，现在并非所有的技术提供商都提供其中一些高级 OPC UA 功能，这些功能将是 PackML 配套规范所需要的。具体来说，方法处理与有限状态机建模并不具有广泛的支持。根据目前的预估，发布商业上可行的 PackML 配套规范实施所需的时间范围为 6 个月至 3 年不等。这些预估并不一定体现所涉及的技术障碍的性质，而是因为软件开发周期较长。

2010 年，OPC Foundation 会与 PLCopen 的一个联合工作小组发布了 [2] 一项代表 IEC 61131-3 体系结构模型的 OPC UA 信息模型<sup>1</sup>。尤其是，PLCopen 配套规范定义了如何在 OPC UA 地址空间表示（即“映射”）PLC 的变量。本文档将按这一通用模型的特例考虑如何以这种方式表示 PackTag。另外，虽然 PLCopen 文档在名义上适用于 Programmable Logic Controllers (PLC)，但我们认为，其中定义的与 PackTag 的表示有关的技术应同样适用于包装或相关行业中所用的任何类型的控制器（PC、PAC 等）。这包括不以 IEC 61131-3 语言为基础的平台。

本文档分为两个主要部分。第一部分，**§3 变量、属性与特性**，介绍 OPC UA 服务器内 PackTag 的基本要求。其余章节仅供参考，介绍如何将一整套 PackTag 构建成嵌套数据结构的一个序列。为了在 OPC UA 服务器中以正确的嵌套结构显示 PackTag，大多数技术提供上要求在 PLC 程序内同时显示类似结构。因此，在多数情况下，需要由机器生产商或 OEM（例如 PLC 程序设计人员）在应用程序级别小心地进行正确地实施。

有关 ISA-TR88.00.02-2015 的一般介绍，请参阅 [7]。有关 OPC Unified Architecture 的详细描述，请参阅 [5]。

## 1.2. 目标

这本 PackML implementation guide 将定义实施 OPC UA PackTag interface 的最低要求。例如，针对如何设置 OPC UA 服务器以满足处理 Admin、Command 与 Status PackTag 的要求，提供多个应用实例。它还介绍了如何完成不同的 OPC UA 实施。该指南提供示例，解释如何在 OPC UA 服务器内配置 PackTag。

最终用户和机器供应商都可以获得以下好处：

- 通过标准 OPC UA 服务器轻松集成 Unit/Machines
- 降低集成成本
- 从监控系统来看，连接 PackML Unit 的 OPC UA 接口相同
- 更快的接口规范
- 可靠的数据（例如 OEE、能源数据）
- 使用标准 OPC UA 接口更换生产线上的一个 Unit/Machine
- 等等

---

<sup>1</sup> 另请参阅 <https://opcfoundation.org/markets-collaboration/plcopen/>

### 1.3. 背景

OMAC 最终用户希望通过统一的接口与车间中的 Unit/Machine 交互。提供该统一接口有利于轻松集成监控控制系统。所有单元都使用类似的接口，而且这些单元都具有相同的数据结构接口。

根据 PackML State Model 应用统一的用户接口后，操作员即便在面临计划外停机时也能妥善应对。此外，所有单元都使用通用事件处理机制，有利于最终用户采用适用于所有单元的方式查明停机原因。

本文所述的接口基于 International Society of Automation (ISA) 定义的国际参考模型 PackML 构建。该接口遵循 ANSI/ISA TR88.00.02-2015 中所述的 PackML State Model。

阅读这本最佳做法建议之前必须对 TR88.00.02 技术报告有着基本了解和认识。事实上，我们强烈建议在阅读本文时将该报告放在手边，以便随时 TR88.00.02 技术报告。

Unit/Machine 的定义是执行一项或多项重要处理功能的一系列物理设备和控制功能。Unit/Machine 可以是单台机器或者整个包装生产线的一个子集。本文档中，Unit 和 Machine 两词可以互换。

机器供应商的定义是提供 PackML 兼容 Unit/Machine 的公司或组织。最终用户的定义是在生产设施中使用 PackML 兼容 Unit/Machine 的公司或组织。

### 1.4. 编写中的相关 PACKML 文档

本文档是指导企业打造 PackML Unit/Machine 系列文档的第二部分。后续的文档<sup>2</sup> 将陆续推出，并侧重于：

- Part 1: PackML Interface State Manager (可用)
- Part 2: Standard Representation of PackTag in an OPC UA Server (可用)
- Part 3: PackML Machine Software Code Structure (正在编写)
- Part 4: PackML Network Connections (正在编写)
- Part 5: PackML and Safety (正在编写)
- Part 6: PackML and Line Integration (正在编写)
- Part 7: PackML User Interface - HMI (正在编写)

---

<sup>2</sup>列表内容不完整，文档标题可能会有所更改。

## 2. 术语与定义

---

**AddressSpace:** OPC UA 服务器为客户提供的对象与相关信息的集合叫做其 AddressSpace。

**Attributes:** OPC UA 定义八个不同的 NodeClass。分别是：Object、Variable、Method、View、ObjectType、VariableType、DataType 和 ReferenceType。每个 NodeClass 包含一组固定的信息，称之为属性。所有 NodeClass 共有的属性叫做 Base Attribute（基本属性），分别是：NodeId、NodeClass、DisplayName、Browse Name、Description、WriteMask 和 UserWriteMask。分别定义如下。

**BrowseName:** 用于在内部标记节点的固定（即非本地化）字符串。客户在浏览时使用 BrowseName 识别节点。

**Description:** 用于提供与节点有关的额外信息的本地化字符串。

**DisplayName:** 用于在用户界面上显示节点名称的本地化字符串。此处的“本地化”意思是 DisplayName 可能因用户的本地（或配置的）语言而异。

**NodeClass:** NodeClass 属性包含与节点本身的用途或固有功能相关的信息。定义一组固定的枚举值：Object、Variable、Method、ObjectType、VariableType、ReferenceType、DataType 和 View。

**NodeId:** 一个 AddressSpace 内的每个节点以其 NodeId 作为唯一标识，而 NodeId 是由 NamespaceIndex (UInt16)、IdentifierType（数值、字符串、GUID 或 Opaque）以及标识符组成的。为标识符指定的值的格式取决于 IdentifierType。

**Nodes:** OPC UA 以节点标识 AddressSpace 内的特征。每个节点都有一个不同的 NodeClass，NodeClass 的主要类型有：Object、Variable 和 Method。例如，方法以 NodeClass Method 的一个节点标识。AddressSpace 内的每个节点以其 NodeId 作为唯一标识。为方便起见，我们有时将具有 NodeClass Object 的节点称为 *Object* 节点或者将具有 NodeClass Variable 的节点称为 *Variable* 节点。

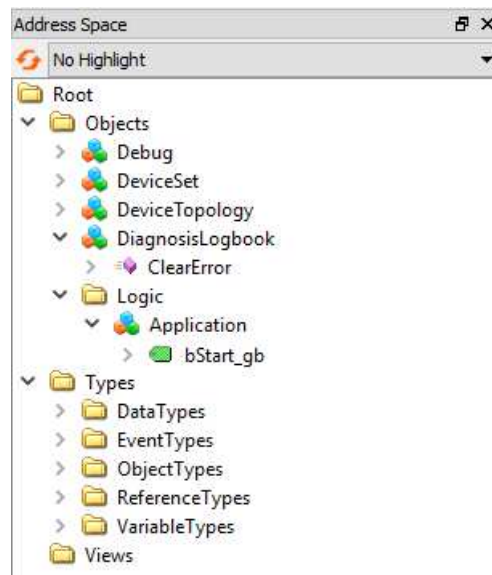


图 1 OPC UA AddressSpace 示例。所示的每个项目都是一个节点。节点之间的引用构成了自然的层次结构。例如，FolderType 对象 Root 具有指向 Objects、Types 和 Views 的引用（“组织”）。同样，DiagnosisLogbook 包含对 ClearError 方法的引用，而 Application 包含对 bStart\_gb 变量的引用。

**Object:** OPC UA 使用对象提供数据与相关事件或方法的正式表示。对象可以通过引用进行链接。支持著名的面向对象的类型层次结构和继承概念。

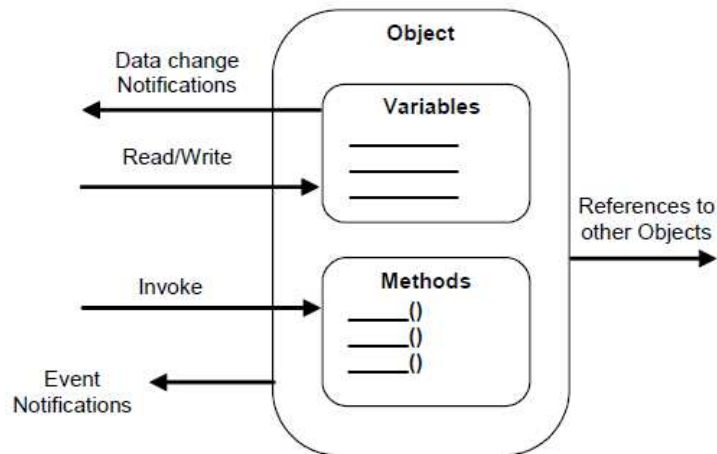


图 2 OPC UA 对象模型将变量、方法等封装在单一正式实体中。还定义了对其他对象的引用。如需更多信息，请参阅 [4]，§ 4.2. 经许可使用的图。

**PackTag:** 如 ISA-TR88.00.02-2015 中的定义所述，PackTag 为包装机器中常用的数据元素提供一套命名约定。*UnitName.Status.CurMachSpeed* 是 PackTag 的一个示例，在这种情况下指的是机器的当前速度，以每分钟内包装数为单位。单一值，即 Simple type 的各 PackTag 都具有明确定义的 IEC-61131-3 的数据类型（DINT、REAL、STRING 等）。互相具有自然联系的 PackTag 被分组到数据结构（或 *complex type*）中。例如，*UnitName.Status.EquipmentInterlock* 是一种 Complex type，它具有个体元素 *UnitName.Status.EquipmentInterlock.Blocked* 和 *UnitName.Status.EquipmentInterlock.Starved*。*Blocked* 和 *Starved* 均是 Boolean 元素，因此它们是 Simple type。

要注意的是作为一个现实问题，PackTag 可能被视为 PLC 或 PC 进程控制器的变量的保留名称。本文档的目的就是明确说明如何在 OPC UA 地址空间中表示这些变量。

PackTag 的完整列表见附录。

**References:** AddressSpace 内的节点可能通过引用而互相关联。支持分层和不分层引用类型。引用提供了用于表示类型层次结构和继承概念的正式机制。

**Root Node:** 每个 AddressSpace 都具有叫做其根的唯一节点。一个地址空间内的所有节点都是通过一个引用序列直接或间接地链接到根。

**UserWriteMask:** UserWriteMask 显示目前连接至服务器的用户可以修改哪些节点属性。

**Variable:** 计算机程序设计人员使用 *Variable* 一词表示与内存地址有关的符号名称。根据我们此处的目的，变量始终包含一种类型，它限制变量可代表的信息的种类。例如，如果 myBool 是一种 Boolean 变量，那么 myBool 既可以是 TRUE，也可以是 FALSE，但是它不会取整数值 123。在 PLC 编程中，变量可以与术语标签互换使用。Variable 也可以用在 OPC UA 上下文中，用于描述特定的 NodeClass。如同我们将在下一节看到的一样，所有单 Single-valued programming variable，尤其是所有 Elementary type 的 PackTag 都会与 NodeClass Variable 的节点相关联。

**Unit/machine:** ISA-TR88.00.02-2015 使用“Unit/Machine”指代一台自动化机器，具体来说，一台包装机。我们为每一台 Unit/Machine 指定了一组 PackTag。每一组都与一种名为 UnitName 的字符串标识符相关联。UnitName 可被视为顶级 PackTag。

**WriteMask:** WriteMask 指示 OPC UA 客户可以修改哪些节点属性。



## 3. 变量、属性与特性

本节内容紧接 [2]，§ 5 中提供的材料。关键主题是在 simple 或 Single-valued programming variable、Complex 或用户定义的结构化数据和阵列的 OPC UA AddressSpace 内进行建模。这些规则使我们能够以一致的方式在 OPC UA AddressSpace 内提供一组完整的 PackTag。

注意，虽然 [2] 是专为 Programmable Logic Controllers 编写的，但我们认为，其中锁定的变量的建模规则应广泛应用，以将嵌入式 PC 或其他非 PLC 平台包含在内。与这些设备类型中任何一种相关联的 OPC UA 服务器应使用与此处介绍的 PLC 方法相同的方式提供 PackTag。

### 3.1. 共同节点属性

所有 NodeClass 共有的属性分别是：NodeId、NodeClass、BrowseName、DisplayName、Description、WriteMask 和 UserWriteMask。对每一种属性的要求如下所述。

#### 3.1.1. NODEID

指定 AddressSpace 中的每个节点都有一个标识符，叫做 NodeId。PLCopen 配套规范 [2] 未对此结构作任何要求。图 3 所示的示例展示了技术提供商如何处理该属性。注意，标识符类型是 String，相关的标识符字符串则以规范的方式体现了嵌套式 PackTag 结构。这可以视为最佳做法，强烈建议用户遵循这一模式。

Attribute	Value
▼ NodeId	NodeId
NamespaceIndex	2
IdentifierType	String
Identifier	Application.PackMLObjects.Machine_01.Status.CurMachSpeed

图 3 NodeId 定义示例。标识符的字符串的格式并不规范，且因供应商而异。在这种情况下，相关的 PackTag 是 Machine\_01.Status.CurMachSpeed。

#### 3.1.2. NODECLASS

所有 Elementary type (DINT、REAL、STRING 等) 的 PackTag 将以 NodeClass 变量的节点在 AddressSpace 中表示。Elementary type 的 PackTag 示例：UnitName.Status.CurMachSpeed (REAL)、UnitName.Admin.Parameter[#].Unit (STRING) 或者 UnitName.Status.StateCurrent (DINT)。

建议也使用变量的节点表示 Complex type，但是也可以仅将这些表示为对象。将 Complex type 表示为变量有利于 UA 客户在单次调用中读取底层 complex 数据。附录 1 和 2 中比较了 Complex type 建模的两种基本方法。§ 3.5 Mapping of structure data types 中也详细讨论了这一主题。

#### 3.1.3. BROWSENAME

PackTag 的 BrowseName 包含指定嵌套结构内最内层成员的名称。

例如，PackTag UnitName.Status.CurMachSpeed 的 BrowseName 必须包含字符串“CurMachSpeed”。

BrowseName 还包括对 Namespace index 的引用。对于 PackTag，Namespace index 是服务器/供应商特有的；本文档对 Namespace index 没有任何额外限制。请参考 [2]，§ 7.3 了解更多信息。

#### 3.1.4. DISPLAYNAME

DisplayName 是用于显示节点名称的本地化字符串。由于 ISA-TR88.00.02-2015 不描述本地化 PackTag 名称，所以建议 DisplayName 匹配上述 BrowseName。

#### 3.1.5. DESCRIPTION

Description 是一种可选属性，用于提供与节点有关的额外信息。并未指定其用途。

#### 3.1.6. WRITEMASK

WriteMask 是一种可选属性，描述客户可以更改哪些节点属性。并未指定其用途。

#### 3.1.7. USERWRITEMASK

UserWriteMask 是一种可选属性，描述特定用户可以更改哪些节点属性。并未指定其用途。

### 3.2. MAPPING OF ELEMENTARY DATA TYPES

所有 single-valued PackTag 由以下 IEC 61131-3 Elementary type 构成: BOOL、DINT、STRING 和 REAL。下表展示了这些类型与它们的 OPU UA built in data type 的映射。关于所有 IEC 61131-3 类型的完整映射, 请参考 [2], 表 26。

表 1 ISA-TR88.00.02、IEC 61131-3 与 OPC UA Elementary type 的比较。

<i>ISA-TR88.00.02 elementary data types</i>	<i>IEC 61131-3 elementary data types</i>	<i>OPC UA built in data types</i>	<i>Comment</i>
Boolean	BOOL	Boolean	单一比特值 (True 或 False)
Integer (32 位)	DINT	Int32	32 位带符号整数值
实数 32 位 IEEE 754 标准 浮点值格式	REAL	Float	OPC UA 定义: IEEE-754 单精度 (32 位) 浮点值。  IEC 61131-3 定义: 实数 (32 位), 为基本的单一幅度的浮动小数点格 式, 其值在 IEC 60559 规定的范围内
String、Null 结尾的 ASCII, 大小任意, 但不 超过 80 个字符	STRING	String	PLCopen 特定 OPC UA Simple data type STRING 源自于 Built in data type STRING。它描述用作可变长度 单字节字符串的类型。

### 3.3. ENUMERATED DATA TYPES

ISA-TR88.00.02-2015 并未明确定义任何 enumerated data type, 因此, 所有 Single-valued PackTag 将根据其 Elementary type 进行映射。包括 Enumerated data type 可能为有效的 PackTag。具体而言, 这意味着 *UnitMachine.Status.StateCurrent*、*UnitMachine.Status.UnitModeCurrent*、*UnitMachine.Command.CntrlCmd* 应作为 Simple DINT 的值映射, 而不是它们相关的标准枚举。

### 3.4. 变量节点属性

如前文所述，OPC UA 服务器中所有表示 Single-valued PackTag 的节点都具有 NodeClass=Variable。OPC UA 将这一 NodeClass 的固定属性集定义如下：Value、DataType、ValueRank、ArrayDimensions、AccessLevel、UserAccessLevel、MinimumSamplingInterval 和 Historizing。表 1 对每一种属性进行了定义，并针对它们在 PackTag<sup>3</sup> 中的使用提供了相关的额外信息。

表 2 变量 NodeClass 的属性

属性	用途	DataType	说明
Value	Mandatory	见 DataType	节点的活动值。忽略更新延迟，我们希望该寄存器与 PLC、PC 或 PAC 控制器中的相关 PackTag 的值相等。
DataType	Mandatory	NodeId	Value 属性的 DataType。对于 single-valued PackTag，仅使用四个 BaseDataTypes：Boolean [1]、Int32 [6]、Float [10] 和 String [12]。在这里，标准 OPC UA 节点设置定义了数值标识符。对于 complex data，必须定义 BaseDataType>Structure 的相应子类型。
ValueRank	Mandatory	Int32	ValueRank 指示 Value 属性是否为一个阵列。如果 Value 是一个标量（即不是阵列），那么 ValueRank = -1。否则，ValueRank 将被指定为 Value 阵列的维数。所有 PackTag 的 ValueRank 均等于 -1、1 或 2。
ArrayDimensions	Optional	UInt32[]	ArrayDimensions 指示每个阵列维度的长度。如果 Value 是一个标量，则 ArrayDimensions 为未定义状态或 Null。
AccessLevel	Mandatory	Byte	AccessLevel 指示 Value 属性是否可读取和/或可写入。Command 集中包含的所有 PackTag 必须均可写入。Admin 以及 Status 集中包含的所有 PackTag 必须均可读取。如果支持 Historizing（见下文），还可以定义对历史数据流的访问。
UserAccessLevel	Mandatory	Byte	AccessLevel 的用户特定设置。供应商可能视需要限制指定用户对 PackTag 的读取或写入访问。

<sup>3</sup> 参考 [2] § 5.3 了解关于一般情况的额外信息。

属性	用途	DataType	说明
MinimumSamplingInterval	Optional	Duration	MinimumSamplingInterval 指示 OPC UA 服务器特定节点的最小更新速度（以毫秒为单位）。MinimumSamplingInterval=0 表示伺服器连续监控项目。  MinimumSamplingInterval=-1 表示抽样率未确定。  该属性的值是服务器特定的。
Historizing	Mandatory	Boolean	指示服务器目前是否正在收集变量值的历史记录。对值历史记录的支持是服务器特定的。 PackTag 的 Historizing 不是必需项。

### 3.5. MAPPING OF STRUCTURED DATA TYPES

Complex type 或 Structured data type 的建模有两种可以接受的方法。首选方法是将 Complex type 建模为变量节点。这种节点的相关 Value 属性包括一整套底层数据。这一属性值可由 OPC UA 客户在单次调用中读取。另一种方法是将 Complex type 建模为对象节点。在这种情况下，这种对象节点只有组织功能：它引用结构数据本身的所有子类型。接下来我们详细介绍每一种方法。

#### 3.5.1. 变量方法

如果将 Complex data type 作为一个变量节点映射，那么相关 IEC 61131-3 Structure data type 应作为 OPC UA DataType Structure 的一个子类型映射。见图 4。这种 OPC UA type 将被指定为节点的 DataType 属性。注意，OPC UA 服务器必须明确地描述如何对 Structured DataType 进行编码，并向客户提供这一信息以正确地读取或写入数据。OPC UA 服务器必须至少支持二进制编码（“默认二进制”）。如需更多信息请参阅 [2]，§ 5.2.3.4。

使用变量方法时，用于构建整套 PackTag 的全部数据结构都将定义为 Standard type *Structure* 的子类型。标题为 *PackML Base Structure Data Types* 的后续章节中提供了完整的 PackML base type 清单。

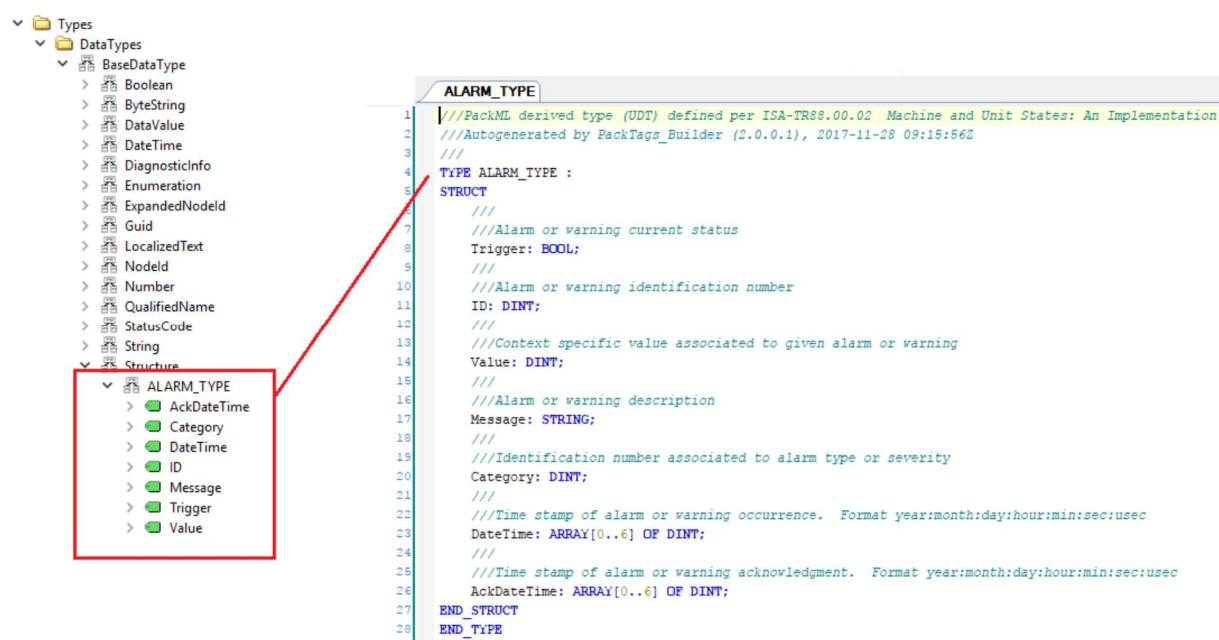


图 4 将 PackML 基本结构映射到 OPC UA 结构子类型。

建议 OPC UA 服务器也提供每个结构的组成部分数据作为一组子变量。这可以使不支持复杂数据的 OPC UA 客户访问每个单独的标量值。图 4 和 5 展示了如何映射示例数据结构及其单独的元素。注意，这种方法需要另外将每个 IEC 61131-3 Structure 建模为变量类型。示例直接取自 [2] 中。

我们将 IEC61131-3 Data type 定义如下：

```
TYPE ExampleIEC611313Structure:
STRUCT
    IntStructureElement: INT;
    RealStructureElement: REAL;
    BoolStructureElement: BOOL;
END_STRUCT;
END_TYPE
```

将与 C# 同等的数据结构进行定义：

```
Struct ExampleIEC61131-3
{
    public short IntStructureElement;
    public float RealStructureElement;
    public bool BoolStructureElement;
}
```

图 5 提供了相关的映射 OPC UA。

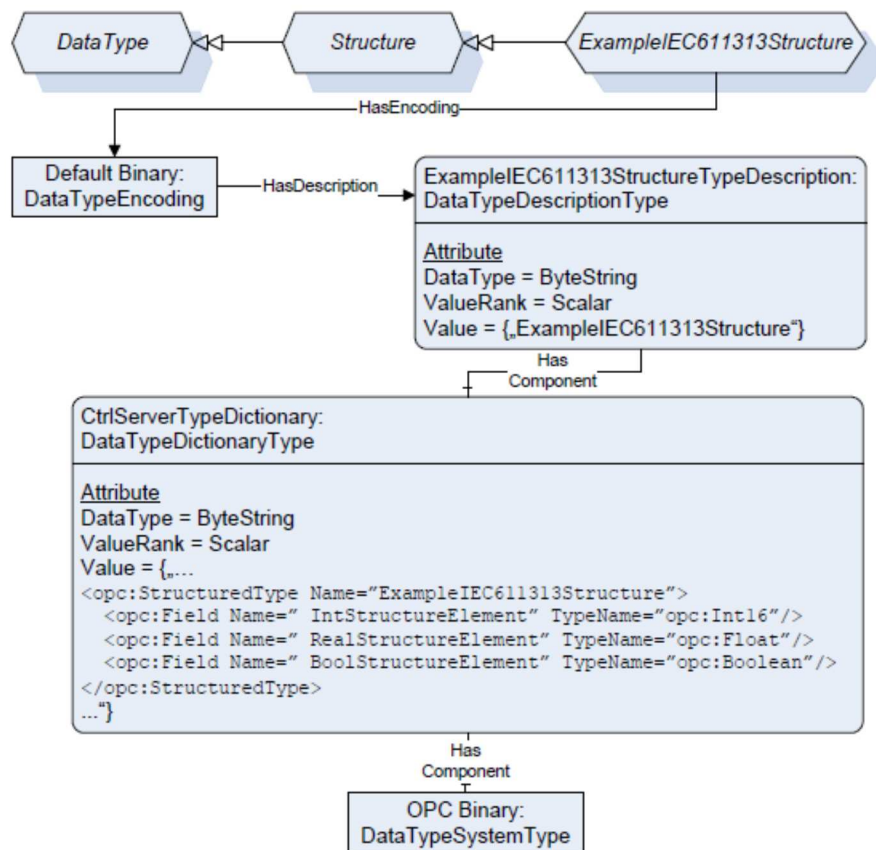


图 5 映射 Sample structure data type。

图 6 展示了进一步将结构映射到子变量。

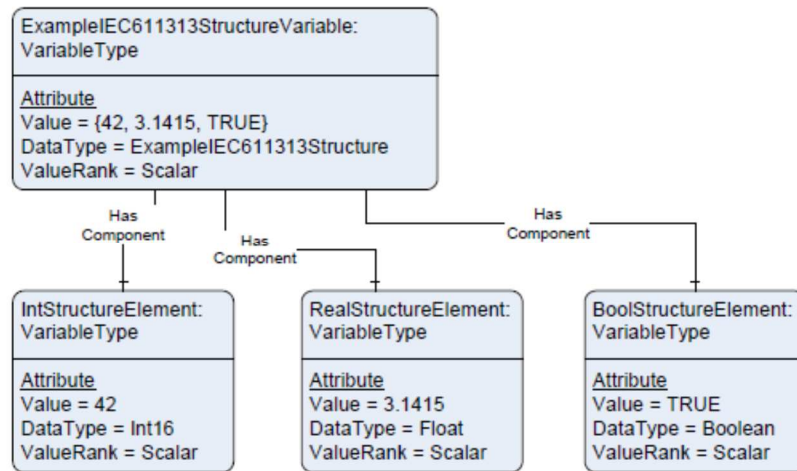


图 6 将 Structure data type 映射到单独的组成部分

举个具体的例子，*UnitName.Status.EquipmentInterlock* 被定义为类型 EQUIPMENTINTERLOCK type 的 PackTag。参考 § 4.4 EQUIPMENTINTERLOCK。为了将此 PackTag 建模为 VariableNode，我们定义了结构的子类型，如图 7 所示。

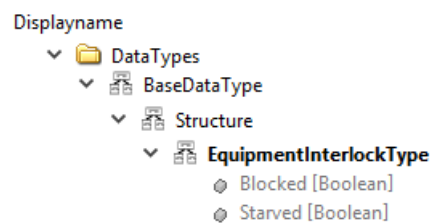


图 7 来源于 BaseDataType>Structure 的 EquipmentInterlockType 以 Boolean 的组成部分 Blocked 和 Starved 进行定义。

此外，我们对 VariableType，特别是 BaseDataVariableType 的子类型进行定义，如下所示：

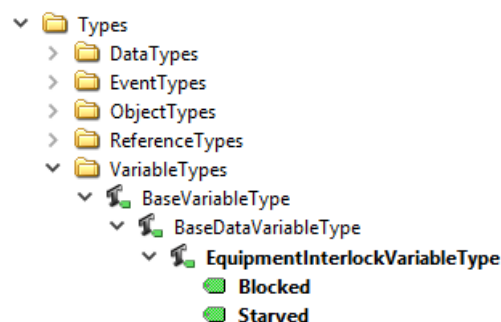


图 8 映射作为变量类型的 PackML EQUIPMENTINTERLOCK type。EquipmentInterlock VariableType 的组成部分 Blocked 和 Starved 均是 Boolean。

变量节点本身被定义为 TypeDefinition=EquipmentInterlockVariableType、DataType=EquipmentInterlockType，如图 9 所示。



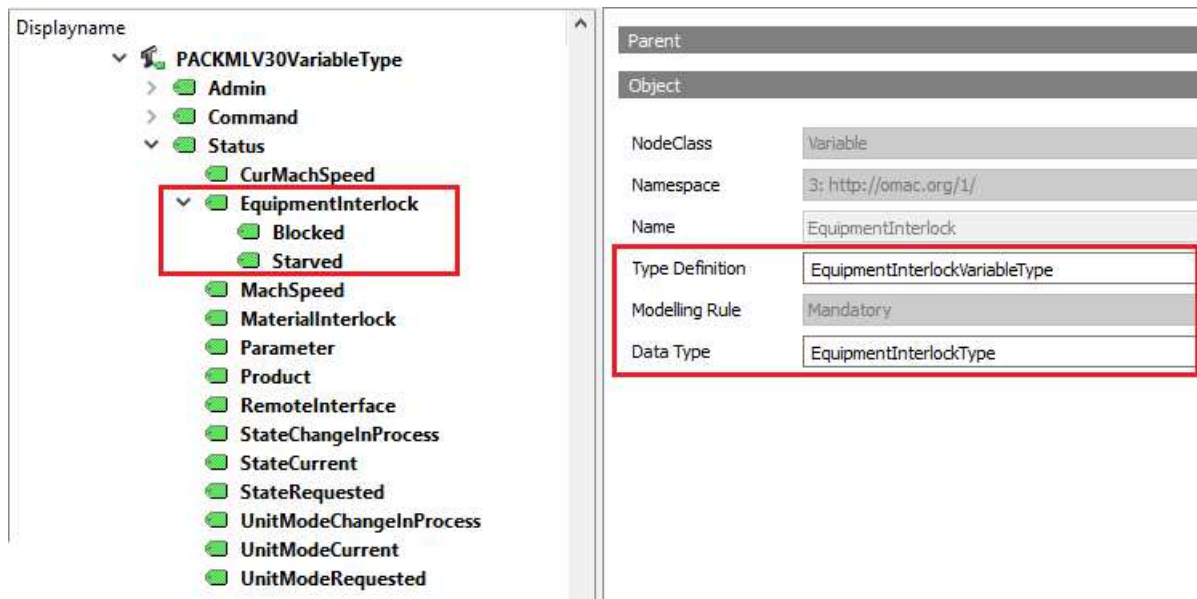


图 9 映射作为变量节点的 `UnitName.Status.EquipmentInterlock`。在这里，`EquipmentInterlockVariableType` 和 `EquipmentInterlockType` 相似，但是 `EquipmentInterlockVariableType`（源自于 `BaseVariableType`）适合作为变量节点的 `TypeDefinition`，而 `EquipmentInterlockType`（源自于 `BaseDataType`）适合作为其相关 `Value` 的 `DataType`。

### 3.5.2. 对象方法

Complex data 也可建模为对象节点。这种方法允许在 OPC UA 服务器内将包含复杂数据的子类型分组在一起。但是，如前文所述，在这种情况下，OPC UA 客户在单次调用中无法读取底层 Complex data。相反，必须单独读取每个基本子类型。将 ISA-TR88.00.02 Complex type 建模为一个对象需要定义类似的 OPC UA `ObjectType`。例如，回到上一节所举的例子，如果将 `UnitName.Status.EquipmentInterlock` 建模为一个对象节点，那么应定义以下 `ObjectType`：

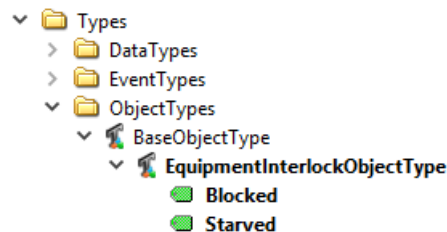


图 10 将 PackML type EQUIPMENTINTERLOCK 作为 `ObjectType` 映射。

节点本身将被定义为 `EquipmentInterlockObjectType`。

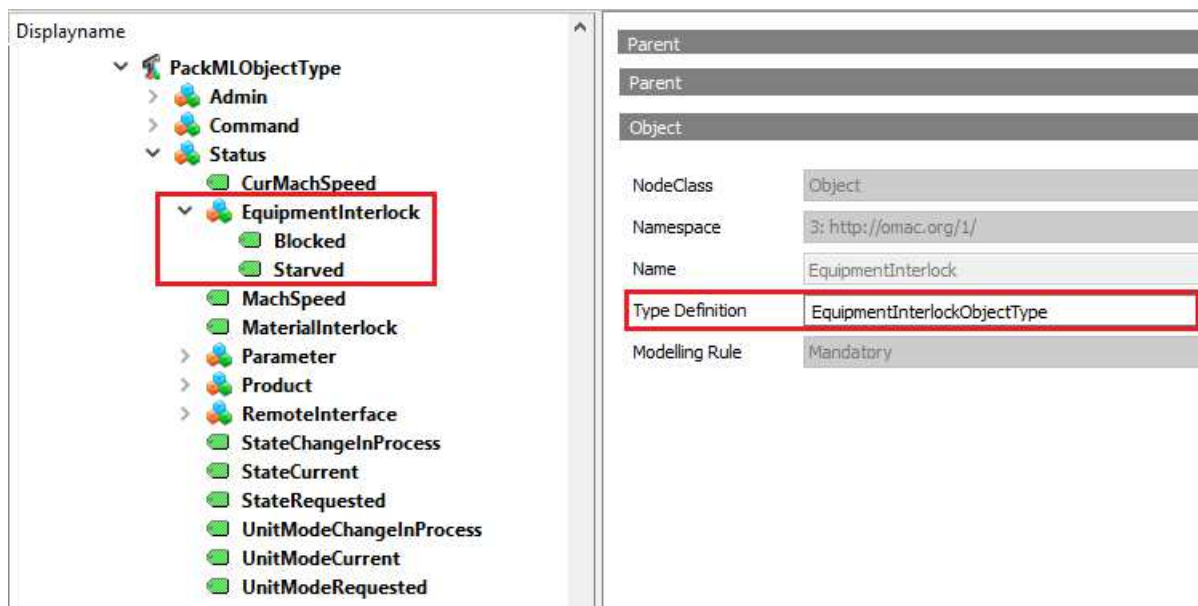


图 11 将 complex PackTag 作为对象节点映射需要定义合适的 ObjectType。注意，在这种情况下，没有与节点本身相关的值。只可以读取基本子类型（Blocked 和 Starved）。

### 3.6. MAPPING OF ARRAY DATA TYPES

OPC UA 允许通过对 ValueRank 和 ArrayDimensions 的属性进行相应的定义将任何变量节点视为一个阵列。例如，如果指定的变量节点表示一个一维阵列，那么将 ValueRank 设为 1 并将 ArrayDimensions[0] 设为阵列本身中的元素数。OPC UA 定义所有阵列从索引 0 开始。

IEC 61311-3 的阵列方法更加灵活。具体来说，用户可以自由定义阵列索引的上限和下限。为了保留这一灵活性，[2] 添加了对 Property type Dimensions、IndexMax 以及 IndexMin 的额外引用，其中 Dimensions 表示阵列的维数，IndexMax 和 IndexMin 分别表示阵列索引的上限和下限。

例如，使用标准 IEC 61311-3 符号，我们可以将一个二维阵列定义如下：

```
VAR MyArray : ARRAY [1..2, 3..5] OF DINT; END_VAR
```

此阵列包含元素 {MyArray[1,3], MyArray[1,4], MyArray[1,5], MyArray[2,3], MyArray[2,4], MyArray[2,5]}。在 [2] 之后，OPC UA 服务器中相应的变量节点包括特性 Dimensions、IndexMax 以及 IndexMin，其值如下：

```
Dimensions.Value = 2
IndexMin.Value[0] = 1, IndexMax.Value[0] = 2,
IndexMin.Value[1] = 3, IndexMax.Value[1] = 5
```

下方所示为 [2] 中定义的一般情况。

表 3 一般情况定义

属性	Value
BrowseName	<Variable Name> 或 <IEC Data Type Name>

表 4 任何 VariableType 或 Data Type Node 的实例

References	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
HasProperty	Variable	Dimensions	UInt32	PropertyType	Mandatory
HasProperty	Variable	IndexMin	UInt32 [ ]	PropertyType	Mandatory
HasProperty	Variable	IndexMax	UInt32 [ ]	PropertyType	Mandatory

注意，ISA-TR88.00.02-2015 不指定 PackTag 接口中包含的大部分阵列的上限和下限。在这些情况下，只要 IndexMin.Value <= IndexMax.Value，就可以随意定义 IndexMin 和 IndexMax。作为对比，必须将 DateTime 和 AckDateTime<sup>4</sup> 定义为 IndexMin.Value = 0 且 IndexMax.Value = 6 的 DINT 阵列。

一些用户可能希望通过实施所有阵列从索引 0 开始这一规则来简化 PackTag 接口。其他用户可能会被迫遵守这一规则。例如，C# 从索引 0 开始所有阵列。在这种情况下，[2] 定义的其他特性从本质上是冗余的，在非 PLC 实施中可以安全地忽略。

<sup>4</sup>DateTime 和 AckDateTime 均是 § 4.1 中定义的 ALARM 的元素。

## 4. PACKML BASE STRUCTURE DATA TYPES

---

ISA-TR88.00.02-2015 定义了构建整套 PackTag 所需的 7 种基本 Structure data type。如下所示。我们的符号遵循 IEC 61131-3 所用的规范来定义 Complex data type。注释会在前方添加“///”。注意，类型名称本身不规范。

### 4.1. ALARM

Alarm type 包含描述个别 alarm 或 warning 所需的信息。该类型的 PackTag 包括 *UnitName*.Admin.Alarm[#]、*UnitName*.Admin.AlarmHistory[#]、*UnitName*.Admin.Warning[#] 等。

TYPE ALARM:  
STRUCT

```
///
///Alarm or warning current status
Trigger: BOOL;
///
///Alarm or warning identification number
ID: DINT;
///
///Context specific value associated to given alarm or warning
Value: DINT;
///
///Alarm or warning description
Message: STRING;
///
///Identification number associated to alarm type or severity
Category: DINT;
///
///Time stamp of alarm or warning occurrence. Format year:month:day:hour:min:sec:µsec
DateTime: ARRAY[0..6] OF DINT;
///
///Time stamp of alarm or warning acknowledgment. Format year:month:day:hour:min:sec:µsec
AckDateTime: ARRAY[0..6] OF DINT;
```

END\_STRUCT  
END\_TYPE

### 4.2. COUNT

Count type 包含追踪指定操作发送次数所需的信息。操作本身也可以进行描述。该类型的 PackTag 有 *UnitName*.Admin.ProdConsumedCount[#] 和 *UnitName*.Admin.ProdProcessedCount[#]。

TYPE COUNT:  
STRUCT

```
///
///Product identification number
ID: DINT;
///
///Product description
Name: STRING;
///
```

```

        ///Product unit of measure
        Unit: STRING;
        ///
        ///Count since last operator reset
        Count: DINT;
        ///
        ///Count since last accumulative counter reset
        AccCount: DINT;
    END_STRUCT
END_TYPE

```

### 4.3. DESCRIPTOR

Descriptor type 用于描述 Unit/Machine 中的参数。该类型的示例有 *UnitName*.Status.Parameter[#] 和 *UnitName*.Command.Product[#].Ingredients[#].Parameter[#]。

```

TYPE DESCRIPTOR:
STRUCT
    ///
    ///Parameter identification number
    ID: DINT;
    ///
    ///Parameter description
    Name: STRING;
    ///
    ///Parameter units of measure
    Unit: STRING;
    ///
    ///Numeric value of parameter
    Value: REAL;
END_STRUCT
END_TYPE

```

### 4.4. EQUIPMENTINTERLOCK

Equipment interlock type 在 Unit/Machine Blocked 或 Starved 时发出信号。该类型的唯一示例为 *UnitName*.Status.EquipmentInterlock。

```

TYPE EQUIPMENTINTERLOCK:
STRUCT
    ///
    ///Machine stoppage due to downstream fault
    Blocked: BOOL;
    ///
    ///Machine stoppage due to upstream or supply fault
    Starved: BOOL;
END_STRUCT
END_TYPE

```

## 4. 5. INGREDIENT

Ingredient type 包含描述指定产品配方中所用原材料所需的标签集合。该类型的示例为 *UnitName.Command.Product[#].Ingredients[#]*。

TYPE INGREDIENT:

STRUCT

```
///  
///Ingredient identification number  
IngredientID: DINT;  
///  
///Parameters associated to given ingredient  
Parameter: ARRAY[] OF DESCRIPTOR;
```

END\_STRUCT

END\_TYPE

## 4. 6. INTERFACE

Interface type 包括用于协调机器之间 State machine transition 的数据。也可能交换参数信息。该类型的示例为 *UnitName.Command.RemoteInterface[#]*。

TYPE INTERFACE:

STRUCT

```
///  
///Identification number of upstream or downstream unit machine  
Number: DINT;  
///  
///Control command for upstream or downstream machine  
ControlCmdNumber: DINT;  
///  
///Control command value associated with ControlCmdNumber  
CmdValue: DINT;  
///  
///Parameter data required for peer-to-peer coordination  
Parameter: ARRAY[] OF DESCRIPTOR;
```

END\_STRUCT

END\_TYPE

## 4. 7. PRODUCT

Product type 包含制作制定产品所需的所有标签，这包括工艺特定数据以及所有配方数据。该类型的示例有 *UnitName.Command.Product[#]* 和 *UnitName.Status.Product[#]*。

TYPE PRODUCT:

STRUCT

```
///  
///Product identification number  
ProductID: DINT;  
///  
///Parameters required for processing given product  
ProcessVariables: ARRAY[] OF DESCRIPTOR;  
///  
///Ingredients required for processing given product  
Ingredients: ARRAY[] OF INGREDIENT;
```

END\_STRUCT

END\_TYPE

## 5. TOP-LEVEL ORGANISATION

---

PackTag 包含一组嵌套数据类型，在 ISA-TR88.00.02-2015 将其顶级定义为 **PACKMLV30**。PACKMLV30 有三个组成部分，Command、Status 和 Admin，分别定义为子类型 PMLc、PMLs 和 PMLa。这些类型如下所示。注意 ISA-TR88.00.02-2015 并未要求在指定机器上实施 PMLc、PMLs 和 PMLa 的所有组成部分。PMLc、PMLs 和 PMLa 的强制组成部分**加粗**显示。

TYPE PACKMLV30 :

STRUCT

```
///
///Command data produced by the 'Information Sender' (i.e. HMI, machine peers, etc.)
Command: PMLc;
///
///Status data produced for 'Information Sender' (i.e. HMI, machine peers, etc.)
Status: PMLs;
///
///Administration data produced for higher level supervisory system.
Admin: PMLa;
```

END\_STRUCT

END\_TYPE

TYPE PMLa :

STRUCT

```
///
///Parameters associated to the local interface (e.g. HMI)
Parameter: ARRAY[] OF DESCRIPTOR;
///
///Listing of all currently active alarms
Alarm: ARRAY[] OF ALARM;
///
///Extent of the Alarm array
AlarmExtent: DINT;
///
///Chronological alarm history, with most recent given as AlarmHistory[0]
AlarmHistory: ARRAY[] OF ALARM;
///
///Extent of AlarmHistory array
AlarmHistoryExtent: DINT;
///
///Alarm associated to the primary cause of machine stoppage ('first fault') (required)
StopReason: ALARM;
///
///Extent of StopReason array
StopReasonExtent: DINT;
///
///Listing of all currently active warnings
Warning: ARRAY[] OF ALARM;
///
///Extent of warning array
WarningExtent: DINT;
///
///Elapsed time in seconds of current mode.
ModeCurrentTime: ARRAY[] OF DINT;
```



```

///
///Cumulative time in seconds in each defined mode.
ModeCumulativeTime: ARRAY[] OF DINT;
///
///Elapsed time in seconds of current [mode, state] pair.
StateCurrentTime: ARRAY[,] OF DINT;
///
///Cumulative time in seconds in each defined [mode, state] pair.
StateCumulativeTime: ARRAY[,] OF DINT;
///
///Count of raw materials used or consumed during production
ProdConsumedCount: ARRAY[] OF COUNT;
///
///Count of total products (i.e. good + defective) made during production (required)
ProdProcessedCount: ARRAY[] OF COUNT;
///
///Count of defective products made during production. (required)
ProdDefectiveCount: ARRAY[] OF COUNT;
///
///Accumulative time since last reset of timer arrays.
AccTimeSinceReset: DINT;
///
///Maximum design speed of the machine within given environment.
MachDesignSpeed: REAL;
///
///Bitwise indication of inactive states in current mode.
StatesDisabled: DINT;
///
///Current date and time of local processing unit (PLC, PC, PAC, microcontroller, etc.)
PLCDateTime: ARRAY[0..6] OF DINT;

END_STRUCT
END_TYPE

TYPE PMLc :
STRUCT

///
///Numeric representation of the commanded unit mode (required)
UnitMode: DINT;
///
///Signals request that the current value of Command.UnitMode be activated (required)
UnitModeChangeRequest: BOOL;
///
///Machine speed setpoint in units of [primary packages/min] (required)
MachSpeed: REAL;
///
///Bitwise indication that materials are ready for processing.
MaterialInterlock: ARRAY[1..32] OF BOOL;
///
///Numerical representation of the state commands (1=Reset, 2=Start, etc.) (required)
CntrlCmd: DINT;
///
///Signals request that the current value of Command.CntrlCmd be activated (required)
CmdChangeRequest: BOOL;
///

```

```

    ///Use to coordinate activities between peers (i.e. upstream or downstream unit machines.)
    RemoteInterface: ARRAY[] OF INTERFACE;
    ///
    ///Values or parameters associated to the end user supervisory interface.
    Parameter: ARRAY[] OF DESCRIPTOR;
    ///
    ///Recipe or process data associated to the various products run on the unit machine.
    Product: ARRAY[] OF PRODUCT;

END_STRUCT
END_TYPE

TYPE PMLs :
STRUCT
    ///
    ///Numeric representation of the active unit mode (required)
    UnitModeCurrent: DINT;
    ///
    ///Handshaking bit acknowledging that unit mode change request has been made.
    UnitModeRequested: BOOL;
    ///
    ///Signals that unit mode change request has been accepted; switching is actively in process.
    UnitModeChangeInProgress: BOOL;
    ///
    ///Numeric representation of the active state (required)
    StateCurrent: DINT;
    ///
    ///Handshaking value acknowledging that state change request has been made.
    StateRequested: DINT;
    ///
    ///Signals that state change request has been accepted; switching is actively in process.
    StateChangeInProgress: BOOL;
    ///
    ///Active machine speed setpoint in units of [primary packages/min] (required)
    MachSpeed: REAL;
    ///
    ///Actual machine speed in units of [primary packages/min] (required)
    CurMachSpeed: REAL;
    ///
    ///Bitwise indication that materials are ready for processing.
    MaterialInterlock: ARRAY[0..31] OF BOOL;
    ///
    ///Signals that the machine is either blocked or starved (required)
    EquipmentInterlock: EQUIPMENTINTERLOCK;
    ///
    ///Use to coordinate activities between peers (i.e. upstream or downstream unit machines.)
    RemoteInterface: ARRAY[] OF INTERFACE;
    ///
    ///Current values or parameters associated to the end user supervisory interface.
    Parameter: ARRAY[] OF DESCRIPTOR;
    ///
    ///Current recipe or process data associated to the various products run on the unit machine.
    Product: ARRAY[] OF PRODUCT;

END_STRUCT
END_TYPE

```

PACKMLV30, PMLc、PMLa 和 PMLs 以及上一节所述的 base type，为附录 1 和 2 中所列的整套 PackTag 提供了基础。

在 ISA-TR88.00.02-2015 中，使用“Unit/Machine”的术语表示用于不同生产的机器。我们为每一台 Unit/Machine 指定了唯一的一组 PackTag。每一组都与一种名为 UnitName 的字符串值标识符相关联。在当前的上下文中，这意味着我们将 UnitName 与 PACKML30 的一个实例关联起来。例如，如果我们机器的 UnitName 为 Machine\_01，那么使用标准 IEC61131-3 符合，我们将宣布 PACKML30 的实例如下：

```
Machine_01: PACKMLV30;    /// IEC61131-3
```

类似地，在 C# 中，我们可以定义

```
PACKMLV30 Machine_01;    // C#
```

建议在可以轻松识别的 Objects（文件夹）下组织所有 PackML 数据。在下方的示例中，Machine\_01 提供的 PackTag 显示在 PackML Objects 下方。

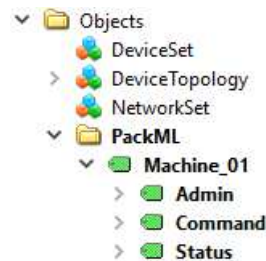


图 12 在专用的 Objects（文件夹）下组织来自单一 Unit/Machine Machine\_01 的 PackTag。推荐使用名称 PackML，但不是必须使用该名称。

如果 OPC UA 服务器包含来自多台 Unit/Machine 的 PackML 数据，建议也在同一个 Objects 下组织这一数据。如图 13 所示。

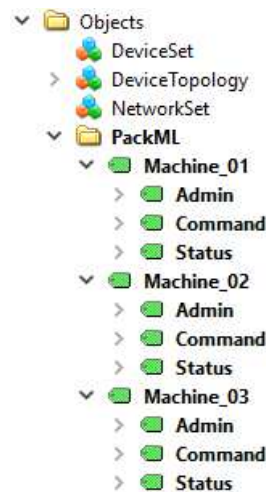


图 13 在一个共用的 Objects（文件夹）下组织来自 Unit/Machine Machine\_01、Machine\_02 和 Machine\_03 的 PackTag。

图 14 展示了 PackML 文件夹的相应结构。注意，在此示例中，将每种 Complex data type 作为一个变量节点进行建模。

Parent

Instance

NodeClass
Object
Namespace
3: http://omac.org/1/
Name
PackML
Type Definition
FolderType

Additional Attributes

NodeId
Numeric
5004

DisplayName
PackML

BrowseName
3
PackML

Description

EventNotifier
0 (None)

Children / Organized Instances

NodeClass	Name	TypeDefinition	ModellingRule	DataType		
> Variable	Machine_01	PACKMLV30VariableType	No ModellingRule	PACKMLV30	+	x
> Variable	Machine_02	PACKMLV30VariableType	No ModellingRule	PACKMLV30	+	x
> Variable	Machine_03	PACKMLV30VariableType	No ModellingRule	PACKMLV30	+	x

图 14 使用“Organizes” ReferenceType 构建 PackML 文件夹。使用 Unified Automation GmbH 的 UaModeler 创建 OPC UA 模型。

## 6. 参考信息

### 6.1. 定义和缩写

表 5 定义和缩写列表

缩写	说明	解释
FAT	Factory Acceptance Testing	交付前对 Unit/Machine 开展的测试。开展该测试是为了确保交付的 Unit/Machine 符合约定的要求。
FMEA	Failure Mode and Effect Analysis	FMEA 通常是系统可靠性研究的第一步。它涉及审查尽可能多的组件、总成和子系统，以便排查故障模式及其成因和影响。
HMI	Human Machine Interface	Human machine interface 是 Unit/Machine 的组成部分，用来处理人机交互。
I/O test	Input / Output test	I/O test 是在安装地开展的电气连接测试。
ISA	The International Society of Automation	ISA 是一个非营利性技术协会，其成员包括从事、研究或有兴趣了解工业自动化及其愿景的工程师、技术人员、企业家、教育工作者和学生。
SCS	Supervisory Control System	<p>Manufacturing Execution System (MES) 用于集成 ERP 系统中的客户订单和对 SCADA 与 PLC 系统的序列进行编程的系统软件。</p> <p>Supervisory Control and Data Acquisition (SCADA) SCADA 系统通常会记录来自 Unit/Machine 的日志数据并监控其状态。SCADA 往往用作生产线上集成的一套 Unit/Machine 的 HMI。</p> <p>Line Controller (LC) Line Controller 可以是 Supervisory Control System 的一部分。Line Controller 用来控制生产线中的若干 Unit。</p> <p>Supervisory Control System 可以由 PLC 担任，也可以是现有 PLC、PC 或 Server 的一部分。</p>
OEE	Overall Equipment Effectiveness	OEE 经常与生产效果一起用作 Key performance indicator (KPI)，用来衡量生产成功与否。
OEM	Original Equipment Manufacturer	OEM 是生产的产品或组件被另一家公司购买并贴上购买公司品牌或公司名称的生产商。
OMAC	Organization for Machine Automation and Control	OMAC 是一个由自动化和制造业专业人士组成的组织，其宗旨是支持制造业对机器自动化和运营的需求。
PackML	Packaging Machine Language	PackML 的主要目的是为所有机器提供通用“界面外观”和操作一致性。
PackML GW	PackML Gateway	表示 PackML Interface State Manager 的 gateway，它会映射至特定机器 (Unit/Machine) 的现有 Machine State Manager。

缩写	说明	解释
PackML SDS	PackML Software Design Specification	Unit/Machine 的功能规范，它规定了 Unit/Machine 的行为，也指定了从 PackML 角度与 Unit/Machine 交互的接口。
PackTags	PackML data identifier	PackTag 是由 OMAC Packaging Workgroup 制定的数据元素，用来简化包装机器之间的数据交换。
PLC	Programmable Logic Controller	控制仪表、马达、阀门等零部件的计算机。
SAT	Site Acceptance Testing	在安装 Unit/Machine 的现场对其开展的测试。开展该测试是为了确保交付的 Unit/Machine 符合约定的要求。
TR 88.00.02	Technical Report	Technical Report 是关于在不同机器上实施 S88 的信息文档，它也是 PackML 的规范。

## 6.2. 参考资料

表 6 参考资料列表

参考编号	文档
[1]	A. Griffen, “PackML on OPC UA, Implementation Guide”, Mettler-Toledo International (内部文件), 2016
[2]	<i>PLCopen and OPC Foundation: OPC UA Information Model for IEC 61131-3 - Release 1.00</i> , March 24, 2010
[3]	<i>Machine and Unit States: An Implementation Example of ISA-88</i> , ISA-TR88.00.02-2015, 2015
[4]	<i>OPC Unified Architecture Specification Part 3: Address Space Model</i> , Release 1.03, July 10, 2015
[5]	W. Mahnke, S. Leitner and M. Damm, <i>OPC Unified Architecture</i> , Springer, 2010
[6]	C. Nøkleby, <i>PackML Unit/Machine Implementation Guide</i> , Ver. 1.00, December 5, 2016
[7]	Wikipedia contributors, “PackML,” <i>Wikipedia, The Free Encyclopedia</i> , <a href="https://en.wikipedia.org/w/index.php?title=PackML&amp;oldid=805578184">https://en.wikipedia.org/w/index.php?title=PackML&amp;oldid=805578184</a> (2017 年 12 月 6 日访问)

### 6.3. 参与编写本指南的人员

以下人士积极参与了本技术报告的编写工作：

表 7 参与编写本技术报告的人员名单

名字	姓氏	公司
Carl	Bostrom	Bosch Rexroth
Lee	Smith	Mettler Toledo
Carsten	Nøkleby	SESAM-World
Malte	Schlüter	Mitsubishi

本文的编写得到了 Bosch Rexroth、Mettler Toledo、SESAM-World 与三菱电机的大力支持。

## 7. 附录

### 7.1. PackTag: 作为变量节点映射 COMPLEX DATA STRUCTURE (即 *NodeClass=Variable*)

表 8 作为变量节点映射 Complex data structure (即 *NodeClass=Variable*)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type <sup>5</sup>
Variable	PACKMLC30 VariableType	PACKML30	"en-US ", "UnitName"	index, "UnitName"	UnitName	PACKMLV30
Variable	PMLc VariableType	PMLc	"Command"	"Command"	UnitName.Command	PMLc
Variable	BaseVariableType	6 [Int32]	"UnitMode"	"UnitMode"	UnitName.Command.UnitMode	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"UnitModeChangeRequest"	"UnitModeChangeRequest"	UnitName.Command.UnitModeChangeRequest	Bool
Variable	BaseVariableType	10 [Float]	"MachSpeed"	"MachSpeed"	UnitName.Command.MachSpeed	Real
Variable	BaseVariableType	1 [Bool]	"MaterialInterlock"	"MaterialInterlock"	UnitName.Command.MaterialInterlock[#]	Bool
Variable	BaseVariableType	6 [Int32]	"CntrlCmd"	"CntrlCmd"	UnitName.Command.CntrlCmd	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"CmdChangeRequest"	"CmdChangeRequest"	UnitName.Command.CmdChangeRequest	Bool
Variable	InterfaceVariableType	Interface	"RemoteInterface"	"RemoteInterface"	UnitName.Command.RemoteInterface[#]	Interface
Variable	BaseVariableType	6 [Int32]	"Number"	"Number"	UnitName.Command.RemoteInterface[#].Number	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ControlCmdNumber"	"ControlCmdNumber"	UnitName.Command.RemoteInterface[#].ControlCmdNumber	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"CmdValue"	"CmdValue"	UnitName.Command.RemoteInterface[#].CmdValue	Int (32-bit)
Variable	Descriptor VariableType	Descriptor	"Parameter"	"Parameter"	UnitName.Command.RemoteInterface[#].Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Command.RemoteInterface[#].Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Command.RemoteInterface[#].Parameter[#].Name	String

<sup>5</sup> 数据类型符合 IAA-TR88.00.02-2015。



Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	<i>UnitName.Command.RemoteInterface[#].Parameter[#].Unit</i>	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	<i>UnitName.Command.RemoteInterface[#].Parameter[#].Value</i>	Real
Variable	Descriptor VariableType	Descriptor	"Parameter"	"Parameter"	<i>UnitName.Command.Parameter[#]</i>	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	<i>UnitName.Command.Parameter[#].ID</i>	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	<i>UnitName.Command.Parameter[#].Name</i>	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	<i>UnitName.Command.Parameter[#].Unit</i>	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	<i>UnitName.Command.Parameter[#].Value</i>	Real
Variable	ProductVariableType	Product	"Product"	"Product"	<i>UnitName.Command.Product[#]</i>	Product
Variable	BaseVariableType	6 [Int32]	"ProductID"	"ProductID"	<i>UnitName.Command.Product[#].ProductID</i>	Int (32-bit)

表 9 作为变量节点映射 Complex data structure (即 NodeClass=Variable)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	Descriptor VariableType	Descriptor	"ProcessVariables"	"ProcessVariables"	<i>UnitName.Command.Product[#].ProcessVariables[#]</i>	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	<i>UnitName.Command.Product[#].ProcessVariables[#].ID</i>	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	<i>UnitName.Command.Product[#].ProcessVariables[#].Name</i>	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	<i>UnitName.Command.Product[#].ProcessVariables[#].Unit</i>	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	<i>UnitName.Command.Product[#].ProcessVariables[#].Value</i>	Real
Variable	Ingredient VariableType	Ingredient	"Ingredients"	"Ingredients"	<i>UnitName.Command.Product[#].Ingredients[#]</i>	Ingredient
Variable	BaseVariableType	6 [Int32]	"IngredientID"	"IngredientID"	<i>UnitName.Command.Product[#].Ingredients[#].IngredientID</i>	Int (32-bit)
Variable	Descriptor VariableType	Descriptor	"Parameter"	"Parameter"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#]</i>	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].ID</i>	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].Name</i>	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].Unit</i>	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].Value</i>	Real

表 10 作为变量节点映射 Complex data structure (即 NodeClass=Variable)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	PMLs VariableType	PMLs	"Status"	"Status"	UnitName.Status	PMLs
Variable	BaseVariableType	6 [Int32]	"UnitModeCurrent"	"UnitModeCurrent"	UnitName.Status.UnitModeCurrent	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"UnitModeRequested"	"UnitModeRequested"	UnitName.Status.UnitModeRequested	Bool
Variable	BaseVariableType	1 [Bool]	"UnitModeChangeInProgress"	"UnitModeChangeInProgress"	UnitName.Status.UnitModeChangeInProgress	Bool
Variable	BaseVariableType	6 [Int32]	"StateCurrent"	"StateCurrent"	UnitName.Status.StateCurrent	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StateRequested"	"StateRequested"	UnitName.Status.StateRequested	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"StateChangeInProgress"	"StateChangeInProgress"	UnitName.Status.StateChangeInProgress	Bool
Variable	BaseVariableType	10 [Float]	"MachSpeed"	"MachSpeed"	UnitName.Status.MachSpeed	Real
Variable	BaseVariableType	10 [Float]	"CurMachSpeed"	"CurMachSpeed"	UnitName.Status.CurMachSpeed	Real
Variable	BaseVariableType	1 [Bool]	"MaterialInterlock"	"MaterialInterlock"	UnitName.Status.MaterialInterlock[#]	Bool
Variable	EquipmentInterlock VariableType	Equipment Interlock	"EquipmentInterlock"	"EquipmentInterlock"	UnitName.Status.EquipmentInterlock	Equipment Interlock
Variable	BaseVariableType	1 [Bool]	"Blocked"	"Blocked"	UnitName.Status.EquipmentInterlock.Blocked	Bool
Variable	BaseVariableType	1 [Bool]	"Starved"	"Starved"	UnitName.Status.EquipmentInterlock.Starved	Bool
Variable	InterfaceVariableType	Interface	"RemoteInterface"	"RemoteInterface"	UnitName.Status.RemoteInterface[#]	Interface
Variable	BaseVariableType	6 [Int32]	"Number"	"Number"	UnitName.Status.RemoteInterface[#].Number	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ControlCmdNumber"	"ControlCmdNumber"	UnitName.Status.RemoteInterface[#].ControlCmdNumber	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"CmdValue"	"CmdValue"	UnitName.Status.RemoteInterface[#].CmdValue	Int (32-bit)
Variable	Descriptor VariableType	Descriptor	"Parameter"	"Parameter"	UnitName.Status.RemoteInterface[#].Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.RemoteInterface[#].Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.RemoteInterface[#].Parameter[#].Name.	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.RemoteInterface[#].Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.RemoteInterface[#].Parameter[#].Value	Real
Variable	Descriptor VariableType	Descriptor	"Parameter"	"Parameter"	UnitName.Status.Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.Parameter[#].ID	Int (32-bit)

表 11 作为变量节点映射 Complex data structure (即 NodeClass=Variable)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.Parameter[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.Parameter[#].Value	Real
Variable	ProductVariableType	Product	"Product"	"Product"	UnitName.Status.Product[#]	Product
Variable	BaseVariableType	6 [Int32]	"ProductID"	"ProductID"	UnitName.Status.Product[#].ProductID	Int (32-bit)
Variable	Descriptor VariableType	Descriptor	"ProcessVariables"	"ProcessVariables"	UnitName.Status.Product[#].ProcessVariables[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.Product[#].ProcessVariables[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.Product[#].ProcessVariables[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.Product[#].ProcessVariables[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.Product[#].ProcessVariables[#].Value	Real
Variable	Ingredient VariableType	Ingredient	"Ingredients"	"Ingredients"	UnitName.Status.Product[#].Ingredients[#]	Ingredient
Variable	BaseVariableType	6 [Int32]	"IngredientID"	"IngredientID"	UnitName.Status.Product[#].Ingredients[#].IngredientID	Int (32-bit)
Variable	Descriptor VariableType	Descriptor	"Parameter"	"Parameter"	UnitName.Status.Product[#].Ingredients[#].Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].Value	Real

表 12 作为变量节点映射 Complex data structure (即 NodeClass=Variable)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	PMLa VariableType	PMLa	"Admin"	"Admin"	UnitName.Admin	PMLa
Variable	Descriptor VariableType	Descriptor	"Parameter"	"Parameter"	UnitName.Admin.Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.Parameter[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Admin.Parameter[#].Value	Real
Variable	AlarmVariableType	Alarm	"Alarm"	"Alarm"	UnitName.Admin.Alarm[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.Alarm[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.Alarm[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.Alarm[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.Alarm[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.Alarm[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.Alarm[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.Alarm[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AlarmExtent"	"AlarmExtent"	UnitName.Admin.AlarmExtent	Int (32-bit)
Variable	AlarmVariableType	Alarm	"AlarmHistory"	"AlarmHistory"	UnitName.Admin.AlarmHistory[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.AlarmHistory[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.AlarmHistory[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.AlarmHistory[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.AlarmHistory[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.AlarmHistory[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.AlarmHistory[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.AlarmHistory[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AlarmHistoryExtent"	"AlarmHistoryExtent"	UnitName.Admin.AlarmHistoryExtent	Int (32-bit)

表 13 作为变量节点映射 Complex data structure (即 NodeClass=Variable)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	AlarmVariableType	Alarm	"StopReason"	"StopReason"	UnitName.Admin.StopReason[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.StopReason[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.StopReason[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.StopReason[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.StopReason[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.StopReason[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.StopReason[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.StopReason[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StopReasonExtent"	"StopReasonExtent"	UnitName.Admin.StopReasonExtent	Int (32-bit)
Variable	AlarmVariableType	Alarm	"Warning"	"Warning"	UnitName.Admin.Warning[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.Warning[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.Warning[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.Warning[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.Warning[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.Warning[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.Warning[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.Warning[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"WarningExtent"	"WarningExtent"	UnitName.Admin.WarningExtent	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ModeCurrentTime"	"ModeCurrentTime"	UnitName.Admin.ModeCurrentTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ModeCumulativeTime"	"ModeCumulativeTime"	UnitName.Admin.ModeCumulativeTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StateCurrentTime"	"StateCurrentTime"	UnitName.Admin.StateCurrentTime[#, #]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StateCumulativeTime"	"StateCumulativeTime"	UnitName.Admin.StateCumulativeTime[#, #]	Int (32-bit)
Variable	CountVariableType	Count	"ProdConsumedCount"	"ProdConsumedCount"	UnitName.Admin.ProdConsumedCount[#]	Count
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.ProdConsumedCount[#].ID	Int (32-bit)

表 14 作为变量节点映射 Complex data structure (即 NodeClass=Variable)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.ProdConsumedCount[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.ProdConsumedCount[#].Unit	String
Variable	BaseVariableType	6 [Int32]	"Count"	"Count"	UnitName.Admin.ProdConsumedCount[#].Count	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccCount"	"AccCount"	UnitName.Admin.ProdConsumedCount[#].AccCount	Int (32-bit)
Variable	CountVariableType	Count	"ProdProcessedCount"	"ProdProcessedCount"	UnitName.Admin.ProdProcessedCount[#]	Count
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.ProdProcessedCount[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.ProdProcessedCount[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.ProdProcessedCount[#].Unit	String
Variable	BaseVariableType	6 [Int32]	"Count"	"Count"	UnitName.Admin.ProdProcessedCount[#].Count	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccCount"	"AccCount"	UnitName.Admin.ProdProcessedCount[#].AccCount	Int (32-bit)
Variable	CountVariableType	Count	"ProdDefectiveCount"	"ProdDefectiveCount"	UnitName.Admin.ProdDefectiveCount[#]	Count
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.ProdDefectiveCount[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.ProdDefectiveCount[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.ProdDefectiveCount[#].Unit	String
Variable	BaseVariableType	6 [Int32]	"Count"	"Count"	UnitName.Admin.ProdDefectiveCount[#].Count	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccCount"	"AccCount"	UnitName.Admin.ProdDefectiveCount[#].AccCount	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccTimeSinceReset"	"AccTimeSinceReset"	UnitName.Admin.AccTimeSinceReset	Int (32-bit)
Variable	BaseVariableType	10 [Real]	"MachDesignSpeed"	"MachDesignSpeed"	UnitName.Admin.MachDesignSpeed	Real
Variable	BaseVariableType	6 [Int32]	"StatesDisabled"	"StatesDisabled"	UnitName.Admin.StatesDisabled	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"PLCDateTime"	"PLCDateTime"	UnitName.Admin.PLCDateTime[#]	Int (32-bit)

## 7.2. PACKTAG: 作为对象节点映射 COMPLEX DATA STRUCTURE (即 NODECLASS=OBJECT)

表 15 作为对象节点映射 Complex data structure (即 NodeClass=Object)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type <sup>6</sup>
Object	PACKML30 ObjectType	---	"en-US ", "UnitName"	index, "UnitName"	UnitName	PACKMLV30
Object	PMLc ObjectType	---	"Command"	"Command"	UnitName.Command	PMLc
Variable	BaseVariableType	6 [Int32]	"UnitMode"	"UnitMode"	UnitName.Command.UnitMode	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"UnitModeChangeRequest"	"UnitModeChangeRequest"	UnitName.Command.UnitModeChangeRequest	Bool
Variable	BaseVariableType	10 [Float]	"MachSpeed"	"MachSpeed"	UnitName.Command.MachSpeed	Real
Variable	BaseVariableType	1 [Bool]	"MaterialInterlock"	"MaterialInterlock"	UnitName.Command.MaterialInterlock[#]	Bool
Variable	BaseVariableType	6 [Int32]	"CntrlCmd"	"CntrlCmd"	UnitName.Command.CntrlCmd	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"CmdChangeRequest"	"CmdChangeRequest"	UnitName.Command.CmdChangeRequest	Bool
Object	InterfaceObjectType	---	"RemoteInterface"	"RemoteInterface"	UnitName.Command.RemoteInterface[#]	Interface
Variable	BaseVariableType	6 [Int32]	"Number"	"Number"	UnitName.Command.RemoteInterface[#].Number	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ControlCmdNumber"	"ControlCmdNumber"	UnitName.Command.RemoteInterface[#].ControlCmdNumber	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"CmdValue"	"CmdValue"	UnitName.Command.RemoteInterface[#].CmdValue	Int (32-bit)
Object	DescriptorObjectType	---	"Parameter"	"Parameter"	UnitName.Command.RemoteInterface[#].Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Command.RemoteInterface[#].Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Command.RemoteInterface[#].Parameter[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Command.RemoteInterface[#].Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Command.RemoteInterface[#].Parameter[#].Value	Real
Object	DescriptorObjectType	---	"Parameter"	"Parameter"	UnitName.Command.Parameter[#]	Descriptor

<sup>6</sup> 数据类型符合 IAA-TR88.00.02-2015。

Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	<i>UnitName.Command.Parameter[#].ID</i>	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	<i>UnitName.Command.Parameter[#].Name</i>	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	<i>UnitName.Command.Parameter[#].Unit</i>	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	<i>UnitName.Command.Parameter[#].Value</i>	Real
Object	ProductObjectType	---	"Product"	"Product"	<i>UnitName.Command.Product[#]</i>	Product
Variable	BaseVariableType	6 [Int32]	"ProductID"	"ProductID"	<i>UnitName.Command.Product[#].ProductID</i>	Int (32-bit)

表 16 作为对象节点映射 Complex data structure (即 NodeClass=Object)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Object	Descriptor	---	"ProcessVariables"	"ProcessVariables"	<i>UnitName.Command.Product[#].ProcessVariables[#]</i>	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	<i>UnitName.Command.Product[#].ProcessVariables[#].ID</i>	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	<i>UnitName.Command.Product[#].ProcessVariables[#].Name</i>	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	<i>UnitName.Command.Product[#].ProcessVariables[#].Unit</i>	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	<i>UnitName.Command.Product[#].ProcessVariables[#].Value</i>	Real
Object	Ingredient	---	"Ingredients"	"Ingredients"	<i>UnitName.Command.Product[#].Ingredients[#]</i>	Ingredient
Variable	BaseVariableType	6 [Int32]	"IngredientID"	"IngredientID"	<i>UnitName.Command.Product[#].Ingredients[#].IngredientID</i>	Int (32-bit)
Object	Descriptor	---	"Parameter"	"Parameter"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#]</i>	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].ID</i>	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].Name</i>	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].Unit</i>	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	<i>UnitName.Command.Product[#].Ingredients[#].Parameter[#].Value</i>	Real



表 17 作为对象节点映射 Complex data structure (即 NodeClass=Object)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Object	PMLs ObjectType	---	"Status"	"Status"	UnitName.Status	PMLs
Variable	BaseVariableType	6 [Int32]	"UnitModeCurrent"	"UnitModeCurrent"	UnitName.Status.UnitModeCurrent	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"UnitModeRequested"	"UnitModeRequested"	UnitName.Status.UnitModeRequested	Bool
Variable	BaseVariableType	1 [Bool]	"UnitModeChangeInProgress"	"UnitModeChangeInProgress"	UnitName.Status.UnitModeChangeInProgress	Bool
Variable	BaseVariableType	6 [Int32]	"StateCurrent"	"StateCurrent"	UnitName.Status.StateCurrent	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StateRequested"	"StateRequested"	UnitName.Status.StateRequested	Int (32-bit)
Variable	BaseVariableType	1 [Bool]	"StateChangeInProgress"	"StateChangeInProgress"	UnitName.Status.StateChangeInProgress	Bool
Variable	BaseVariableType	10 [Float]	"MachSpeed"	"MachSpeed"	UnitName.Status.MachSpeed	Real
Variable	BaseVariableType	10 [Float]	"CurMachSpeed"	"CurMachSpeed"	UnitName.Status.CurMachSpeed	Real
Variable	BaseVariableType	1 [Bool]	"MaterialInterlock"	"MaterialInterlock"	UnitName.Status.MaterialInterlock[#]	Bool
Object	EquipmentInterlock ObjectType	---	"EquipmentInterlock"	"EquipmentInterlock"	UnitName.Status.EquipmentInterlock	Equipment Interlock
Variable	BaseVariableType	1 [Bool]	"Blocked"	"Blocked"	UnitName.Status.EquipmentInterlock.Blocked	Bool
Variable	BaseVariableType	1 [Bool]	"Starved"	"Starved"	UnitName.Status.EquipmentInterlock.Starved	Bool
Object	InterfaceObjectType	---	"RemoteInterface"	"RemoteInterface"	UnitName.Status.RemoteInterface[#]	Interface
Variable	BaseVariableType	6 [Int32]	"Number"	"Number"	UnitName.Status.RemoteInterface[#].Number	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ControlCmdNumber"	"ControlCmdNumber"	UnitName.Status.RemoteInterface[#].ControlCmdNumber	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"CmdValue"	"CmdValue"	UnitName.Status.RemoteInterface[#].CmdValue	Int (32-bit)
Object	DescriptorObjectType	---	"Parameter"	"Parameter"	UnitName.Status.RemoteInterface[#].Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.RemoteInterface[#].Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.RemoteInterface[#].Parameter[#].Name.	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.RemoteInterface[#].Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.RemoteInterface[#].Parameter[#].Value	Real
Object	DescriptorObjectType	---	"Parameter"	"Parameter"	UnitName.Status.Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.Parameter[#].ID	Int (32-bit)

表 18 作为对象节点映射 Complex data structure (即 NodeClass=Object)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.Parameter[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.Parameter[#].Value	Real
Object	ProductObjectType	---	"Product"	"Product"	UnitName.Status.Product[#]	Product
Variable	BaseVariableType	6 [Int32]	"ProductID"	"ProductID"	UnitName.Status.Product[#].ProductID	Int (32-bit)
Object	DescriptorObjectType	---	"ProcessVariables"	"ProcessVariables"	UnitName.Status.Product[#].ProcessVariables[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.Product[#].ProcessVariables[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.Product[#].ProcessVariables[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.Product[#].ProcessVariables[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.Product[#].ProcessVariables[#].Value	Real
Object	IndredientObjectType	---	"Ingredients"	"Ingredients"	UnitName.Status.Product[#].Ingredients[#]	Ingredient
Variable	BaseVariableType	6 [Int32]	"IngredientID"	"IngredientID"	UnitName.Status.Product[#].Ingredients[#].IngredientID	Int (32-bit)
Object	DescriptorObjectType	---	"Parameter"	"Parameter"	UnitName.Status.Product[#].Ingredients[#].Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Status.Product[#].Ingredients[#].Parameter[#].Value	Real

表 19 作为对象节点映射 Complex data structure (即 NodeClass=Object)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Object	PMLa ObjectType	---	"Admin"	"Admin"	UnitName.Admin	PMLa
Object	DescriptorObjectType	---	"Parameter"	"Parameter"	UnitName.Admin.Parameter[#]	Descriptor
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.Parameter[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.Parameter[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.Parameter[#].Unit	String
Variable	BaseVariableType	10 [Float]	"Value"	"Value"	UnitName.Admin.Parameter[#].Value	Real
Object	AlarmObjectType	---	"Alarm"	"Alarm"	UnitName.Admin.Alarm[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.Alarm[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.Alarm[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.Alarm[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.Alarm[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.Alarm[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.Alarm[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.Alarm[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AlarmExtent"	"AlarmExtent"	UnitName.Admin.AlarmExtent	Int (32-bit)
Object	AlarmObjectType	---	"AlarmHistory"	"AlarmHistory"	UnitName.Admin.AlarmHistory[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.AlarmHistory[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.AlarmHistory[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.AlarmHistory[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.AlarmHistory[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.AlarmHistory[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.AlarmHistory[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.AlarmHistory[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AlarmHistoryExtent"	"AlarmHistoryExtent"	UnitName.Admin.AlarmHistoryExtent	Int (32-bit)

表 20 作为对象节点映射 Complex data structure (即 NodeClass=Object)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Object	AlarmObjectType	---	"StopReason"	"StopReason"	UnitName.Admin.StopReason[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.StopReason[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.StopReason[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.StopReason[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.StopReason[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.StopReason[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.StopReason[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.StopReason[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StopReasonExtent"	"StopReasonExtent"	UnitName.Admin.StopReasonExtent	Int (32-bit)
Object	AlarmObjectType	---	"Warning"	"Warning"	UnitName.Admin.Warning[#]	Alarm
Variable	BaseVariableType	1 [Bool]	"Trigger"	"Trigger"	UnitName.Admin.Warning[#].Trigger	Bool
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.Warning[#].ID	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"Value"	"Value"	UnitName.Admin.Warning[#].Value	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Message"	"Message"	UnitName.Admin.Warning[#].Message	String
Variable	BaseVariableType	6 [Int32]	"Category"	"Category"	UnitName.Admin.Warning[#].Category	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"DateTime"	"DateTime"	UnitName.Admin.Warning[#].DateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AckDateTime"	"AckDateTime"	UnitName.Admin.Warning[#].AckDateTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"WarningExtent"	"WarningExtent"	UnitName.Admin.WarningExtent	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ModeCurrentTime"	"ModeCurrentTime"	UnitName.Admin.ModeCurrentTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"ModeCumulativeTime"	"ModeCumulativeTime"	UnitName.Admin.ModeCumulativeTime[#]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StateCurrentTime"	"StateCurrentTime"	UnitName.Admin.StateCurrentTime[#, #]	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"StateCumulativeTime"	"StateCumulativeTime"	UnitName.Admin.StateCumulativeTime[#, #]	Int (32-bit)
Object	CountObjectType	---	"ProdConsumedCount"	"ProdConsumedCount"	UnitName.Admin.ProdConsumedCount[#]	Count
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.ProdConsumedCount[#].ID	Int (32-bit)

表 21 作为对象节点映射 Complex data structure (即 NodeClass=Object)

OPC UA					ISA-TR88.00.02-2015	
Node Class	HasType Definition	Data Type Identifier	DisplayName	BrowseName	PackTag	Type
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.ProdConsumedCount[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.ProdConsumedCount[#].Unit	String
Variable	BaseVariableType	6 [Int32]	"Count"	"Count"	UnitName.Admin.ProdConsumedCount[#].Count	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccCount"	"AccCount"	UnitName.Admin.ProdConsumedCount[#].AccCount	Int (32-bit)
Object	CountObjectType	---	"ProdProcessedCount"	"ProdProcessedCount"	UnitName.Admin.ProdProcessedCount[#]	Count
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.ProdProcessedCount[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.ProdProcessedCount[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.ProdProcessedCount[#].Unit	String
Variable	BaseVariableType	6 [Int32]	"Count"	"Count"	UnitName.Admin.ProdProcessedCount[#].Count	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccCount"	"AccCount"	UnitName.Admin.ProdProcessedCount[#].AccCount	Int (32-bit)
Object	CountObjectType	---	"ProdDefectiveCount"	"ProdDefectiveCount"	UnitName.Admin.ProdDefectiveCount[#]	Count
Variable	BaseVariableType	6 [Int32]	"ID"	"ID"	UnitName.Admin.ProdDefectiveCount[#].ID	Int (32-bit)
Variable	BaseVariableType	12 [String]	"Name"	"Name"	UnitName.Admin.ProdDefectiveCount[#].Name	String
Variable	BaseVariableType	12 [String]	"Unit"	"Unit"	UnitName.Admin.ProdDefectiveCount[#].Unit	String
Variable	BaseVariableType	6 [Int32]	"Count"	"Count"	UnitName.Admin.ProdDefectiveCount[#].Count	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccCount"	"AccCount"	UnitName.Admin.ProdDefectiveCount[#].AccCount	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"AccTimeSinceReset"	"AccTimeSinceReset"	UnitName.Admin.AccTimeSinceReset	Int (32-bit)
Variable	BaseVariableType	10 [Real]	"MachDesignSpeed"	"MachDesignSpeed"	UnitName.Admin.MachDesignSpeed	Real
Variable	BaseVariableType	6 [Int32]	"StatesDisabled"	"StatesDisabled"	UnitName.Admin.StatesDisabled	Int (32-bit)
Variable	BaseVariableType	6 [Int32]	"PLCDateTime"	"PLCDateTime"	UnitName.Admin.PLCDateTime[#]	Int (32-bit)

## 8. 版本记录

---

表 22 版本日志

日期 [YYYY-MM-DD]	缩写	说明	版本
2018/4/3	CAB/CAN	第一版	01