PackML Unit Implementation Guide

History

Carsten Nøkleby, SESAM-World & Malte Schlüter, Mitsubishi Electric
VALUE PROPOSITION OF STANDARDIZED AUTOMATION

PACKML & LINE CONTROLLER

PackML & Line Controller

PackML, a standardization way to communicate with machines on the factory floor. It gives a common interaction language for the machines. This makes it possible to standardize the overall control of machines in a production line.

Benefits in CAPEX

- Reduced the integration cost with a total 4 times. The reason is a technical and organizational standardization model. Easy to integrate and test.
- Reduced the specification cost with a factor 4. The reason is the usage of use cases (SCHEMATICS AND TECHNICAL SPECIFICATIONS)
- The interface is standardized and the interfaces can be evaluated by the machine builder itself using a live database before commissioning.
- Training of operators can be standardized and reused across the company.

Benefits for OPEX

- The OPEX benefits are not fully obtained yet, but there are clear indications that the following is to be foreseen.
  - The same OPEX calculations for an machines than commissioning:
  - Energy data to be analyzed
  - Data quality related to production and products. Reducing labeling errors, packaging
  - The visualization of the performance on the line and the individual units. It is expected to give a 2-3% increase productivity.

What has been achieved?

- Standard integration cost reduced 4 times
- Reduced specifications cost reduced 4 times
- Commissioning time reduced by 50%

Expectations for OPEX

- OEE on all machines during commissioning
- Increase productivity with 5-10%
- Quality in production data

Figure 1: The yellow line indicates until 2014 the development in integrations costs. The white line is the estimated result after standardization.

Integration cost per unit that follows the standardized PackML and Line Controller concept from 1st to 5th project

Figure 2: The yellow line indicates the integration cost for one unit during 5 packaging projects based on a standardized application platform based on PackML and Line Controller. The PackML and Line Controller concept and modules are even developed during the projects. Figures are in Euro.
Benefits in CAPEX

The following benefits have been achieved:

- Reduced the integration cost with a factor 4. The reason is a technical and organisational standardized model. Easy to integrate and test.
- Reduced the specification cost with a factor 5. The reason is the reuse of use case scencarios and technical specifications.
- The interface test is standardized and the interface can be evaluated by the machine builder itself using a test suitcase before commissioning.
- Training of operators can be standardized and reused across the company.
Benefits for OPEX

The OPEX benefits are not really obtained yet, but there are clear indications that the following is to be fulfilled:

☑ The same OEE calculations for all machines from commissioning.
☑ Energy data can be logged.
☑ Data quality related to production and products. Reducing labelling errors, packaging material errors.
☑ The visualisation of the performance on the line and the individual units, is expected to give 5-20% increased productivity.
OMAC PackML State Machine

The machine is turned on, and is in the 'STOPPED' state. The Operator can either press the E-Stop button or press the 'Reset' button to enable the machine.
End User

OMAC
Line Integration Strategy

Open protocol for PackML tag exchange

OMAC State Model

PackTags
OMAC State Model

PackTags
OMAC State Model

PackTags
OMAC State Model

PackTags
OMAC State Model

I/O Machine Network

Intellectual Properties
Machine Builder A

I/O Machine Network

Intellectual Properties
Machine Builder B

I/O Machine Network

Intellectual Properties
Machine Builder C

I/O Machine Network

Intellectual Properties
Machine Builder D

Copyright © 2016 OMAC. All rights reserved.
OEM

- Higher development cost due to X software versions
- Higher maintenance cost due to machine lifecycle of 15 – 25 yrs
- Longer time to market

End User

- Higher machine price
- Higher TCO
- Enhanced downtime risk during maintenance
PackML Implementation Guide

- Arla Foods
- Nestlé
- P&G
- Mars
- Velux
- Coloplast

Kickoff 1. OMAC PackML Conference/Workshop in June 2015 at Schneider Electric
PackML Implementation Guide V 0.1

OMAC PackML Conference/Workshop
13.- 14.01.2016 at Mitsubishi Electric Ratingen

End User
Nestlé, P&G, Mars, Arla, Velux, Coloplast, Sesam

OEM
Bosch Packaging, Multivac, Tetrapak, PakTech, Mettler Toledo, Krones/KHS, Meurer, SIG

Supplier
Siemens, Schneider, B&R, Lenze, Beckhoff, LTI, Omron, Mitsubishi Electric
Next Steps

1. March - April adding changes

6. Presentation at PackExpo 2016 in Chicago

7. Presentation at Interpack 2017 in Düsseldorf
Agenda – version 1.0

• Purpose, Goals and Background
• Scope of Implementation Guide
• PackML Interface State Manager
• The ISA 88 Physical Model & Recipe Management
• Define a Unit
• The PackML Interface State Model
  • PackML State Model
  • Unit/Machine parameter – Batch/production order
Purpose, Goals and Background

Purpose
This PackML Unit/Machine Implementation Guide is a **Best Practice** Recommendation based on **ANSI/ISA TR88.00.02-2015**.

This PackML implementation guide is addressing machine builders, System Integrators and end users at the same time.

Background
The OMAC end users want to achieve a unified way of interfacing with units/machines on the factory floor. Providing this unified Interface will ensure an easy implementation of a supervisory control system. There will be similar interfaces to all units, and the units will have the same data structure available.
## Purpose, Goals and Background

### Goals – End users
- Easy integration
- Reduction of integration cost – spec., implementation & commissioning
- Same view on units from supervisory system
- Less risk & Less uncertainty in commercial contract
- Reliable data

### Goals – OEMs
- Standardized interface – not end user specific
- Migration path to Full PackML Unit
- Less risk & Less uncertainty in commercial contract
- Reduced FAT and SAT (time to marked)
All leading OMAC members have been involved
Part 1: PackML Interface State Manager
- Part 2: PackML Machine software code structure
- Part 3: PackML network connections
- Part 4: PackML and Safety
- Part 5: PackML and Line integration
- Part 6: PackML User interface - HMI

The list is not complete and there could be changes in the document titles.
Scope of Implementation Guide

The PackML specification is an implementation of the ANSI/ISA-TR88.00.02-2015, Machine and unit/machine States. As an implementation example of ANSI/ISA-88.00.01, it covers the following two application examples:

1. PackML Interface State Manager
2. PackML Machine State Manager (Not part of this document)

The focus of the PackML unit/machine Implementation Guide is the “PackML Interface State Manager”, equal to the blue part of the figure.
Scope of Implementation Guide

Machine Suppliers wanting to use the PackML state model for their internal Machine State Manager have two scenarios shown in:

- **PackML gateway unit/machine**
- **Full PackML compliant unit/machine**
The ANSI/ISA-88.00.01-2010 [Ref. S88-1] and the ANSI/ISA-TR88.00.02-2015 [TR88] define the levels of equipment and machines via a Physical Model.
The ISA 88 Physical Model & Recipe Mng.

- Process related unit/machine parameters
- Control recipe parameters.
The ISA 88 Physical Model & Recipe Mng.

• One **Control Recipe** per production order that describes the parameters which need to be set in the unit/machine for the production of one specific product. An example could be a palletizer unit/machine that receives the three process related parameters for a production order: Packing pattern, number of layers and interleaf between layers.

• As an example **Heat Sealing can have the process related unit/machine control recipe parameters**: speed, *time temperature*, pressure, contact area, temperature of material, variation of foil thickness, material layer.
Unit/machine mapping to one PackML interface
PackML implementation levels

- **Integration levels**
  - A: PackML State manager Gateway
  - B: PackML State manager
  - C: PackML State manager

- **Mapping**
  - 1:N

- **Make2Pack**
  - ISA 88.00.05-Draft

- **“Other code”**
  - Legacy equipment

PackML / ISA TR88.00.02
Define a Unit

A PackML unit/machine combined of different system elements
The PackML Interface State Model

Two main elements:

• Commands

• States
The PackML Interface State Model
The PackML Interface State Model
The PackML Interface State Model

The figure illustrates the two different stop situations the unit/machine is producing.
Mapping an existing unit to PackML

A PackML gateway unit/machine

Mapping 1:N

Supplier dependent Machine State Manager

PackML Interface State Manager [Ref. TR88]
PACKML INTERFACE (RESETTING)

The Production order/Batch is being Reset, and a new Production order/Batch can be started.

MACHINE CONTROL

Supplier dependent Machine State Manager

The Unit/Machine is being reset and ready for starting production.

Reset
PACKML INTERFACE (HOLDING from internal hold)
The reason for moving a unit/machine into the HOLDING state can be:
1) Minor issue caused by a hold situation on the Unit/Machine

MACHINE CONTROL

Supplier dependent Machine State Manager
The Unit/Machine can stop because of a minor error and enter the unit/machine state Holding.
PACKML INTERFACE (HOLDING from internal stop)
The reason for moving a unit/machine into the HOLDING state can be:
2) A stop of the Unit/Machine

MACHINE CONTROL

Supplier dependent Machine State Manager
The Unit/Machine can stop because the operator stops the unit and the unit/machine enter the machine state Stopping.
The PackML Interface State Model

- **Command Tags** – Used for control of the unit/machine.
- **Status Tags** – Used for getting status information.
- **Admin Tags** – Used to getting performance information from the unit/machine.
## PackTags – Status tags

<table>
<thead>
<tr>
<th>PackTag type</th>
<th>PackTag</th>
<th>Example of End user term</th>
<th>Datatype</th>
<th>TR 88.00.02 Minimum tags</th>
<th>End user Minimum tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>StateCurrent</td>
<td>State</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status</td>
<td>UnitModeCurrent</td>
<td>Mode</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status</td>
<td>MachSpeed</td>
<td>Nominal Speed</td>
<td>REAL</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status</td>
<td>CurMachSpeed</td>
<td>Current Speed</td>
<td>REAL</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status</td>
<td>EquipmentInterlock.Blocked</td>
<td>Blockage</td>
<td>BIT</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status</td>
<td>EquipmentInterlock.Starved</td>
<td>Starvation</td>
<td>BIT</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status</td>
<td>Parameter [#].ID</td>
<td>Machine data/parameter</td>
<td>Array Structure</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Parameter[#].Name</td>
<td>Name of parameter</td>
<td>STRING</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Parameter[#].Unit</td>
<td>Unit of measure</td>
<td>STRING[5]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Parameter[#].Value</td>
<td>Value of parameter</td>
<td>User Defined</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>RemoteInterface.Parameter[#].ID</td>
<td>Additional production data</td>
<td>Structure</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>RemoteInterface.Parameter[#].Name</td>
<td>Parameter ID</td>
<td>INT(32)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>RemoteInterface.Parameter[#].Name</td>
<td>Name of parameter</td>
<td>STRING</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>RemoteInterface.Parameter[#].Unit</td>
<td>Unit of measure</td>
<td>STRING[5]</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>RemoteInterface.Parameter[#].Value</td>
<td>Value of parameter</td>
<td>REAL</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
## PackTags – Admin tags

<table>
<thead>
<tr>
<th>PackTag type</th>
<th>PackTag</th>
<th>Example of End user term</th>
<th>Datatype</th>
<th>TR 88.00.02 Minimum tags</th>
<th>End user Minimum tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin</td>
<td>Warning[#]</td>
<td>Warning</td>
<td>Array Structure</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Admin</td>
<td>Warning[#].Trigger</td>
<td>Trigger</td>
<td>Bool</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Admin</td>
<td>Warning[#].ID</td>
<td>ID</td>
<td>Int (32bit)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Admin</td>
<td>Warning[#].Value</td>
<td>Value</td>
<td>Int (32bit)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Admin</td>
<td>ProdDefectiveCount</td>
<td>OEE.Bad count</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Admin</td>
<td>ProdProcessedCount</td>
<td>OEE.Total count</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Admin</td>
<td>StopReason.ID</td>
<td>Event and stop reason</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Admin</td>
<td>StopReason.Value</td>
<td>Detailed Error Information</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### PackTags – Command tags

<table>
<thead>
<tr>
<th>PackTag type</th>
<th>PackTag</th>
<th>Example of End user term</th>
<th>Datatype</th>
<th>TR 88.00.02 Minimum tags</th>
<th>End user Minimum tags</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command</strong></td>
<td>CntrlCmd</td>
<td>Command</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>Parameter [#]</td>
<td>Machine data/parameter</td>
<td>Array Structure</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>Parameter[#].ID</td>
<td>Parameter ID</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>Parameter[#].Name</td>
<td>Name of parameter</td>
<td>STRING</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>Parameter[#].Unit</td>
<td>Unit of measure</td>
<td>STRING[5]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>Parameter[#].Value</td>
<td>Value of parameter</td>
<td>User Defined</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>RemoteInterface.Parameter [# ]</td>
<td>Additional Production data</td>
<td>Array Structure</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>RemoteInterface.Parameter[#].ID</td>
<td>Parameter ID</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>RemoteInterface.Parameter[#].Name</td>
<td>Name of parameter</td>
<td>STRING</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>RemoteInterface.Parameter[#].Unit</td>
<td>Unit of measure</td>
<td>STRING[5]</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>RemoteInterface.Parameter[#].Value</td>
<td>Value of parameter</td>
<td>REAL</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>UnitMode</td>
<td>Mode</td>
<td>INT(32)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>UnitModeChangeRequest</td>
<td>Change mode</td>
<td>BOOL</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>MachSpeed</td>
<td>Mach Speed</td>
<td>REAL</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td>CmdChangeRequest</td>
<td>Change command</td>
<td>BOOL</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
EXAMPLE OF UNIT CONTROLLER FUNCTIONALITY

- Press button: Operator action—normal operation
- SC Command: Unit/machine commands
- PackTags: Information to and from external systems
- Fix problem: Operator action—Alarm situation
**Example of Unit Controller Functionality**

External system monitors unit states.

The External system can either in STOPPED or IDLE state download machine parameters.

Operator start Unit/Machine from External system.

The External system can either in STOPPED or IDLE state download machine parameters.

External system send Start command to unit.

Unit 1

- State = STOPPED
- Parameter - Unit 1
- State = IDLE
- Parameter - Unit 1
- Command = Start
- State = STARTING
- State = EXECUTE

Unit 1 is starting and when the state actions are complete the unit performs a "State Completed" (SC) and enters into state EXECUTE. See state description below.
Example of Unit Controller Functionality

External system monitors unit states.

- Unit 1
  - State = STOPPED
  - State = IDLE
  - State = STARTING
  - State = EXECUTE

Unit 1

Start command

Parameter - Unit 1

Operator can enter the unit/machine parameters via HMI

Operator starts Unit/Machine from Unit/Machine system

Unit 1 is starting and when the state actions are complete the unit performs a "State Completed" (SC) and enters into state EXECUTE. See state description below.
EXAMPLE OF UNIT CONTROLLER FUNCTIONALITY
Held unit

External system monitors unit states.

Unit 1

State = EXECUTE

Error on unit

Hold command

Unit has an internal error

Unit 1 is Holding and when the state actions are complete the unit performs a "State Completed (SC) and enters into state HELD.

Operator fixes problem on Unit 1.

Operator re-starts unit 1 by pressing a button on unit.

SC

Unit 1

State = HELD

SC

State = Unholding

Fix problem

Un-Hold command

Unit 1 is Unholding and when the state actions are complete the unit performs a "State Completed (SC) and enters into state EXECUTE.
Purpose, Goals and Background

**Goals – End users**
- Easy integration
- Reduction of integration cost – spec., implementation & commissioning
- Same view on units from supervisory system
- Less risk & Less uncertainty in commercial contract
- Reliable data

**Goals – OEMs**
- Standardized interface – not end user specific
- Migration path to Full PackML Unit
- Less risk & Less uncertainty in commercial contract
- Reduced FAT and SAT (time to marked)

Use the PackML unit/machine Implementation Guide to get started on your PackML journey
Malte Schlueter
Mitsubishi Electric
Malte.Schlueter@meg.mee.com

Carsten Nøkleby
SESAM-World
can@sesam-world.dk
Thank You
END