Proposed Updates To *PackSpec*
To Enhance PackML Requirements

Tom Doney – Nestlé
Pack Expo East 2015
Tom Doney  
Expert Engineer  
Nestec - PTC - Marysville  

Career Summary

• 10/98 to Present – Nestle PTC New Milford & PTC Marysville  Expert Engineer, Packaging Development, RTD Technology (current)
• 2/89 to 10/98 -  Kraft Foods, Global Technology, Tarrytown NY  
  Staff Engineer, Packaging Equipment Development
• 7/74 to 12/84 – Leeds & Northrup Inc., Project Engineer, Design, Fabricate & Test Electric Utility Steam Generation Control Systems

Education

• BS Electrical Engineering, Drexel University
• MS Manufacturing Systems Engineering, Lehigh University
• MS Computer Engineering, Manhattan College.

Technical Interests/Expertise

• Packaging Machinery & Packaging Systems
• Equipment Development
• Machine Vision Systems
• Robotics
• RFID
• PLC Programming
How Nestlé Assimilates PackML

Table of Contents

1. Introduction ........................................................................................................................................... 2
2. State Model and States definition ......................................................................................................... 4
3. PackTags .............................................................................................................................................. 8
4. Modes .................................................................................................................................................. 10
5. Modularity .......................................................................................................................................... 12
6. Light Tree VS machine buttons .......................................................................................................... 13
7. Mandatory and optional requirements ................................................................................................. 14
Proposal – Two Tier PackML Compliance

• Level B – Lesser degree of Compliance
  – PackTags communicate machine status
  – Operator Interface conforms to common “look and feel”
  – Visual alarms (stack lights on machines) comply with a standard structure and meaning

• Level A – The Full “Enchilada”
  – A unit operation’s control programming is fully compliant with PackML
  – PackTags communicate machine status
  – Control software fully compliant to support state model control and communication
  – Operator Interface conforms to common “look and feel”
  – Visual alarms (stack lights on machines) comply with a standard structure and meaning
Reason For PackML – End User

• For an end user, PackML enables information integration with other machines in the packing line, with a line controller, and with a MES layer of control by means of a common communication language.
Reason For PackML – End User (Cont’d)

• PackML also provides for common status, operation information, and fault information on HMI’s – a common look and feel on the Packaging Line from filler to stretch wrapper

Proposed addition to PackSpec
Reason for PackML – End User (cont’d)

- Can facilitate a common look and feel for visual status indication along the entire packing line with a standard pushbutton and status light indication

<table>
<thead>
<tr>
<th>Machine States vs Light States</th>
<th>Aborting</th>
<th>Aborted</th>
<th>Clearing</th>
<th>Stopping</th>
<th>Stopped</th>
<th>Resetting</th>
<th>Idle</th>
<th>Starting</th>
<th>Execute</th>
<th>Holding</th>
<th>Held (no production)</th>
<th>Unholding</th>
<th>Suspending</th>
<th>Suspended</th>
<th>Unsuspending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Light Flashing</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red light Steady</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amber Light Flashing (Starved)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amber Light Steady (Blocked)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green Light Flashing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green Light Steady</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue Light Flashing (Does not changed a model state)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Blue Light Steady</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buzzer 2 sec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Button Flashing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Button Steady</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset Button Flashing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proposed addition to PackSpec
Full Compliance

5.1 Definition

Modular programming means that the software code is organized into reusable blocks of code. The ISA88 physical hierarchy is the most commonly used method to decompose a machine and define what blocks of code will be used, for example:
A Question on Integration to Integrator Community

• How To Verify Minimal Compliance?
  – What should an OEM be prepared for?
  – What is the process for verification at an FAT?
    – Screen Content
    – Status Lamp Behavior
    – Communication Testing