Stacking up the advantages of robotic palletizing software

September 2017
Stacking up the advantages of robotic palletizing software
Doug Stoll

Product manager — Palletizing, Intelligrated

Doug is responsible for product strategy, development and ongoing management of Intelligrated’s automated palletizing, robotic palletizing and depalletizing solutions. He has more than 18 years of experience in the material handling industry, with more than 15 at Intelligrated. His background is in project engineering and execution, with extensive management experience in a variety of engineering disciplines, including electrical, controls, software and mechanical engineering.

He holds a bachelor’s degree in electrical engineering from the University of Missouri and an MBA from Saint Louis University.
Agenda

1. Introduction
2. Key industry drivers — robotic palletizing
3. Key considerations — robotic palletizing software
4. Palletizing software — overview
5. Functionality and key attributes
6. Architecture overview
7. Integration considerations
8. Total cost of ownership
9. Summary/wrap-up
Why use robotic palletizing software?

- Flexibility is in the hands of the end user
- Easily adapt and optimize operations when changes are introduced
- Reduced training and skill sets required (operator re-training)
- Lower total cost of ownership
- Increased quality and performance — optimization
- Improved execution and implementation times
- Improved changeover capabilities
Key drivers — robotic palletizing

- Labor costs and available work force
- Outgrowing existing palletizing processes
- Keeping pace with increasing order volumes
- **Adapting to evolving packaging trends/case design**
- SKU explosion — creating the needs for greater flexibility
- Desire to improve operating efficiency and/or capacity
- More complex load configurations (patterns, sheets, wrapping, etc.)
Question #1

What role has you seeking a deeper understanding of robotic palletizing software?

A. End user — new to robotic palletizing
B. End user — with existing robotic palletizing in place
C. OEM
D. Integrator/implementer
E. Other
Key considerations — robotic palletizing software

- **Software capabilities and needs are expanding based on industry trends.**
  - Development, implementation and operational considerations need to be factored in.
  - Choose software with the right focus, fit and feature set for your purpose/application.

<table>
<thead>
<tr>
<th>Developers/Integrators</th>
<th>Power Users/Implementers</th>
<th>End Users/Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Reduce delivery times</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Cell layout/integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Setup/configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality and performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Simulation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Capacity/utilization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Cost-effective, timely execution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Simple setup/configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Future expansion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Flexibility/adaptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Total cost of ownership</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Adaptable and flexible cell setup/config.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• EOAT options and config.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Flexible tools enabling effective and optimized load sequencing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Minimize setup/startup time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Common/integrated solution, scales to multiple cells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Maintenance supportability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Simple product changes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Reduced changeover time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Effective, integrated user interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• System and cell status/control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality and performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Simple, fast pattern changes and/or configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Simple, fast, flexible load changes and/or configuration</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Key elements and features — software

• Key elements
  - Simulation and/or visualization
  - Cell setup, layout and configuration
  - Robot tooling setup and configuration
  - Pallet, product and pattern configurations
  - Load building sequence and configuration (pick and place planning)

• Key features
  - **Flexible:** Pattern revisions, additions and/or line changeover
  - **Adaptable:** Handles various cell layouts, elements, tools and load types
  - **Performance tools:** Tools such as row forming, multi-drop and case orientation increase speed and capability
Question #2

Given the broad spectrum of palletizing software capabilities, what feature sets do you see providing the greatest benefit in your application(s)?

A. Simulation, cell design, tool configuration, system validation

B. New pattern/load configurations and updates, optimization, load validation

C. Daily operations, search/select patterns/loads, production change-overs
End user solution considerations

- End user solution profiles based on project type
  - What tools, functions, features and capabilities do I really need as an end user?
  - Balance current and future needs to find a cost-effective solution

**Simple**
(Single robot/cell, unlikely change, few SKUs/sizes, mid/low speeds)
- Focus on ease of use
- Simplify solution, keep it basic
- Focus on integration, leverage core hardware
- Very simple applications may only require a teach pendant

**Moderate**
(Mult. robots/cells, some change, mult. SKUs/sizes, mid speeds)
- Effective user interface needed for status and utility
- Quick change-out capabilities to accommodate production changes
- Configurability to account for changes in product/load types
- Tools to optimize rate and meet production demands

**Complex**
(Many robots/cells, likely change, many SKUs/sizes, higher speeds)
- Advanced configuration and teaching utilities
  - Future line expansion/addition
  - Equipment additions
  - Tooling changes
- Simulation capabilities
- Capacity and utilization tools
- Flexibility of tooling types
- Consistency within organization
  - Site-to-site file sharing
Functionality and key attributes

• User help tools
  – Help screens, videos, pop-up messages

• Change notifications, version control features

• Simple integration/architecture
  – Common interface
  – Simple data exchange

• Intuitive interface
  – Simple to use, visual understanding of process steps

• Reasonable training requirements
  – Understood/usable with reasonable training
    (given core understanding of load build process)
  – Intuitive/usable, even if infrequent changes or new loads
Architecture overview — considerations

• Common user interface vs. dedicated interfaces
• How does this fit in with my system now?
• How flexible will my system be in the future based on trends?
• Flexibility, scalability and ease of use
• Maintenance, updates, obsolescence
# Architecture overview

## General System Architecture – Robotic Palletizer

<table>
<thead>
<tr>
<th>Typical System Components</th>
<th>Typical Purposes</th>
</tr>
</thead>
</table>
| Laptop                    | • Pattern/Load Configs.  
                           | • Offline Edits      |
|                           | • Storage/Transfer  |
|                           | • Troubleshooting   |
| PC                        | • Simulation       |
|                           | • Cell/tool Configs.|
|                           | • Pattern/Load Configs.|
|                           | • HMI/Visualization|
| Robotic Controller and Pendant | • Pattern/Load Configs.  
                                  | • Motion control      |
|                           | • System level control (limited) |
| System level Controller   | • Conveyor control  |
|                           | • System level control|
|                           | • Cell/tool Configs. |
|                           | • Pattern/Load Configs.|
| System level HMI          | • HMI/Visualization  |
|                           | • Online Edits      |
|                           | • Pattern/Load Configs.|

In some cases, robot controller includes PC element.
Architecture examples

"Moderate" complexity example

Example #1
- Pattern/Load Configs.
- Offline Edits
- Storage/Transfer
- Simulation
- Cell/tool Configs.
- Motion Planning
- Motion control
- Conveyor control
- System level control
- HMI/Visualization

Example #2
- Pattern/Load Configs.
- Offline Edits
- Storage/Transfer
- Motion Planning
- Motion control
- Conveyor control
- System level control
- HMI/Visualization

Example #3
- Motion Planning
- Motion control
- Conveyor control
- System level control
- Cell/tool Configs.
- Pattern/Load Configs.
- HMI/Visualization
- Online Edits
- Pattern/Load Configs.
Question #3

For a current or future project under consideration, what would you consider the complexity to be?

A. Simple
B. Moderate
C. Complex
D. Not sure
Integration considerations

• Reduce the amount of complexity and variables
  – Leverage a common PLC and HMI when possible
  – Support and/or control ancillary equipment such as row forming, case turners, sheet handling, wrappers and stackers
  – Online vs. offline pattern and load configurations

• Look from a full system perspective (# cells, supporting equipment)
  – PC, PLC, HMI, controllers, interfaces, complexities/interfaces
  – Multi-line systems/cells
  – System-level machines and hardware such as sortation

• Integration factors — software and IT
  – Understand implications/dependencies of adding PC-based elements to an overall solution
  – Evaluate software and hardware dependencies from an IT perspective
    • Updates, patches, access, licensing
Total cost of ownership

• Implementation costs
  – Training
  – Software and licensing costs
  – Hardware costs (PLCs, HMIs, laptops, etc.)

• Operational and maintenance costs
  – Pattern and load changes
  – Technical support, vendor site visits
  – Maintenance

• Long-term costs
  – Upgrades (HW, SW)
  – Obsolescence
  – Compatibility updates/upgrades
    • PC/laptop, software, OS, service packs, updates
  – Production/capacity expansions
Summary — considerations for palletizing software

• Think big picture, integration
  – How do the software pieces fit together? Mix and match?
  – How many elements make up the solution (PCs, HMIs, controllers, etc.)?
  – What can be leveraged to simplify the solution?

• What level of software do I really need?
  – Think back to complex/moderate/simple solution profiles
  – Frequent changes in products?
  – Likely increases in production, possible line additions?
  – Think about the software user perspective. Where do you fit?
    • Developer, power user, operator use
  – What trained personnel do I have? What will I need going forward?
  – Licensing costs, updates, IT upgrades, tech support
Summary — advantages for palletizing software

• Range and flexibility in the hands of the end user
  – Wide range of features and configuration options

• Easily adapt and optimize operations when changes are introduced
  – Search, store, share and apply pattern and load configurations easily

• Reduced training and skill sets required (operator re-training)
  – Do not need advanced robotic skill sets on a regular basis

• Lower total cost of ownership
  – Fewer vendor visits, reduced maintenance involvement

• Increased quality and performance
  – Simulation, advanced optimization tools and features

• Improved execution and implementation times
  – Tools and utilities streamline processes and configurations
  – Improved change-over capabilities
Questions?

Doug Stoll
Product manager — Palletizing
Phone: 314.995.2328

Email: doug.stoll@intelligrated.com