

Interview

# Do you speak PackML?

The common machine language defined by the PackML standard allows packaging machines, their operators and higher-level information systems to communicate with unprecedented clarity. Machine builders, system integrators and end users alike are already experiencing the benefits. Developed by OMAC and following in the footsteps of the S88 standard on which it is based, PackML is experiencing near exponential growth in acceptance. Recently, OMAC Chairman Bryan Griffen explained to automotion why PackML has been so successful and who stands to benefit. He also shared his vision of a not-so-distant future where there is only one answer to the question "Do you speak PackML?"



Bryan Griffen  
OMAC Chairman



### ***Can you start by explaining the concept behind PackML as an open standard?***

In creating the standard, we're defining a consistent way of controlling discrete machinery. If we define things like the state model, the modes of operation and the information that is exchanged between machines, then any two machines that apply these definitions can communicate with each other, regardless of who built them and what control technology they used.

And as an open standard, PackML is not tied to a particular technology or a particular company. It's freely available to anyone in the industry. Any machine that complies can communicate with any other machine that complies, and we no longer need large systems in-between doing translation.

### ***So are we just talking about communication at the machine-to-machine level or also on a machine-to-MES or SCADA level?***

We're looking in both directions. Let me be clear that what we're not talking about is standardizing the network protocol used internally in each machine. What we are specifying is how those machines are going to talk to each other across a common Ethernet-based network protocol. Which specific protocol is used for any given line is a decision that can be made by the end user. The idea is that the machines can put information out on that network in a format that all the other machines can pick up.

The second step is then to get that information up into an ERP system so we can get recipe information down into the machines and reports from production back up into the ERP layer based on events in the line. This lets us record stoppages, for example, and use that stoppage analysis to determine what's causing the most downtime over a specific period of time.

This new communication lets us target efficiency improvements not only at the machine level, but for the entire line as well.

### ***So it's not just the communication layer that is standardized, but also the information to be exchanged?***

Yes, the information is contained in Pack-Tags – packets of information with a specific format and a specific function. So if we take a speed tag for example, everybody knows that it's called "CurMachSpeed". They know it contains information in the format "primary units per minute" of whatever it is that's being made, and they know it's going to be a 32-bit number. Every machine in the line knows these things, so when they get something that says "CurMachSpeed", they don't have to figure out "Now is that feet per second or meters per minute?" because they know exactly what they're dealing with.

### ***Why are these standards so important for an end user?***

As an end user, the problem you have when you build a line is that you buy a filler from one company, a coupon inserter from somebody else, a labeler, a caser, a palletizer and they're each made with a different engineering methodology. So when everything arrives at your factory, you face the task of making the machines work together.

You spend a lot of money with a system integrator who has to go talk to each machine builder and get into the inner workings of each machine. It gets very expensive and very difficult. And where do most of the problems pop up during commissioning? Not in any individual machine, but in making the machines work together.

With a standard methodology in place, the task of integrating the machines becomes much easier. Rather than spending all this

time down in the guts of the PLC code aligning bits and bytes, the system integrator is focused on higher-level applications – making the line more efficient and providing tools that make the operator and supervisor more effective.

The end user benefits from having the system integrator and OEM each focusing on what they do best. If you dictate to the machine builder what technology to use, you're guaranteed a suboptimal machine. That's like going to Ford or Volkswagen and telling them how to build an engine. You don't do that. You specify the functionality and performance, you specify PackML and say, "OK, give me your best machine that complies." Then you get exactly what you really want, which is a best-in-class machine with the best possible performance.

### ***Are there already end users supporting OMAC's initiative and using PackML?***

There are quite a number of end users who are direct members of OMAC such as Nestlé, Procter & Gamble and PepsiCo. Then there are other large end users who, while not members of OMAC, are still specifying PackML as a requirement for machines coming into their factories, such as Arla Foods, Mars and others.

### ***So the end user saves on system integration, gets a smoother commissioning process and sees overall improvements in line performance. What's in it for the OEM?***

There are typically three areas of benefit that we see. The first is improved development time. Say, for example, that two end users are both buying a case packer from the same company. One comes in and says it must use Technology Provider A, and the other says it must use Technology Provider B, so the OEM has to develop the same machine with two different platforms just to

service multiple customers. They also need technicians who understand both platforms. This is not only expensive and difficult, it bogs down their capacity to innovate, because they're spending their time keeping the technology platforms and technicians at the same level instead of enhancing performance.

There's also an improvement in terms of after-sales support. No longer do they send out a technician who ends up sitting there scratching his head because he knows there was something different about this machine but can't remember exactly what. With standards in place, service calls become much less of a headache. Yet it seems to me that the biggest benefit is in the retention of intellectual property. When the end user buys a line that integrates equipment from multiple OEMs, the system integrator has to go down into the PLC code of each machine to make them line up and communicate. The OEM is forced to give away intellectual property to a system integrator, which of course nobody likes to do.

With PackML it can be a black box. The permissives are encapsulated in specifically-coded pieces of information, the Pack-Tags. The system integrator knows what the state model is, so he doesn't need to know the inner workings; it's practically plug-and-play. It's a huge improvement for a machine builder to keep that intellectual property locked up.

***So for the OEM, PackML brings improvements in development, support and protection of their intellectual property. What about the system integrators?***

One of the things that a system integrator struggles with is bidding on a project in-

volving a large number of machines. You just never know how long it will take to get it working. So if we can take that out of the equation with a standard, the bidding process becomes much easier and they can focus on higher-level functions.

Also, instead of everything being one-off, they can start making standard packages. They can make a middleware package for reporting line efficiency to a plant manager, for example, and they know they can use it in any factory because it is built on an accepted standard.

***If we're talking about line integration, safety communication is going to come into play. Are there any PackML developments that deal with the issue of safety?***

There are several active subcommittees within the OMAC packaging workgroup. One of them is focused revising the standard itself to make it more usable. Another is PackSafety, who is working on providing the same functionality as PackML at a safety level to communicate information between machines about safety events.

***What is OMAC's strategy for deploying these standards on the market?***

Right now, we're in the final phases of a revision to the PackML standard aimed at improving clarity and usability. We're working with various technology providers to make sure libraries are available for all the different types of PLCs. At PACK EXPO 2013 we had a PackML meeting with 8 different technology providers showcasing PackML tools already implemented in their programming software. Beyond that, we have the push from OMAC members such as Nestlé, Procter & Gamble and PepsiCo, who are establishing PackML as one of their fixed requirements. That's

certainly going to drive the industry, and with technology providers also making it easier to implement, the adoption will move even more quickly.

Some of the larger OEMs like ProMach and Bosch Packaging are also adopting PackML as their standard.

***How does a company become an OMAC member, and what are the benefits?***

One of the primary benefits is having input on what's happening with the standard. It's a very active community. Getting involved now puts a company on the forefront of that wave, allows them to help define the future, gives them a first look at what is going to be required to comply and helps them get a step ahead of the competition.

Membership is pretty straightforward, ranging from \$1,000 to \$3,000 per year depending on the size of the company. For us, the most important thing is the participation. It's the ideas that come out of the member companies. This collaboration makes the standard stronger and more applicable across the board.

All of the details for joining can be found at OMAC.org.

***Sounds like you have an exciting future ahead of you. We wish you all the best – thanks for your time! ↩***