



ZIGGY™ LEAN - Modular Process Units

The Modular Process Unit is the workhorse of the ZIGGY Architecture.

The operational concept is to process an article by sending it through a series of manufacturing steps. Each step is performed by a Modular Process Units (MPU).

Since different articles require different manufacturing steps, the MPU is selected—or designed—to meet the needs of the article being manufactured.

For example, if the article being manufactured is candy, the MPUs will be different from ones used to make lightbulbs.

Accordingly, WRC works with the end user to select a machine builder or systems integrator that can supply the MPUs needed for their process. The critical consideration is that certain common characteristics are required in order for the MPUs to be integrated into a ZIGGY Architecture.

The common characteristics include:

- Standardized physical interface to allow easy interchangeability of MPUs.
- Controls embedded within the MPU
- Self-contained MPU including sensors, actuators, power supply
- Standardized external interfaces including Ethernet connections, power, air, hydraulics, etc.
- The MPU and the Modular Conveyance are to operate independently
- Either a HOLOCON controller or an IEC 61499 Proxy interface to the MPU.

The article being manufactured is to be moved between MPUs in order for the processing steps to be performed. The controls are to be designed to independently perform their operation when instructed by the AGILE Controller and the Mass Customization Controller.

Status and error messages will be passed to the AGILE Controller, the Mass Customization Controller, and to other MPUs.

During start-up, software negotiations will be conducted between the MPUs, the AGILE Controller, and the Mass Customization

Controller addressing the capability of the MPU and its status. This negotiation provides the AGILE Controller and the Mass Customization

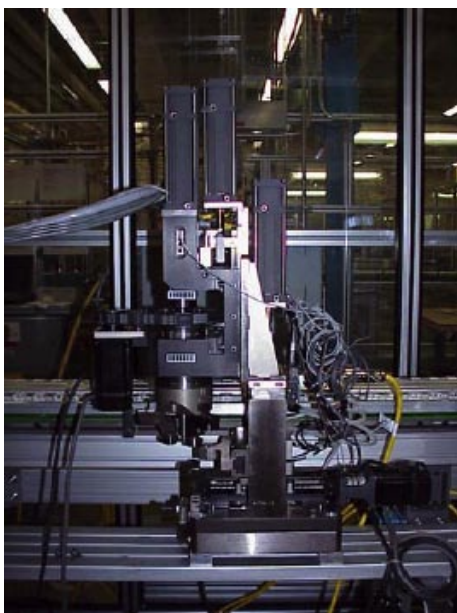
Controller with the information they need to manage the overall production line. These negotiations are implemented using Agent-Based Software and are resident in the HOLOCON Controller or IEC 61499 Proxy.

The MPU rules are established for a number of reasons:

- The objective is AGILE Manufacturing—as a minimum.
- Possibly, Mass Customization is desired
- In either case, we can be guaranteed that at some point in time a process or product change will be required.
- The primary objective behind the ZIGGY MPU rules is to be able to implement changes without disrupting operations elsewhere in the process.

As an illustration, refer to the two photos of sealing MPUs. A client had been using sealer A. It did the job. It was in place. But sealer A operated by stamping. There was some indication that the seal would be stronger and last longer if it was rolled instead of stamped.

In a conventional assembly line, it would be disruptive to consider making such an incremental change. Production would be interrupted for days. Other equipment would have to be moved with the potential for introducing damage and additional delays. And if the new design did not work out for some reason, it would be disruptive to return production to the original condition.



Sealer A



Interchangeable Sealer B made possible by ZIGGY

With ZIGGY, the customer built up a new sealer B that rolled the part closed.

Since A and B followed the same mechanical footprint, they were interchangeable.

Since they followed the same standards for electrical, communications, and air connections, they were interchangeable. And since they both followed the Agent-based software interface standard, one could be substituted for the other.

In fact, with only 5 minutes of downtime, the two units were swapped and the line was back in operation.

Western Reserve Controls is an established manufacturer of industrial controls. WRC provides the ZIGGY architecture and components to Machine Builders and Systems Integrators that know your production and process requirements.

ZIGGY is protected by the following

US patents:

- 6,615,091
- 6,325,198
- 6,478,134
- 6,681,915