

Work Process Management for Adaptive Manufacturing



WHITE PAPER #36

A MESA International and GE Intelligent Platforms
co-branded white paper.
21 October 2010

Table of Contents

ABSTRACT.....	1
1. INTRODUCTION TO WORK PROCESS MANAGEMENT.....	1
2. ISA-95 MODELS & TERMINOLOGY FOR WORK MODELING.....	3
RESOURCE MODELS AND PROCESS MODELS	3
THE BASIC STRUCTURE OF AN ISA-95 OBJECT.....	5
THE ISA-95 ROLE-BASED EQUIPMENT HIERARCHY MODEL.....	7
3. ISA-95 BASED WORK MODELS FOR OPERATIONS MANAGEMENT.....	8
LEVEL 3 MOM WORK MODEL OBJECTS	8
4. PUTTING THE ISA-95 MODELS TOGETHER	14
DETERMINING THE GRANULARITY OF THE PRODUCTION MODEL	14
BUILDING A PRODUCTION LINE FROM THE GROUND UP	15
BUILDING THE REST OF PRODUCTION.....	17
OUTPUT DATA STRUCTURE DESIGN.....	17
5. BUILDING WORKFLOWS FOR WPM.....	18
BASIC ORGANIZING PRINCIPLES REVIEWED	18
WORKFLOW TYPES AND SAMPLES.....	21
6. IMPLEMENTING WORKFLOWS IN AN ISA-95 DATA STRUCTURE.....	23
THE BIG PICTURE.....	23
A VERY SIMPLE USE CASE – MAKING A BICYCLE	24
7. INTERFACING ERP WITH WPM SYSTEM IN MOM LEVEL 3	32
B2MML TRANSACTION PROCESSING OF MOM WORKFLOWS.....	32
TRANSACTION MODELS	33
INTERFACE WORKFLOW TYPES AND SAMPLES.....	34
SERVICE INTERFACE	36
CONCLUSION	37
GLOSSARY, ACRONYMS, AND ABBREVIATIONS	38
AUTHORS	40
CONTRIBUTING EDITOR.....	40
REVIEWERS.....	40

ABSTRACT

Manufacturing Operations Management (MOM) solutions are growing in demand as manufacturers seek to streamline plant operations and align those operations with ever-changing supply chain processes. These manufacturers are discovering that using modern computing technology combined with continuous improvement methods dramatically accelerate time-to-benefit. Optimizing manufacturing operations requires a work process management (WPM) system to define, model and coordinate each plant's workflows and resources across shop floor, MOM and supply chain applications. An operations workflow engine in a WPM system is the key unifying application for aligning business and plant processes for rapid response to normal and abnormal tasks and for optimizing all real-time conditions. Workflow applications drive proper notification and acknowledgement of alerts, alarms and events to trigger alternative and corrective workflows and tasks through integrated real-time transactions.

This white paper explains how to implement ISA-95 workflow applications on a manufacturing enterprise platform to execute plant tasks through the MOM and physical process levels while coordinating them with supply chain tasks. The business benefits of optimizing operations are realized by structuring the workflows in ISA-95 models as the foundation for manufacturing architecture. This enforces effective data structure, definition, integrity and governance across manufacturing applications.

1. INTRODUCTION TO WORK PROCESS MANAGEMENT

MES (manufacturing execution systems) evolved dramatically over the last 20 years, to be further defined and replaced by the functional activity models and the term Manufacturing Operations Management (MOM) in ISA-95 Part 3, MOM Activity Models. The current generation of MOM systems is now performing near real-time root cause analysis while tracking order routes, batch processes, material transformations and movement, and the status of plant resources.

Manufacturing Operations Management (MOM) solutions are growing in demand as manufacturers seek to streamline plant operations and align those operations with ever-changing supply chain processes. These manufacturers are discovering that using modern computing technology combined with continuous improvement methods dramatically accelerate time-to-benefit. Optimizing manufacturing operations requires a work process management (WPM) system to define, model and coordinate each plant's workflows and resources across shop floor, MOM and supply chain applications. An operations workflow engine in a WPM system is the key unifying application for aligning business and plant processes for rapid response to normal and abnormal tasks and for optimizing all real-time conditions. Workflow applications drive proper notification and acknowledgement of alerts, alarms and events to trigger alternative and corrective workflows and tasks through integrated real-time transactions.

This white paper explains how to implement ISA-95 workflow applications on a manufacturing enterprise platform to execute plant tasks through the MOM and physical process levels while coordinating them with supply chain tasks. The business benefits of optimizing operations are realized by structuring the workflows in ISA-95 models as the foundation for manufacturing architecture. This enforces effective data structure, definition, integrity and governance across manufacturing applications.

After a plant's work and resources definitions are modeled in an extended ISA-95 plant model in a WPM application, workflow tasks and supporting applications are then contextualized in the plant's operations metadata model. This foundation makes

possible effective near real-time analysis of plant Level 3 operations, Level 2 control execution and Level 0/1 physical activities.

The ISA-95 models are broad and generic in scope, which gives them the flexibility required to accurately define and model numerous plant environments and forms. However, this strength of ISA-95 models makes the learning curve somewhat difficult to understand and apply unless they are applied as the data definitions and structure within a workflow engine application. To accurately characterize the high complexity of granular tasks and their parent operations workflows in a plant, the ISA-95 common metadata models must be extended to address a plant's workflows and associated tasks, dependencies, and condition based responses.

ISA-95 models are primarily designed to provide data exchange interfaces based on a unified MOM canonical definition for a given plant to higher level ERP (enterprise resource planning) systems and to Level 2 control systems. As mentioned, an ISA-95 based workflow engine permits the alignment of MOM systems to execute co-dependent operations processes in a very adaptable manner. Processes that are aligned across MOM systems are able to rapidly respond to changing resource states and reorder demand priorities to reduce the costly effects of non-value-added activities while enabling "pull" make-to-order supply chains. This aligned workflow foundation allows rapid development of MOM applications, interfaces, reports, and notification/acknowledgement of alarms and events. By defining and contextualizing MOM workflow applications, a work process management system enables adaptive manufacturing.

To effectively manage production and support activities across plants and their supply chains, the WPM system is a services oriented architecture (SOA) platform containing components and functionality specifically designed to build out applications. A WPM system consists of the following components:

- Tools for building data models based on ISA-95 standards
- A workflow execution engine that uses a graphical editor for simplifying the construction of complex logic. Workflows are programs used to execute activities and respond to events and data changes and then write out values 1) to a database based on the metadata models or 2) through external connectors.
- A client console hosting editors for building MOM applications
- HMI (human-machine interface) screens for monitoring process control and physical activities
- Displays of workflow tasks to operators on the plant floor
- WPM operator interface forms to accept input to route to workflows and clients throughout the enterprise
- A configurable event engine that initiates workflows and other code based on many different kinds of internal and external triggers
- A security infrastructure allowing objects to be secured by role and location

In combination, these WPM tools are the core of a complete suite of applications built to facilitate the manufacturing work process management and analysis of activities in the enterprise. This is an Event Driven Architecture as part of an SOA for manufacturing designed for the real-time requirements of high transactions with high parametric data loads.

As general guide for this white paper:

- Section 2 explains the basic ISA-95 model
- Section 3 introduces work models
- Section 4 describes build-up of the production line example
- Section 5 introduces the general types of workflow needed
- Section 6 ties it all together using the ISA-95 by giving examples of building workflow in example WPM application
- Section 7 explains the B2MML interface for a WPM application

Note: This white paper assumes a good general knowledge of ISA-95; to get the most out of this paper the novice reader is referred to the following sources to gain a basic understanding of ISA-95: 1) MESA White Paper #25: *An Overview and Comparison of ISA-95 and OAGIS* and 2) the *MESA Guidebook for SOA for Manufacturing*. For actual implementation details, the recommendation is for the reader to study the ISA-95 family of standards.

This white paper was produced as part of the MESA/ISA-95 Best Practices Working Group and is considered a description of best practices for developing ISA-95 MOM Standards-based manufacturing operations management systems.

2. ISA-95 MODELS & TERMINOLOGY FOR WORK MODELING

This discussion starts with modeling manufacturing operations management with ISA-95 models. Plant production and operations modeling is the first design activity required before implementing the higher level features of an industrial WPM system.

Two major ISA-95 models are the foundation for plant production modeling:

- Resource models, which include equipment-role (production roles of equipment, assets and buildings of an enterprise), materials (raw, intermediates, produced and consumed) and personnel (operations personnel involved in work)
- Process models for the operations and activities of the enterprise to quantify plant capability and production capacity

Resource Models and Process Models

ISA-95 models as applied in a WPM system fit into two broad categories: Resource Models and Process Models.

ISA-95 Resource Models

An ISA-95 resource is defined as an entity within the enterprise that provides some or all of the capabilities required for the execution of a unit of work such as an operation, step or phase in an enterprise activity and/or business process.

The ISA-95 Resource Models are divided into equipment, material, and personnel. The Role-based Equipment Hierarchy Model is a hierarchical plant structure representing the co-dependent role of fixed assets in production workflows from the enterprise down to the smallest hardware on the floor. The material and personnel models have class-instance structure. An industrial WPM system allows some organization of these objects to facilitate navigating and locating them in a structured program.

ISA-95 Process Models

For the purposes of this paper, the following ISA-95 Part 2 and proposed Draft Part 4 models are referred to as the ISA-95 Process Models:

- *Process Segment model*
- *Product Definition model (Product Segment, original Part 2)*
- *Operations Definition model (Operations Segment, updated Part 2)*
- *Work Process Segment model (Work Process Segment, proposed Part 4)*
- *Work Segment model (Work Segment, proposed Part 4)*

The *process segment* is for the Level 3 to 4 interfaces, while *work process segment* is for the Level 3 operations exchanges where a collection of *process or work segments* defines the operations processes and capabilities inside the plant. These are “capability segments” that define a unit or collection of work such as each generic route or recipe, and possibly as granular as an operation, step or phase of work. A segment of work is defined in terms of its resources per the Resource Model definitions, their specifications, parameters, performance data, etc. To define plant capability, a WPM system may extend the work process definition into a hybrid form to define work processes and exchanges in and across all Levels 1 to 4 and 5.

Once plant capability is defined, instance-specific definitions of the production and work order are defined based on the plant’s generic process capability models. *Product segment* and *operations segment* define order-specific information across the Level 3 to Level 4 interfaces while *work segment* defines order-specific information for Level 3 functions and tasks.

A Note Regarding ISA-95 Terminology

The original ISA-95 Parts 1 and 2 standards only address production operations and not support operations; consequently, these early version of ISA-95 referred to *product segment* only as production objects: *product definition*, *production request*, *production response* and so on. In updated versions, ISA-95 was extended to encompass all operations processes—including quality, maintenance, inventory and energy flow—having to do with plant operations. Most books and articles written about ISA-95 prior to 2009 use the original versions’ terminology.

The Part 3 MOM Activity Models have expanded this scope to encompass more than just production activities. ISA-95 Parts 1 and 2 were updated in 2009 to reflect MOM activity models for the support operations of maintenance, inventory and quality operations management. ISA-95 is now used to model and coordinate all operations work in a plant including line changeovers, quality tests, environmental tests, power utilization, and maintenance. As a result, many of the terms have updated names, which may be confusing if you are already familiar with ISA-95 since the standard still permits the production-centric terms for backward compatibility. Consequently, a *product segment* can also be modeled as an *operations segment* with an operations type attribute of production operations. Table 1 lists the various ISA-95 based terms proposed in draft ISA-95 Part 4 as of January 2010. These terms are part of the Level 3 MOM object models for work definitions. These terms will change in the final Part 4 standard but the structure and dependencies from Part 1 will be preserved.

Original Part 1 Name	New Proposed Part 4 Level 3 Name
<i>process segment</i>	<i>work process segment</i>
<i>product segment</i>	<i>work segment</i>
<i>production rule</i>	<i>(not identified as Draft 4)</i>
<i>production request</i>	<i>work request</i>
<i>production response</i>	<i>work response</i>
<i>production schedule</i>	<i>work schedule</i>
<i>production capability</i>	<i>work capability</i>
<i>production performance</i>	<i>work performance</i>

Table 1: Expanded ISA-95 Terminology as of January 2010

The Basic Structure of an ISA-95 Object

An ISA-95 object is essentially a collection of attributes and property attributes defined by attributes to describe the important inputs, outputs, and other fundamental characteristics.

ISA-95 Attributes of Resource Properties

Core ISA-95 objects are collections of properties. Each property has the following attributes as shown in Table 2.

Attribute Name	Description	Example
ID	A unique identification of a specific property.	1999-10-27-a67-B6653
Description	Additional information about the <i>property</i> .	Pallet 2 of 6
Quantity	The quantity of the <i>property</i> .	40
Quantity Unit of Measure	The unit of measure of the associated quantity, if applicable.	Sheets

Table 2: Attributes for ISA-95 Properties for Objects

Manufacturing Operations Modeling with ISA-95 Classes

The first step in modeling objects is to determine the common properties between them and build *Equipment Classes* to match.

To make the process modeling easier, ISA-95 *Classes* organize properties for plant resources into common collections used as groups for scheduling and performance analysis.

Equipment such as machines often has common properties. For example, every machine that draws power may have a current draw property for monitoring energy use in a plant. Rather than add this property to every machine object, it is preferable to build a Powered Machine Class that contains those common properties. By adding this *Class* to every machine that uses power, they will all have copies of these properties. This not only makes the process of building models easier, it ensures common data across machines for reporting and analysis by *Class*.

For example, the equipment in a plant has the following *properties*:

Mixer 1:

- RPM Setting
- RPM Output
- Current Draw
- Operating Voltage

Mixer 2:

- RPM Setting
- RPM Output
- Current Draw
- Operating Voltage
- Mix Temperature
- Filter Screen Number

Wave Solder 1:

- Current Draw
- Operating Voltage
- Board Temperature
- Wave Height

Note: All three machines have a Current Draw and Operating Voltage *property*. To model, an *Equipment Class* called Powered Equipment is created to define those two *properties*. Then each machine in the model is assigned to the *Class*.

Note: The mixers both have an RPM Settings and RPM Outputs *property*. To model, an *Equipment Class* called Basic Mixer is created to define those common *properties*.

Finally, the unique *properties* left over are directly assigned to each *Equipment Definition*.

The result of this process looks like this:

Powered Equipment Class:

Properties: Current Draw, Operating Voltage

Basic Mixer Class:

Properties: RPM Setting, RPM Output

Mixer 1:

Classes: Powered Equipment, Basic Mixer

Mixer 2:

Classes: Powered Equipment, Basic Mixer

Properties: Mix Temperature, Filter Screen Number

Wave Solder 1:

Classes: Powered Equipment

Properties: Board Temperature, Wave Height

This modeling method of breaking down objects into common property sets is used throughout the ISA-95 Resource Models of equipment, materials and personnel.

Classes are also used to specify shared resources or resources that can be substituted. For example, by notifying a production controller or shift scheduler that equipment of the Basic Mixer Class is required for given *production order*, the scheduler then queries all available equipment that is a mixer and schedules one. Likewise, requesting a *Class* of operator called Mixer Operator allows the scheduler to find a worker who belongs to that *Class*.

The ISA-95 Role-based Equipment Hierarchy Model

The general structure of the ISA-95 *Role-based Equipment Hierarchy Model* is shown in Figure 1.

The ISA-95 *Role-based Equipment Hierarchy Model* is a hierarchy of various logical entities that are used to model aspects of production.

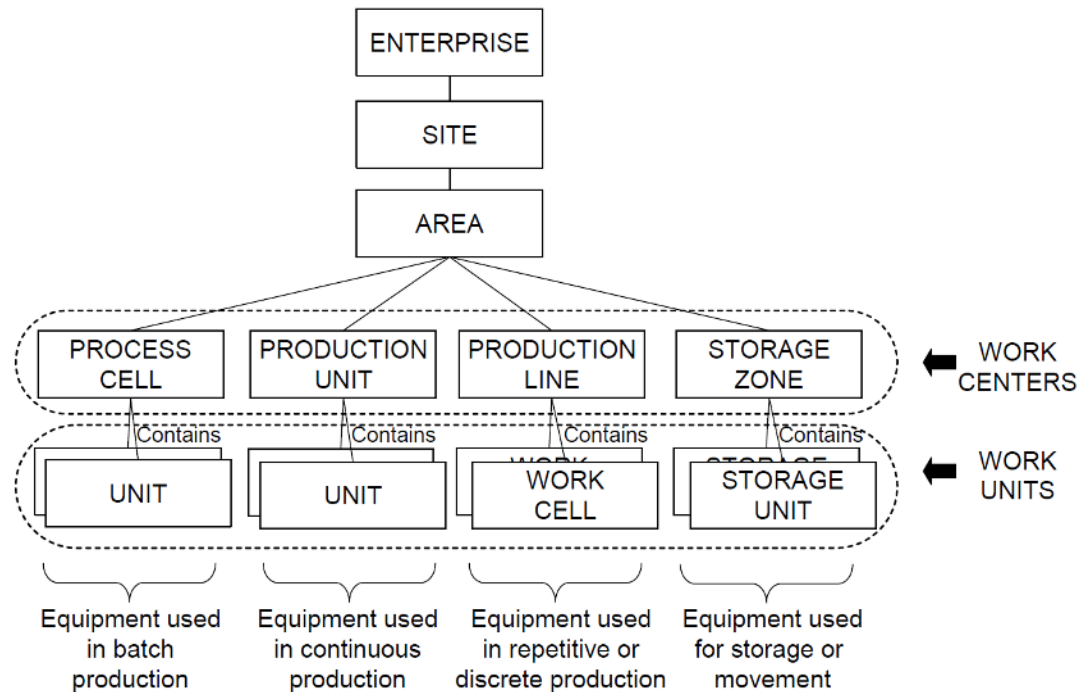


Figure 1: ISA-95 Role-based Equipment Hierarchy Example

The *Enterprise* represents an entire company or a division within a company.

The *Site* is typically a physical location of a business unit within the company.

The *Area* is a physical location within a *Site* that encapsulates a certain aspect of production based on the modeling of operations scheduling and performance tracking.

Segmentation below the *Area* contains *Process Cells*, *Production Units*, *Production Lines*, and *Storage Zones*. These are *Work Centers*. A *Work Center* is a grouping of equipment that you want to schedule, dispatch, report on, control or otherwise treat as a logical entity within an *Area*.

A *Work Unit* is a smaller entity that is a logical grouping of equipment and is used for reporting, scheduling, or controlling execution.

Some highlights of the *Role-based Equipment Hierarchy Model*:

- The concept of containing other equipment. Per the standard, an enterprise must have at least one *Site*, but a *Site* does not require an *Area*. A *production line* must have at least one *work cell*, but a *work cell* does not necessarily require an *equipment module*. Refer to UML Diagrams in the standard for modeling rules.
- The similarities between *production line*, *production unit*, and *process cell*. These are essentially the same level of entity applied to batch, continuous, or discrete production, respectively. When building out a plant that has only discrete production activities, the plant model would only include the *production line* and its child objects and would not use objects associated with other manufacturing types.

- *Storage zones* and *storage units* are independent of the type of production and represent any kind of storage entity: parts bins, warehouses, storage tanks for liquids, tank farms, and so on.
- *Equipment modules* are reciprocal and may contain other equipment modules, just as reciprocal *control modules* in an equipment model may contain other control modules. In this way, plant models are able to represent and build a complex, hierarchical model of each piece of equipment according to its production role for scheduling and performance tracking.
- Equipment modules and control modules are not actually a part of the base ISA-95 standard since they are Level 2 entities. They are added as extensions of the model to unify the model with the ISA-88 standard, which includes these objects. This allows a WPM system to model scheduling, dispatching, execution and tracking activities down to the control layer and control actions.

3. ISA-95 BASED WORK MODELS FOR OPERATIONS MANAGEMENT

The ISA-95 Resource and Role-based Equipment Hierarchy Models describe the assets and physical objects in an *enterprise* in the context of the role of operations in producing a set of products. The Process Models for *product definition*, *operations definition*, *process segment*, *work definition* and *work process segment* describe the processes, work definitions, and data flows within the *enterprise* and *site*.

Actual work associated with Level 3 MOM operations is defined in the *Work Definition Model* in Draft Part 4. Level 3 plant work is managed through *work requests* for specific operations activities for production, inventory, quality, and maintenance. *Work requests* are actual scheduled instances of a *work segment(s)*. Depending on scheduling methods, a scheduled *work segment* may be the single parent to recursive referenced collections of *work segments* down into Level 2 activities. As shown in Figure 2, when a *work request* is completed, it returns a *work response* containing pertinent data for recording what occurred when the work was done.

Level 3 MOM Work Model Objects

ISA-95 Parts 1 and 2 models described above in this paper are pseudo-static models because they describe the interfaces between Level 3 and 4 for logistics, planning and scheduling. They describe resource entities used in manufacturing, such as materials, buildings, and machines, for planning. As the products change in the plant, the plant model and its associated manufacturing operations models adapt, based on SKU count, process types and scale. As the product mix changes, manufacturing roles change, and then the resource and equipment hierarchy models change.

The Level 3 MOM Work Models describe actual operations management activities; the Work Models define how the work execution is defined, finite scheduled, dispatched, etc. including tasks, processes, recipes and information flows. The *Product/Operations Definition Models* and associated *Process Segment Models* determine how and when the resource entities described in resource models are used; *work segment(s)* and associated *work process segment(s)* are the executable tasks and processes.

Work Process Segment

In Level 3, a *work process segment* is defined as the smallest element of manufacturing activities that is visible to MOM processes and that corresponds to a detailed element of work that is scheduled, displayed, executed, and tracked by Level 3 activities. Work carried out in a plant is defined as a collection of such segments. *Work process segments*

are defined to be as general as possible, so they can be reused against more than one specific form of the operation. This allows *work process segment* reuse and common reporting and analysis of data returned by the segment.

A *work process segment* contains properties describing the parameters required to set up and carry out the specific process. It is defined by the equipment, materials, and personnel required to carry out the process. The *resource segment specifications* may reference specific objects or they may define an entire *class* of objects. By defining *resource segment specifications* by *class*, the process may be made flexible enough to handle variations in scheduling and dispatching or to provide different equipment for different variations of the process. Resource *parameters* and *segment specifications* are defined to allow the same process to be used in different ways for different applications.

The *work response*, which is the parent structure of the actual data collection outputs and response timing, is defined by the *Work Performance Model* but referenced in a *work request* by a unique *ID* identifier. The work response is a collection of *work segment responses* where each has an *ID* identifier and is associated with a specific *work request* and *work process segment*. Consequently, a specific *work process segment ID* may have many *work segment response IDs* based on the data collection segment specifications of a specific *work request* and associated *work segment*.

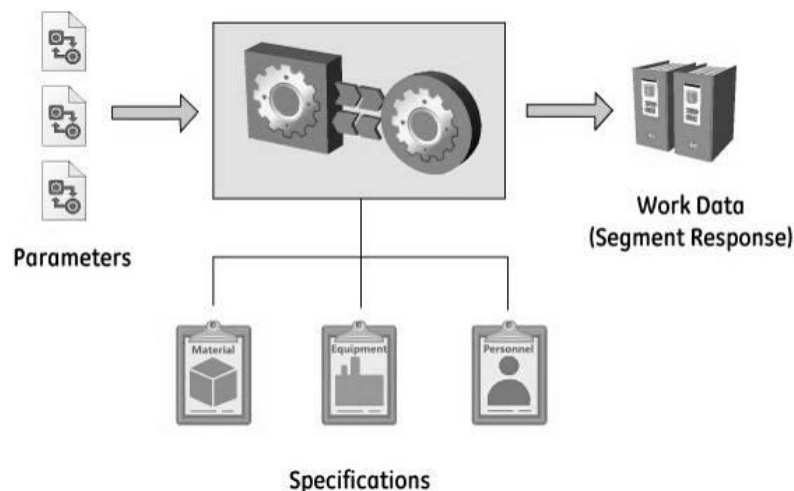


Figure 2: Work Process Segment Elements Relate to Work Segment Response

The resource descriptions of the materials, personnel, and equipment are *segment specifications* in the *work process segment*. In an industrial WPM application, the definition of the information that the segment needs to execute its work are defined by the *parameters* of the segment, and the *work segment response* data returned by the *work process segment* is defined by the work data definition.

A *work process segment* is recursive and may be modeled down to the specific unit procedure or operation within a unit procedure. However, a *work process segment* typically does not model below the unit procedure as described in ISA-88 due to the required level of expanded detail. The physical operation is broken out as an instruction and rules set in an ISA-95 *production rule* by an *ID* identifier for a specific product's *work segment* to carry out the process, recipe procedures, operator instructions or other execution controls. In an industrial WPM application, *production rule* information is contained in the workflow's *tasks* and sub-processes to control execution of the *work process segment*.

For example, consider a medical device product. A *work process segment* may be defined for the *work cell* that inserts components into a circuit board. The *resource segment specifications* are one grade-2 technician, a *material definition* for each component to be inserted, the *parameters* that specify the inputs to the process (insertion force, settings for the jig that holds the board, and temperature for the soldering guns). The *work process segment* defines the *parameters* and the *resource segment specifications* while the workflow (or ISA-95 *production rule*) contains the work instruction and business rules logic to act upon inputs and disposition the next *task*.

A later section of this white paper shows how to combine *work process segments* together to define a complete unit of work.

Work Segment

The structure of a *work segment* is specific to a given product and references a generic *work process segment* while a *work definition segment* supplies product-specific *parameters* and *resource specifications* for a *work process segment*.

The plant capability abstraction layer in ISA-95 is the *Product and Operations Definition* and *Process Segment Models* at the interfaces for Level 3 to 4. The reason for this is that many processes may be the same, or almost the same, across multiple products or processes. This method of plant modeling provides the database filtering for performance and root cause analysis across product line and process types. It is time consuming to copy every *work process segment* and make slight changes for each variation in the process; doing so would make data collection, analysis and tracking very custom and expensive.

In Figure 3, consider a *work process segment* that runs a wave soldering machine. Wave soldering operations have enough commonality between products that the process can be defined once for all of them. However, different products may require different fluxes. Some products may require an additional cleaning step. Different types of circuit boards may need to be pre-heated to different values. Different equipment may be used in the process; for example, a Mil-spec board needs a different cleaning step than a standard board, to include a high-quality cleaning machine instead of a standard machine and an operator trained to solder Mil-spec devices.

The goal is to write a *work process segment* once, write a workflow for task and sub-process instructions and rules once, and then parameterize the *work process segment* in a way that allows multiple *work segments* to be constructed to define the process differently for each different type of product or work.

Each *work segment* is associated with one *work process segment*. You cannot use the same *work segment* for multiple *work process segments*, nor can you have a *work segment* that exists on its own. However, a *work process segment* can be associated with more than one *work segment*.

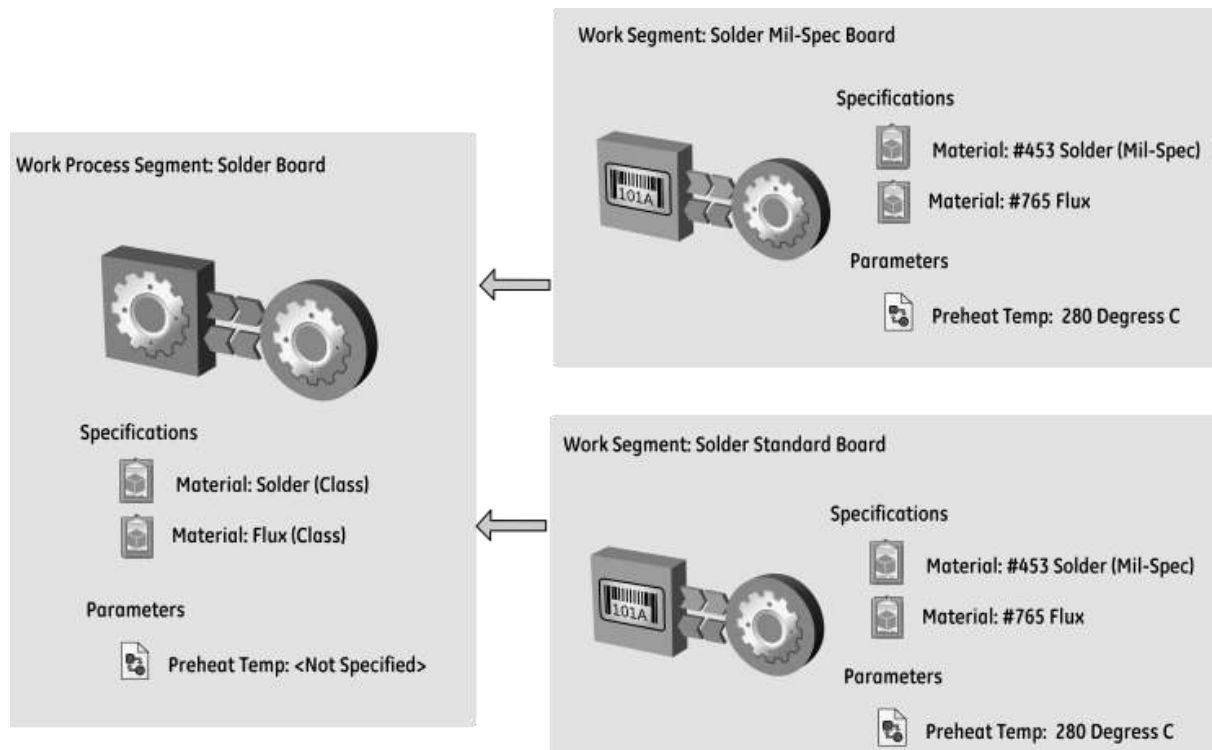


Figure 3: Process/Product Segment Relationships

Work Definition

A *work definition* is a collection of *work segments*; brought assembled together to represent the resources and data required to carry out a complete unit of work to produce a specific product in specific site or plant.

Note: The *work definition* itself does not describe *how* to do the work; it only defines the resource specifications, the *work segment* sequence and scheduling dependencies, and the *work segment response* information. The actual instructions for executing the work are built into workflow logic, contained in workflow forms that are presented to operators or written in work instructions that a workflow can provide when a task needs to be carried out.

A unit of work is any defined *task* or series of *tasks* required to carry out an activity in a plant. If the *work definition* is describing a production activity for a specific product, the parent object is called a *product definition* per original ISA-95 Part 2 or an *operations definition* with a *production* attribute per Part 2 update. An *operations definition* contains parent and child *work segments* for everything required to build the specific product. A *work definition* for a product has an *ID* identifier associated with an inventory number, a UPC (universal product code), or another identification of a specific product that is sold.

If a *work definition* is defined for a maintenance activity, it defines a parent *work segment* such as changing over a specific production line from one product to another, or replacing the blade of a cutting wheel when it becomes dull.

If a *work definition* is a quality-related activity, it defines a parent *work segment* such as taking a specific kind of air quality sample or removing specific sample products from an assembly line and carrying out a specific quality control test on them.

Work definitions are used to generate *work requests*. *Work requests* process work orders sent down by a scheduling system or entered by an operator, which triggers the activity in the plant. A *work request* packages up all the data associated with the *work definitions* and its *work segments* and makes it available to the process or to higher level business functions. *Work requests* are covered in more detail in the next section.

To determine the scope of a *work definition*, think about the activities in the plant that need to be individually monitored, scheduled, and reported on. Since a *work definition* is a collection of *work segments*, it applies to a specific thing or a specific *task*.

As an example, consider building a bicycle.

- A *work process segment* is Paint Frame and specifies that some type of enamel paint must be used
- A *work segment* for a Model #12 bike is Paint Model #12 Frame and specifies that the paint be high-quality gloss enamel for this bike
- A *work definition* would then be Build #12 bicycle and would contain *work segments* such as Paint Model #12 Frame, Attach Model #12 Leather Seat, and Package Bicycle in Model #12 box

Work Request

A *work request* is a request to carry out a specific *work definition* for a given work order. For example, a *work request* for 20 devices is issued against the *work definition* for Build A13 Medical Device, or there is a maintenance activity to test the air quality in the plant that needs to be performed once a day. Air Quality Test is defined as a *work definition* and then sets up a daily event that creates a new *work request* against this *work definition ID*.

In a WPM application, *work requests* may be created through a service provider call by an external system. For example, a master scheduler in a plant or an alarm system triggers *work requests* in response to unusual conditions through *workflow events*. For another example, the *work request* is created manually on the plant floor by a process engineer through a WPM console.

Regardless of how it is created, work requests listed above are all event driven in the nature, thus it is Event Driven Architecture.

Work Response

Work requests generate *work segment responses* from the parent object, *Work Response*. A *work request* provides *resource requirements* and *parameters*; the *work segment responses* return actual measured data as defined in the *segment data* and *resource actuals* objects.

Work responses are collections of *work segment responses* where each is associated with an executable *work segment* of a *work request*. Each *work segment* determines the *work response* data to be returned when the *work segment* is executed through the structure of the work data object associated with it in a WPM application. These *work segment responses* are organized into one *work response* in the WPM database.

For example, a *work process segment* is created that specifies that a circuit board be pre-heated to 400 degrees, soldered in a wave soldering machine, and then cleaned. For a specific circuit board product, CB1, a *work segment* (associated with the *work process segment*) defines *work response* data to store the actual temperature measured during

pre-heat, the duration of the soldering stage and the duration of the cleaning stage, and respond with that information when the process finishes.

If board failures are experienced in the field, a root cause analysis will be performed. Each failed board is tracked by way of the *work request* that created it; the *work segment responses* are looked up to discover patterns, such as that boards are failing when the pre-heat temperature is too low. Another root cause analysis may be based on the recorded specifics of the material used, since there may have been a substitution on the floor that correlates failures to the use of a certain brand of solder.

Work segment response information can also be used for inventory tracking. For example, a *work segment* may include a material specification to use two resistors of Part #324A, but one was found to be defective during installation and was replaced with another one. The operator brings up a workflow form on a WPM system console when a material quantity is changed and records the value in the form per the *work segment response*. The workflow application tracks actual amounts used versus amounts requested, then updates inventory accordingly while another workflow application simultaneously tracks and alarms spoilage and waste by product and process.

Figure 4 shows the relationship between a *work definition*, a *work request*, and a *work response*. The *work definition* provides the structure of the *work request*. The *work request* gets the product-specific *parameters* from the *work definition* where optional additional parameters are able to be added to the *work request*.

The *work request* executes each *work segment*, with the result being a *work segment response*. The executed *work segment response* is structured as a *work data object*. All of the *work segment responses* are collected together as a single *work response ID*.

Another way to think of it is this: A *work request* is made against a *work definition*. The result of the work being completed is the actual data that was returned. For example, a *work request* has a *material requirement* of two (2) Part #743A resistors. The *work segment response* records that three (3) Part #743A resistors were used because two (2) were consumed into the finished circuit board and one (1) was spoiled during production. The *work request* also has an *equipment requirement* that the pre-heater must be heated to 400 degrees. The *work segment response* records an actual measurement of 387 degrees.

The *work request* specifies the *resource requirements* for what is to happen in each *work segment*. The *work segment response* specifies *resource actuals* that are to be measured in each *work segment*. The cumulative *work response* indicates what actually happened and what was actually measured.

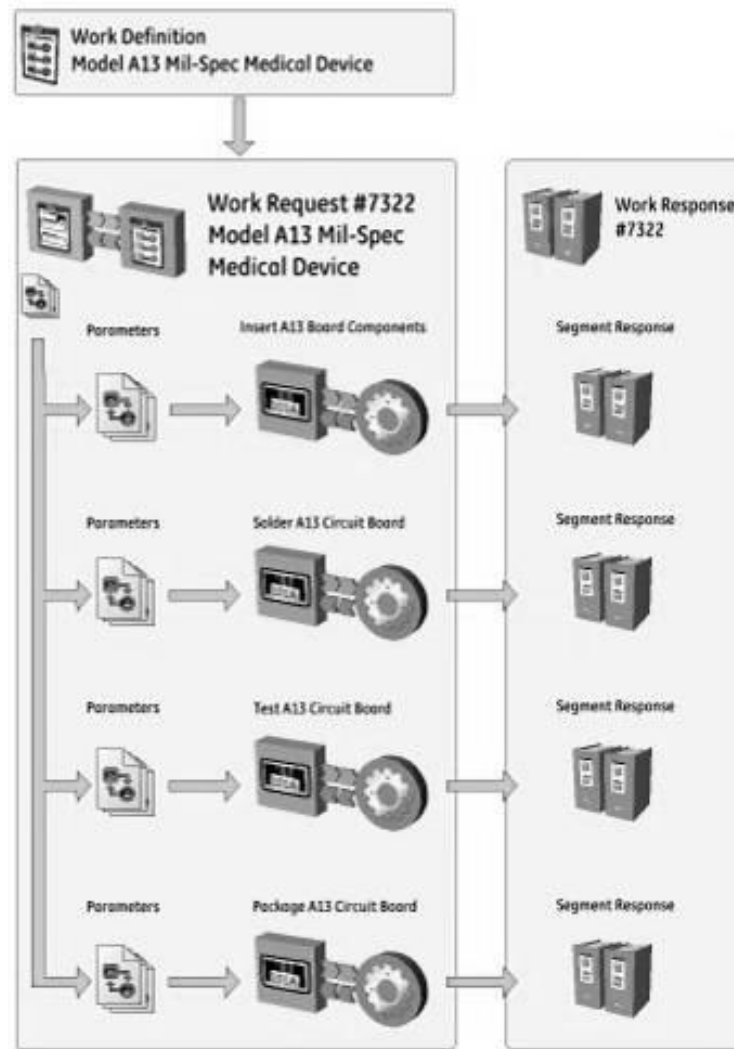


Figure 4: Work Request to Work Response

4. PUTTING THE ISA-95 MODELS TOGETHER

This section discusses how to break down the process of MOM modeling into the following areas and then relate them to each other:

- Role-based equipment hierarchy
- Material class and definition
- Personnel
- Plant processes
- Activities

Determining the Granularity of the Production Model

When deciding how to model a plant, the process engineer first needs to decide how granular to make the model, based on many consideration such as regulatory compliance, process complexity, quality tolerances, important costing principles, customer genealogy requirements, reporting, interface data, analytics for alarms and events, etc. For example, the process engineer could model an entire production line as one object or could break the production line into hundreds of smaller units, equipment modules, control modules, and so on. The choice of how granular to make the model

depends on what functionality set the model must serve. If visibility is required into a specific machine on the production line, the machine must be modeled separately.

In some cases, the process engineer builds the plant Production Model based on the *Production/Operations Definition* and *Process Segment Models* all the way down to each individual sensor and input on the plant floor. In other cases, if there is no business or operations need for seeing or controlling deeper into the process, the plant Production Model stops at some operations management Level 3 above the actual process.

The decision as to how far to take the model is often driven by the role of the equipment used. For example, in a flexible manufacturing line where each individual piece of equipment can be removed, reconfigured, and scheduled, the entire line is modeled down to each machine's equipment modules and control modules. This allows detailed *work response* or data collection that includes data from the process, operations routing, and quality for all materials types. In another line that is fixed in function and produces a single make-to-stock, strongly characterized product where parts are never individually scheduled or monitored, there is no business or operations requirement to model the lower level equipment.

Building a Production Line from the Ground Up

From the ISA-95 definition, a *production line* is a *work center* that contains equipment dedicated to the manufacture of a discrete product or family of products. A *production line* contains all the equipment required to make a discrete product or a component of a product.

A *production line* consists of multiple pieces of equipment; it is a good example of building an entire Production Model or plant model.

For this example, a process engineer builds a *production line* for manufacturing a circuit board for a medical device. Figure 5 represents the application's *Role-based Equipment Model*.

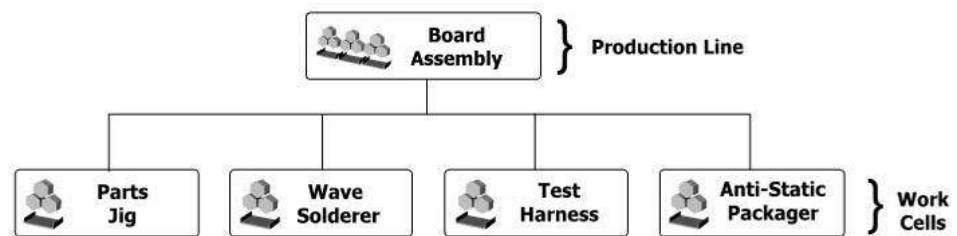


Figure 5: PC Board Assembly Line Model

This is a simplified model of the various stations on the assembly line. However, it is still useful for demonstrating the logical decision-making in the design process for modeling a plant.

Note: Some of these *work cells* do not represent a piece of equipment, as is the case in all plant models.

Test Harness is a *work cell* containing work units board holder and multimeter.

Parts Jig is a *work cell* containing work units jig bench, board holder and many parts bins.

Wave Soldering is a *work cell* containing work units: Fluxer Spayer, Conveyer, Wave Tank, Flux Cleaner and others.

Wave Soldering Machine Model

In Figure 6, the Wave Soldering work cell is modeled:

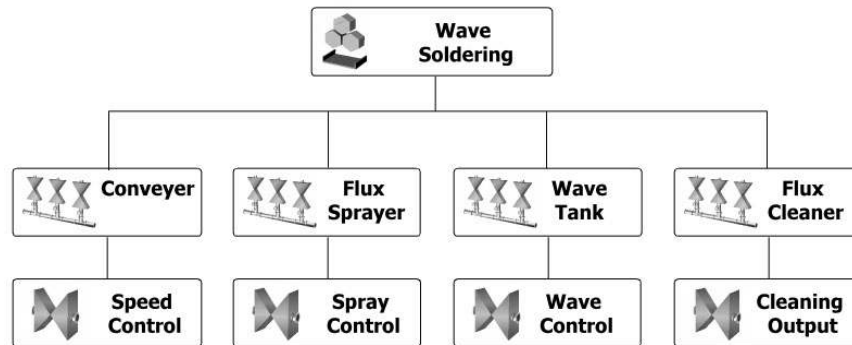


Figure 6: Wave Soldering Machine Model

The wave soldering machine is the core of a *work cell*. It consists of multiple *equipment modules* that are each subsystems of the wave soldering process. Each of those in turn contains *control module(s)* with the low level representation of the inputs and outputs they connect to.

The decision as to where to draw the boundaries depends on what is done at each station and where *work response data* is required for reporting, analysis, alarming, event notification, and operations considerations. Remember from the definition of a *work cell* that it is a *work unit* specific to discrete manufacturing, and a *work unit*:

- Typically has a recipe or procedure associated with it
- Is a component of an assembly line that adds value to the product or finished a defined step in a *work definition*
- Is a collection of equipment that may be scheduled together as a unit, has quality tests associated with it, or is individually monitored and/or controlled

Material Model

The *Material Model* in ISA-95 defines and tracks all of the materials created or consumed in the process of producing things. The physical amounts of the materials are modeled as properties of the *storage zones* and *storage units*. This is an acceptable approach if the quality of each individual lot of material does not need to be tracked but only requires random quality sampling, or if an audit trail of all materials and their quality tests for a specific product produced is not required to be maintained.

Personnel Model

The *Personnel Model* in ISA-95 is used to schedule, dispatch, and track the correct personnel resources used across operations activities in a plant. By fully implementing a *Personnel Model*, a key resource in the complete product history is tracked to know who was involved in any operations work request. The *Personnel Model* provides the ability to schedule personnel based on their specified capabilities and *availability*. Many WPM systems have task routing features to enable instruction or data forms to be routed to the system consoles in a *work cell* where dispatched users are logged-in and tracked. Dispatching may be done for individual users or for *classes* of users such as a shift or team.

Work Model

The Work Model for the wave soldering process is built of two parts:

1. *Work process segments* and their *resource segment specifications* are written generically to control the operations of the various components of the *work cell*
2. *Work segments* provide the *resource specifications* to the *work process segments* for the specific needs of a product for a specific work order type

The exact breakdown of where to define *work process segments* and how many *parameters* are provided depends on the specifics of the installation; that is, how many different products are made, how much commonality there is between them, how much individual scheduling needs to be done of the various components, and so on.

Building the Rest of Production

Following the example of the assembly line above, other assembly lines, storage zones and units, and other equipment used in production and in production support operations may be modeled.

When considering *Storage Zones* and *Storage Units*, each of the stations on the *production line* may have a number of *storage units* near them for necessary consumables. However, these *storage units* are not child objects of the *work cells*, even though they are physically located within them. *Storage units* are child objects of *storage zones* and do not have to be physically located within the *storage zone*. The important relationship is that the *storage unit* is *controlled* by the *storage zone*. The *storage zone* has the responsibility of maintaining inventory levels and the flow of materials. For example, a delivery truck on a highway is considered a *storage unit* of a *storage zone* 1000 miles away as long as its contents are controlled by the *storage zone*.

In some cases, a *storage unit* may be associated with a *work cell* even though there is no logical relationship between them. The association is made through the *process segment's equipment segment specifications*. By specifying the parts bins as necessary equipment for a given process, material transfers from the parts bins are tracked within workflows, tracking their quantities as process data in the *work response*.

Output Data Structure Design

ISA-95 Work Responses

ISA-95 *work responses* typically have several components:

- Values from *Resource actual* specifications in *work segment responses* that relate to *resource specifications* in *work segments*
- A *work data* block comprised of *segment data* from *work segment responses* that typically are actual values for selected parameters in a *work segment*. However, *segment data* may be product- or process-specific data specified in the *work request*.

Work responses consist of multiple *work segment responses*. A *work segment response* can also respond multiple times to each *work request*. For example, a *work request* has an equipment requirement for a given temperature; the *work response: work segment response* contains multiple actual measured values of the temperature. If the temperature is tracked every 5 seconds while a process is running and is then reported

as the entire array of values, each measurement would be its own *work segment response*.

Work Data Classes

In a WPM application, the *work data* block is used to structure each *work segment response*. The work data is a *class* like other ISA-95 classes. A *work data class* has no limit on the number of properties it may have. This *class* can then be associated with the *work process segment* as its *work segment response* in the Production Model in a WPM application. Workflow applications running against this *work process segment* can then populate the *work data class* with actual data and return it as part of the *work response/work segment response*.

Defining the structure of a *work data* block in this way makes it easy to ensure consistent reporting of data and also makes it easier to write reports and do analysis of the data.

5. BUILDING WORKFLOWS FOR WPM

After the resources and *work definitions* have been modeled, the process of writing the workflows required to carry out the *tasks* and collect the data needed for plant operations may begin. This white paper does not cover all the details of building a workflow, but it describes how workflows can be used in conjunction with ISA-95. This best practice should not be taken as the only accepted way to work with ISA-95 in an industrial WPM system and application, but rather as some helpful design patterns to illustrate the concepts.

Basic Organizing Principles Reviewed

A quick review of ISA-95 as modeled in a WPM system is useful in thinking about how to organize workflows:

- **Work Process Segment:** The lowest level of work processes exposed for business purposes. *Work process segments* can be applied to many products and are therefore somewhat generalized or generic.
- **Work Segment:** Essentially a template that provides specific executable information to *work process segments* in order to configure them for specific products or types of work.
- **Work Definition:** A collection of *work segments* that when combined; completely describe the process for carrying out a *task* in the *enterprise* or for building a specific product to a *work request* as an intermediate or finished good. For example, if a company makes ten different products, it would have ten work definitions, one for each product. If it has five different maintenance activities, it would have five work definitions, one for each activity.
- **Work Request:** A *work request* is a request to execute a specific *work definition*. If you have a *work definition* for building a car, a *work request* would be issued to initiate the process of building that car.

From these descriptions, a process engineer sees some general patterns emerging for plant operations workflows. Because *work process segments* are generic, typically workflows are written against them with the logic designed to handle the various parameters of the *work definition*. In this way, workflows only have to be written once. If one large workflow is written for the entire *work definition*, it will be less reusable when other *work definitions* are created that share some of the same processes.

Since *work definitions* contain all the work segment information without dependency or timing/sequencing information, process engineers consider writing a *master workflow* that corresponds to the *work definition* and contains the logic that triggers the various *work segment* workflows when they need to be executed. This *master workflow* would also create the *work response* and pass it to the *work segment* workflows so they can add their *work segment responses* to it.

The *work request* contains all the information needed to populate all the *work segments* and provides the interface between the various child workflows. Each workflow accepts the entire *work request* and is responsible for pulling its own *segment parameters* from the *work request* by ID. An industrial WPM system requires built-in functionality for managing *work requests* inside workflows.

Finally, unless *work requests* are driven directly from events, the next step is to create a scheduling workflow that generates *work requests* and triggers the appropriate workflow for each *work definition*.

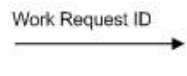
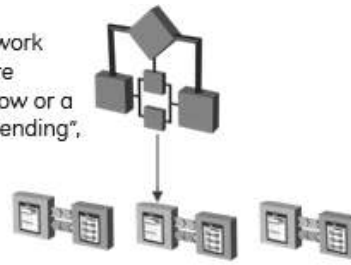
The Importance of the Work Request

As indicated in Figure 7, the *work request* ties everything together. By passing the *work request* from one workflow to the next, each workflow has access to all the data it needs. In addition, all data resulting from the execution of a *work request* is stored in a *work response*, which allows all process data to be associated with products, operations, work segments, and resources applied and created (equipment, personnel, and material).

IMPORTANT: All work requiring data collection should be initiated from a *work request* whenever possible.

There is nothing to stop a process engineer from simply writing workflows, connecting the inputs and outputs to data, and carrying out work. However, if the engineer takes the time to build proper ISA-95 structures and issues *work requests* against the work, work data is stored in a way that enables more effective, low cost system governance and data management. Structured data stores the data within the context of your operations and resources to provide an adaptable, user friendly environment for developing data collection, analytics, reporting, and interfaces/transactions.

A simple form for a product order generates work request(s) for each product ordered. These are marked "Unscheduled". Later, another workflow or a button on the form changes the request to "Pending", which is the signal to begin work.



A condition event fires when the new Work Request has a status change to "Pending", calling a dispatch workflow that determines which work definition the request belongs to.

Work Request ID

The dispatching workflow retrieves a Work Definition ID from the Work Request, and starts the appropriate Workflow Schedule for the Product Definition

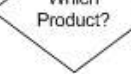


Definition schedule for Product A

Definition schedule for Product B

Work Request ID

Work Request ID



Work Request ID

Work Request ID



Definition Workflow for Product A

Product Definition Workflows call segment workflows



Definition Workflow for Product B

Work Request ID

Work Request ID

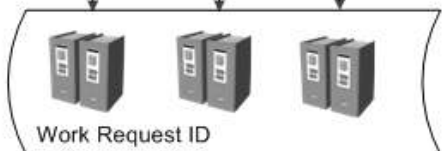
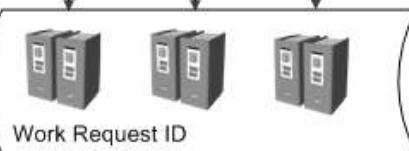


Segment workflows execute.



Work Data Work Data Work Data

Work Data Work Data Work Data



Work Response for Product A

Work Response for Product B

Figure 7: Typical Workflow Structure for ISA-95 Implementation of a WPM Application

Workflow Types and Samples

Basic Process Segment Workflows

These workflows control low level machine operations, present forms and work instructions to operators for data entry and acknowledgement, or monitor processes at the *process segment* level. For example, a simple workflow running on a *work process segment* for a *work cell* presents an operator with a form on the system console. The form provides instructions for attaching a part to a device and a button for acknowledging the completion of the task. The workflow then writes *work segment response* data containing production information to the *work response* and signals “complete.”

The example workflows below are typically associated with *work process segments* in the ISA-95 based Production Model and take a *work request* and a *work segment response* as inputs.

Sequencing Workflows (Work Definition Workflows)

A sequencing workflow monitors lower level workflows to validate that they have completed successfully and then triggers *events* to cause the next workflows in sequence to run (or to repeat the previous workflow or do other activities based on the internal logic of the workflow).

These workflows are typically associated with a *work definition* and are responsible for triggering the *work process segment* workflows at the right time.

These workflows also create the *work response* object, set the start time, and pass its *ID* to each *work process segment* so the segments can add their *work segment responses*.

Figure 8A shows a simple sequencing workflow that is the top level of the workflow. Each block has additional code underneath it. If the sequence of *work request* segments is exposed, the basic logic flow underlying the sequencer is shown in Figure 8B.

The workflow reads each *resource segment requirement* from the *work request*; for each requirement, it loops through a branch that reads the *work segment* identification and calls the appropriate workflow schedule.

Dispatching Workflows

A dispatching workflow executes when *work requests* need to be executed. Typically, this is triggered by an *event* which is filtered by a *condition event* that calls this workflow when the status changes to a level that indicates the *work request* is ready to be processed.

A dispatcher is required because many different *work requests* can be executed against the same *work definition*. Workflows are written specific to each *work definition*, so a dispatcher is needed to examine every *work request*, check the *work definition ID* to determine which sequencing workflow to call, and then call that workflow.

Scheduling Workflows

A scheduling workflow responds to orders sent down from a higher level ERP system or from some other insertion of a *work request* into the queue. This workflow then changes the status of the *work request* from Unscheduled to Pending; the status change is picked up by the dispatching workflow. A scheduling workflow is not required. *Work requests* may be generated that need to be executed as soon as they are generated; in

which case, the process engineer simply creates them, sets their properties, then changes their status to one that is picked up by the sequencer.

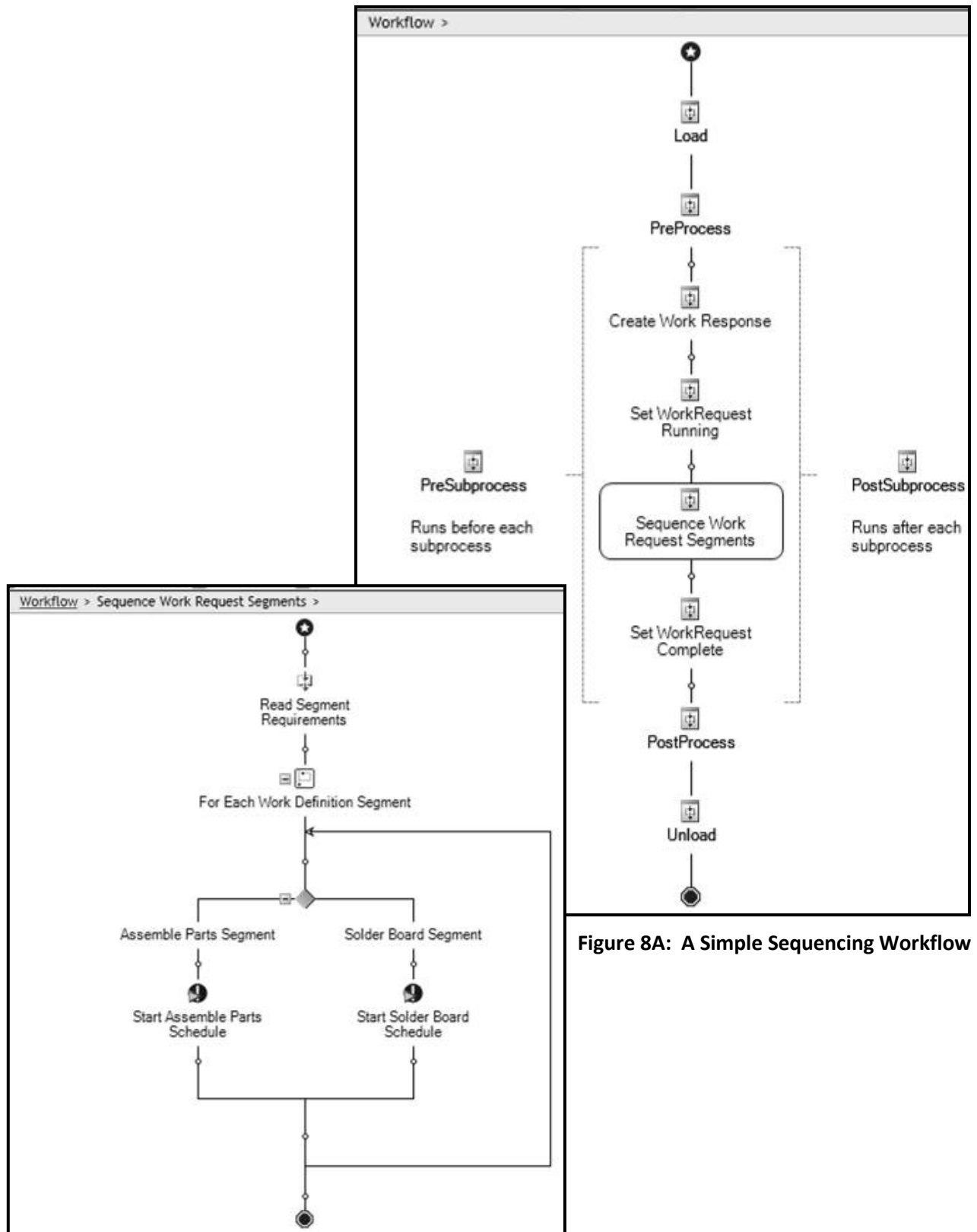


Figure 8B: A Simple Sequencing Subprocess

Other Workflows

In addition to the workflows associated with *work segments*, *work definitions* and *work requests*, other workflows can be running to monitor or control other aspects of production. For example, a workflow can be written to monitor inventory levels and when certain thresholds are reached, it automatically injects a *work request* to trigger an order for new material. There can be hundreds of active workflows executing in a WPM workflow application at any given time. Not all workflows need to be associated with ISA-95 objects except workflows carrying out structured activities for which data is tracked. Low level workflows that are performing orchestration between various data sources or monitoring plant activities may be written independent of the ISA-95 models.

6. IMPLEMENTING WORKFLOWS IN AN ISA-95 DATA STRUCTURE

This section explains:

- The process overview of connecting various ISA-95 objects together on a WPM platform
- The use of the various WPM components to enable the models to function in an executable application

The Big Picture

Before getting into the details of building an ISA-95 plant model and workflow application on a WPM platform, the following sections provide an overview of how the pieces fit together:

ISA-95 Equipment-Role and Resource Models

These data definitions are created in a WPM configuration tool. In some cases, more than one type of data object is added from a given workflow node.

1. Connections to Data Collection Sources

The WPM platform requires external data connectors. These connectors allow workflow applications to read data from historians, SQL data sources, service providers from MOM applications or other data sources provided by third party services. Properties in the ISA-95 models, events properties, workflow parameters, and other objects are bound to external data through these connections.

2. ISA-95 Work (Definition) Models

ISA-95 models are created in a WPM ISA-95 editor to define the processes, resources, products, MOM support operations (maintenance, inventory and quality operations activities), and other work you want to manage or analyze through the WPM platform.

- Work model objects are created under the Plant (Equipment-role) Model in a WPM configuration tool
- Work data classes are also created
- Work data classes are added to *work process segments* to define the output data structures of the *work response* for use in alarming, analysis, tracking genealogy, reporting, and interfaces

3. Workflow Applications

Workflow applications are the core of a WPM platform/system. A workflow application uses a graphical programming language, typically based on business process modeling notation (BPMN) or similar to write workflows to:

- Execute logic
- Read and write data from and to databases and the plant floor
- Send forms, notifications, and work instructions directly to operators
- Start or synchronize other workflows to carry out more complex operations

Workflows are extended with custom code activities that should be written in a modular fashion for reuse in many workflows.

A WPM configurator should design a library of built-in, reusable standard activities to offer a high level of functionality, such as the ability to interface with web services and connect with standard-based (such as B2MML or OAGIS) MOM applications, enterprise manufacturing intelligence (EMI) portals, and other available services.

4. Events

A WPM event engine is used to configure specific events to trigger workflows and pass appropriate time-based data to them. These events can be timed, fired by external events, or combined with expressions to make the events conditional.

5. Reporting

The WPM reporting database stores work data from manufacturing processes to allow a WPM reporting engine to automatically create reports on the stored data. Of course, the database allows output of data in other ways as well.

By putting all these pieces together, a process engineer is able to make and change complex MOM and control applications with a fraction of the effort required by more traditional development methods. Rapid change management and rapid response to abnormal conditions or loss of planned resources are the primary requirements for making adaptive manufacturing a reality.

A Very Simple Use Case – Making a Bicycle

To illustrate how the above components may be integrated into a workflow application, a very simple use case is considered that contains at least one element from each of the above categories of functionality. The use case is a hypothetical bicycle manufacturer, ACME Frames, who is trying to improve the process of making bicycles. One identified problem is paint inconsistency on the frame. Another is difficulty of assembly—sometimes parts do not fit the frame well. To this end, the ACME Frames want to use a WPM system with a platform approach to optimize this assembly process.

Step 1 – Determine ISA-95 Role-based Equipment Model

The bicycle assembly line consists of three pieces of equipment: 1) Frame Bender, 2) Paint Spray Booth and 3) Frame Assembly where the gears and handlebars are attached. Figure 9 illustrates this model application.

This Role-based Equipment Hierarchy Model for the ACME Frames plant starts at the *Work Center* level with the Frame Production Line, but in the full model for a company, there would be *Area*, *Site*, and *Enterprise* objects above it to complete a hierarchical Plant Production Model in the ISA-95 Role-based Equipment Model.

The *Work Center* level breaks down to the *Work Unit* level which can be a work cell, equipment unit, or storage unit. For the plant workflows for the Frame Bender and Frame Assembly Work Units, modeling Level 2 control processes was determined not to be necessary for the application because these processes do not require data connections into and out of Level 2 except for start/complete status. The processes for these units are determined by sending forms to the plant floor console to operators to provide work instructions and to indicate start/complete status.

Only the Spray Booth unit object has an *Equipment Module* (Spray Modulator object) and a *Control Module* (Nozzle Control object) level to illustrate the extended Equipment-role model where these ISA-88 objects extend the ISA-95 Equipment-role Model objects for a given workflow application. In this case, the model is connected to real-world data in a plant data historian, so the Spray Booth is modeled down to the Level 2 control processes where the Nozzle Control object requires the addition of properties for the settings; the properties are then bound to tags in the historian to read the actual settings values.

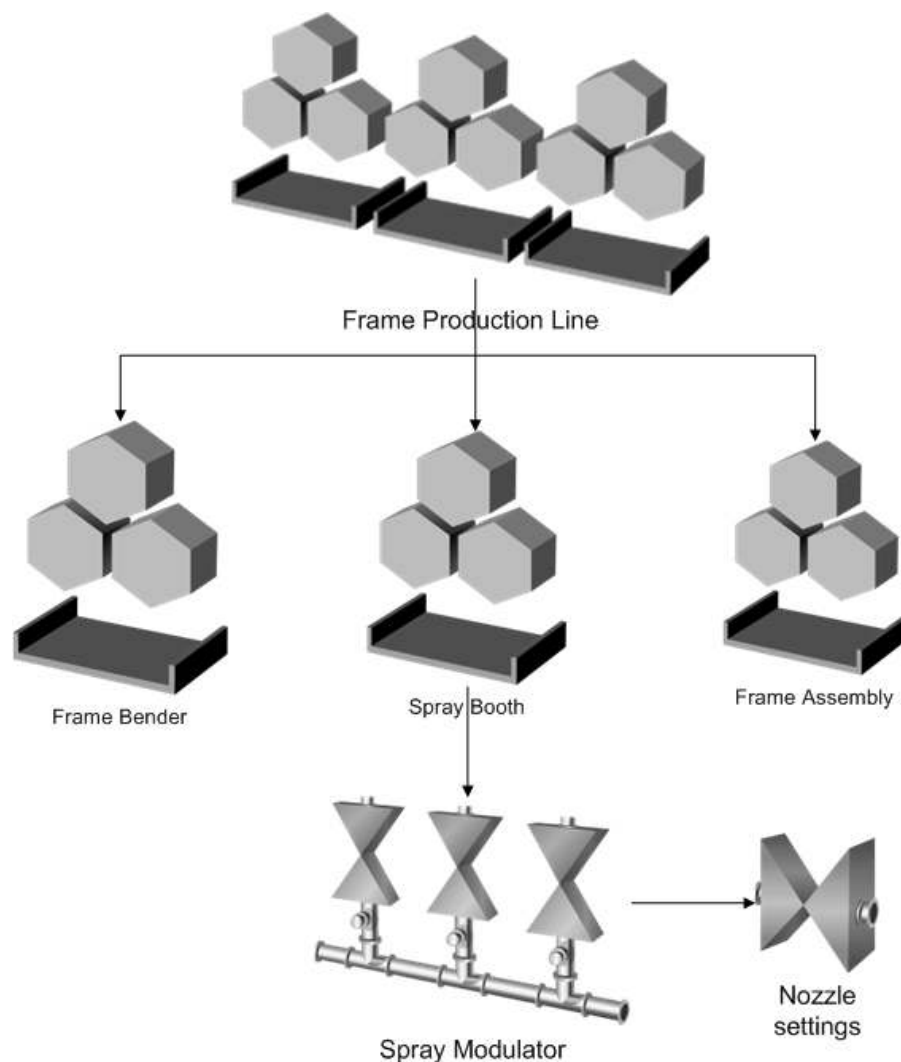


Figure 9: A Simple Role-based Equipment Model of a Bicycle Production Line

Create the ACME Plant Production Model in a WPM system:

In a WPM application editor for ACME Frames, configure the company's Role-based Equipment Model:

1. Add an *Enterprise* location with name ID and description of the *Enterprise* object
2. Below the *Enterprise* level, add *Site* objects, and then add *Area* objects to each *Site* object
3. For this application, add the parent Frame Production Line object for the bicycle frame to the *Area* object
4. Add three child *work unit* objects of the Frame Production Line: Frame Bender, Spray Booth and Frame Assembly work units
5. For the Spray Booth Work Unit, add a Spray Modulator Equipment Module to it
6. For the Spray Modulator Equipment Module, add a Nozzle Control Control Module to it
7. For the Nozzle Control Control Module, add properties for the input and output values objects to be used later when building workflows and data connections

Step 2 – Add Personnel and Material Resource Models

Now that you have ACME's Role-based Equipment Model, create the ACME Frames *Work Model*. The *work process segments* are associated with resource specifications for material, personnel, and equipment for each *work process segment*. Because a *material class* needs to be used as one of the specifications, create a *material class* object and a *material definition* object, and link the class to the definition.

Create a material object for the Frame Bender Work Unit:

In the WPM application editor for ACME Frames, create a *Material model*:

1. Add a *material class* object with the ID name of Aluminum Tubing
2. For the Aluminum Tubing material class, add *material definition* objects and each object's *properties* objects to this object that are common to all the different types of aluminum tubing used in manufacturing all of ACME's bicycles
3. For the Aluminum Tubing material class, add a *material definition* object as single instance with ID name 21" Frame Tubing
4. For the 21" Frame Tubing material definition, add actual *properties* objects of the material—something roughly akin to a material data sheet. It's up to the process engineer to decide how much information these definitions contain

Create the personnel object for the Frame Bender Work Unit:

In the WPM application editor for ACME Frames, create a *Personnel model*:

1. Add a *personnel definition* object as single instance with ID name called Welder Bob
2. For the Welder Bob personnel definition, add the actual *properties* objects of person

Step 3 – Connect to External Data

In the WPM application model for the Spray Booth *work cell*, add a *Control Module* called Nozzle Control. Connect this object to external data.

A WPM system and its applications should have the ability to bind object properties to real world data through various system and third party connectors (interfaces/APIs [application programming interfaces]/drivers). In this sample case, a historian data source connects a property of the object to a tag in the historian. The WPM system receives current values from the spray booth through one of the data connectors. The steps for connecting to data sources vary based on the needs of the data source, but should be similar.

Once the historian server is configured and tags are selected, these tags now bind to object properties to other properties in workflows, events, and other locations in the WPM system. In the WPM application editor for ACME Frames, select the equipment resource object to connect to external data.

If the process engineer only wants to bind object properties to current values, this configuration is done.

From this point on, when the value of that property is read, the WPM application will retrieve the value from the historian tag.

Step 4 – Add a *Work Data Class*

A workflow process retrieves real-time data. When a *work data class* object is created, a consistent structure is created for that data to make reporting and analysis easier. Create the class and then add it to the *work segment response* object later in the application configuration.

Create the *work data class* object for the Frame Bender *Work Unit* object:

In the WPM application editor for ACME Frames, add *Work Data Class* objects to the *Production Model*:

1. Add a *work data class* object called Frame Bender Actuals with an *ID* name and description of the *work data class*
2. Add a property called Seat Height Int32 as its data type

Step 5 – Determine the *Work Model*

Through Step 4, a representation of the ACME Frames' *Production Model* (*Equipment-role model*) is completed in the WPM application. The next step is to configure how the equipment is used in each workflow in the plant in a *Work Model*. The modeling process is not straightforward. The process engineer must choose how to represent workflow activities around each equipment object in the *Production Model* depending on the role of each equipment entity and whether it's used for multiple products or a single product, is scheduled separately, and so on.

The *Work Process Segment*

The lowest level of detail in a WPM *Work Model* is the *work process segment*. The *work process segment* is the smallest unit of definable work that needs to be exposed to your business processes. A *work unit* is typically the smallest piece of equipment tracked individually for business purposes such as scheduling, ERP, and asset management. Start by creating all *work process segments* for each *work unit* object such as Frame Bender.

For adaptive manufacturing, the *work process segments* are defined for the *work unit* in a generic form for reuse across all product types specified in *work segments*.

The structure of a *work segment* is specific to each product (intermediate or finished good) that utilizes generic *work process segments*. In an actual Level 3 *work request* from a Level 4 *production request*, a *work definition segment* then supplies product-specific *parameters* and *resource requirements* for *work process segment parameters* and *resource specifications*.

Work segments are created with specific information defined for a defined piece of work for a product type. For example, the Frame Bender *work unit* handles different tube diameters and has adjustable bend points to accommodate different bend radii or lengths. Each specific type of frame (product) requires values (*material specifications*) for tube diameter, bend point, bend radius and tube length, but at the *work process segment* level, common parts must be defined by value type and ranges so it works across a class of products such as a frame (a finished good material class). This property structure allows for reuse of the *work process segment* and any workflows associated with it.

For example, the Frame Bending *work process segment* has a *material specification* for tube length and tube diameter where the length and diameter depend on the type of frame being made. Frame Bending has a *personnel specification* for a Grade 1 welder and an *equipment specification* for the Frame Bender *work unit* defined earlier. The Frame Bender and Grade 1 Welder resource specifications do not change depending on the type of frame being made; however, the material specifications do and so do the values for tube diameter, bend radii and tube length. *Parameters* objects are created for these values in the *work process segment* so the *work process segment* works for any type of frame being built.

Next, consider the *work data* associated with the Frame Bending *work process segment*. For every frame, there are four measurements to make:

1. Measured height of the cross bar
2. Distance between the wheel mounts
3. Height of the seat mount
4. Distance between the pedal mount and the rear wheel mount

Create a *work data class* defining the values and associate the *work data class* with the Frame Bending *work process segment*:

Now, regardless of which frame is made, a workflow form from the WPM application is displayed to the operator to enter the measured values and store them with each frame for later use.

Create a *work process segment*:

In the WPM application editor for the ACME Frames application, configure the Production Model by adding *work process segment* objects:

1. Add a *work process segment* object with an *ID* name, Frame Bending, and a description of the work process segment
2. Add *specifications* for the following resource objects: equipment (Frame Production Line), material class (Aluminum Tubing), and personnel (Welder Bob)
3. *Parameter* values will be added later

Figure 10 shows the various elements of the *Frame Bending* work process segment. Frame Bender #1 and Welder Bob are required to carry out this work process segment. However, the material specification is only to the class level, which allows the work segment for each specific frame to define the exact *material definition* for a material instance to use.

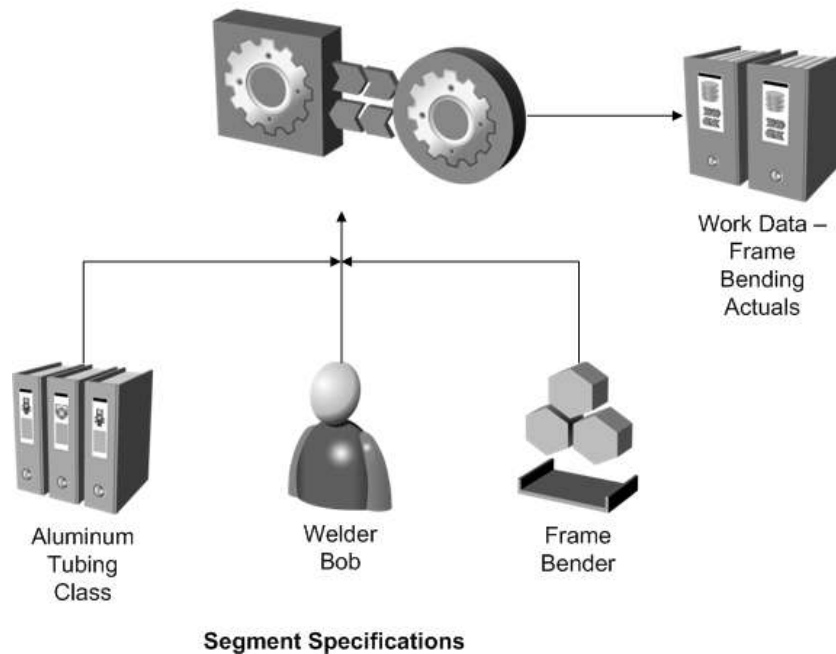


Figure 10: Work Process Segment: Frame Bending

The *Work Segment*

The *work segment* is associated with a *work process segment*. When a *work segment* is created, the associated *work process segment* must be chosen. When you create the *work segment*, the specifications defined for the *work process segment* are inherited; specifications may be further narrowed down in the *work segment* if they are specified to the class level in the *work process segment*.

Create a *work segment*:

In the WPM application editor for ACME Frames, configure the Production Model by adding *work segment* objects:

1. Add a *work segment* object, Model #12 Frame, with an *ID* name and description of the *work segment*
2. Associate the Frame Bending work process segment to the Model #12 Frame work segment object
3. The Model #12 Frame work segment inherits the resource specifications defined for the associated Frame Bending work process segment
4. *Parameter* values will be added later

The *Work Definition*

The *work definition* is a collection of *work segments* required to build a specific product to a Level 3 *work request* from a Level 4 *production request*. *Work requests* are issued against *work definitions* which define the resource requirements in actual setting for the quantity to be made. The Level 3 *work request* passes the *work definitions* to the

workflow application where the resource requirements are read into the specifications and parameters of the required *work segments*. A *work definition* would then be Build #12 Bicycle and contains *work segments* such as Model #12 Frame, Attach Model #12 Leather Seat and Package Bicycle in Model #12 box.

Create a *work definition*:

In the WPM application editor for the ACME Frames application, configure the Production Model by adding *Work Definition* objects:

1. Add a *work definition object*, Build #12 Bicycle, with an *ID* name and description of the *work definition*
2. Associate the Model #12 Frame *work segment* object with the Build #12 Bicycle *work definition*
3. For the resource requirements of the Build #12 Bicycle *work definition*, the structure of resource specifications is inherited from the associated Frame Bending *work process segment*
4. *Parameter* values will be added later

The *work process segment*, *work segment*, and *work definition* have detailed properties for Level 4 Production Scheduling. These are optional properties that are configured in the WPM application for advanced planning and scheduling and are beyond the scope of this white paper.

The *Work Request*

Now that a *work definition* has been created, *work requests* are created and issued against it. *Work requests* contain references to all the work objects in the *work definition*. *Work requests* can also trigger events when they are created or modified, which in turn can trigger workflows to execute logic. The workflows reference the *work request* so that they can read specifications, parameters, and other data associated with the *work request*. These features are discussed in the next section.

Work requests are created by workflows or created manually.

Manually create a *work request*:

In the WPM application editor for ACME Frames application, configure the Production Model by adding a *work request*:

1. Add a *work request* with an *ID* name, Order #1 – 21 inch Bicycle and description of the *work request*
2. Associate work definition, Build #12 Bicycle, with the work request, Order #1 – 21 inch Bicycle
3. Add specific Resource Requirements for all *work segments* in the associated Build #12 Bicycle *work definition* for the Order #1 – 21 inch Bicycle *work request* with controls for each of the requirements. Fill in any missing specifications or parameters for these requirements.

Step 6 – Configuring Events

An industrial WPM system has an event engine that contains built-in events and allows events to be defined. Of the many event types built into an industrial WPM event engine, five types of the most critical events are described:

1. Time Events
2. Condition Events
3. Workflow Events
4. Production Events
5. Defined Events

An *event* can trigger workflows. *Events* also pass attributes data to workflows or schedules when the *event* occurs. Different events can have different attributes. For example, the New Work Request event has an attribute containing the Work Request ID. The Work Request Status Changed event has an attribute containing the Old Status and the New Status.

Time Events

A *time event* is a defined event occurring during a specified time interval. A *time event* is set to occur once or to occur repeatedly by the month, week, day, or a set time interval. *Time events* are set to occur indefinitely or to stop on a specified date and time.

Condition Events

A *condition event* is a compound event. A *condition event* has a *control event* to triggers it. When triggered, the *condition event* evaluates one or more expressions, and if the defined conditions are met, it then fires itself.

For example, inventory levels are checked once an hour and trigger a *work request* to order more inventory when it reaches a certain threshold. A *time event* is created to fire every hour and then to assign it as the *control event* to the *condition event*.

Condition events are very powerful as they allow workflows to trigger only when needed, and to route *work requests* to the proper workflows when they come in.

The variables used in the expression. Each *event* can have variables defined for it; these variables can be bound to properties on other ISA-95 objects in the Production Model. Variables can be configured three ways:

1. Bind the variable to external data: Configure a data item to a property on the equipment and bind to the equipment
2. Bind the variable to an attribute of the Event Trigger: Configure the Attribute, Data Type, and Default Value
3. The Variable represents a constant value: Configure the Data Type and Value

Workflow Events

Table 3 describes the common *workflow events* in an industrial WPM application that fire when certain actions occur in a workflow.

Event	Description
Task Expired	Fires when an expiry time is reached before a task completes.
Task Step Expired	Fires when an expiry time is reached before a task step completes

Table 3: Common Workflow Events

Production Events

Table 4 describes the common *production events* in an industrial WPM application that are designed to aid in implementing ISA-95 *work requests*:

Event	Description
New Work Request	Fires when a new work request is created.
Work Request Status Changed	Fires when the status of a work request changes.

Table 4: Common Production Events

The New Work Request event is fired when a new *work request* is created, but the *work request* may not be filled out yet. This is useful for some advanced purposes, but would not typically be used to trigger the workflows that carry out the work. For that, use the Work Request Status Changed event and use this event as a trigger for a *condition event*. The *condition event* examines the status of the *work request* to determine if it is ready to be processed.

Typically, the statuses on the *work request* are: Unscheduled, Pending, Running, Paused, Completed, Failed, Canceled, and Unknown.

By creating *condition events* comparing these statuses, workflows are triggered to respond to failed processes or to notify a scheduler when a *work request* is completed, and so on.

For example, a *condition event* is created to look for the status of any *work request* changing from Pending to Failed; it is used to trigger a workflow to report the failure or to re-inject the *work request*. Alternatively, the *condition event* could test for a status of Completed, which when confirmed then triggers to sequence the next *work request*.

Defined Events

All WPM installations have their own set of built-in events. If they are built on a service-oriented architecture, custom application-specific services expose their own events; these become available for use just as the built-in events are.

7. INTERFACING ERP WITH WPM SYSTEM IN MOM LEVEL 3

B2MML Transaction Processing of MOM Workflows

The WPM system's work processes require external input and outputs to execute each work task that make up data elements as the basic pieces of an information exchange. Data element mapping is the final stage in specifying WPM systems integration. Detailed interface definitions must include data element mappings at the lowest levels. This requires specific elements from source systems, such as table and field elements, and specification of how they map to specific target WPM system elements.

Business to manufacturing integration has come a long way from the early days of ERP (Enterprise Resource Planning) and MES/MOM systems. Projects that once took multiple calendar years and tens of man-years, yet had low success rates, have given way to projects that require less than one man-year and have a high probability of success. This has been due in part to industry adoption of the ISA-95 Enterprise/Control System Integration standard. The WBF Business-To-Manufacturing-Markup-Language (B2MML) standards are XML schemas that implement the ANSI/ISA-95 standards (Parts 1, 2 and 5) and are designed specifically to interface to Manufacturing IT systems.

As more companies are now using B2MML for systems integration projects, issues with data formatting and data structuring have mostly been identified and are well understood. What remains in integration projects are the agreements and specification of integration patterns and specific data elements. Integration patterns define the methods, protocols and patterns to be used in the integration of applications.

The ISA-95 Part 5 standard wraps the resource and capability models from Parts 1 and 2 in a set of transactional models using B2MML that include action verbs. The transaction verbs in Table 5, when combined with the Parts 1 & 2 B2MML schemas, denote the type of processing as well as responses expected.

The WPM system must have the ability to preprocess and validate the Part 5 transaction messages, possibly alert personnel to the receipt of or prior to sending messages. These messages, collectively, interact with the Scheduling workflows to launch Dispatching Workflows which then process data in the transaction messages with a WPM Work application by the creation/modification of *work requests*.

Transaction Verb	Description	Response Verb	Description
Process	Receiver is expected to Add contained elements to target system	Acknowledge	Respond with resulting elements in target system, including any errors.
Change	Receiver is expected to modify contained elements in target system	Respond	Respond with resulting elements in target system, including any errors
Get	Receiver is expected to fetch contained elements from target system	Show	Respond with elements contained in target system
Cancel	Receiver is expected to remove contained elements from target system	(none)	Respond with a general Trans Response, if requested.
Sync(Add, Change, Cancel)	Receiver is expected to carry out sub verb transaction based on a subscription.		

Table 5: Part 5 Transaction Verbs

Transaction Models

In addition to defining the transactions involved, ISA-95 Part 5 and B2MML define transaction models for communication between the ERP, MOM systems and their WPM systems. Three basic types of transaction models are typically defined:

- **Push:** The transaction is sent from the source system to the target system for processing. An ERP sending a MOM (and WPM) system a ProcessProductionSchedule message is an example of a Push model. A MOM system sending a ProcessProductionPerformance message to an ERP system is another example of a Push model.
- **Pull:** The target system sends a request to the source system for the data needed. An ERP sending a MOM (and WPM) system a GetProductionPerformance message is

an example of a Pull model, while a MOM system sending an ERP system a `GetProductionSchedule` message is also a Pull model transaction.

- Sync: Both the source and target system subscribe to certain types of messages on a message bus and wait for such messages to appear and then process them, as opposed to using a call method and return. An example here would be to utilize the WPM system's Event Architecture to encapsulate the message in an event that can be subscribed to.

Choosing only one transaction model is not necessary. Based the business processes and operations work processes, a combination of transaction models are typically used to best adapt implementation for both the ERP and MOM systems. The WPM workflows should be designed to at least support both the Push and Pull models.

Interface Workflow Types and Samples

Inbound Transaction Processing Workflows

The interface workflows are designed to process a specific incoming transaction message for a WPM application, such as `ProcessProductionSchedule` (Push), `ChangeProductionSchedule` (Push), `GetProductionPerformance` (Pull), and so forth. Some automated preprocessing of the message is typical to apply business rules specific to a WPM application. For example, a business rule is applied through an interface workflow to not allow the modifying a *work request* once it's move past a particular stage in production.

Additionally, the WPM application must notify plant personnel that a transaction has been received and is pending processing against the ISA-95 data models of the WPM system. The interface workflow presents the contents of the message to the supervisor or production controller in a graphical form which allows them to drill down to various levels within the message and review the contents of the message in terms of scheduling operations workflows. The graphical UI allows the user to make necessary modifications to message data to correct any errant data. Finally, the user approves or rejects the transaction for execution by the WPM application.

If the transaction is approved then processing of the message continues in WPM application to execute the data per the ISA-95 models. Generally, this message processing includes either the creation or modification of *work requests* in the WPM system fire the events that trigger the dispatching workflows or similar tasks.

Once the contents of the message are processed into the WPM ISA-95 models, the results of the transaction are compiled into the appropriate response message for the transaction; the interface workflow sends the response transaction to the ERP system.

Inbound Message Watcher Workflows

Depending on the type of B2M interfaces with an ERP system, e.g. Web Services, FTP, etc., the WPM application typically requires a workflow that is triggered to run when an event occurs in the WPM system that indicates receipt of a transaction message. This type of workflow is similar to the dispatching workflows in that it examines the message to determine the type of transaction to be processed. This so called Message Watcher workflow then calls the appropriate Inbound Transaction Processing Workflow for the transaction type. This is just a simplified example of a WPM application may interact with external systems to process B2MML transactions.

A Watcher workflow may not be necessary if the WPM system exposes an interface via the SOA platform with enough granularities to allow the ERP system to direct the transaction messages to the appropriate Inbound Transaction workflow.

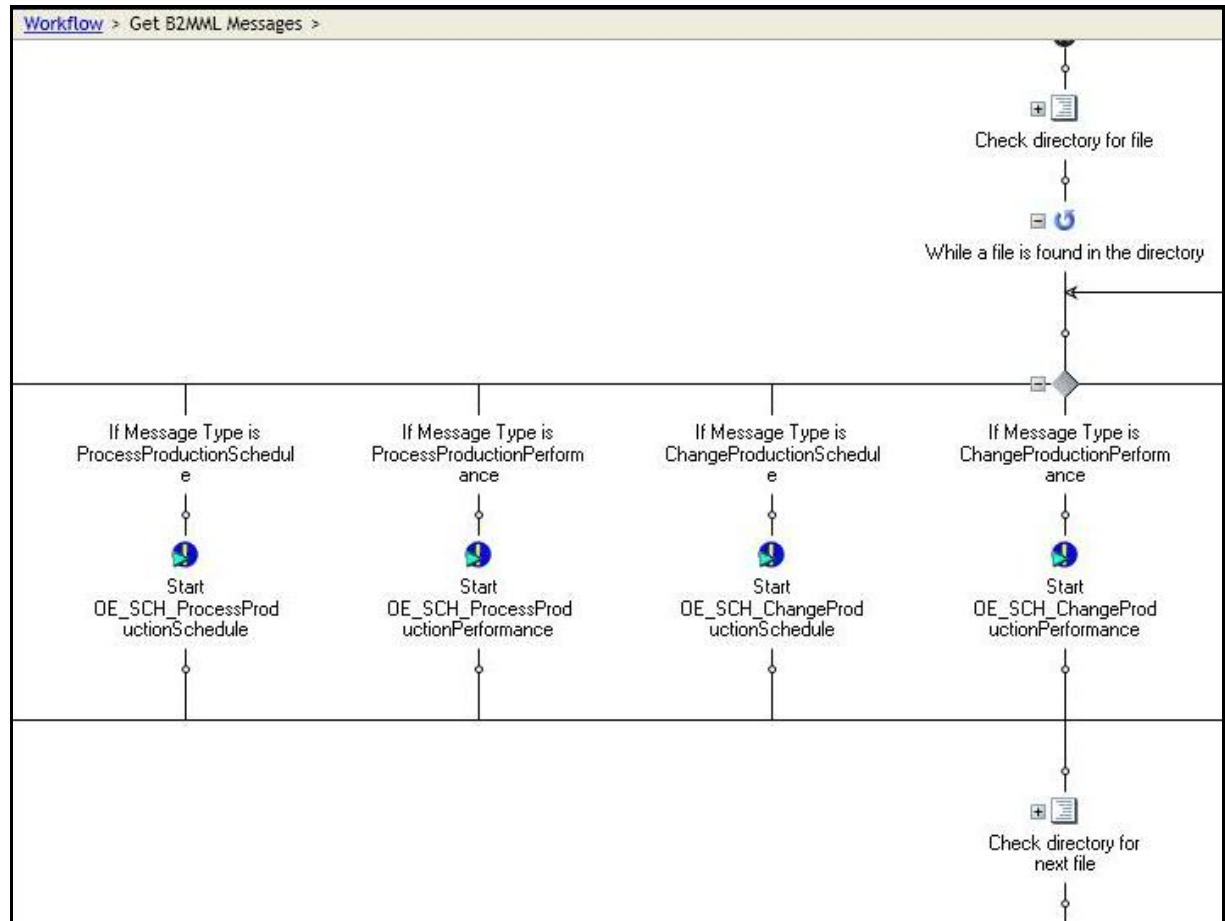


Figure 11: Example of an FTP-based Message Watcher Workflow

Outbound Transaction Processing Workflows

These workflows are similar the Inbound workflows since they are designed around specific types of send transaction messages, such as ProcessProductionPerformance (Push), ChangeProductionPerformance (Push), and GetProductionSchedule (Pull) to the ERP system. These messages are triggered by events which are fired by the changes made to the *work responses* by the Basic Process Segment and Sequencing workflows.

Their purpose is to extract data from the WPM ISA-95 models and compile it into a transaction message and send it to the ERP system, such as ProcessProductionPerformance. The Basic Process Segment workflows can trigger incremental uploads of ProductionPerformance data to the ERP system while the *work request* is still being processed in production. This allows the ERP system to monitor and track progress of the *work request* through the production process. The Scheduling workflow initiates the final ProductionPerformance upload of a completed *work response*.

Either on a periodic or event driven basis due to lack of work, the WPM system may initiate a Pull model transaction, such as GetProductionSchedule, on the ERP system to request more work.

Service Interface

Within an SOA framework, a higher level interface it may be designed around the production schedule and performance transactions to simplify the processing of messages into the actual ISA-95 models of the MOM system. The implementation of such an interface abstracts the lower level interface methods that must be called to process or extract data from the models.

The work definitions contain a sufficient level of detail necessary to create the basic *work request* in the MOM system. However, the ERP system applies specifics to the *ProductionRequest* that need to be applied to the basic *work request* that is generated. For example, a production parameter value for paint color might be specified by the ERP system in the production request. Additionally, a production request specifies particular equipment instances and raw material lots to be used based on the ERP production scheduling. This can require several transaction method calls on various interfaces to alter the *work request* attributes, override segment requirements, and parameters. Implementing a higher level interface in this way can simplify the development of the transaction processing workflows.

The method prototype below is specific to a transaction. It accepts the transaction message as an input and then outputs the appropriate response as well as an optional message transaction confirmation. The return type is a Boolean to indicate if there are any errors in processing the message. A return value of 'false' indicates that the response and transaction confirmation should be examined for detailed error information.

```
Bool ProcessProductionSchedule  
(in ProcessProductionScheduleType, out AcknowledgeProductionScheduleType,  
optional out TransConfirmType)
```

Ideally, the interface and parameter types support SOA standards for communication, such as Web Services, in which the parameters are easily serialized and de-serialized across environment boundaries. For example, parameter types that support XML serialization of B2MML.

CONCLUSION

The ISA-95 models are broad and generic in scope, which gives them the required flexibility to fit into numerous plant environments. However, this strength of ISA-95 models makes them somewhat difficult to understand and apply in the real world unless they are applied within the plant model of an industrial WPM workflow application.

- To characterize the complexity of granular tasks and their parent operations workflows within Levels 2 and 3, the ISA-95 common metadata models must be extended to characterize a plant's workflows and associated tasks, dependencies, and condition based responses.
- A WPM MOM workflow application based on extended ISA-95 definitions permits the alignment of MOM systems to execute co-dependent operations process in a very adaptable method.
- Aligned MOM systems rapidly respond to changing resource states and order demand priorities to lower costs of non-value-added activities while enabling "pull" make-to-order supply chains. This aligned workflow foundation allows the rapid development of MOM applications, interfaces, reports, and notification/acknowledgement of alarms and events.
- The implementation of the ISA-95 standard defines and contextualizes WPM applications for adaptive manufacturing.

Optimizing manufacturing operations requires each plant's workflows and resources to be modeled into and aligned across integrated real-time transaction database applications.

An operations workflow engine is the key unifying application to align business and operations processes across MOM applications for rapid notification and acknowledgement of alerts, alarms and events to trigger alternative and corrective workflows and tasks. This new generation of workflow based MOM systems is capable of performing near real-time root cause analysis while tracking order routing and batch processes and the movement and status of plant resources.

GLOSSARY, ACRONYMS, AND ABBREVIATIONS

WORD/TERM/ACRONYM	DEFINITION
API	Advanced Programming Interface. Interface technology that enables calling functions directly inside another software program.
B2MML	See Business To Manufacturing Markup Language. See also: B2MML Schema.
Best Practices	The most efficient (least amount of effort) and effective (best results) way of accomplishing a task, based on repeatable procedures that have proven themselves over time for large numbers of people.
BPM	See Business Process Management and Business Process Modeling.
BRD	See Business Requirements Document.
Business Process Management	The development and automation of new and integrated business processes to assist in real-time business visibility.
Business Process Modeling	Analysis and schematic drawing of the business processes of a company.
Business To Manufacturing Markup Language	A common XML data representation of the ANSI/ISA-95 family of standards (ISA-95). It brings together various manufacturing data models such as production schedule, product definition, material, equipment, labor, etc. See B2MML Schema.
CMMS	Computerized Maintenance Management Software
CoA	Certificate of Analysis
CRM	Customer Relationship Management
CSV	Comma Separated Values. Commonly used file format where the information is separated by commas.
DCS	Distributed Control System
ERP	Enterprise Resource Planning : a planning system for manufacturing, order entry, accounts receivable and payable, general ledger, purchasing, warehousing, transportation and human resources.
FIFO	First In First Out
I/O	Input/output
Level 1	In ISA-95, defines the activities involved in sensing the production process and manipulating the production process.
Level 2	In ISA-95, defines the activities of monitoring, supervisory control and automated control of the production process. It deals with time frames on the order of hours, minutes, seconds, and subseconds.
Level 3	In ISA-95, defines the activities of work flow, stepping the process through states to produce the desired end products. It deals with maintaining records and optimizing the production process. Level 3 deals with time frames of days, shifts, hours, minutes, and seconds.
Level 4	In ISA-95, defines the activities of establishing the basic plant schedule—production, material use, delivery, and shipping. It deals with determining inventory levels and making sure that materials are delivered on time to the right place for production. Level 4 deals with time frames of months, weeks, days, and shifts.
LIMS	Laboratory Information Management System
LIFO	Last In First Out. How things are put into and taken out of, for example, a stack or queue.
Manufacturing Operations Management	Activities within level 3 of a manufacturing facility that coordinate the personnel, equipment, and material in manufacturing.
MOM	Manufacturing Operations Management. Also Message Oriented Middleware: software that resides in both portions of client/server architecture and typically supports asynchronous calls between the client and server applications.
MRP	Material Requirements Planning—predecessor to MRPII
MRPII	Manufacturing Resource Planning—predecessor to ERP

WORD/TERM/ACRONYM	DEFINITION
OAGIS	Open Applications Group Integration Standard
OPC	OLE for Process Control is a communication standard for Control Equipment based on OLE concepts. The foundation's goal is to develop an open and interoperable interface standard based on the functional requirements of OLE/COM and DCOM technology that fosters greater interoperability between automation/control applications, field systems/devices, and business/office applications.
PDM	Product Data Management
PDM/PLM	Production Data Management/Product Lifecycle Management
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
Production Capability	Information exchanged on required, committed and available resource capacity. The amount of output a process stage segment is capable of producing. The data collected under the Production Capability model can include capability information about equipment, material, personnel, and process segments.
Production Performance	Information exchanged on what was made and what resources were used.
Production Rule	Information exchanged on how to make a product.
Production Schedule	Information exchanged on what to make, when and where to make it, and what resources to use.
Production Units	
QA	Quality Assurance
QMS	Quality Management System
SCADA	Supervisory Control and Data Acquisition
SKU	Stock Keeping Unit (number)
SOA	Services Oriented Architecture
SQL	Structured Query Language. Programming language used to interact (e.g. select or insert data) with databases.
Tracing	Also: traceability. The attribute of a software system that allows the determination of an assembly item location or process to be determined at any given time
Tracking and Tracing	See Tracking. See Tracing. The two attributes of software systems are inter-dependent since software systems can only trace data that has been tracked.
WBF	Formerly World Batch Forum, now WBF, the Organization for Production Technology.
WMS	Warehouse Management System
WPM	Work Process Management
UML	Unified Modeling Language
UPC	Universal Product Code

AUTHORS

Dan Hanson

Senior Software Engineer
GE Intelligent Platforms
780-420-2113
Dan.Hanson@ge.com

Greg Millinger

Proficy Workflow - Product Manager
GE Intelligent Platforms
419-341-1427
greg.millinger@ge.com

Nick Miller

Senior Software Engineer
GE Intelligent Platforms
518-464-4503
Nick.Miller@ge.com

CONTRIBUTING EDITOR

Charlie Gifford

President and Chief Manufacturing Consultant
21st Century Manufacturing Solutions LLC
208-788-5434
charlie.gifford@cox.net

REVIEWERS

Kwangnam (Kevin) Cho

Enterprise Solution Architect
Samsung Electronics Co., Ltd.
82-31-20-87240
kwangnam.cho@samsung.com

Arne Svendsen

Head of Manufacturing Services & Automation
Arla Foods Global IT (GBSIT)
Arla Foods
+45 2071 7380
arne.svendsen@arlafoods.com

Jan Kudsk

Senior Business Manager
Arla Foods Global IT (GBSIT)
Arla Foods
+45 8938 1891
jan.bo.kudsk@arlafoods.com



About GE Intelligent Platforms: GE Intelligent Platforms is an experienced high-performance technology company and a global provider of software, hardware, services, and expertise in automation and embedded computing. We offer a unique foundation of agile and reliable technology providing customers a sustainable competitive advantage in the industries they serve, including energy, water, consumer packaged goods, government & defense, and telecommunications. GE Intelligent Platforms is headquartered in Charlottesville, VA. www.ge-ip.com.



About MESA: MESA promotes the exchange of best practices, strategies and innovation in managing manufacturing operations and in achieving plant-floor execution excellence. MESA's industry events, symposiums, and publications help manufacturers, systems integrators and vendors achieve manufacturing leadership by deploying practical solutions that combine information, business, manufacturing and supply chain processes and technologies. Visit us online at <http://www.mesa.org>.