Model-based Operational Control Methods for Smart Manufacturing Systems

Timothy Sprock
Conrad Bock
Systems Integration Division,
National Institute of Standards and Technology,
Gaithersburg, MD
Why is good “scheduling” so hard?

Implementing smart operational control systems is often hampered by:
- Heterogeneous information sources
- Heterogeneous decision support tools
- Heterogeneous execution mechanisms (shop floor “actuators”)

**Information Sources:**
- Part
- Process
- Resource
- Planning
- Orders
- Shop Status

**Decision Support:**
- Priority Rules
- Scheduling Software
- MES? MRP?
- Optimization Methods
  - Static? Dynamic? Robust?
- Simulation

**Execution:**
- Robotic Arms
- Conveyors
- Overhead Transports
- Automated Guided Vehicles (AGV)
- Humans
Model-based Operational Control

Three components of successful development and deployment of model-based operational control:

• **Standard model of operational control**

• Analysis models and tools properly implementing that standard (interoperability)

• System-analysis integration methods providing automated, inexpensive access to those analysis tools
Operational Control Model Overview

Goal: Standard way of describing the base system and operational control of each “functional unit”
Operational Control Model Overview

Manipulating flows of tasks and resources through a system.

- Which tasks get serviced? (Admission/Induction)
- When \{sequence, time\} does a task get serviced? (Sequencing/Scheduling)
- Which resource services a task? (Assignment/Scheduling)
- Where does a task go after service? (Routing/Dynamic Process Planning)
- What is the state of a resource? (task/services can it service/provide)
Operational Control Model Library

**Functional Capabilities and Resource Roles:** Building blocks for assembling models of system capable of implementing operational control
Reference Patterns

Templates guide implementation of operational control building blocks in a system design.

For existing systems, patterns guide discussions on how operational control works: Where/when is the decision made? How is it made? How is the control decision executed?

Pseudo-checklist for system designers to provide resources (system objects) to fulfill these operational control roles.
Standard Decision-support Interfaces

Controllers are configured with algorithms that provide decision support for each control decision.
# Patterns for Modeling Operational Control

Link decision support in the controller to behaviors and actuators on the shop floor.

<table>
<thead>
<tr>
<th>Question</th>
<th>“In what order (sequence, time) should tasks be served?”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sequence</strong></td>
<td>Task → N</td>
</tr>
<tr>
<td>Decision Function</td>
<td>Sequence(TaskSet) := sort(TaskSet, sequenceIndex) = TaskSet’</td>
</tr>
<tr>
<td>Decision Expression</td>
<td>$x_{lk} = 1$, if task $l$ is serviced $k^{th}$</td>
</tr>
</tbody>
</table>

### Sequencing

**Strategy**

sequencing(out sequenceIndex : Integer [1..], taskSet : Task [1..])

**Assignment**

“Which resource is assigned to service the task?”

$\text{Assign: Task} \times \text{Resource(s)} \mapsto \text{Resource(s)}$

$\text{Assign}(\text{Task}, \text{Resource}) :=$

Task.nextProcessStep.requiredInputResource ← Resource

$x_{lm}^{m} = 1$ if resource $m$ is assigned to execute the next process step of task $l$

$x_{lj}^{j} = 1$ if resource group $R$ is assigned to execute the $j^{th}$ process step of task $l$

### Assignment

**ResourceAssignment**

assignment(availableTask : Task [1..], availableResources : Resource [1..], out resourceAssignment : Resource [1..])

**ResourceAcquirer**

inTask : inDELSTask [1..]$

\begin{block}
\text{Queue}
\end{block}

outTask : outDELSTask [1..]$

\begin{block}
\text{ResourceAcquirer}
\end{block}

\begin{block}
\text{availableResource : inDELSTResource [1..]}
\end{block}
Goal: Standard Interface Enables Interoperability

package Scheduling [ schedulingDecisionSupport ]

DecisionSupport
  context Interface(
    context
  )

Sequencing
  sequencingInterface
    1..*
  sequencing()

ResourceAssignment
  assignmentInterface
    1
  assignment()

Scheduling
  assignment( availableTask : Task [1..*], availableResources : Resource [1..*], out resourceAssignment : Resource [1..*] )
  sequencing( availableTask : Task [1..*], out sequenceIndex : Integer [1..*] )
  scheduling( availableResources : Resource [1..*], availableTask : Task [1..*], out sequenceIndex : Integer [1..*], out resourceAssignment : Resource [1..*] )

CLK_ITP
ClarkWright_VRP
Example: “Adapt” Algorithms to Interface

Existing analysis models, such as those for scheduling, don’t naturally conform to the standard and need to be adapted to become “plug-and-play.”

```python
classdef Scheduling_ClarkWrightVRP < SchedulingInterface  
    %implements SCHEDULING_STRATEGY  
    %using Clark Wright VRP algorithm
    properties
        end
    methods
        function Scheduling(self, TaskList, ResourceSet)
            % 1. Add Depot
            % 2. Make Cost Matrix & Capacity Array
            % 3. Call Clark Wright Savings
            % 4. Assign ordered tasklists to resources
                for j = 1: min(length(ResourceSet), length(loc))
                    ResourceSet(j).TaskList = TaskList(loc(j));
                    ResourceSet(j).TaskListCapacityRequirement = TC(j);
                end
        end
        end
```
System-Analysis Integration Methods

• Use a common representation of the system under control (system model) to integrate multiple sources of information already defined and/or represented in other ways, often from heterogenous systems in incompatible formats, to create an integrated model of the system.

• Integrate system models with many kinds of analysis models, such discrete event simulation.
Goal: Enable Simulation-based Methods

• [Design] Standard system models and supporting analysis methods will enable simulation-based methods to be routinely applied during the (re-)design process to test and validate control logic in high-fidelity simulations before deploying to the system.

• [Operation] Simulation can also be integrated with optimization and heuristic methods to provide online decision support.

• [Goal] Pathway from design and analysis of control to testing, validation, and deployment.
Need for Model-Based Methods for Smart Manufacturing

• Current methods and tools are limited for production systems engineering
  • Formal specification & analysis automation
  • Design and teaching

• Documentation & Organization of Knowledge
  • Existing Systems Models (industry)
  • Existing Analysis Models (academia)

• Bridge between system and analysis models
  • Interoperability between different analysis models of the same system
  • Greater reusability of analysis: collaboration and automation
  • Modeling & Simulation Interoperability (MSI)
Model-based Operational Control

Challenges:
• Heterogeneous information sources
• Heterogeneous decision support tools
• Heterogeneous execution mechanisms (shop floor “actuators”)

Approach:
• Standard model of operational control
• Analysis models and tools properly implementing that standard (interoperability)
• System-analysis integration methods providing automated, inexpensive access to those analysis tools

Information Sources:
- Part
- Process
- Resource
- Planning
- Orders
- Shop Status

Decision Support:
- Priority Rules
- Scheduling Software
- MES? MRP?
- Optimization Methods
- Static? Dynamic? Robust?
- Simulation

Execution:
- Robotic Arms
- Conveyors
- Overhead Transports
- Automated Guided Vehicles (AGV)
- Humans
More Information

• System-Analysis Integration (SAI) Project
  • Conrad Bock, project lead  conrad.bock@nist.gov
  • http://www.nist.gov/el/msid/syseng/smsi.cfm

• Discrete event logistics systems (DELS)
  • Tim Sprock, timothy.sprock@nist.gov
  • INCOSE Production and Logistics Systems Modeling Challenge Team
    • http://www.omgwiki.org/MBSE/doku.php?id=mbse:prodlog