SDCWorks: A Formal Framework for Smart Manufacturing Systems

MATTHEW POTOK, CHIEN-YING CHEN, SAYAN MITRA, **SIBIN MOHAN**
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Introduction

Unscheduled downtimes in manufacturing systems

- Affect Overall Equipment Effectiveness (OEE)
- Cyberattacks, machine failures, random faults

Production changes

- Introduction of new product line
- Demand fluctuations

Software-defined control (SDC)

- Inspired by software-defined networks (SDN)
- Global view of manufacturing plant
- Separates the control (logical) plane from rest of system

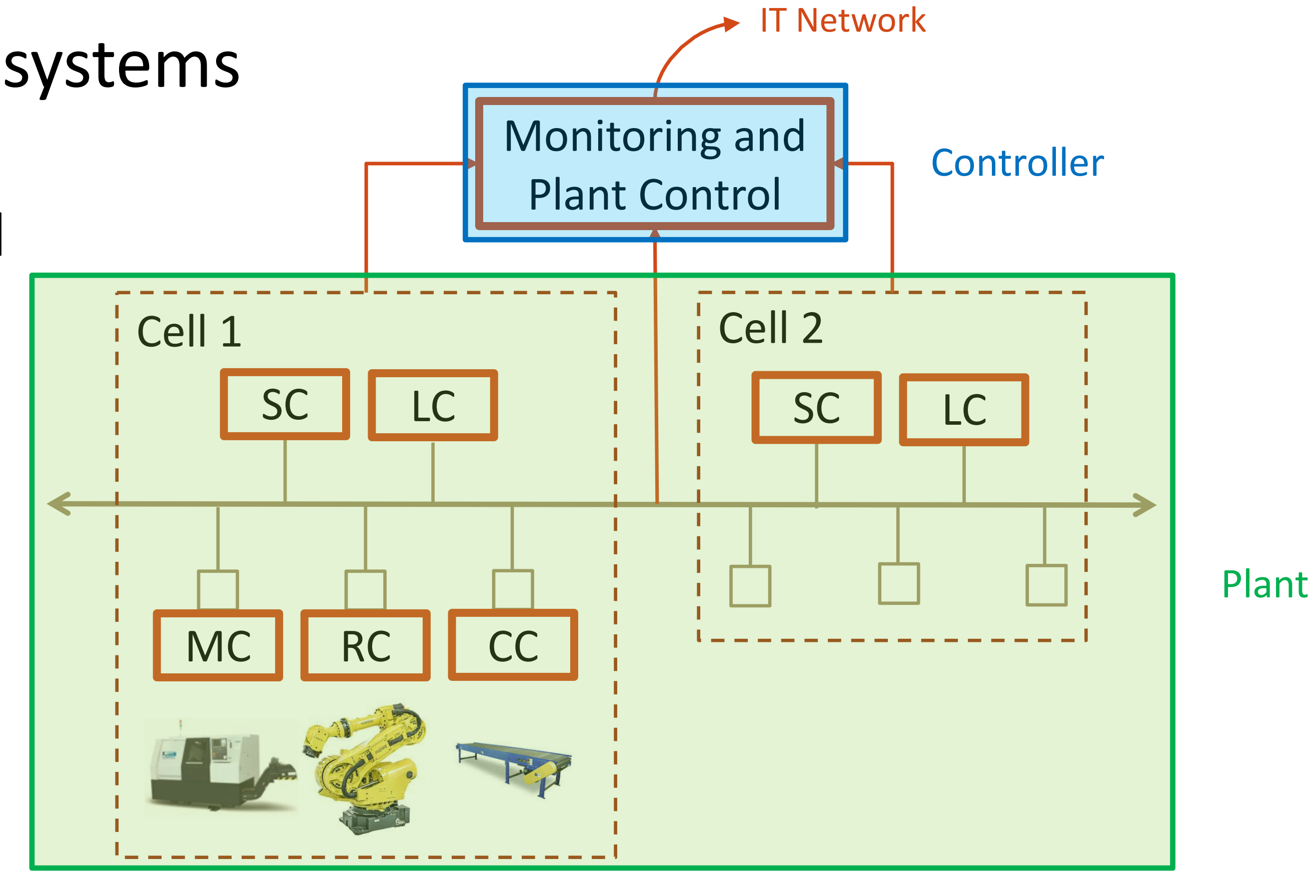
Modern [Discrete] Manufacturing Systems

Components of manufacturing systems

- Material handling devices
- Controllers: low-level or high-level

Abstractions

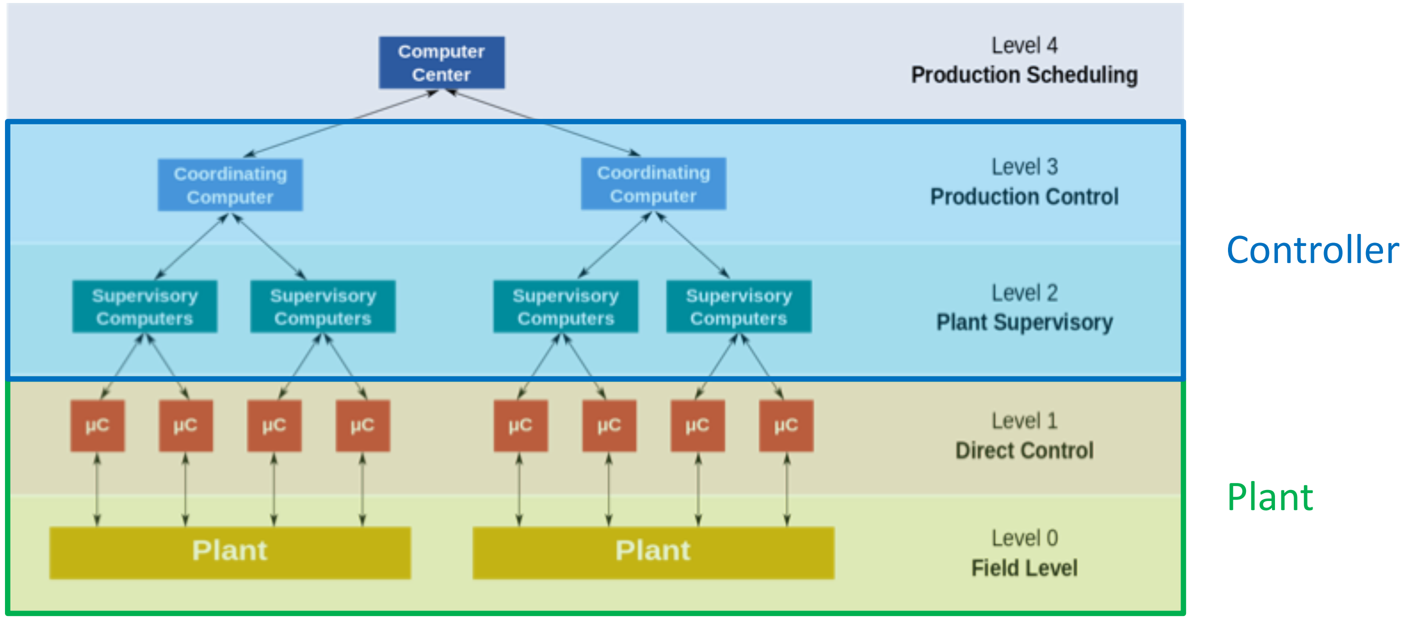
- Controller
- Plant



Modern Manufacturing Hierarchies

Levels

- Level 0: Production processes
- Level 1: Sensing and manipulation
- Level 2: Monitoring, supervisory and automated control
- Level 3: Work flow control and optimization
- Level 4: Business Logistics



https://commons.wikimedia.org/wiki/File:Functional_levels_of_a_Distributed_Control_System.svg

Overview

Formal modeling framework for discrete manufacturing systems

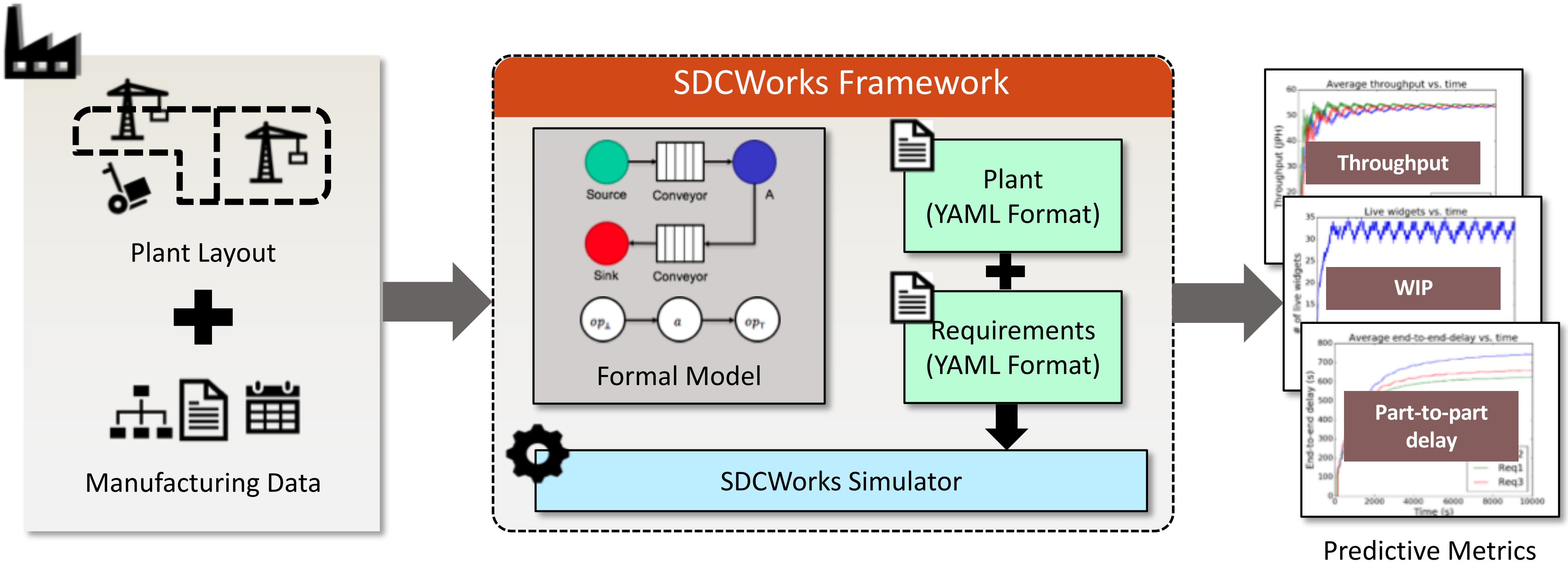
- Captures **cyber and physical aspects** of systems
- Enables verification of controller properties
- Expressive to model variety of plant assets and configurations

Simulator implementation as **instance of framework**

- Provides analysis and performance metrics
- Flexibility to support custom controllers
- Open source

SDCWorks

SDCWorks Framework



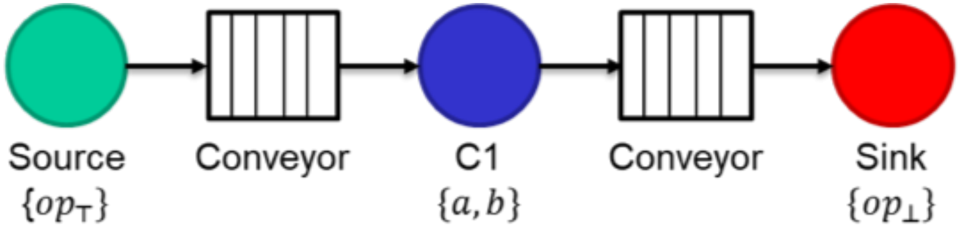
SDCWorks Framework Overview

Discrete Transition System

- Plant
- Controller

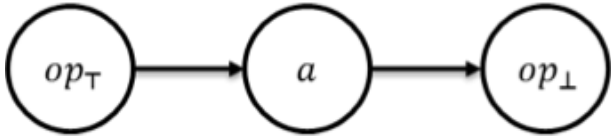
Major components:

- Controller
- Parts
- Plant
- Requirements



Operation table

	C1
a	10
b	20

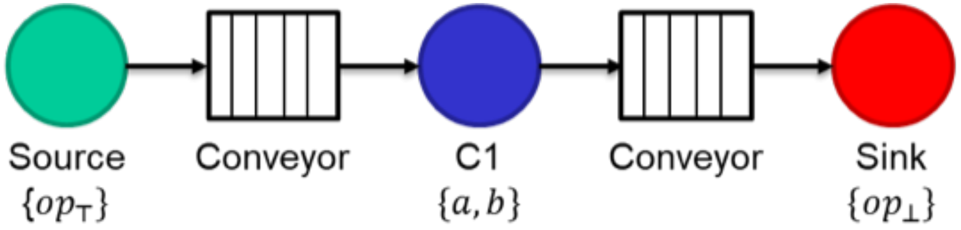


Cells and Plant

Cells: any **devices** that can **perform** some **operation** on a **part**

- Individual machines, robots, conveyors
- Set of multiple devices or even other cells

Plant: models **floor plan** of a manufacturing system → layouts of cells & connections



Operation table

	C1
a	10
b	20

Plant formally defined as $\boldsymbol{P} = \langle \boldsymbol{G_P}, \boldsymbol{L_P}, \boldsymbol{T_P}, \boldsymbol{Q_P} \rangle$

- A graph of the cells, $G_P = \langle V_P, E_P \rangle$
  - V_P is the set of cells
  - E_P defines connections between cells
- Set of operations that can be performed at each cell $L_p: V_p \mapsto 2^{\overline{\mathrm{OP}}}$
- The amount of time for each operation $T_p: V_p \times \overline{\mathrm{OP}} \mapsto \mathbb{N}$
- The maximum number of widgets waiting at each cell $Q_p: V_p \mapsto \mathbb{N}$

Parts, Requirements, and Controller

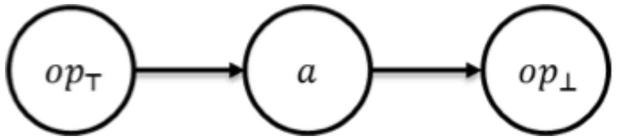
Parts/widgets: the materials that traverse the plant and have operations on them

- Uniquely identified

Requirements: the sequence of operations to be performed on parts

Each requirement defined as

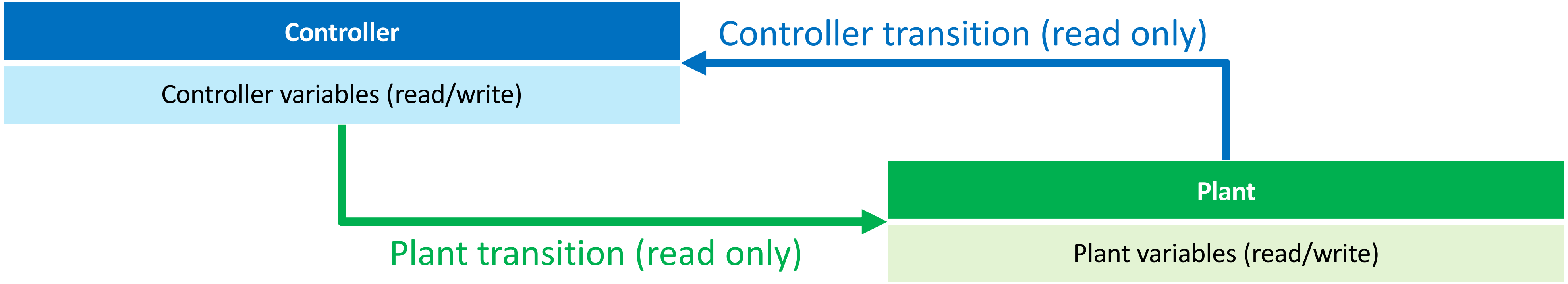
- Directed acyclic graph $\boldsymbol{R} = \langle \boldsymbol{V_R}, \boldsymbol{E_R} \rangle$
  - V_R is a set of operations
  - E_R defines a sequence of operations with precedence constrains
- Mapping of vertices in R to operations $\boldsymbol{L_R}: \boldsymbol{V_R} \mapsto \overline{\boldsymbol{OP}}$



Controller:

- Plans a sequence of operations according to requirements
- Commands cells to perform certain actions

SDCWorks Discrete Transition System



Discrete transition systems defined as $\langle \boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{A}, \boldsymbol{D} \rangle$

- finite set of variables partitioned into controller and plant variables ,$X = X_C \cup X_P$
  - Valuations of X, $val(X)$ map each $x \in X$ to a value and are called *states*
- set of initial states, $\Theta \subseteq val(X)$
- finite set of actions partitioned controller and plant actions, $A = A_C \cup A_P$
- set of discrete state transitions, $D \subseteq val(X) \times A \times val(X)$

Baseline Controller

Plans

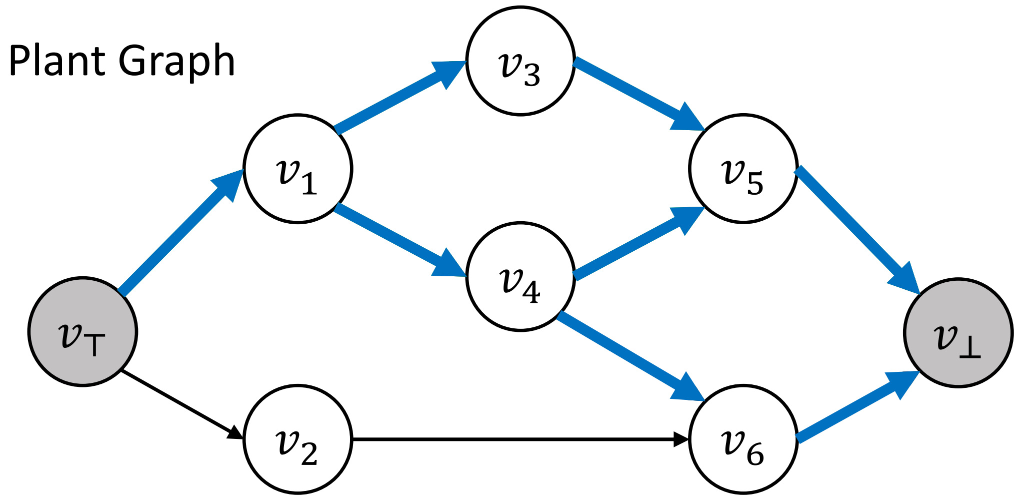
- Mapping from requirement to actual plant
- Given $G_P = \langle V_P, E_P \rangle$ and $G_R = \langle V_R, E_R \rangle$, a plan is a forward simulation from V_R to V_P, or a relation $C \subseteq V_R \times V_P$

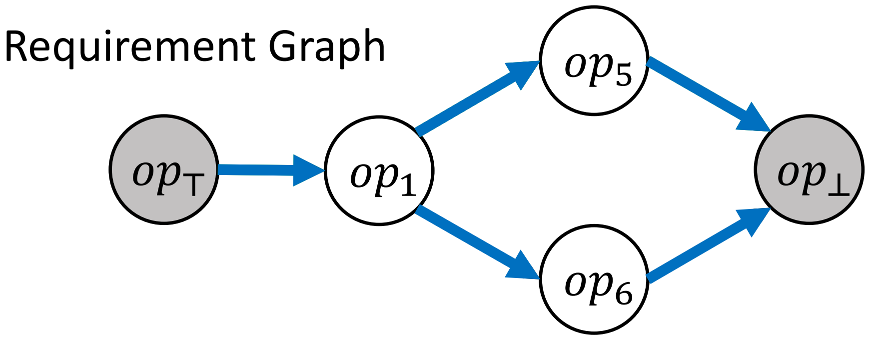
Feasible Paths

- Finding corresponding paths in requirement/plant
- Given a plan C and path $\pi \in G_R$, a feasible path is the corresponding path in G_P

Feasible Graph

- All possible paths in the plant for a specific requirement
- Set of all paths corresponding to R in G_P, denoted by $F_{R,P}$





Properties

Properties that can be captured by formal model

- Mutual exclusion: no two active parts in the plant collide with each other
  - For any ordinary cell $v \in V_P \setminus \{v_\top, v_\bot\}$, and any two distinct parts $w_1, w_2 \in W$, if $w_1, w_2 \in bag(v)$, then $pos(w_1) \neq pos(w_2)$
- Correctness: all requirements completed for any part that exits a plant
  - For any sink $v_\bot$, and any widget $w \in bag(v_\bot)$, the widget is completed, i.e. $completed(w)$ is a path in $requirement(w)$
- Bounded time: all requirements for parts entering plant is completed within a finite amount of time
  - For any acyclic requirement and any widget $w \in V_0$, there exists a $k > 0$ such that $widget\_time(w) \geq k$ then $w \in bag(v_\bot)$

SDCWorks Simulator

Flexible, open-source discrete event simulator

- Capable of simulating arbitrary SDCWorks models
- Python3 and Graphviz and Matplot libraries

Inputs

- Plant and requirement files in YAML format

Every time step

- Simulator moves parts around in the plant
- Updates the state of each cell
- Logs various **metrics**: **throughput**, **end-to-end delay** and **number of live parts** in the system

Output

- Log file → state of every cell and number of live parts in the system

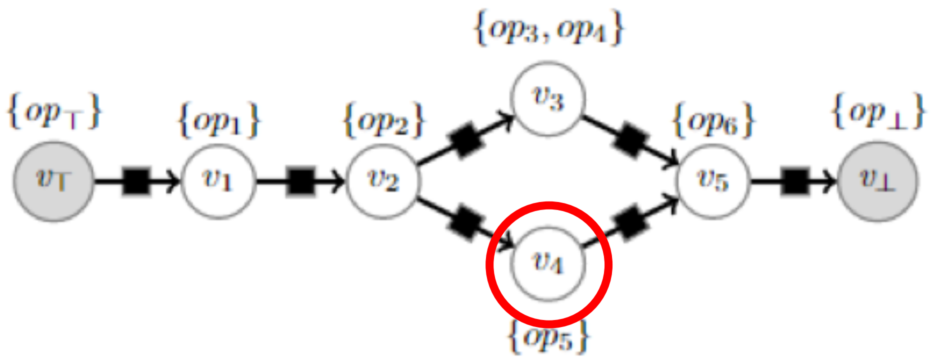
Case Study 1: Synthetic Linear Model

Description

- Linear manufacturing system with single branch
  - 3 requirements
  - 14 cells total (7 conveyors, 7 machines)
  - 6 operations

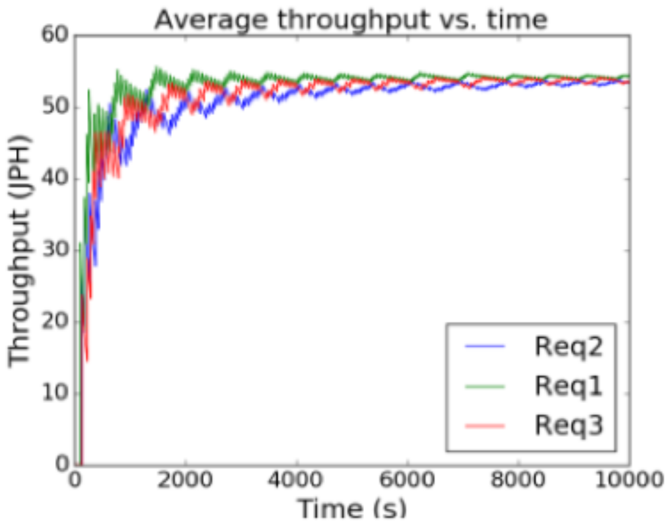
	Operations
R_1	$op_1 \rightarrow op_2 \rightarrow op_3 \rightarrow op_6$
R_2	$op_2 \rightarrow op_5$
R_3	$op_1 \rightarrow (op_4 \ or \ op_5) \rightarrow op_6$

- Bottleneck at v_4



Operation Table					
	v_1	v_2	v_3	v_4	v_5
op_1	10	-	-	-	-
op_2	-	20	-	-	-
op_3	-	-	40	-	-
op_4	-	-	35	-	-
op_5	-	-	-	50	-
op_6	-	-	-	-	15

Case Study 1: Simulation Results



Throughput

converges at 55.6 JPH

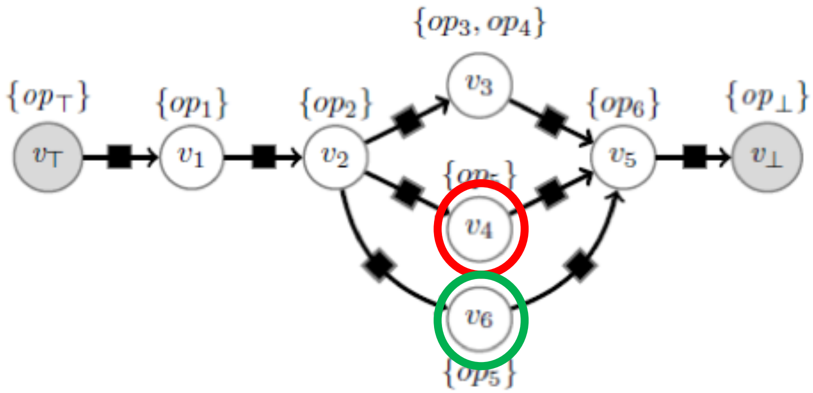
Case Study 1.5: Linear Model Modified

Description

- Linear manufacturing system with single branch
  - 3 requirements
  - 17 cells (9 conveyors, 8 machines)
  - 6 operations

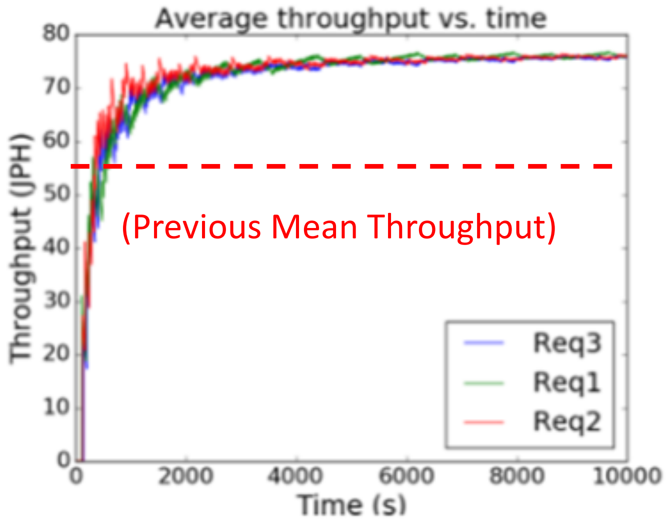
	Operations
R_1	$op_1 \rightarrow op_2 \rightarrow op_3 \rightarrow op_6$
R_2	$op_2 \rightarrow op_5$
R_3	$op_1 \rightarrow (op_4 \text{ or } op_5) \rightarrow op_6$

- Bottleneck at v_4
- **Addition of v_6 to relieve bottleneck at v_4**



Operation Table						
	v_1	v_2	v_3	v_4	v_5	v_6
op_1	10	-	-	-	-	-
op_2	-	20	-	-	-	-
op_3	-	-	40	-	-	-
op_4	-	-	35	-	-	-
op_5	-	-	-	50	-	50
op_6	-	-	-	-	15	-

Case Study 1.5: Simulation Metrics



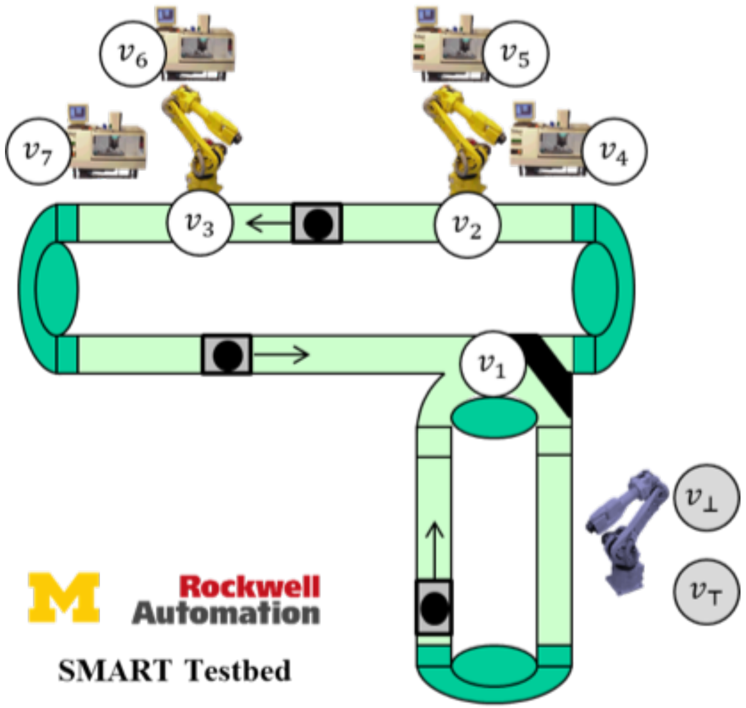
Throughput

converges at **75 JPH**

[compare: 55.6]

Case study 2: SMART [U. of Mich. Testbed]

PHYSICAL PLANT LAYOUT



Hybrid serial-parallel line manufacturing testbed

- University of Michigan at Ann-Arbor
- 3 cells and 2 conveyor lines connected with a controllable pneumatic diverter
- Three industrial robots
- Four CNC milling machines

Case Study II: SMART Testbed Models

CNC Machine Timings

	CNC 1 (v_4)	CNC 2 (v_5)	CNC 3 (v_6)	CNC 4 (v_7)
op_1	20	25	35	–
op_2	50	30	–	10
op_3	25	–	15	30
op_4	–	10	25	30

* all values are scaled to transition time ticks

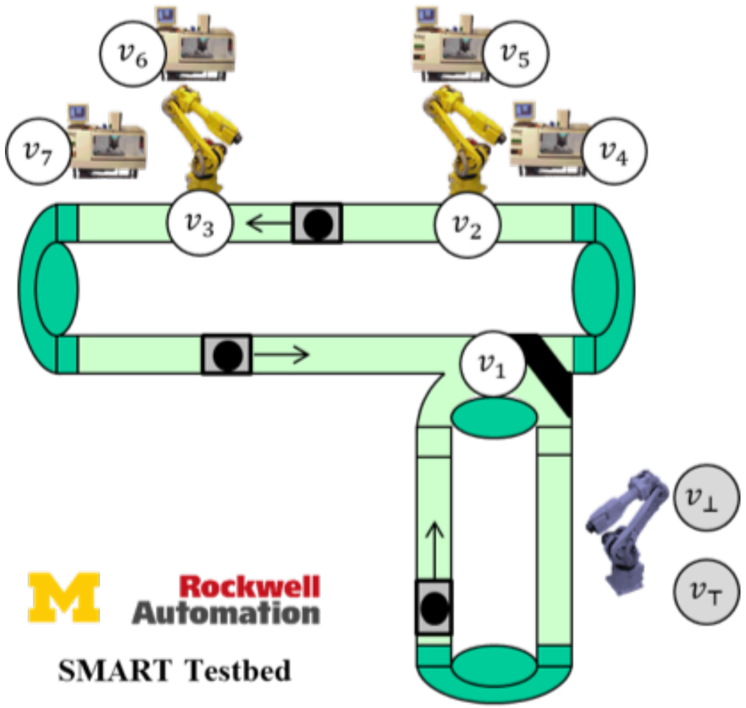
* "–" indicates that the operation is not supported on this machine

Operational Requirements

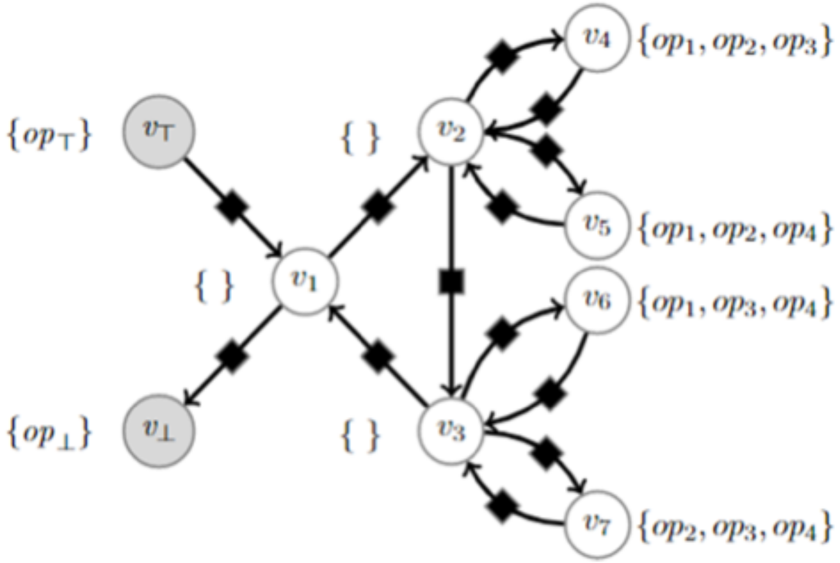
	Operations
R_1	$op_\top \to op_1 \to op_2 \to op_3 \to op_4 \to op_\bot$
R_2	$op_\top \to op_1 \to op_3 \to op_1 \to op_\bot$
R_3	$op_\top \to op_2 \to (op_3\ ,\ op_4) \to op_1 \to op_\bot$

Case study 2: SMART [U. of Mich. Testbed]

PHYSICAL PLANT LAYOUT



PLANT GRAPH



Conclusion

SDCWorks

- Formal framework for modeling and analyzing discrete manufacturing systems
- Implementation of framework provides simulation capabilities
- Open-source implementation at **https://github.com/SDC-UIUC/SDCWorks**

Future work

- Tests with larger realistic manufacturing system configurations
- Bridge simulator with real manufacturing systems for real-time monitoring and predictive purposes
- Integrate monitoring and synthesis capabilities into implementation

Thanks!

Questions?

sibin@illinois.edu