Services in smart manufacturing: comparing automated reasoning techniques for composition and orchestration

Flavia Monti, Luciana Silo, Francesco Leotta, Massimo Mecella

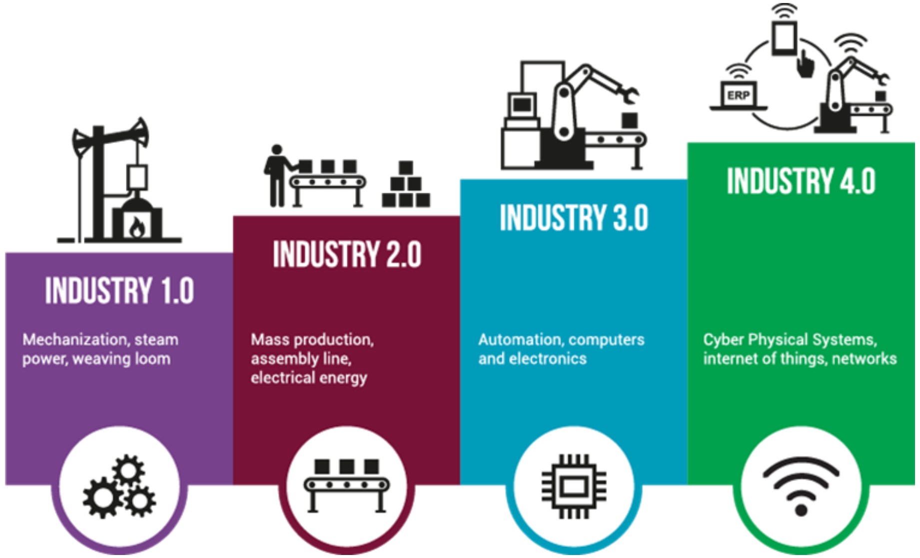
{monti, silo, leotta, mecella}@diag.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE ANTONIO RUBERTI

Smart Manufacturing

- Evolution of the traditional manufacturing practice
- Digitization of factories
  - E.g., introduction of IoT technologies and Cyber Physical Systems
- Intelligent and agile systems
  - Increase productivity and quality
  - Able to adapt to changes
  - React to breaks
  - Ease workers lives



Problem statement

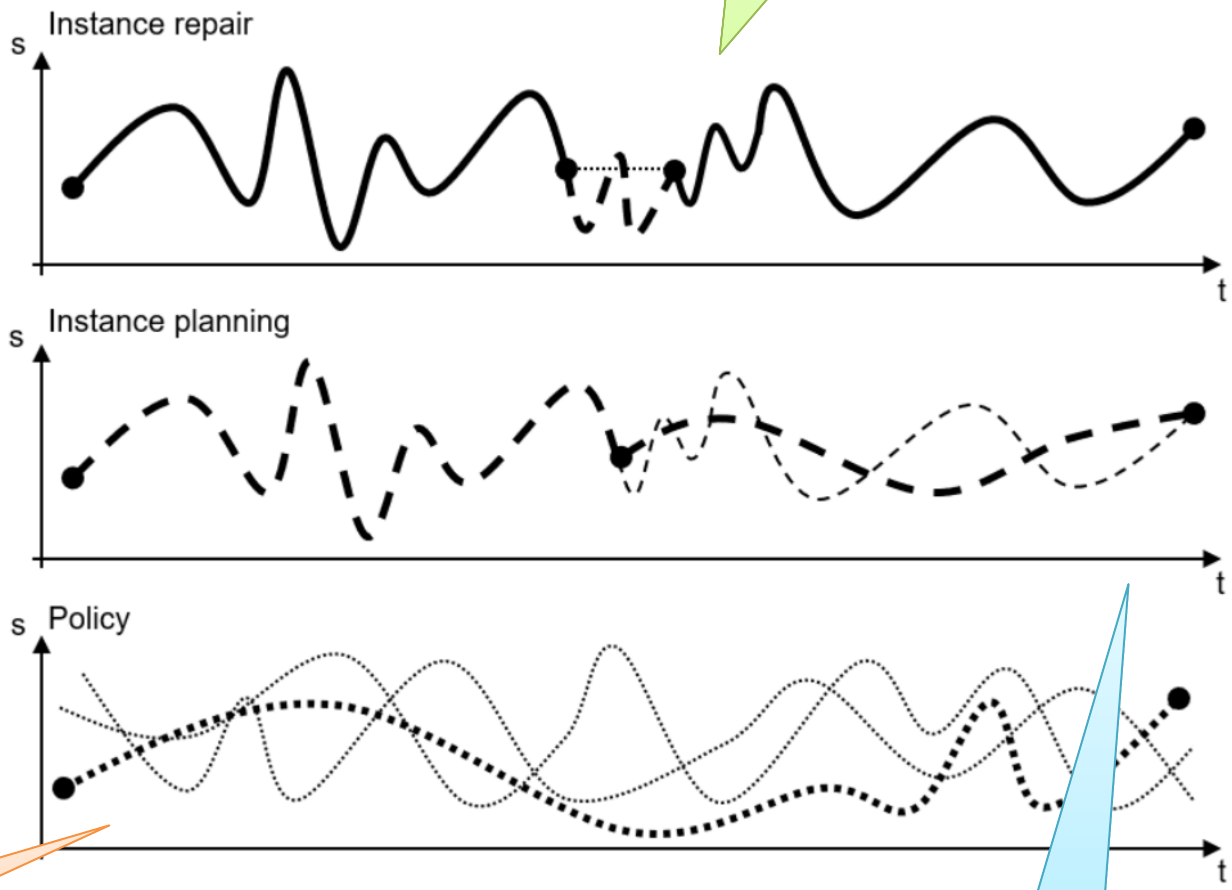
- In Smart Manufacturing, actors along the supply chain coordinate towards a unique manufacturing goal
- **Problem:** actors' conditions may vary during execution leading to
  – scarce quality of the multi manufacturing results, or
  – disruption of the supply chain
- **Solution:** represent actors as services monitored through Industrial APIs

  – actors include machines, humans, entire organizations
  – use *automated reasoning* (AR) techniques to provide adaptivity and resilience to multi-party processes in smart manufacturing

Adaptive strategies

Andrea Marrella, Massimo Mecella, Sebastian Sardiña: Supporting adaptiveness of cyber-physical processes through action-based formalisms. AI Commun. 2018
Andrea Marrella, Massimo Mecella, Sebastian Sardiña: Intelligent Process Adaptation in the SmartPM System. ACM Trans. Intell. Syst. Technol. 2017

- **Instance repair**
  - Process is well defined and upon exceptions the state of resources is taken back to the expected one employing AR techniques
- **Instance planning**
  - AR techniques are employed every time a new process instance is needed and upon disruption
- **Policy-based**
  - AR techniques are employed every time a new process instance (policy) is needed. All the possible states are computed.

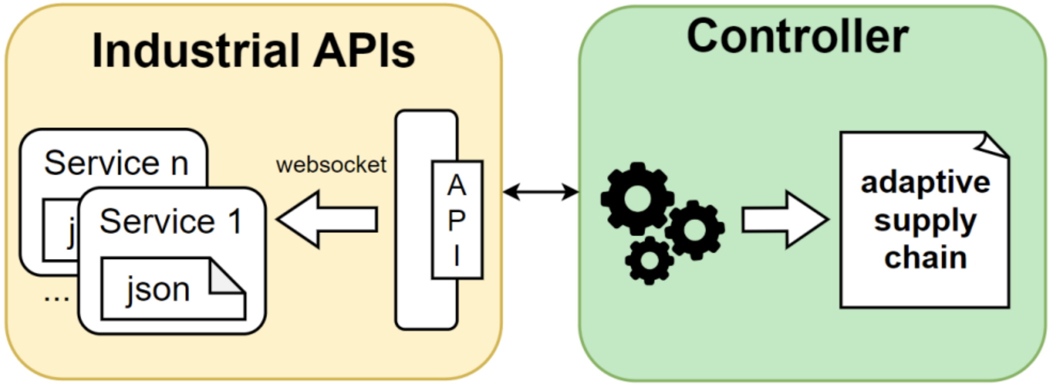


Giuseppe De Giacomo, Marco Favorito, Francesco Leotta, Massimo Mecella, Luciana Silo: Digital twin composition in smart manufacturing via Markov decision processes. Comput. Ind. 2023

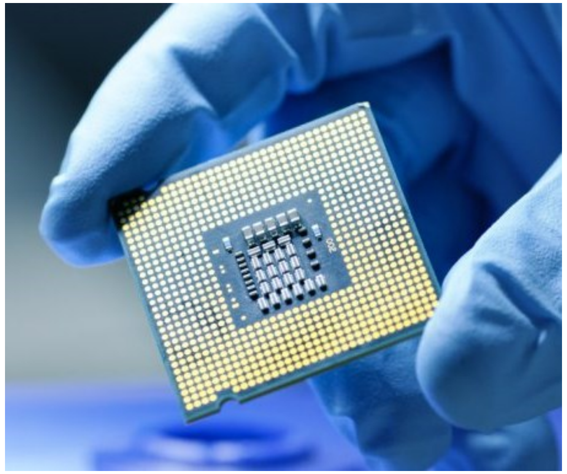
Agnes Koschmider, Francesco Leotta, Jerin George Mathew, Massimo Mecella, Flavia Monti: On the Application of Process Management and Process Mining to Industry 4.0. Submitted to SOSYM, special issue on BPMDS 2022

A service-based adaptive framework

- **Industrial APIs** to represent actors like services
  - Access the selected actors, monitor the behavior and status information, invoke commands, access RUL, costs and probability of failures
    - strongly relying on actor's Digital Twins (DT)
  - Platform: websocket and HTTP servers, services JSON descriptions
- **Controller** employing automated reasoning techniques
  - Black box taking as input specification of actors (Industrial APIs) and final manufacturing target
  - Provide as output an adaptive process



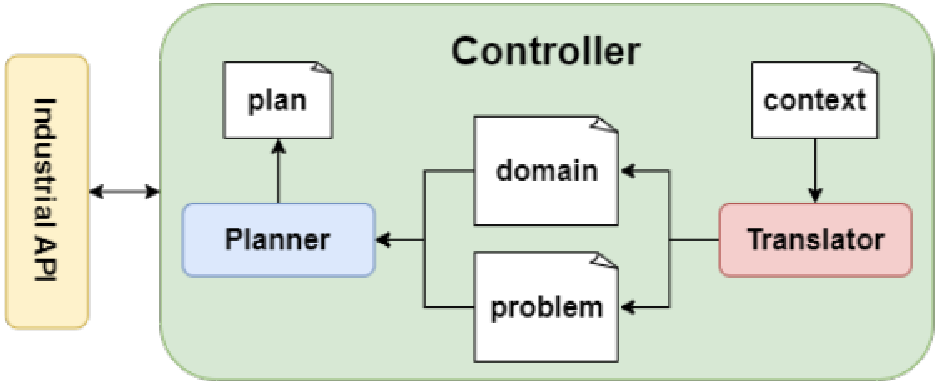
Case study: *ChipChain*



- Integrated chip manufacturing supply chain
  - Raw materials and design assortment phase
  - Manufacturing process
- Multiple alternatives (i.e., services) for the same actor
  - Different locations (i.e., countries) and costs upon execution of an operation
  - Actors' performance change over time and may become unavailable at any time

1° approach: planning-based

- Classical planning
  - Find a sequence of actions (plan) that transforms an initial state into a goal state
- A translator generate the *domain* and *problem* file given:
  - The list of services (modeled in a PDDL-like fashion)
  - The description of a production goal
- Use *FastDownward* planner to generate the plan
- Services are continuously monitor through Industrial APIs

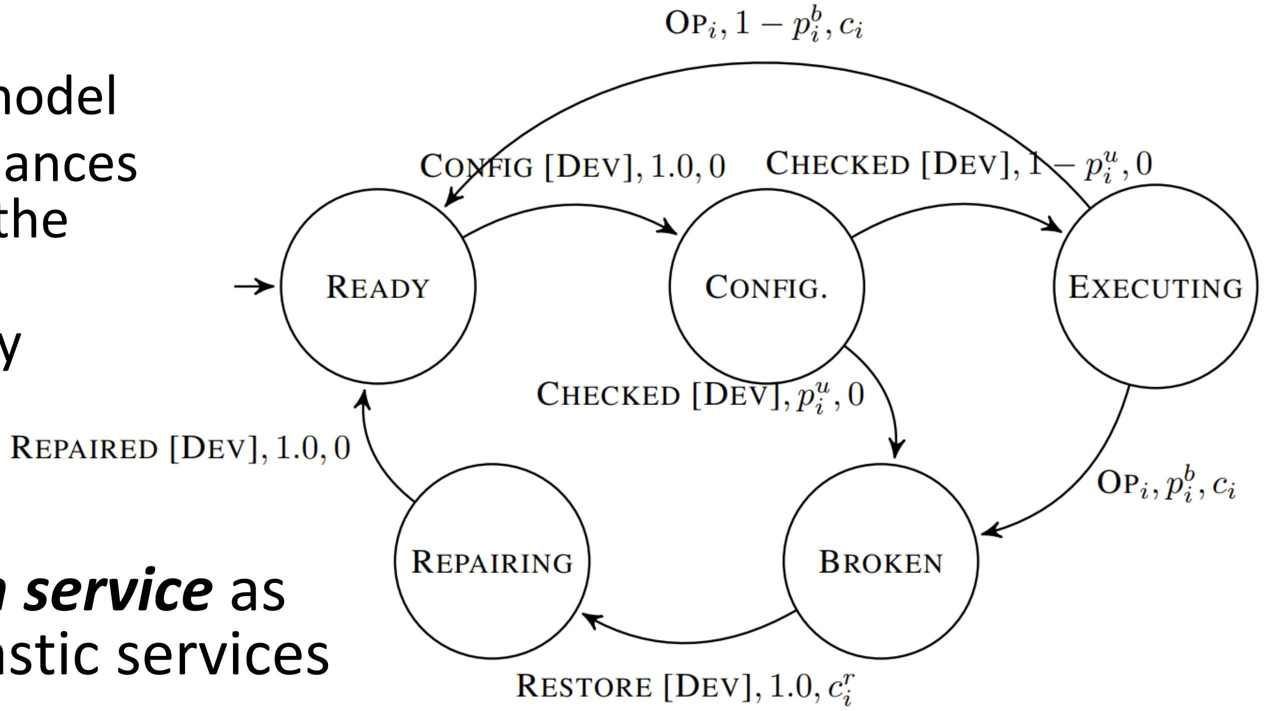


2° approach: stochastic-based with MDP and automata

- Each actor is modeled as a ***stochastic service*** via MDP

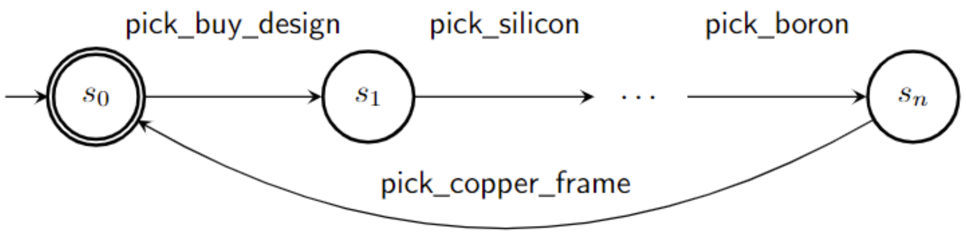
$$\tilde{S} = \langle \Sigma_s, \sigma_{s0}, F_s, A, P_S, R_s \rangle$$

  - **a fully observable, probabilistic** state model
  - the transition probability models the chances that applying an action in a state takes the service in another state (e.g., broken)
  - the reward models the predicted quality of the result obtained by performing an action
- Given a community of stochastic services we define a ***stochastic system service*** as an MDP which combines all the stochastic services in the community



2° approach: stochastic-based with MDP and automata (cont.)

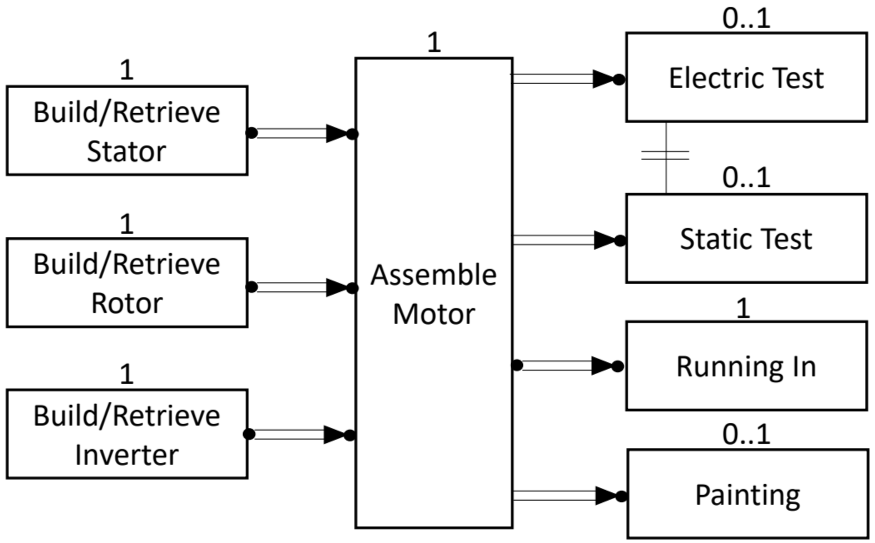
- The ***target process*** (manufacturing target) is modeled as a particular MDP
  $$T = \langle \Sigma_t, \sigma_{t0}, F_t, A, \delta_t, P_t, R_t \rangle$$



- ***Composition MDP*** combines the stochastic system service and the target service
  - A solution is a function *policy* that assigns a service to each action according to the state
  - The *value of a policy* is the expected sum of rewards when selecting actions based on the policy
  - The *optimal policy* maximizes the value of a policy
    - It represents a production chain

3° approach: stochastic-based with MDP and DECLARE

- Services modeled as *stochastic services* composing the *stochastic system service* as in the second approach
- The *target process* is modeled using **DECLARE**
  - language and framework for the declarative, constraint-based modeling of processes and services
  - transformed in a **LTL_f formula** from which we can generate **the target DFA**
- The production plan is generated by finding the optimal policy of the ***composition MDP***



Preliminary results

- Preliminary experiments based on *automated planning* and *stochastic* approaches
- Automated planning – classical planning
  - Take into account neither non-determinism nor probabilities in the execution of the operation
  - Resulting in potentially non-optimal or risky plans
  - Slowly growing execution time and memory consumption when increasing the number of services employed thanks to efficient heuristics
- Stochastic approaches – Markov Decision Processes (MDPs), automata and DECLARE formulas
  - Take into account the stochastic behaviors of the actors
  - Solutions are functions (policies) mapping states into actions and minimizing expected cost to goal
  - Memory consumption and execution time increase exponentially with the number of services employed

Preliminary results

- Experiments performed on a server with 1TB of RAM and 2 AMD processors EPYC Rome 7742 (64 cores each)
- Manufacturing process phase:
  - Small case
    - A total of 14 services (an extra copy for 2 services)
  - Manageable case #1
    - A total of 17 services (an additional copy for 5 services)
  - Manageable case #2
    - A total of 22 (an additional copy for 10 services)
  - Complex case
    - A total of 60 services (5 copies of each services)

	Small case		Manageable case #1		Manageable case #2		Complex case	
	Time (s)	Memory (MiB)	Time (s)	Memory (MiB)	Time (s)	Memory (MiB)	Time (s)	Memory (MiB)
Instance planning	0.14	26.4	0.14	26.5	0.18	26.4	0.31	26.7
Stochastic policy	16.43	2 245.53	905.51	119 057.86	7 834.65	845 834.16	Failed	Failed
Stochastic constraint-based policy	85.86	134 721.56	4 067.72	293 288.93	Failed	Failed	Failed	Failed

Thanks for your attention
and see you tomorrow at the poster session! ☺

SAPIENZA
UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA,
AUTOMATICA E GESTIONALE ANTONIO RUBERTI