Synthesis of controllers for cyber-physical systems

Natasha Alechina

Utrecht University

Vardifest 2022

Joint work

- Giuseppe De Giacomo, Moshe Vardi, Paolo Felli, Brian Logan, Natasha Alechina, Synthesis of Orchestrations of Transducers for Manufacturing, AAAI 2018

- Natasha Alechina, Tomáš Brázdil, Giuseppe De Giacomo, Paolo Felli, Brian Logan, Moshe Vardi, Unbounded Orchestrations of Transducers for Manufacturing, AAAI 2019

- and on-going joint work on fail-safe specifications with Louise Dennis, Angelo Ferrando and Brian Logan

Problem setting

Given

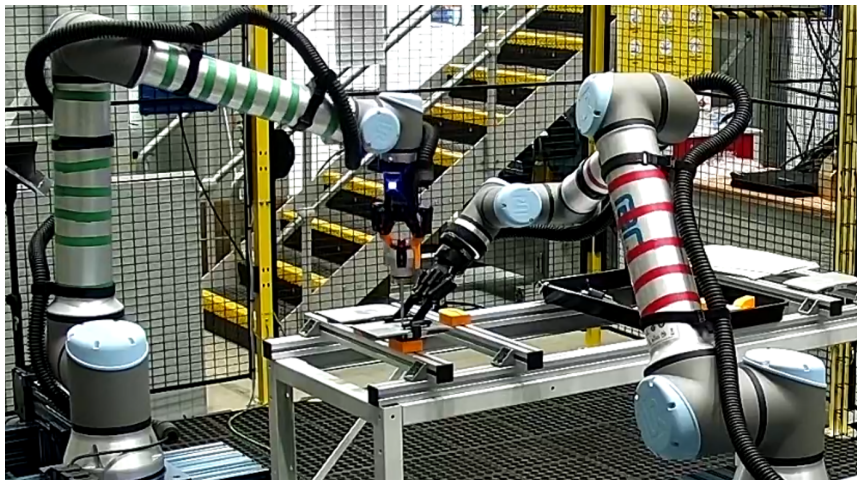
- a high level plan of what needs to be done: target behaviour
- a set of machines, robots etc. that can do be used to simulate the target behaviour: resources

is there

- a concrete plan assigning actions to resources, or orchestration of resources to simulate the target behaviour

Applications

autonomous manufacturing



autonomous inspection & maintenance

Resources

Resources are modelled as **multi-transducers**

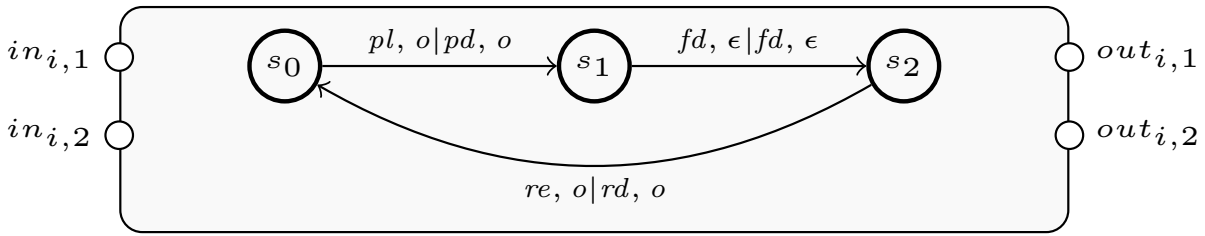
Definition (Deterministic Multi-Transducer)

A deterministic multi-transducer $T = (\Sigma, S, s_0, f, g, k, \ell)$ is a transition system with k **inputs** and l **outputs**, where:

- Σ is the **alphabet** (of both inputs and outputs)
- S is the non-empty finite **set of states**, and s_0 the **initial state**
- $f : S \times \Sigma^k \longrightarrow S$ is the **transition function**, which takes a state and k input symbols and returns the successor state
- $g : S \times \Sigma^k \longrightarrow \Delta^\ell$ is the **output function**, which returns ℓ output symbols.

Target

- The target is also modelled as a **multi-transducer**.
- An example target transducer that attaches a part



Port binding

- several resources may be required to realise an operation specified by T
- a **port binding** is a pair of the form $(out_{x',y'}, in_{x,y})$ which represents a connection between the output port y' of transducer x' and input port y of transducer x
- the set of port bindings must be consistent with a set of **binding constraints** $\mathcal{B}$, specified as boolean combinations of atoms of the form $(out_{x',y'}, in_{x,y})$, e.g., a physical output port can be bound only to one (physical) input port
- the set of all legal port binding sets is denoted by $Cntl$

Orchestration

- a **controller** C for T, $T_1, \ldots, T_m$ is a strategy $C : \Sigma^{k^+} \longrightarrow Cntl$, i.e., a mapping from a finite sequence of input tuples to port bindings
- the **orchestration problem** can then be stated as: 'is there a C for $T_1, \ldots, T_m$ such that C realises T'

Definition (Orchestration Problem)

Given a set of multi-transducers $T_1, \ldots, T_m$, a set of binding constraints $\mathcal{B}$ and a target T, the orchestration problem is the question whether there is a controller C for $T_1, \ldots, T_m$ that realises T.

Theorem (AAAI 2018)

The orchestration problem for a set of multi-transducers $\{T_1, \ldots, T_m\}$ and a target T is in 2EXPTIME.

Unbounded Orchestration

Given resource **types** $\{T_1, \ldots, T_m\}$ and a target T, **are there numbers** n_1**, . . . ,** n_m such that the orchestration problem has a solution for n_i copies of each transducer types, that is, for $\{T_1^1, \ldots, T_1^{n_1}, \ldots, T_m^1, \ldots, T_m^{n_m}\}$

Theorem (AAAI 2019)

The unbounded orchestration problem for a set of multi-transducers $\{T_1, \ldots, T_m\}$ *and a target* T *is undecidable.*

For unitransducers (one input and one output port) the unbounded orchestration problem is in 2EXPTIME.

New Problem: Ensuring Safety

- potentially dangerous events are not in the alphabet of the target but need to be responded to
- for example, the target of manufacturing a hinge or fixing a tile to a wall of the reactor does not specify what to do in the event of flooding
- resulting behaviours (like shutting down the plant) are not specified in the target
- we add an additional specification F for the given set of resources: how to respond to abnormal events (F is a deterministic multi-transducer with u ports)
- a **fail-safe controller** for (T, F) allocates inputs from two streams of events: when the 'abnormal' stream is empty, it acts as a controller above; when there are inputs on the 'abnormal' stream, switches to orchestrating response to the abnormal events (until receiving a signal to resume or restart)

Fail-Safe Orchestration

- a **fail-safe controller** C for (T, F), $T_1, \ldots, T_m$ is a strategy $C : (\Sigma^k \times \Sigma^{F^u})^+ \longrightarrow Cntl$, i.e., a mapping from a finite sequence of input tuples to port bindings

Definition (Fail-Safe Orchestration Problem)

Given a set of multi-transducers $T_1, \ldots, T_m$, a set of binding constraints $\mathcal{B}$ and a fail-safe target (T, F), the orchestration problem is the question whether there is a fail-safe controller C for $T_1, \ldots, T_m$ that realises (T, F).

Theorem

The orchestration problem for a set of non-deterministic multi-transducers $\{T_1, \ldots, T_m\}$ and a fail-safe target (T, F) is in 2EXPTIME.