Modelling an implementation

&

implementing a model

thanks to the organisers

Apologies to the audience

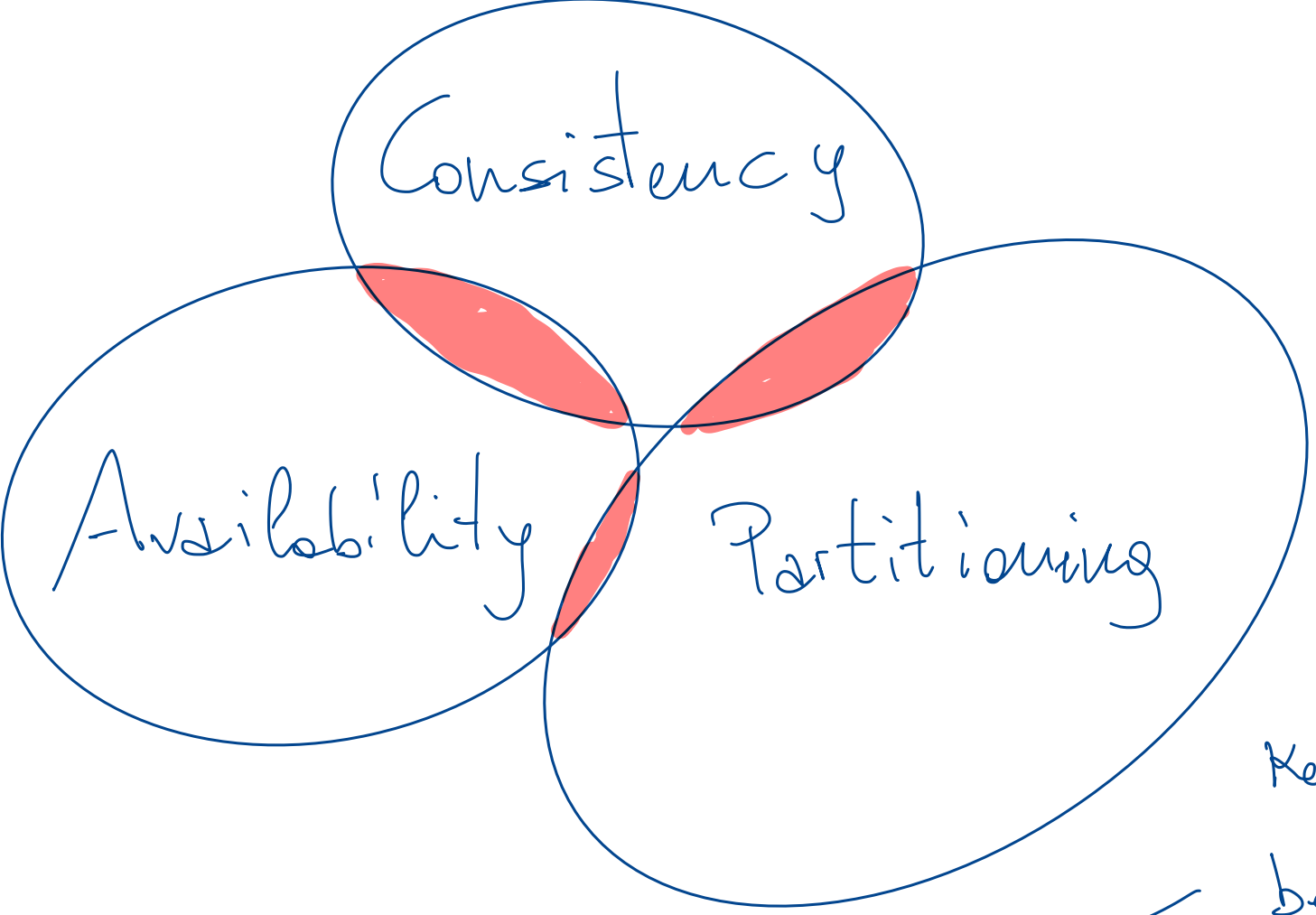
thank Michele Pinna

Modelling an implementation

Joint work with

Roland Kuhn @ Actyx

Hernán Melgratti @ UBA



- Systems don't stop
- Partitioning is intrinsic
- Availability is not negotiable €

⇓

Keep going and don't be afraid to make mistakes

LOCAL-FIRST

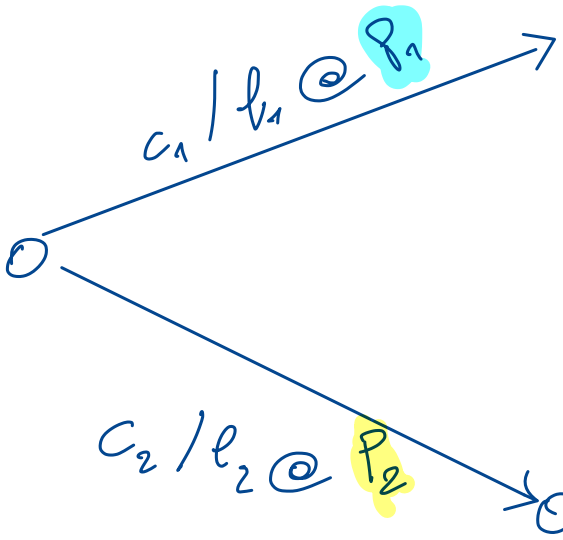
@ Actyx (*)

- Platform to develop applications for controlling / managing industrial production processes
- Command execution generates events
- Pub-Sub middleware (asynchronously) replicates logs of events by merging newly generated events
- Temporary partitioning
- Need for eventual consistency of replicated logs

(*) and other application domains too

Top down

- Global type (and well-formedness conditions) to handle protocols where



with $p_1 \neq p_2$ and arbitrary num. of inst.

- Local type are FSAs too

projection op. parametrised by subscriptions

Some comments / results ①

- Session infidelity
- Projections yield correct realisations
- Deadlock freedom by construction !
- DROPPING events is harmless in W.F. protocols

① Details in Kuhn, Melgratti, T. @ ECOOP 23)

Open Problems

- Generalise WF
- Optimise
- Support to "design" subscriptions

- Collaborative vs. Adversarial
- Failures
- Unreliable pub-sub
- impure / non-compensetable events

Implementing a model

Joint work in progress with

Philipp Heller @ ETH Hernán Melgratti @ UBA

Alceste Scalas @ DTU

Dagstuhl, September 2021

the join calculus is a powerful abstraction for concurrent programming:

banana(b) & apple(a) if a=b ▷ fruitSalad(b, a)

lettuce(l) & apple(a) & Lives(b) ▷ salad(l, a, b)

Q: Why hasn't the join calculus become popular?

A: It's implementations are not very efficient

Idea: reuse partial matches

Fix a signature Σ and a set of variable symbols X

μ range on $Term_{\Sigma,X}$

γ range on propositions on X

$Q \in Term_\Sigma^*$

pattern construction

some boolean exp with vars

queues of values

Pattern matching

$\pi = \mu_1 \& \dots \& \mu_n$ if γ ← conditional pattern

$Q \vDash_\sigma \pi$ when

- $Q \in Term_\Sigma^n$
- $\forall 1 \le i \le n: Q[i] = \pi[i]$
- $\gamma\sigma$

Chasing matches

Given $Q \in \mathrm{Term}_\Sigma^n$ and $\pi = \mu_1 \& \dots \& \mu_m$

candidate indexes

$$\mathrm{cidx}(Q, \mu) = \{ i \mid \exists j, \sigma : Q[i] = \mu[j]\sigma \}$$

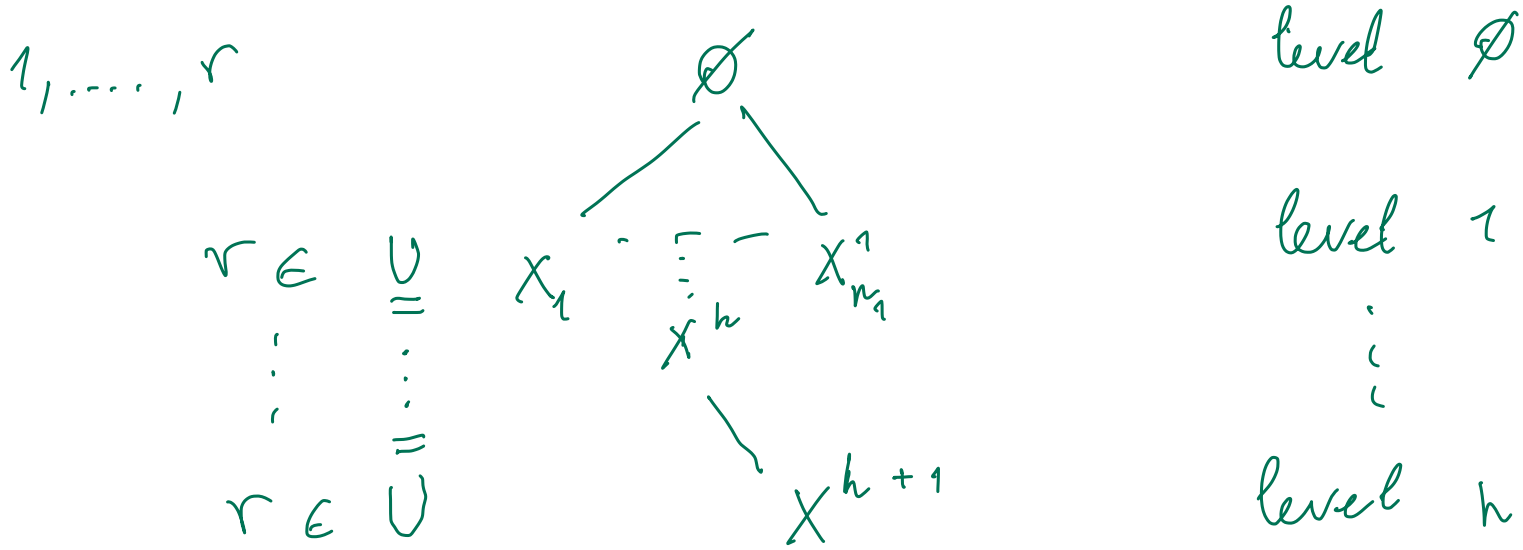
assignments

$$\mathcal{A} = \{ a \in \{1, \dots, m\}^I \mid I \subseteq \mathrm{cidx}(Q, \mu) \text{ and } a : i \mapsto j \Rightarrow \exists \sigma : Q[i] = \mu[a(i)]\sigma \}$$

$a \sqsubseteq b \iff \min(\mathrm{dom}\, a \setminus \mathrm{dom}\, b) \leq \min(\mathrm{dom}\, b \setminus \mathrm{dom}\, a)$ or $\mathrm{dom}\, a = \mathrm{dom}\, b$

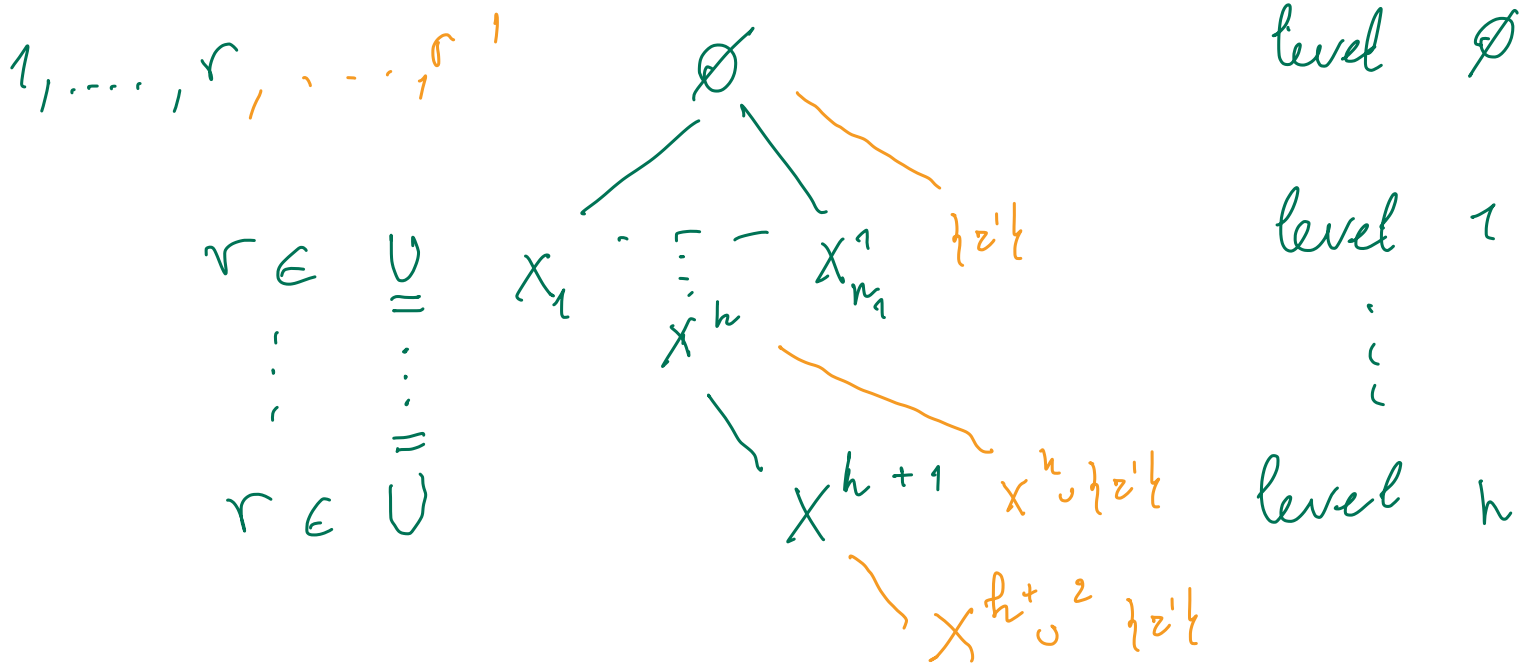
Find $a \in \mathcal{A}$ "minimal" s.t.

$\exists \sigma : \forall i \in \mathrm{dom}\, a : Q[i] = \mu_{a(i)}\sigma$ and $\gamma\sigma$ holds



X father Y $\Rightarrow$ $\min X \leq \min Y$

$\max X \neq \max Y$



X father Y $\Rightarrow$ $\min X \leq \min Y$

$\max X \neq \max Y$

Open Problems

- Does this work better?
  - Experiments
  - Benchmarks (eg. SAVINA)
- the algorithms is exponential in the worst case, but how realistic pathological cases are?
- Are there other interesting kinds of join patterns?
- Other notions of "best" match?