

On the Expressiveness of Event-Based Coordination Media

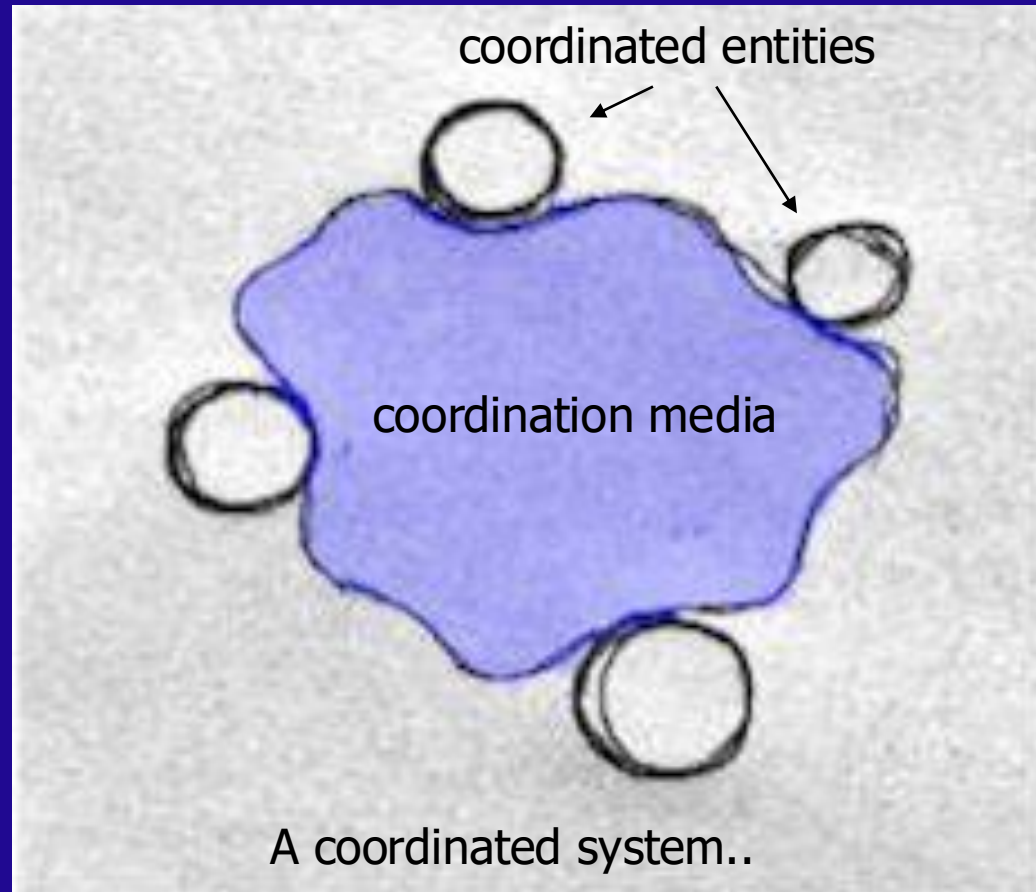
Mirko Viroli, Andrea Omicini, Alessandro Ricci

DEIS, Università degli Studi di Bologna, Cesena, Italy

{mviroli,aomicini,aricci}@deis.unibo.it



Coordination



Outline

- A Paradigm Shift in Coordination models
 - from closed systems, parallel applications
 - to distributed, open, event-based systems
 - coordination as a language / coordination as a service
 - an impact either to applications and to formal semantics
- The traditional notion of expressiveness of coordination models
- A new notion of expressiveness
 - expressiveness as a coordination medium interactive behaviour
 - full-expressiveness of a formalism for coordination media
 - the σ RST framework for defining coordination media

Coordination as a language

- The prototype coordination model is Linda
 - tuple-based blackboard,
 - allowing operations out, in, rd, inp, rdp
 - supports spatial and temporal decoupling
- Traditional application
 - as a language for coordinating activities in closed systems
 - the tuple space
 - is implicitly created at compile-time
 - is part of the application
 - is merged with activities
 - has no counterpart either as
 - design-time
 - run-time abstraction

Paradigm shift

- Today complex systems
 - feature multi-application setting in the same host environment
 - ⇒ calls for infrastructures addressing common needs
 - ⇒ providing effective and reliable services
 - require abstractions to be formally described
 - ⇒ supporting the engineering process down to deployment
- Coordination is a critical and complex activity
 - coordination medium abstraction as first-class entity
 - represented as an interactive component / abstraction
 - supporting engineering from design down to deployment
- At run-time
 - infrastructure providing coordination as an event-based service

Coordination as a service

- Coordinated system
 - coordinated entities: entities subject to coordination
 - coordination media: abstraction providing coordination
 - interaction space: supporting event production and consumption
- Coordination schema
 - coordinated entities post a request in the interaction space
 - a coordination medium
 - consumes the request
 - produces replies
 - replies are then consumed by some coordinated entities
- Coordination medium as interactive abstraction
 - e.g. represented by transition system $\langle M, \rightarrow, \{ie, oe_1, \dots, oe_n\} \rangle$

The change of perspective

- Has impact to applications
 - new kinds of coordination infrastructures (JavaSpaces, Lime, ...)
 - featuring new coordination mechanisms
 - event-notification, timeouts, transactions, mobility, ...
- Has impact to theory
 - which formal characterisation?
 - which notion of expressiveness?
- From:
 - [Busi et.al], [Brogi&Jacquet,Coord99], ...
- To:
 - [Gelernter&Zuck,Coord97], [Omicini, SAC99], [Viroli&Omicini,Coord02], [Rowstron, SAC00]

Traditional notion of expressiveness

- Coordination language characterised by:
 - the Turing-like transformations it can encode
- Relative notion:
 - more expressive $\sim$ able to represent more computations
- Absolute notion:
 - Turing-equivalence of a coordination language
 - result: Linda(out,in,rd,inp,rdp) is Turing-equivalent
- Does it make sense in the context of event-based systems?
 - May a Linda tuple space realise all kinds of coordination activity?
- The notion of expressiveness has to be reconsidered!!!

The Coordination Medium

- In the context of event-based, interactive systems
 - coordination cannot be described by Turing-like transformations
 - need for a framework based on Interactive Computing
- CM as a Sequential Interactive Machine [Wegner, CACM 1997]
 - gets an input: a request $i = \langle id, req \rangle$,
 - produces an output: one or more replies $o = (\langle id_n, rep_n \rangle)_n$
 - characterised by its admissible interaction histories $\langle i_n, o_n \rangle$
- An equivalent viewpoint in the process algebra field
 - a medium as a process described by transition system $\langle M, \rightarrow, \{ie, oe_1, \dots, oe_n\} \rangle$
 - with trace semantics as observation criteria

Coordination Medium Expressiveness

- Given two coordination media C_m and C_m'
 - let $SIM = \langle M, \rightarrow, \{ie, oe_1, \dots, oe_n\} \rangle$ be the description of C_m
 - let $SIM' = \langle M', \rightarrow', \{ie, oe_1, \dots, oe_n\} \rangle$ be the description of C_m'
- Naturally: C_m more expressive than C_m' if
 - SIM admits more interaction histories than SIM'
- Meaning:
 - C_m provides more services
 - the services provided by C_m are more general / less specific
- From the process algebra viewpoint
 - C_m' is an “implementation” of C_m , a *horizontal* refinement

Full-expressiveness

- Defined considering a formalism for describing media behaviour
 - $CM_\sigma = \langle M, \rightarrow, \{ie, oe_1, \dots, oe_n\} \rangle_\sigma$, parametric in specification σ
- The formalism CM_σ is full-expressive if
 - for any SIM m , it exists a spec s s.t. medium CM_s admits the same histories of m
- Equivalently:
 - for any computable function $f: Ie^* \rightarrow Oe^*$
 - it exists a spec s , so that after applying ie^* to CM_s , it replies $f(ie^*)$
- Meaning: any event-based behaviour can be described!
- Can be a foundation for programmable coordination media
 - examples: Tuple Centres, MARS, Lime, SpaceTub
 - they provide flexible coordination infrastructures
 - $\Rightarrow$ whose behaviour can be targeted to special needs

The σ RST formalism

- We provide a full-expressive, minimal formalism
 - inspired by ReSpecT language for programming tuple centres
 - borrowing the style from logic languages (e.g. Prolog)
 - a program for the medium as a set of reaction specifications
 - are fired by input events
 - executes a sequence of goals
 - may put, drop, read a tuple from an internal dataspace
 - may fire a new reaction
 - may send an output event
- It is proved to be fully expressive
 - each SIM machine can be simulated
- A primitive for explicit production of output events is necessary
 - This is not supported by ReSpecT
 - $\Rightarrow$ “*out_tc*” is meant to fill this lack

Examples

- A Linda tuple space:
 - showing basic “data-oriented” features
- Notification:
 - registering the interest of an observer
 - a publisher emits notification of a tuple
 - the medium notifies interested observers
- Tuple-expiration:
 - Time passing modelled as inputs reception (tic)
 - Each tuple carries a “lease” time, after which it expires

Modelling Linda

```
 $\langle \text{input}(\text{Id}, \text{linda}(\text{X}, \text{T})), \quad \Diamond \text{ev}(\text{X}, \text{T}, \text{Id}) \rangle$   
 $\langle \text{ev}(\text{in}, \text{T}, \text{Id}), \quad \uparrow \text{tup}(\text{T}); \text{T} \Rightarrow \text{Id} \rangle$   
 $\langle \text{ev}(\text{in}, \text{T}, \text{Id}), \quad \neg \text{tup}(\text{T}); \downarrow \text{qry}(\text{in}, \text{T}, \text{Id}) \rangle$   
 $\langle \text{ev}(\text{inp}, \text{T}, \text{Id}), \quad \uparrow \text{tup}(\text{T}); \text{T} \Rightarrow \text{Id} \rangle$   
 $\langle \text{ev}(\text{inp}, \text{T}, \text{Id}), \quad \neg \text{tup}(\text{T}); \text{null} \Rightarrow \text{Id} \rangle$   
 $\langle \text{ev}(\text{rd}, \text{T}, \text{Id}), \quad \updownarrow \text{tup}(\text{T}); \text{T} \Rightarrow \text{Id} \rangle$   
 $\langle \text{ev}(\text{rd}, \text{T}, \text{Id}), \quad \neg \text{tup}(\text{T}); \downarrow \text{qry}(\text{rd}, \text{T}, \text{Id}) \rangle$   
 $\langle \text{ev}(\text{rdp}, \text{T}, \text{Id}), \quad \updownarrow \text{tup}(\text{T}); \text{T} \Rightarrow \text{Id} \rangle$   
 $\langle \text{ev}(\text{rdp}, \text{T}, \text{Id}), \quad \neg \text{tup}(\text{T}); \text{null} \Rightarrow \text{Id} \rangle$   
 $\langle \text{ev}(\text{out}, \text{T}, \text{Id}), \quad \downarrow \text{tup}(\text{T}); \Diamond \text{serve}(\text{T}) \rangle$   
 $\langle \text{serve}(\text{T}), \quad \uparrow \text{qry}(\text{rd}, \text{T}, \text{Id}); \Diamond \text{ev}(\text{rd}, \text{T}, \text{Id}); \Diamond \text{serve}(\text{T}) \rangle$   
 $\langle \text{serve}(\text{T}), \quad \neg \text{qry}(\text{rd}, \text{T}, \text{Id}); \uparrow \text{qry}(\text{in}, \text{T}, \text{Id}); \Diamond \text{ev}(\text{in}, \text{T}, \text{Id}) \rangle$ 
```

Conclusions

- "*Coordination as a service*" as new foundational framework
- Promotes an engineering approach for coordination media
- In the context of formal semantics:
 - has impact on various notions, including expressiveness
- Future works
 - comparing to existing notion of expressiveness
 - evaluating σ RST as a unifying description tool
 - comparing to ReSpecT