The internal–local–remote dependency model for distributed collaborative sessions

Jose Martin Molina Espinosa (*), Khalil Drira (**), Jean Fanchon (**)

(*) ITESM–CCM, Mexico
(**) LAAS–CNRS, France

Generic coordination in distributed collaborative sessions

The internal–local–remote dependency model for
Centralised control -> Distributed control

Management of Collaborative Sessions

Coordinate Distributed Communicating Entities

Open Close Data Access

Edward & all, Texier & all: control of data access by participants and activities
Rodriguez & all: relations between participants w.r.t. data, application access
Tata: control of data access by participants and activities

Modeling, specification, verification, implementation, control

General properties
Activity flow

Open Close --- Data Access

Generalised control --- Distributed control

Management of Collaborative Sessions

Coordinate Distributed Communicating Entities
Decidability.

A satisfaction relation between a session model and a set of constraints.

Rules: constraints on the order in which events occur.

Sessions: sets of events (occurrences of actions: open, write, ...)

Executed by actors (participants, tools, ...).

Modeling sessions and specifying their properties.

Examples:

Specification of rules or constraints on collaborative sessions.
The model of the system

A satisfiability relation between system and property

properties and model depend on each other

causality, concurrency

reachability, liveness, safety

branching properties

relation between events / actions, states / configurations

branching structure on sequential behaviours: trees, graphs, automata

branching structure on concurrent behaviours: labelled partial orders

set of sequential behaviours

set of concurrent behaviours: labelled partial orders

The property to express
Session, behaviour set of events, occurrences of actions executed by entities

Verifiability, implementatio by finite automata

Polynomial decision

Satisfiability

Can be extended to MSO logic and modal logics

Precedence, inhibition, concurrency, enabling...

Expressivity

First order formula on partial order and labelling.

Property of a labelled partial order

Rule, property of the behaviour

a labelled partial order

causality relation between events: a partial order

set of events, occurrences of actions executed by entities

Session, behaviour
System Components

**P**: set of actors or agents. Human participants or software components.

**C**: set of categories. Human participants or applications (Graphic tool, floor control, etc).

**S**: set of sites. Concrete sites (machines, LAN) or abstract sites.
The partial ordering of the events in a behaviour.
\( \forall e \in E \) is an occurrence of the event name \( \chi_e \)

\( \lambda : \mathcal{E} \times P \leftarrow \mathcal{E} \times \mathcal{E} \) is a partial order relation (causal, enabling)

\( \mathcal{E} \) a set of events

\( (\mathcal{E}, \lambda, \mathcal{P}) = \mathcal{S} \)

A session on \( \mathcal{P} \) and \( \mathcal{A} \):

A set of agents, \( \mathcal{A} \) a set of actions

Collaborative sessions: Labelled partial orders
\[ (h = z \iff h \supset z > x) \land A \lor h > x \equiv h \leftarrow x \] 

Immediate precedence:

\[ \ldots I, x, x) \varphi = | (u \in \ldots I, x, x) \varphi \land (x) \varphi \text{ for some } s \in B \] 

\[ \rho \supset e \iff h \supset x = | (e, x) \varphi \land (x) \varphi \text{ for } s \in B \] 

\[ (a, u) = (e) \varphi \land (x) \varphi \text{ for } s \] 

\[ \varphi = | s \text{ then } (s \varphi \supset s \varphi) = s \] 

let

\[ \varphi \cdot x \in | \varphi \land \varphi | \varphi \leftarrow | h \supset x | (x) \varphi \text{ for } s \] 

Properties: First order logic on LPO
\[
((h)_p \triangleright x \cdot hA) \iff (x)^p \cdot xA \equiv (\ell \cdot \alpha)_{p \cdot \ell}
\]
\[ (x > y \land y > x) \iff (\forall_{\mathcal{P}} (y) \lor (\forall_{\mathcal{P}} x)) \equiv (\forall_{\mathcal{P}} (\alpha, \nu) \exists_{\mathcal{P}} \bar{x}) \]

**Non-concurrency**

\[ (((\forall_{\mathcal{P}} (y) \lor y \leftarrow x), \forall_{\mathcal{P}} y) \iff (\forall_{\mathcal{P}} (y) \lor (\forall_{\mathcal{P}} x)) \equiv (\forall_{\mathcal{P}} (\alpha, \nu) \exists_{\mathcal{P}} \bar{x}) \]

**Immediate enabling)**

\[ \exists \bar{x} \in \forall_{\mathcal{P}} (\alpha, \nu) \exists_{\mathcal{P}} \bar{x} \]

**Useful predicates**
Internal constraints on the behaviours of single actors (defined per-category) (site symmetry)

Local constraints between the actors on a single site

Remote constraints between the actors of a category
Structuring a specification

Remote or categorical constraints

Site symmetry or site typing:

\[ \text{remote: } (n,a) (m,b) \Rightarrow \text{Cat}(n) = \text{Cat}(m) \]

Relate the actions of actors belonging to the same category

Local constraints

Can be defined for all actors of a category:

\[ \text{local: } (n,a) (m,b) \Rightarrow \text{Site}(n) = \text{Site}(m) \]

Relate the actions of actors located on the same site

Internal constraints

Relate the actions of a single actor

Specific to a category

\[ \phi \text{ is remote: } (n,a) (m,b) \Rightarrow (\phi)(n,a) \geq (\phi)(m,b) \]

Relate the actions of actors of a category

\[ \phi \text{ is local: } (n,a) (m,b) \Rightarrow (\phi)(n,a) \leq (\phi)(m,b) \]

Relate the actions of actors belonging to the same category
The partial ordering of the events in a behaviour.
\[ (P_{\text{pred}}(t, \text{close}), t, \text{delete}) \lor (P_{\text{pred}}(t, \text{open}), t, \text{close}) \lor (P_{\text{pred}}(t, \text{announce}), t, \text{open}) \lor (P_{\text{pred}}(t, \text{create}), t, \text{announce}) \]
\[
\begin{align*}
\text{Communnication (PM and IM categories)}\\
\text{Group membership (SM and PM categories)}\\
\text{Local predicates}
\end{align*}
\]
\[ \text{No two conjecture grant or accept} \]

\[ \forall q' \in \text{grant} \rightarrow \exists q \in \text{grant} \]

\[ ((q', b) (d', v)) \in \text{Inference} \]

\[ \forall \text{grant} \]

\[ (\text{grant} (b) \text{grant} (d)) \]

\[ \text{No concurrent grant and/or accept} \]

\[ \text{Remote predicates} \]
\[(\bar{h} \geq z \geq x \lor (z^{(\mu x \cdot b \cdot d)}) \cdot z_E) \iff \]

\[(\bar{h} \geq x \lor (x^{(\mu x \cdot c \cdot b)}) \lor (x^{(\mu x \cdot c \cdot d)})) \cdot h, x_A = (b',d') \forall c \cap \]

\[(\bar{h} \geq z \geq x \lor (z^{(\mu x \cdot c \cdot b)}) \cdot z_E) \iff \]

\[(\bar{h} \geq x \lor (x^{(\mu x \cdot b \cdot b)}) \lor (x^{(\mu x \cdot b \cdot d)})) \cdot h, x_A = (b',d') \forall c \cap \]
Well defined consistent framework

Concurrent systems model checking

Branching models, event structures, Petri nets

Temporal properties and logic, Second order formulae

Automatic code generation and controller synthesis

Open for extensions and developments

Decidability (polynomial)

Modular decomposition

Expressivity, structure

Conclusion & Perspectives