Scenario Automata

*Theory and Applications*

Jury:
Albert Benveniste
Roland Groz
Paul Gastin
Claude Jard
Martin Leucker
Madhavan Mukund
Sophie Pinchinat
P.S. Thiagarajan

Loïc Hélouët

May 17th 2013
Motivations

Real World

Models
Motivations

Real World
A network of Turing Machines

Highly undecidable

Needs:
Design, validation, monitoring,…

Models
- Automated techniques
- Decidable properties
- Clear Semantics

But:
Close enough to real world?
Motivations

Why scenarios / partial order models?

- Intuitive
- Compact representation of concurrency

Objectives: tools for engineers

- Models for asynchronous & distributed systems behaviors
- Automated verification/analysis
- Avoiding global states computation

An example MSC
Motivations

Why scenarios / partial order models?

- Intuitive
- Compact representation of concurrency

Objectives: tools for engineers

- Models for asynchronous & distributed systems behaviors
- Automated verification/analysis
- Avoiding global states computation
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion
Scenario automata : MSCs

\[ M = (E, \{<_p\}_{p \in P}, \alpha, \mu, \phi) \]

Labelled partial order over a finite set of events \( E \)

- \( P \) : set of processes
- \( \alpha : \Sigma \) : labeling : \( \alpha(e) = \text{Client} !\text{Server}(\text{question}), \ldots \)
- \( \phi : E \to P \) : locality of events
- \( <_p \) : total ordering for each process \( p \in P \)
- \( \mu \) : message pairing

\[
\left( \bigcup_{p \in P} <_p \cup \mu \right)^* \text{ must be a partial order}
\]

\( \text{Lin}(M) = \text{linearizations of } \bigcup_{p \in P} <_p \cup \mu \)
**Concatenation**

\[ MSC \, M_1 \, \circ \, M_2 \]

\[ MSC \, M_1 \]

\[ MSC \, M_2 \]

\[ MSC \, M_1 \, \circ \, M_2 \]

Scenario Automata
High-level MSCs

A scenario automaton defined over a finite set of MSCs $\mathcal{M}$

Path \[ \rho = q_0 \xrightarrow{\mathcal{M}_1} q_1 \xrightarrow{\mathcal{M}_2} \ldots \xrightarrow{\mathcal{M}_k} q_k \]

\[ \rho^\circ = \mathcal{M}_1 \circ \mathcal{M}_2 \circ \ldots \mathcal{M}_k \]

Semantics

- $P_H = \{ \rho = q_0 \xrightarrow{\mathcal{M}_1} q_1 \xrightarrow{\mathcal{M}_2} \ldots \xrightarrow{\mathcal{M}_k} q_k \mid q_k \in Q_{F_i} \}$
- $F_H = \{ \rho^\circ \mid \rho \in P_H \}$
- $Lin_H = \bigcup_{\mathcal{M} \in F_H} Lin(\mathcal{M})$
High-level MSCs

An equivalent transition system
...with infinite state space
Undecidable problems

Let $H_1$, $H_2$ be HMSCs

- $F_{H_1} \cap F_{H_2} = \emptyset$ ?
- $F_{H_1} \subseteq F_{H_2}$ ?
- $F_{H_1} = F_{H_2}$ ?
- $Lin_{H_1} \cap Lin_{H_2} = \emptyset$ ?
- $Lin_{H_1} \subseteq Lin_{H_2}$ ?
- $Lin_{H_1} = Lin_{H_2}$ ?
- $Lin_{H_1}$ Regular ?

Let $R$ be a regular subset of $\Sigma^*$:

- $R \subseteq Lin_{H_1}$ ?
- $Lin_{H_1} \subseteq R$ ?

Not surprising: HMSCs closely related to Mazurkiewicz traces

[Alur et al 99]
[Muscholl et al 99]
[Darondeau et al 00]
[Henriksen et al 05]

Scenario Automata
An HMSC $H$ is **regular** iff for every cycle $\rho$ of $H$ 
$(\phi(E_\rho), \phi(\mu_\rho))$ is a **strongly connected** graph

every message sent is acknowledged after a finite nb of iterations of $\rho$.

**Theorem:**

$H$ regular $\implies$ $Lin_H$ regular subset of $\Sigma^*$

**Consequences:**

$Lin_{H_1} \cap Lin_{H_2} = \emptyset$, $Lin_{H_1} \subseteq Lin_{H_2}$, $Lin_{H_1} = Lin_{H_2}$

$R \subseteq Lin_{H_1}$, $Lin_{H_1} \subseteq R$ decidable
Globally cooperative HMSCs

An HMSC $H$ is **globally cooperative** iff for every cycle $\rho$ of $H$ $(\phi(E_{\rho}), \phi(\mu_{\rho}))$ is a **connected** graph.

Processes do not behave independently in a loop.

**Theorem:**

Let $H_1$ be a HMSC, $H_2$ be a globally cooperative HMSC, then

- $F_{H_1} \cap F_{H_2} = \emptyset$? are decidable
- $F_{H_1} \subseteq F_{H_2}$?

*There is a bound $b \in \mathbb{N}$ s.t all $M \in F_H$ can be run with communication buffers of size $\leq b$. (Existential bound)*
Outline

- Scenario automata
- Extensions
  - Compositional MSCs
  - Causal HMSCs
  - Dynamic MSC Grammars
- Operators
- Verification & partial order logics
- Application
- Conclusion
A drawback of HMSCs

MSC languages generated by HMSCs, GC-HMSCs, regular HMSCs …

…are all finitely generated
Compositional HMSCs

\[ H = (Q, \rightarrow, M, q_0, Q_{Fi}) \]

A partial order automaton defined over a set of cMSCs \( M \)

**Dangling Messages emissions/receptions**

**Glued at composition time**

**Semantics:**

- \( P_H = \{ \rho = q_0 \overset{M_1}{\rightarrow} q_1 \overset{M_2}{\rightarrow} \ldots \overset{M_k}{\rightarrow} q_k \mid q_k \in Q_{Fi} \} \)
- \( F_H = \{ \rho^\circ \mid \rho \in P_H \text{ and } \rho^\circ \text{ is an MSC} \} \)
- \( \text{Lin}_H = \bigcup_{M \in F_H} \text{Lin}(M) \)

An example C-HMSC

[Gunter et al’01, GenestPhD04]
Compositional HMSCs

Generates

Clearly not finitely generated!
Undecidable problems

C-HMScs embed:
- HMScs
- Communicating finite state machines (CFSM)
  ... and all their undecidable problems

The simple message problem:
is there an MSC $M \in F_H$ that contains a message of type $m$?

The emptiness problem:
is $F_H$ empty? are undecidable
Subclasses of cHMSCs

Some paths of a cHMSC may not generate an MSC

- A cHMSC $H$ is safe is for every path $\rho$ of $P_H$, $\rho^\circ$ is an MSC.
  (safe CHMSCs do not embed the whole expressive power of CFSMs)

- Globally cooperative CHMSCs = safe + GC

- Regular CHMSCs = safe + regular
Causal MSCs [GGHTY07, GGHTY09]

Labelled partial order over a set of events

\[ M = (E, \{\sqsubseteq_p\}_{p \in P}, \alpha, \mu, \phi) \]

- \( P, \alpha, \phi \) as usual,
- \( \mu \) : message pairing as usual
- \( \sqsubseteq_p \) : partial order

\[ \left( \bigcup_{p \in P} \sqsubseteq_p \cup \mu \right)^* \] is a partial order

\[ \text{Lin}(M) = \text{linearizations of } \bigcup_{p \in P} \sqsubseteq_p \cup \mu \]
Visual extensions

$Vis(M) = \text{MSCs that are compatible with } M$
Concatenation

**Independence relation** $I_p \subseteq \Sigma_p \times \Sigma_p$ for each $p \in P$
(Symmetric and irreflexive)

\[
I_p = \{ (p!q(n), p!q(m')), (p!q(m'), p!q(n)) \} \\
I_q = \emptyset
\]

Note: $M_1$ or $M_2$ need not respect \{ $I_p$ $\}$ $p \in P$
Causal HMSCs

\[ H = (Q, \rightarrow, M, q_0, Q_{Fi}) \]

+ \{ I_p \}_{p \in P}

A partial order automaton defined over a finite set of Causal MSCs \( M \)

Path

\[ \rho = q_0 M_1 q_1 M_2 \ldots M_k q_k \]

\[ \rho^o = M_1 \circ^o M_2 \circ^o \ldots M_k \]

Semantics:

- \( P_H = \{ \rho = q_0 \overset{M_1}{\rightarrow} q_1 \overset{M_2}{\rightarrow} \ldots \overset{M_k}{\rightarrow} q_k \mid q_k \in Q_{Fi} \} \)

- \( F_H = \{ \rho^o \mid \rho \in P_H \} \)

- \( Vis_H = \bigcup_{M \in F_H} Vis(M) \)

- \( Lin_H = \bigcup_{M \in F_H} Lin(M) \)
Example

\[ I_{\text{client}} = \{(\text{Client}!\text{Server}(\text{Question}), \text{Client}?\text{Server}(\text{Answer}))\} \]
\[ I_{\text{Server}} = \emptyset \]
Results on Causal HMSCs

**Regular CaHMSCs**
- Automaton $A_H$ that recognizes $\text{Lin}_H$ (exponential size)
- $\cap$, $\subseteq$, $=$, regular Model checking decidable

**Globally cooperative CaHMSCs**
- $H$ CaHMSC and $H'$ with same trace alphabet
  - Build CaHMSCs over atoms of $H$ and $H'$
  - $F_H \cap F_{H'} = \emptyset$? PSPACE-complete
  - $F_H \subseteq F_{H'}$ EXPSPACE-complete

**For CaHMSCs $H$ and $H'$ with distinct independence relation**
- $F_H \cap F_{H'} = \emptyset$? undecidable
Comparison of MSC languages

- R-HMSC
- GC-HMSCs
- CaHMSCs
- GC-CaHMSCs
- CHMSCs
- Safe CHMSCs
- GC-CHMSCs
- R-CHMSCs

: Inclusion of MSC Languages

Extensions: Causal HMSCS
Dynamic MSC grammars

A context free grammar with MSCs as terminals

Axiom ::= $M_0$. A  
A ::= $M_1$.B | $M_2$  
B ::= $M_1$.B.$M_2$ | $M_2$

Variant of Dynamic MSCs [Leucker 02]

Extensions: Dynamic MSC Grammars
Dynamic MSC grammars

Generate MSC languages:
- With *dynamic* creation,
- *Over arbitrary* sets of processes
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion
Projection

Project MSC languages on a subset of events

Operators: projection

$X = \{a, b, c, d\}$

$\Pi_X(M)$

[GHM03]
Projection

- **HMSC projections** and safe CHMCSs have the same expressive power
  \[ \Pi(F_H) \cap F_{H'} = \emptyset \] decidable (if \( H' \) G.C.)
- One can decide if a projection is equivalent to an HMSC

Operators: projection
Parallel composition

$M_1 \parallel M_2 = \left\{ M \mid \text{lin}(M) \right\} \text{shuffle of } \text{lin}(M_1), \text{lin}(M_2)
\begin{align*}
\emptyset & \quad \text{if } M_1, M_2 \text{ disagree on order of common evts}
\end{align*}

$F_{H_1 \parallel H_2} = \bigcup_{M_1 \in F_{H_1}} M_1 \parallel M_2$

Problem:

$F_{H_1 \parallel H_2} = \emptyset$? undecidable!

Operators: parallel composition
Parallel Composition

Enforce common events on a single process

- Emptiness for HMSC products decidable (PSPACE)
- \( H_1 \parallel H_2 \) existentially bounded decidable
- If \( H_1, H_2 \) G. C. and \( H_1 \parallel H_2 \) is \( \exists \).bounded, then one can compute an \( \exists \) bounded cHMSC representing \( H_1 \parallel H_2 \) (but of exponential size)
Fibered product

Synchronize : MSCs and events in pairs of MSCs

Very syntactic... but more controllable than $\parallel : F_{H_1 \otimes H_2} = \emptyset$? decidable

Operators : parallel composition
Conclusion

- **HMSC projections** can be of practical use
  
  For verification purpose, to emphasize some causalities,…

- No satisfactory solution for parallel composition: undecidable emptiness, effective product for pairs of GC HMSCs but not beyond, ….

- Interesting subclasses of C/HMSCs are not preserved by $\Pi$, $//$, $\otimes$...
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion
Verification

Regular Model-checking (LTL, CTL, etc.) applies only to regular [H/C/CA]-HMSCs

Undecidable otherwise

HMSC-based verification:
Let $H_{\text{spec}}$, $H_{\text{bad}}$ be two C/Ca/HMSCs. If $H_{\text{bad}}$ is globally cooperative, then use $H_{\text{spec}} \cap H_{\text{bad}} = \emptyset$? (Safety)

Let $H_{\text{spec}}$, $H_{\text{good}}$ be two C/Ca/HMSCs. If $H_{\text{good}}$ is globally cooperative, then $H_{\text{spec}} \subseteq H_{\text{good}}$? (Refinement)

Problem: $H_{\text{spec}}$, $H_{\text{bad}}$, $H_{\text{good}}$ need to be defined at the same abstraction level

$H$ Glob.Coop. $\iff$ $\Pi(H)$ Glob.Coop.
MSCs are non-interleaved models:

reason on non-interleaved representation of behaviors with LOCAL logics (MSO,PDL,TLC,...)

**MSO for MSCs**

[Madhusudan01]

\[
\phi ::= lab_{a}(x) | msg((x, p), (y, q)) | next(x, y) | x \in X | p \in P
\]

\[
\neg \phi | \phi_1 \wedge \phi_2 | \exists x, \phi | \exists X, \phi | \exists p, \phi | \exists P, \phi
\]
Verification

Example: MSCs are FIFO.

\[ \varphi := \neg(\exists x, x', y, y', p, q, \text{msg}((x, p)(y, q)) \land \text{msg}((x, p)(y, q)) \land \text{next}(x, x') \land \text{next}(y, y')) \]

MSO for MSCs is **decidable** for

- HMSCS [Madhusudan01]
- Safe CHMSCs [MadhusudanMeenakshi01]
- Causal HMSCs
- Dynamic HMSCs, Dynamic MSC grammars [Leucker02] [BHH10b]

**Main principle:** build a (tree) automaton that recognizes models for \( \varphi \), intersect with the original model. Automata guess an interpretation for variables, synthesizes facts to recognize sequences of MSCs (parse trees) satisfying \( \varphi \).
Verification

MSO decidability is not so surprising:

all orders produced by these models can be seen as productions of a context free graph grammar.

Can we go further? : specify with logics and drop the automata/grammars/...

It is undecidable whether, given an MSO formula φ, there exists an MSC satisfying φ.

*Specifying with logical statements only is not possible.* [YHG08]
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application: Diagnosis
- Conclusion
Applications: Diagnosis

Objective:
provide information to supervisors of a distributed system starting from:
- a partial observation (log) : O
- a model of the system : M

Questions:
- Is there a run of M that embeds O ? (existence)
- list all runs that embed O ? (Diagnosis)
- is there a faulty run that embeds O ? (fault detect°)

Solutions for: Automata, Petri Nets, ...
Build a product or an unfolding : M x O
Diagnosis: \( H' = H \times O \)

Every M in \( F_{H'} \) embeds O
Embedding

O : Observation

Explanation for O as a MSC
Results

Theorem:

\[ H' = H \times O \] is an HMSC of size at most in

\[ O\left(\frac{|H|}{|O|}P \times |P_{obs}|\right) \] s.t.

- every \( M \) in \( F_H \) that embeds \( O \) is also in \( F_{H'} \)
- every \( M \) in \( F_{H'} \) embeds \( O \)

- Diagnosis/existence are decidable for HMSCs
- Nice associative properties: \((H \times O_1) \otimes (H \times O_2) = H \times (O_1 \parallel O_2)\)

Note: undecidable in general for CHMSCs

Take \( G \) a cHMSC: is there a run that embeds \( O \)

\( O = \) The simple message problem

Application: Diagnosis
Diagnosis as an MSO property

\[
\varphi_O := \exists x, y, z, t, \text{lab}(x, y, z, t) \land \text{caus}(x, y, z, t) \\
\land \text{closed}(x, y, z, t)
\]

\[
\text{lab}(x, y, z, t) := \text{lab}_a(x) \land \text{lab}_b(y) \land \text{lab}_c(z) \land \text{lab}_m(t)
\]

\[
\text{caus}(x, y, z, t) := x \leq y \land x \leq z
\]

\[
\text{closed}(x, y, z, t) := \forall x', x' < x \lor x' < y \lor x < x' < z \lor x' < t
\]
\[
\Rightarrow \text{lab}_{\Sigma_{\text{obs}}}(x')
\]

M embeds O \iff M \models \varphi_O

Diagnosis / existence are decidable for CaHMSCs, dynamic HMSCs,...
Outline

- Scenario automata
- Extensions
- Operators
- Verification & partial order logics
- Application
- Conclusion
Over the last 10 years

- Several extensions: Causal HMSCs, dynamic MSC Grammars, Extended coregions, etc.

- Decidable classes remain incomplete models.
- Composition w/o automaton/grammar structure highly undecidable
- Modularity (///) leads to undecidability too

- MSO decidability / diagnosis easily achieved
- Scenarios well adapted to Diagnosis.
Repeat

\([C/\text{Ca/Dyn}/...]\) HMSCs are incomplete

Propose an extension!

Pb X is undecidable for the extension

Find a subclass to solve the problem

Until ???

?? = (provocation)

- the model is complete enough to be a decent modeling tool
- your models are so ugly that nobody wants to use them
- no one wants to read a paper on MSCs anymore
- you get bored
- ....
Future plans

Scenario specific issues:

- **Generalize** decidability results for scenarios seen as graph grammars
- **Implementation**: for HMSCs, Dynamic MSC grammars
- **Causal HMSCs**: finite generation, equivalence with HMSCs
- **Diagnosis**: Online diagnosis, alternatives to MSO
- **Security**: covert flows detection with probabilities, information theory, etc.
Future plans

- **Time & Robustness**: Check if timed requirements make sense w.r.t. implementation assumptions (architecture, ...)

- **Abstraction**: obtain decidable models using abstractions of real systems

**From Telecoms to services:**

many assumptions on scenarios due to IP-like communications (FIFO, etc)

In **Services**: no FIFO, open and dynamic world, sessions, data...

*new models, new problems, new solutions*

*Closer to real world situations*
Acknowledgements

- The Distribcom/Vertecs/S4 (now SUMO) colleagues
- Students and Postdocs
  - A. Degorre
  - T. Gazagnaire, J. Klein, T. Ziadi, R. Abdallah, ...
  - S. Yang, S. Akshay, B. Masson
- All members of the CASDS, DST, and now DISTOL associated teams, INRIA for funding them

- courir@inria.fr, courir@irisa.fr

And of course....Philippe
Questions ?

Ravitaillement en salle Sein !
Communication graph

An example MSC $M$

Client

question

answer

OK

logged

store

Server

Log

An useful abstraction of the contents of MSCs

Communication graph $CG(M)$
Regular HMSCs

Definition: $H$ is regular iff

$$\forall \rho = q_1 \xrightarrow{M_1} q_2 \xrightarrow{M_2} \ldots \xrightarrow{M_k} q_1$$ cycle of $H$, $CG(\rho^\circ)$ is a strongly connected graph

Theorem:

$H$ regular $\Rightarrow Lin_H$ regular subset of $\Sigma^*$

Consequences:

$$Lin_{H_1} \cap Lin_{H_2} = \emptyset, Lin_{H_1} \subseteq Lin_{H_2}, Lin_{H_1} = Lin_{H_2}$$

$R \subseteq Lin_{H_1}, Lin_{H_1} \subseteq R$ decidable

[Alur et al 99] [Muscholl et al 99]
Globally cooperative HMSCs

**Definition**: $H$ is **Globally Cooperative** iff

$$\forall \rho, \text{ cycle of } H, CG(\rho^\circ) \text{ is a connected graph}$$

**Theorem**: [Genest et al 02]

Let $H_1$ be a HMSC, $H_2$ be a **globally cooperative** HMSC, then

$$F_{H_1} \cap F_{H_2} = \emptyset \quad \text{?,}$$

$$F_{H_1} \subseteq F_{H_2} \quad \text{Decidable}$$
 States recall:
- A node of H
- for every process p, the events of O that have been « seen » by p
Product: transitions
Product : final nodes

M1

qi

EA

O

EC

ED

M2

A

B

C

D

qj

A

B

C

D

O

EC'

ED'

61
Applications : intrusion detection

Idea : use HMSCs to represent « normal behaviors » of a system

Normal (O,H₁,...Hₙ)

for (i=0;i<n;i++){

  if (L(Hᵢ × πₜᵢ (O)) = φ) {
    return true // raise an alarm
  }
}
Return false

If no explanation exists for an observation O, raise an alarm.