Abstraction and Modular Verification of Inter-Enterprise Business Processes

KAIS KLAI

LIPN,
UNIVERSITY PARIS 13, SORBONNE PARIS CITÉ

Journée - Action inter GDR AFSEC, 16 Octobre 2014
Abstraction and Modular Verification of Inter-Enterprise Business Processes

KAIS KLAI

LIPN,
UNIVERSITY PARIS 13, SORBONNE PARIS CITÉ

Joint work with Jörg Desel, Samir Tata and Hanen Ochi

Journée - Action inter GDR AFSEC, 16 Octobre 2014
Outline

1 Context

2 Event-based Symbolic Observation Graph

3 Our approach
   - Deadlock freeness property
   - Other Correction Criteria
   - Application to Composite Web Services

4 Conclusion and Perspectives
Outline

1 Context

2 Event-based Symbolic Observation Graph

3 Our approach
   - Deadlock freeness property
   - Other Correction Criteria
   - Application to Composite Web Services

4 Conclusion and Perspectives
**Business Process**: A series of logically related activities or tasks performed together in order to accomplish a specific organizational goal.
Abstraction and Verification of Inter-Enterprise Business Processes (IEBP)

- **Business Process**: A series of logically related activities or tasks performed together in order to accomplish a specific organizational goal.

- **IEBP**: The collaboration of several local BPs
IEBP: Characteristics and requirements

- Relatively simple models
- Loosely coupled (in general)
- Each local process is designed and analysed locally
IEBP: Characteristics and requirements

- Relatively simple models
- Loosely coupled (in general)
- Each local process is designed and analysed locally
- Need to collaborate with partners
- Ensure correct behaviour of the whole process
- Want to hide the trade secrets and the organisational structure of their local processes
IEBP: Characteristics and requirements

- Relatively simple models
- Loosely coupled (in general)
- Each local process is designed and analysed locally
- Need to collaborate with partners
- Ensure correct behaviour of the whole process
- Want to hide the trade secrets and the organisational structure of their local processes

What is the right abstraction?
IEBP: Characteristics and requirements

- Relatively simple models
- Loosely coupled (in general)
- Each local process is designed and analysed locally
- Need to collaborate with partners
- Ensure correct behaviour of the whole process
- Want to hide the trade secrets and the organisational structure of their local processes

What is the right abstraction?

- Sufficiently detailed for potential partners
- Hides the local organisation/structure (privacy)
- Suitable for checking the correction of the composite process
Outline

1. Context

2. Event-based Symbolic Observation Graph

3. Our approach
   - Deadlock freeness property
   - Other Correction Criteria
   - Application to Composite Web Services

4. Conclusion and Perspectives
Event-Based SOG


Goal: Check event-based LTL $\mathcal{X}$ properties

Principle: Compute a reduced abstraction of the state space

Event based LTL $\mathcal{X}$ formula

Observed events: events occurring in the formula (Obs)

$\mathcal{G}(a = \Rightarrow F b) \rightarrow \text{Obs} = \{a, b\}$

SOG

Hybrid Graph

Nodes (aggregates): sets of explicit states

Symbolic encoding (BDDs)

Symbolic algorithms (deadlock, livelock)
Goal: Check event-based $LTL \setminus X$ properties
**Goal:** Check event-based $LTL \setminus X$ properties

**Principle:** Compute a reduced abstraction of the state space

- Event based $LTL \setminus X$ formula
- Observed events: events occurring in the formula ($Obs$)
  - $G(a \Rightarrow F b) \rightarrow Obs=\{a, b\}$
**Goal:** Check event-based $LTL \setminus X$ properties

**Principle:** Compute a reduced abstraction of the state space

- Event based $LTL \setminus X$ formula
- Observed events: events occurring in the formula ($Obs$)
  - $G(a \implies F b) \implies Obs = \{a, b\}$

**SOG**

- Hybrid Graph
  - Nodes (aggregates): sets of explicit states
    - Symbolic encoding (BDDs)
    - Symbolic algorithms (deadlock, livelock)
  - Edges: labelled by observed events
A LTS
Event-Based SOG


A corresponding SOG

A LTS

\[
\begin{align*}
\text{a LTS} & \quad \text{a corresponding SOG} \\
\tau & \quad \tau \\
\tau & \quad \tau \\
\end{align*}
\]
Event-Based SOG


Observe collaborative actions and hide local ones

A corresponding SOG
Observe collaborative actions and hide local ones
Outline

1. Context

2. Event-based Symbolic Observation Graph

3. Our approach
   - Deadlock freeness property
   - Other Correction Criteria
   - Application to Composite Web Services

4. Conclusion and Perspectives
Approach

Bottom-Up approach [K. Klai, S. Tata and J. Desel BPM’09-DKE’11]

The composite model is unavailable
Bottom-Up approach [K. Klai, S. Tata and J. Desel BPM’09-DKE’11]

The composite model is unavailable

- Model of $P_1$
  - LTS of $P_1$
  - SOG of $P_1$
- Model of $P_2$
  - LTS of $P_2$
  - SOG of $P_2$

$$\text{SOG}_1 \times \text{SOG}_2$$

Correctness Property
Bottom-Up approach [K. Klai, S. Tata and J. Desel BPM’09-DKE’11]

The composite model is unavailable

- Model of \( P_1 \)
- \( \text{LTS of } P_1 \)
- \( \text{SOG of } P_1 \)
- \( \text{SOG}_1 \times \text{SOG}_2 \)
- \( \text{Correctness Property} \)

- Model of \( P_2 \)
- \( \text{LTS of } P_2 \)
- \( \text{SOG of } P_2 \)
Bottom-Up approach [K. Klai, S. Tata and J. Desel BPM’09-DKE’11]

The composite model is unavailable

\[
\begin{align*}
\text{Model of } P_1 &\rightarrow \text{LTS of } P_1 & \text{Model of } P_2 &\rightarrow \text{LTS of } P_2 \\
\text{Model of } (P_1 \times P_2) &\rightarrow \text{LTS of } (P_1 \times P_2)
\end{align*}
\]

\[
\begin{align*}
\text{SOG of } P_1 &\rightarrow \text{SOG of } P_1 \\
\text{SOG of } P_2 &\rightarrow \text{SOG of } P_2 \\
\text{SOG}_1 \times \text{SOG}_2 &\rightarrow \text{Correctness Property} \\
\text{SOG}(P_1 \times P_2) &\rightarrow \text{Correctness Property}
\end{align*}
\]
Outline

1. Context

2. Event-based Symbolic Observation Graph

3. Our approach
   - Deadlock freeness property
   - Other Correction Criteria
   - Application to Composite Web Services

4. Conclusion and Perspectives
Composition of SOGs: Deadlock freeness property

\[ a = \langle a_1, a_2 \rangle \] obtained from \( \lambda(a_1) \) and \( \lambda(a_2) \)

Interlock: Deadlock caused by the interaction
Composition of SOGs: Deadlock freeness property

Interlock: Deadlock caused by the interaction
Composition of SOGs: Deadlock freeness property

\[ \lambda(a_1) : \{\{o_1\}, \{o_2\}, \{\text{term}\}\} \]

\[ \lambda(a_2) : \{\{o_1\}, \{o_2\}, \{\text{term}\}\} \]

- Interlock: Deadlock caused by the interaction
Composition of SOGs: Deadlock freeness property

- $a = \langle a_1, a_2 \rangle$: $\lambda(a)$ obtained from $\lambda(a_1)$ and $\lambda(a_2)$

- **Interlock**: Deadlock caused by the interaction

\[ \lambda(a_1) : \{ \{ o_1 \}, \{ o_2 \}, \{ \text{term} \} \} \]

\[ \lambda(a_2) : \{ \{ o_1 \}, \{ o_2 \}, \{ \text{term} \} \} \]
From BP models to Petri nets

WF-nets [Van Der Aalst 98]

WF-net contractor

WF-net sub-contractor
Event-Based SOGs: Example

SOG contractor

SOG sub-contractor
Event-Based SOGs: Example

explicit size: 38/104

explicit size: 14/22

SOG contractor

SOG sub-contractor
Synchronous Composition

\[
\langle a_0, a'_0 \rangle \{\{\text{order}\}\}
\]

\[
\langle a_1, a'_1 \rangle \{\emptyset, \{\text{cost}\}\}
\]

\[
\langle a_2, a'_3 \rangle \{\{\text{spec}\}\}
\]

\[
\langle a_3, a'_5 \rangle \{\{\text{product}\}\}
\]

\[
\langle a_4, a'_6 \rangle \{\{\text{term}\}\}
\]

Synchronous SOGs product
Synchronous Composition

explicit size: 99/320

Synchronous SOGs product
Asynchronous Composition

Interface medium

Diagram showing the flow of information between two interfaces, $i_1$ and $i_2$, with the following connections:

- $i_1$ to $b_1$ (order)
- $b_1$ to $i_2$ (order)
- $i_1$ to $b_2$ (spec)
- $b_2$ to $i_2$ (spec)
- $i_1$ to $b_3$ (cost)
- $b_3$ to $i_2$ (cost)
- $i_1$ to $b_4$ (product)
- $b_4$ to $i_2$ (product)

Outputs:

- $o_1$ (order)
- $o_2$ (spec, cost, product)
Asynchronous Composition

Interface medium

Interface Graph

GDR AFSEC 12/2014  Abstraction and Modular Verification of Inter-Enterprise Business Processes
Asynchronous Composition

Interface medium

| i₁ → order | b₁ → i₂ → order |
| spec → b₂ | spec → i₂ → spec |
| cost → b₃ | cost → i₂ → cost |
| product → b₄ | product → i₂ → product |
| o₁ | o₂ |

Interface Graph

explicit size: 109/373 vs 9/8
Other Correction Criteria

Model of $P_1$  
$\xrightarrow{\text{LTS of } P_1}$  
$\xrightarrow{\text{SOG of } P_1}$  
$\xrightarrow{\text{SOG}_1 \times \text{SOG}_2}$  
$\xrightarrow{\text{Correctness Property}}$

Model of $P_2$  
$\xrightarrow{\text{LTS of } P_2}$  
$\xrightarrow{\text{SOG of } P_2}$  
$\xrightarrow{\text{SOG}_1 \times \text{SOG}_2}$  
$\xrightarrow{\text{Correctness Property}}$

Model of $(P_1 \times P_2)$  
$\xrightarrow{\text{LTS of } (P_1 \times P_2)}$  
$\xrightarrow{\text{SOG}(P_1 \times P_2)}$  
$\xrightarrow{\text{Correctness Property}}$

GDR AFSEC 12/2014  Abstraction and Modular Verification of Inter-Enterprise Business Processes
Other Correction Criteria

- Soundness variants
Other Correction Criteria

- **Soundness variants**
  - **Soundness** [Van Der Aalst ICATPN'97]
    - Option to complete
    - Proper completion
    - No dead transitions
Other Correction Criteria

- **Soundness variants**
  - **Soundness** [Van Der Aalst ICATPN'97]
    - Option to complete
    - Proper completion
    - No dead transitions
  - **Weak Soundness**
    - Option to complete
Other Correction Criteria

Soundness variants

- **Soundness** [Van Der Aalst ICATPN'97]
  - Option to complete
  - Proper completion
  - No dead transitions

- **Weak Soundness**
  - Option to complete

- **Relaxed Soundness**
  - Each transition belongs to a proper execution
Other Correction Criteria

- **Soundness variants**
  - **Soundness** [Van Der Aalst ICATPN'97]
    - Option to complete
    - Proper completion
    - No dead transitions
  - **Weak Soundness**
    - Option to complete
  - **Relaxed Soundness**
    - Each transition belongs to a proper execution
  - **Easy Soundness**
    - There exists a proper execution
Other Correction Criteria

- **Soundness variants**
  - **Soundness** [Van Der Aalst ICATPN'97]
    - Option to complete
    - Proper completion
    - No dead transitions
  - **Weak Soundness**
    - Option to complete
  - **Relaxed Soundness**
    - Each transition belongs to a proper execution
  - **Easy Soundness**
    - There exists a proper execution

Enrich aggregates with locally computed information
Other Correction Criteria

Soundness variants

- **Soundness** [Van Der Aalst ICATPN'97]
  - Option to complete
  - Proper completion
  - No dead transitions

- **Weak Soundness**
  - Option to complete

- **Relaxed Soundness**
  - Each transition belongs to a proper execution

- **Easy Soundness**
  - There exists a proper execution

Enrich aggregates with locally computed information

Application to Composite Services
Outline

1 Context

2 Event-based Symbolic Observation Graph

3 Our approach
   - Deadlock freeness property
   - Other Correction Criteria
   - Application to Composite Web Services

4 Conclusion and Perspectives
Application to Composition of Web Services

PhD Hanen Ochi

Diagram: A service broker is connected to service consumers and service providers. Service consumers can find services through the service broker, and service providers can register their services with the service broker. The service broker also manages service contracts.
Application to Composition of Web Services

PhD Hanen Ochi

- Publish public description
- Collaboration with potential requesters
- Privacy
Application to Composition of Web Services

PhD Hanen Ochi

- Publish public description
- Collaboration with potential requesters
- Privacy
- Composition of Web Services
**Compatibility criteria** [K. Klai and H. Ochi FSEN’12, + S. Tata ICWS’13, IJWSR’13]

- generic criteria (deadlock, soundness variants)
- specific criteria (expressed with LTL)
Composition of Web Services

Compatibility criteria [K. Klai and H. Ochi FSEN’12, + S. Tata ICWS’13, IJWSR’13]
- generic criteria (deadlock, soundness variants)
- specific criteria (expressed with LTL)

Implementation:
- Deadlock-freeness (integrated to CosyVerif)
- Soundness variants
- LTL modular model checking (not finished yet)

Performs much better than LoLA and better than Woflan
Outline

1. Context

2. Event-based Symbolic Observation Graph

3. Our approach
   - Deadlock freeness property
   - Other Correction Criteria
   - Application to Composite Web Services

4. Conclusion and Perspectives
Conclusion and Perspectives

- Abstraction and Modularity
  - Monolectic Verification
  - Domain-specific Verification
Conclusion and Perspectives

- Abstraction and Modularity
  - Monolethic Verification
  - Domain-specific Verification

- Perspectives
  - Finish and disseminate our tools
Conclusion and Perspectives

- **Abstraction and Modularity**
  - Monolectic Verification
  - Domain-specific Verification

- **Perspectives**
  - Finish and disseminate our tools
  - Time in Business Processes
  - Resources in Business Processes
  - Timed Constraints as Compatibility Criteria for Web Services
Conclusion and Perspectives

- Abstraction and Modularity
  - Monolethic Verification
  - Domain-specific Verification

- Perspectives
  - Finish and disseminate our tools
  - Time in Business Processes
  - Ressources in Business Processes
  - Timed Constraints as Compatibility Criteria for Web Services
  - Service-Based Business Processes in the Cloud
    - Elastic Processes [K. Klai and S. Tata SCC’13, M. Amziani et al [CloudCom’13]]
    - Multi-tenancy, ...
Conclusion and Perspectives

- **Abstraction and Modularity**
  - Monolectic Verification
  - Domain-specific Verification

- **Perspectives**
  - Finish and disseminate our tools
  - Time in Business Processes
  - Ressources in Business Processes
  - Timed Constraints as Compatibility Criteria for Web Services
  - Service-Based Business Processes in the Cloud
    - Elastic Processes [K. Klai and S. Tata SCC’13, M. Amziani et al [CloudCom’13]]
    - Multi-tenancy, ...
  - Security Properties (Opacity [N. Hamdi, K. Klai and N.B-H Alouane WETICE’14])
THANK YOU