A Formal Dynamic Verification of Choreographed Web Services Conversations

Karim Dahmani & Mahjoub Langar & Riadh Robbana

Agenda

1. Outline
2. Choreography Specification Language
3. Security Policy Specification Language
4. Enforcement Approach
5. Conclusion

Outline

- Motivations
- Web services composition
- Security policies
- Formalization of the problem
- The proposed approach
- Example

Motivations

Problem

Ensuring that a service coming from an untrusted source will not compromise the integrity and the good operation of the target system.

Goal

To develop a formal technique that enforces a security policy on a given choreographed services, while providing a proof of validity.

Web Services Composition

- Composition of web services offer complex services.
- Techniques of composition are
  - Orchestration defines an orchestrater that monitor the different implied web services (BPEL).
  - Choreography defines complex tasks to coordinate collaborations between web services (WS-CDL).

Security Policy

Safety Property

- Asserts that nothing bad happens.
- Program must not perform a send on the network after reading a file.

Liveness Property

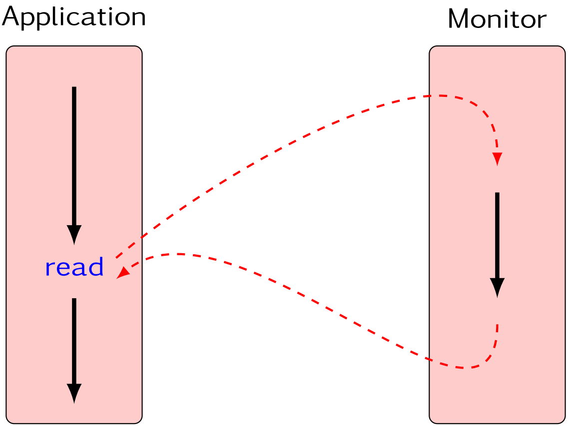
- Asserts that something good eventually happens.
- The program will terminate, the light will turn green.

Program Monitor

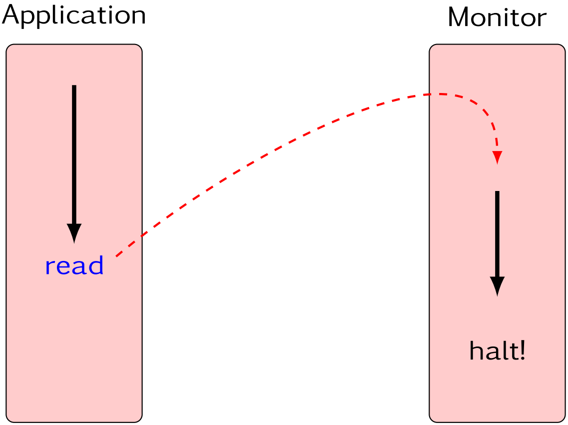
A program monitor is a program that runs in parallel with an untrusted application

- monitors detect, prevent and recover from application errors at run time
- monitor decisions may be based on the history of all actions an application has executed
- we assume monitors have no knowledge of future application actions

Program Monitors : good operations



Program Monitors : bad operations



Formalization of the Problem

$$
\begin{tikzcd}
\text{CS} \arrow[d] & & \text{SP} \arrow[d] \\
P \arrow[dr] & & P_\phi \arrow[dl] \\
& \otimes \arrow[d] & \\
& P' = P \otimes P_\phi &
\end{tikzcd}
$$

Formalization of the Problem

Correctness

- $P' \mid\!\sim \phi$, i.e. P' "satisfies" the security policy ϕ.
- $P' \sqsubseteq P$, i.e. behaviors of P' are also behaviors of P.

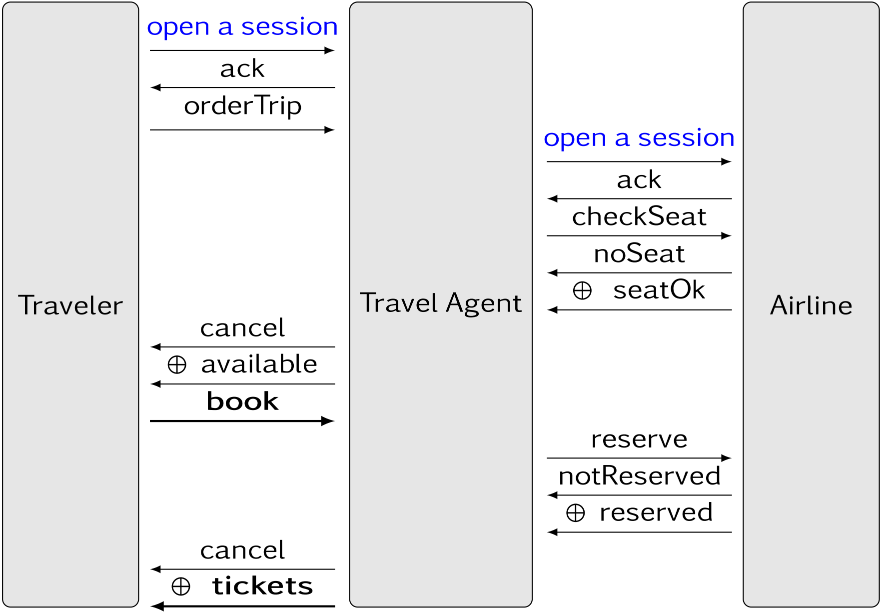
Completeness

- $\forall Q \; : \; ((Q \mid\!\sim \phi) \wedge (Q \sqsubseteq P)) \Longrightarrow Q \sqsubseteq P'$, i.e. all good behaviors of P are also behaviors of P'.

The Proposed Approach

- Security Policy : L_φ Logic
  - Linear temporal logic
  - Safety properties : something bad will not happen
- Program : End-Point Calculus
  - Communication centered systems
  - Based on π-calculus

Airline Reservation System



Agenda

1. Outline
2. Choreography Specification Language
3. Security Policy Specification Language
4. Enforcement Approach
5. Conclusion

End-Point Calculus

- The syntax of EPC is :

$$
\begin{array}{rcl}
P & ::= & !ch(\tilde{s}).P \\
 & | & \overline{ch}(\nu\tilde{s}).P \\
 & | & s \triangleright \sum_i op_i(x_i).P_i \\
 & | & \overline{s} \triangleleft op\langle e\rangle.P \\
 & | & \text{if } e \text{ then } P \text{ else } Q \\
 & | & P_1 \oplus P_2 \\
 & | & P_1 | P_2 \\
 & | & (\nu s)P \\
 & | & \text{rec } X.P \\
 & | & 0
\end{array}
$$

Networks

- A participant A with its behavior P at a local state σ is called a network and denoted by $A[P]_\sigma$.
- Syntax of networks is given by the following grammar :

$$\begin{array}{lcl} N & ::= & A[P]_\sigma \\ & \mid & N|M \\ & \mid & (\nu s)N \\ & \mid & \epsilon \end{array}$$

Semantics of EPC

Semantics of *EPC* are given by :

$$\frac{\sigma_2 \vdash e \Downarrow v}{A[\overline{s} \lhd op_j\langle e\rangle.P]_{\sigma_1} | B[s \rhd \sum_i op_i(x_i).Q_i]_{\sigma_2} \rightarrow A[P]_{\sigma_1} | B[Q_j]_{\sigma_2[x_j \mapsto v]}}$$

$$\frac{A[P_1]_\sigma \rightarrow A[P_1']_{\sigma'}}{A[P_1 \oplus P_2]_\sigma \rightarrow A[P_1']_{\sigma'}} \qquad \frac{A[P[\mathrm{rec}\ X.P/X]]_\sigma \rightarrow A[P']_{\sigma'}}{A[\mathrm{rec}\ X.P]_\sigma \rightarrow A[P']_{\sigma'}}$$

Airline Reservation System

Behaviors of the traveler, the travel agent and the airline reservation system are given in EPC by :

Traveler

$Traveler[\overline{ch_{TA}}(\nu s).s \triangleright ack.\overline{s} \triangleleft orderTrip\langle e_1\rangle.(s \triangleright cancel.0 \oplus s \triangleright available(x_1).\overline{s} \triangleleft book\langle e_2\rangle.(s \triangleright cancelBook.0 \oplus s \triangleright tickets(x_2).0))]_{\sigma_T}$

Travel Agent

$TravelAgent[!ch_{TA}(s).\overline{s} \triangleleft ack.s \triangleright OrderTrip(x_1).\overline{ch_A}(\nu s').s' \triangleright ack.\overline{s'} \triangleleft check\langle e_1\rangle.(s' \triangleright noSeats.\overline{s} \triangleleft cancel.0 \oplus s' \triangleright seatsOK(x_2).\overline{s} \triangleleft available\langle e_2\rangle.s \triangleright book(x_3).\overline{s'} \triangleleft reserve\langle e_3\rangle.(s' \triangleright reserved(x_4).\overline{s} \triangleleft tickets\langle e_4\rangle.0 \oplus s' \triangleright notReserved(x_5).\overline{s} \triangleleft cancelBook.0))]_{\sigma_{TA}}.$

Airline Reservation System

Airline

$\text{Airline}[!ch_A(s').\overline{s'} \triangleleft ack.s' \triangleright$
$check(x_1).\text{if } available(x_1) \text{ then } \overline{s'} \triangleleft seatsOK\langle e_1\rangle.s' \triangleright$
$reserve(x_2).\text{if } available(x_2) \text{ then } \overline{s'} \triangleleft reserved\langle e_2\rangle.0 \text{ else } \overline{s'} \triangleleft$
$notReserved\langle e_3\rangle.0 \text{ else } \overline{s'} \triangleleft noSeats.0]_{\sigma_A}$

Agenda

1. Outline
2. Choreography Specification Language
3. Security Policy Specification Language
4. Enforcement Approach
5. Conclusion

L_φ Logic

Syntax

Table: Syntax of L_φ.

$$
\begin{array}{rcl}
\varphi_1, \varphi_2 & ::= & tt \mid 1 \mid a \mid \varphi_1.\varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi \mid \varphi_1^{*}\varphi_2 \\
a & ::= & \overline{s} \lhd op\langle e\rangle \mid s \rhd op(x)
\end{array}
$$

Airline Reservation System

Security Requirements

In the airline reservation system, the travel agent wants to be sure that his service does not send tickets before the reception of payment details. So we want to ensure that $\overline{s} \triangleleft tickets\langle e_4 \rangle$ does not occur before $s \triangleright book(x3)$.

Airline Reservation System

Security Policy in L_φ

$$(\neg(\overline{s} \triangleleft tickets\langle e\rangle \oplus s \triangleright book(x)))^{*} s \triangleright book(x).tt$$

Agenda

1. Outline
2. Choreography Specification Language
3. Security Policy Specification Language
4. Enforcement Approach
5. Conclusion

Secured EPC^{φ}

Approach

It consists on extending EPC by an enforcement operator $\partial_{\varphi}^{\xi}(P)$ responsible for monitoring a process P with respect to its execution environment ξ and the security property φ.

Syntax

$$\partial_{\varphi}^{\xi}(P)$$

- P is a process from EPC.
- φ is the enforced security property.
- ξ is the execution environment of P. It saves the trace of already executed actions by P.

Secured EPC^{φ}

Approach

It consists on extending EPC by an enforcement operator $\partial_{\varphi}^{\xi}(P)$ responsible for monitoring a process P with respect to its execution environment ξ and the security property φ.

Syntax

$$\partial_{\varphi}^{\xi}(P)$$

- P is a process from EPC.
- φ is the enforced security property.
- ξ is the execution environment of P. It saves the trace of already executed actions by P.

Definitions

- **Normal Form of a Process**
  Every process representing the local behavior of a participant in a web service can be written as an internal sum of processes, which we call the normal form of a process :
  $$\forall P \in \mathcal{P},\ P = \bigoplus_i a_i P_i$$
  where $\mathcal{P}$ denotes the set of processes, a_i range over atomic actions and P_i range over processes of $\mathcal{P}$.
- **Simulation Relation**

$$\frac{P = \bigoplus_i a_i P_i \quad \exists i \in \{1, \ldots, n\} : a = a_i}{A[P]_\sigma \overset{a}{\rightsquigarrow} A[P_i]_\sigma}$$

Definitions

- **Satisfaction Notion**
  Intuitively, a trace may satisfy a security propery when it is a prefix of a trace that satisfies the security property.

$$\xi \vDash \varphi \iff \xi \in [\![\varphi]\!].$$

$$\xi \mathrel{|\!\sim} \varphi \iff \exists \xi' : \xi.\xi' \in [\![\varphi]\!].$$

Semantics of EPC^{φ}

$$\frac{P \overset{s \triangleright op(x)}{\rightsquigarrow} P' \quad \xi.s \triangleright op(x) \vdash \varphi \quad \sigma_A \vdash e \Downarrow v}{A[\partial_{\varphi}^{\xi}(P)|P_1]_{\sigma_A} | B[\overline{s} \triangleleft op\langle e\rangle.Q|R]_{\sigma_B} \rightarrow A[\partial_{\varphi}^{\xi.s \triangleright op(x)}(P')|P_1]_{\sigma_A[x \mapsto v]} | B[Q|R]_{\sigma_B}} \ (\partial\text{-ComIn})$$

$$\frac{P \overset{\overline{s} \triangleleft op_j\langle e\rangle}{\rightsquigarrow} P' \quad \xi.\overline{s} \triangleleft op_j\langle e\rangle \vdash \varphi \quad \sigma_B \vdash e \Downarrow v \quad j \in I}{A[\partial_{\varphi}^{\xi}(P)|P_1]_{\sigma_A} | B[s \triangleright \underset{i \in I}{\circ} op_i(x_i).Q_i|R]_{\sigma_B} \rightarrow A[\partial_{\varphi}^{\xi.\overline{s} \triangleleft op_j\langle e\rangle}(P')|P_1]_{\sigma_A} | B[Q_j|R]_{\sigma_B[x_j \mapsto v]}} \ (\partial\text{-ComOut})$$

Airline Reservation System

Secured Choreography

$$Traveler[P] \mid TravelAgent[\partial_{\varphi}^{\epsilon}(P)] \mid Airline[R]$$

Agenda

Conclusion and Future Works

- We have extended an existing calculus with an enforcement operator $\partial_{\varphi}^{\xi}(P)$ having the role of an IRM that mediates the execution of a choreography of web services by controlling the behavior of each involved participant. Hence, $\partial_{\varphi}^{\xi}(P)$ intercepts communication actions of P and verifies whether their execution adheres to security constraints defined by the formula φ.
- Future work consists on extending L_{φ} for supporting information flow control. It is intended also to optimize this security framework by making $\partial_{\varphi}^{\xi}(P)$ intercept only security-relevant communication actions.

Thanks for your attention!

Questions?