Timed Specification For Web Services Compatibility Analysis

Nawal Guermouche
nawal.guermouche@loria.fr

Joint work with Olivier Perrin and Christophe Ringeissen
Introduction (1): Web services

- Interoperability and integration of applications:
  - Based on standards: WSDL, SOAP,...
  - Design and build complex business applications:
    - Interaction of Web services
Introduction (2): WS Compatibility

- Successful interaction of Web services:
  - Compatibility of Web services

- Two services are compatible:
  - Can terminate correctly when interacting together

- Need for defining a formal compatibility checking framework
Introduction (2): WS Compatibility

Exchanging messages ➔ conversation
Problem: WS Compatibility

- Services conversation involves data:
  - Considering only the exchanged messages types is insufficient ➔ need for considering data flow in the compatibility analysis.

- Web services can have temporal requirements:
  - Need for considering temporal constraints in the compatibility analysis
Problem: WS Compatibility

- A formal compatibility framework:
  - Exchanged messages
  - Data flow and constraints over data
  - Temporal constraints

- Basis for Web services composition
Outline

- Introduction
- Related work
- The model of Web services
- Compatibility checking
- Conclusion and perspectives
Compatibility regarding the WS conversations [Bordeaux-TES04, Benatallah-ER04]

- The framework considers only the supported sequences of messages.
- Do not consider the involved data flow
- Do not consider the requirements, such that:
  - Constraints over data
  - Temporal constraints
Compatibility regarding the temporal constraints (1): [Benatallah-CAISE05]

- Compatibility framework considers:
  - Sequences of exchanged messages
  - A particular form of temporal constraints:
    - An activation constraint over one transition
Compatibility regarding the temporal constraints (2): [Ponge-Inforsid04, Ponge-ER07]

- More expressiveness temporal constraints:
  - Period between exchanging messages

- The model does not consider:
  - Data flow and constraints over data

- Does not consider the possible dependencies between the temporal constraints
Outline

- Related work
- The model of Web services
- Compatibility checking
- Conclusion and perspectives
The model of Web services
[D.Berardi-VLDB05]

- Extended Web Service Timed Transition Systems EWSTTS:
  - Messages flow: WSDL
  - Operations: atomic process OWL-S
  - Data flow capabilities (parameters of messages, and input/output of operations)
  - Constraints over data
  - Temporal constraints

- Web services are modeled as deterministic finite state machines
The model

A Web service is a tuple $Q = (S, s_0, F, M, A, C, X, T)$ such that:

- $S$ is a set of states, $s_0$ is the initial state and $F \subseteq S$ is a set of final states
- $A$ is a set of actions
- $M$ is a set of messages
- $C$ is a set of constraints over data
- $X$ is a set of clocks
- $T \subseteq S \times (M \cup A) \times C \times \psi(X) \times 2^X \times S$
The model

- **Specification of temporal constraints:**
  - Standard timed automata clocks

**Flowchart:**
- **C₀** (Clock reset)
- **C₁** (y:=0)
- **C₂** (checkCCN(ccn;approved))
- **C₃** (Debit(ccn,price))
- **C₄** (If(result≠’ok’) !failure(‘the ccn is invalid’))
- **C₅** (y≤600s)
- **Sending a message**
  - !ticketPaid(ticket,ccn)
- **Performing an operation**
  - If(approved=‘ok’)
  - !ticketPaid(ticket,ccn)
- **Receiving a message**
  - ?ticketToPay(ccn)
- **Data constraint**
- **Payment by credit card number service**
  - z
  - Standard timed automata clocks
  - y ≤ 600s
Outline

- Related work
- The model of Web services
- Compatibility checking
- Conclusion and perspectives
Compatibility checking

- Two web services are incompatible if:
  1. One service waits for a message that is not sent by the other service
  2. The constraints over data associated to the input and output messages are not consistent, i.e., disjoint set of solutions
  3. The corresponding temporal constraints are not consistent, i.e., disjoint set of solutions
Compatibility checking

- The compatibility checking relies on:
  - Consistency of pairs (input message, output message):
  - Consistency classes of two transitions:
    - Absolute consistency
    - Likely consistency
    - Absolute inconsistency
Compatibility checking

1. Absolute consistency:
   ○ An output transition $t_1 = (s, !m, c_1, \psi_{X_1}, Y_1, s')$ is *absolutely consistent* with an input transition $t_2 = (q, ?m, c_2, \psi_{X_2}, Y_2, q')$ if:
     ○ $\text{Sol}(c_1) \subseteq \text{Sol}(c_2)$ and $\text{Sol}(\psi_{X_1}) \subseteq \text{Sol}(\psi_{X_2})$

2. Likely consistency:
   ○ The output transition $t_1$ is *likely consistent* with the input transition $t_2$ if:
     ○ $\text{Sol}(c_1) \cap \text{Sol}(c_2) \neq \emptyset$ and $\text{Sol}(\psi_{X_1}) \cap \text{Sol}(\psi_{X_2}) \neq \emptyset$
The two transitions are absolutely consistent

\[ \text{Sol}(d_1 < 10) \subseteq \text{Sol}(d_1 < 20) \]
\[ \text{Sol}(y_1 \leq 30) \subseteq \text{Sol}(x_1 \leq 60) \]

The two transitions are likely consistent

\[ \text{Sol}(d_1 < 20) \cap \text{Sol}(d_1 < 10) \neq \emptyset \]
\[ \text{Sol}(y_1 \leq 30) \cap \text{Sol}(x_1 \leq 60) \neq \emptyset \]
Compatibility checking

1. Absolute inconsistency:
   - The output transition $t_1$ is *absolutely inconsistent* with the input transition $t_2$:
     - $\text{Sol}(c_1) \cap \text{Sol}(c_2) = \emptyset$ or $\text{Sol}(\psi_{X_1}) \cap \text{Sol}(\psi_{X_2}) = \emptyset$

The two transitions are absolutely inconsistent
Compatibility checking

- The temporal constraints of some transitions can have an impact on other transitions of the same service:
- The local consistency of transitions is not sufficient:
- Check the dependencies between the temporal constraints ➔ Inference of external constraints
Compatibility checking

The two services are compatible!

The two services are not compatible!
Compatibility checking

- Reduction of the compatibility checking into the reachability problem:
  - Cartesian product

- Compatibility analysis based on the inference is less expensive w.r.t the reachability analysis:
  - Do not deal with automata product
Outline

- Related work
- The model of Web services
- Compatibility checking
- Conclusion and perspectives
Conclusion

- A compatibility framework:
  - Messages flow
  - Data flow
  - Operations
  - Constraints over data
  - Temporal constraints:
    - Implicit dependencies between the different temporal constraints
Perspectives

- Studying automatic tool for inferring external constraints
- Enhance the compatibility framework to build a composition.
Reference


Thank you!
Questions??