Verification of Non-functional and Functional Requirements of Web Service Composition

Manman Chen, Tian Huat Tan, Jun Sun, Yang Liu,
Pang Jun, Xiaohong Li

Outline

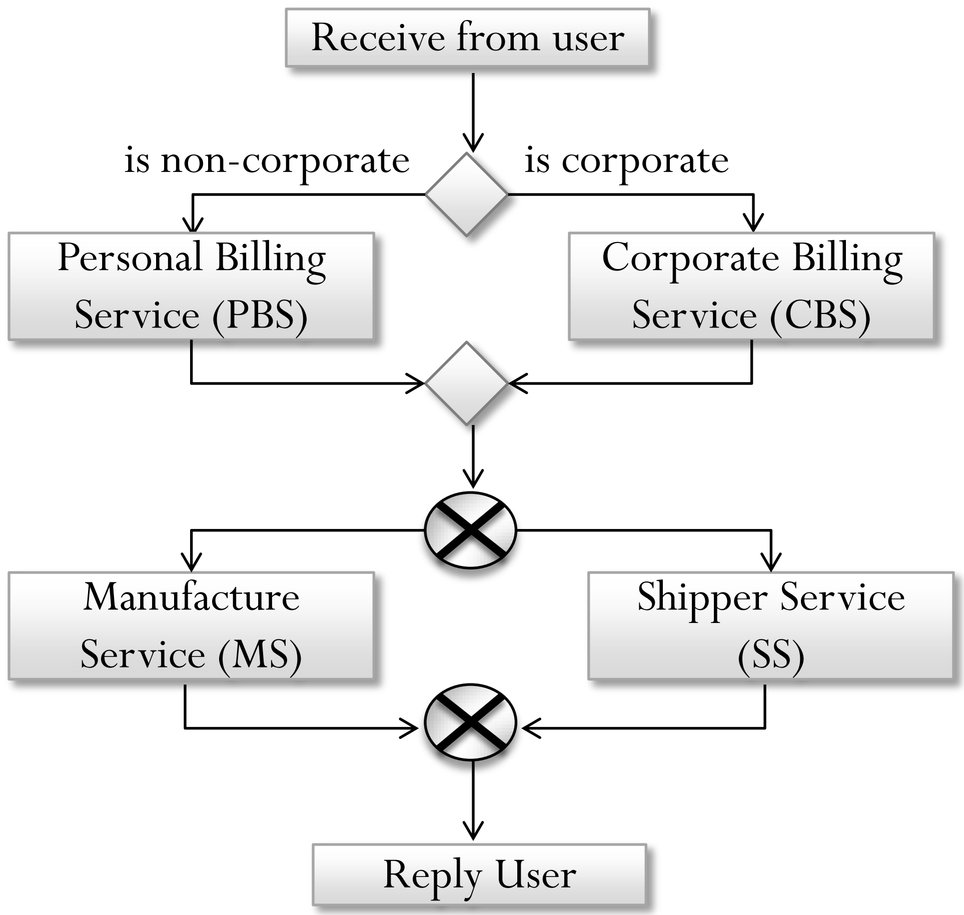
- Introduction
  - Service Composition
  - Functional and Non-Functional Requirements
- Problem Statement
- QoS Compositional Model
  - QoS Attribute Aggregation Function
  - Labeled Transition System
- Verification of Functional and Non-functional Requirements
- Experiments
- Conclusion

Introduction-
What is Service Composition

- A service composition makes use of existing service-based applications as components to achieve a business goal. The service that makes use of existing services to achieve a business goal is called **composite service**.

- Services that made use by a composite service, are called **component services**.

Introduction- What is Service Composition



Computer Purchasing Service (CPS)

Introduction-
Functional and Non-Functional Requirements

- Functional Requirements
  - the functionalities of the Web service composition
  - e.g., computer purchase service will always reply customers with the purchasing results

- Non-Functional Requirements
  - the Quality of Service (QoS)
  - i.e., response time, availability, cost,…
  - e.g., CPS replies to customers within 5s

Problem Statement

Given the non-functional properties for each component of Web service composition, verify the combined functional and non-functional requirements of the Web service composition.

QoS Aggregation Function

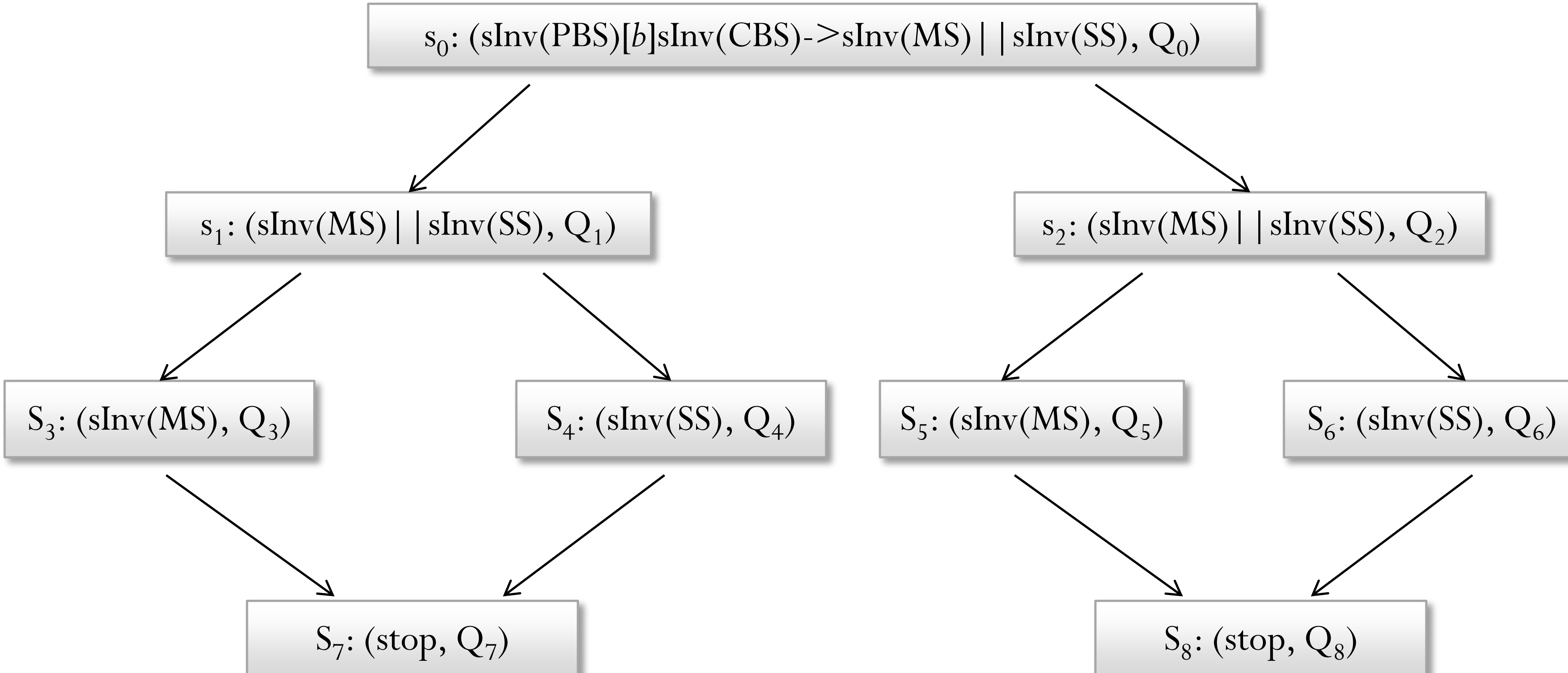
- BPEL syntax
  - **rec(S)** : receive from a service S
  - **reply(S)**: reply to a service S
  - **sInv(S)(aInv(S))**: synchronous (asynchronous) invocation of a service S
  - **P|||Q**: concurrent execution of P and Q
  - **P[b]Q**: conditional activity, where b is a guard condition. If b is evaluated as true, P is executed, otherwise, Q is executed.
- Aggregation Function

QoS Attribute	Sequential	Parallel	Loop	Conditional
Response Time	$\sum_{i=1}^{n} q(s_i)$	$\max_{i=1}^{n} q(s_i)$	$k * (q(s_1))$	$\max_{i=1}^{n} q(s_i)$
Availability	$\prod_{i=1}^{n} q(s_i)$	$\prod_{i=1}^{n} q(s_i)$	$(q(s_1))^k$	$\min_{i=1}^{n} q(s_i)$
Cost	$\sum_{i=1}^{n} q(s_i)$	$\sum_{i=1}^{n} q(s_i)$	$k * (q(s_1))$	$\max_{i=1}^{n} q(s_i)$

Labeled Transition System (LTS)

- **A system state is a tuple (P, V, Q), where**
  - P is the composite service process
  - V is a (partial) variable valuation
  - Q is a vector represents QoS attributes of the composite service (In this work, we consider response time, availability and cost attributes, thus Q =<r, a, c>)
- **An LTS is a tuple L = (S, s_0, Σ, →), where**
  - S is a set of states,
  - $s_0 \in S$ is the initial state,
  - Σ is a set of actions,
  - $\rightarrow: S \times \Sigma \times S$ is a transition relation.

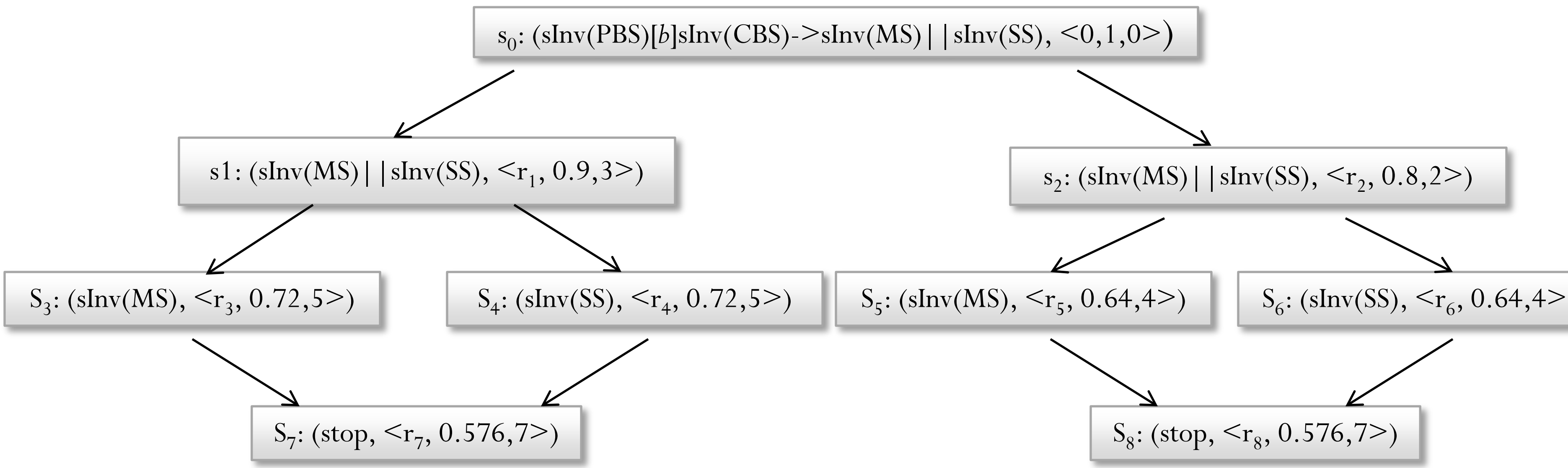
LTS of CPS



LTS of CPS

LTS Integrated with Cost and Availability

- s'.Q(availability) = s.Q(availability) * Availability(a)
- s'.Q(cost) = s.Q(cost) +Cost(a) where (s,a,s') ∈ →

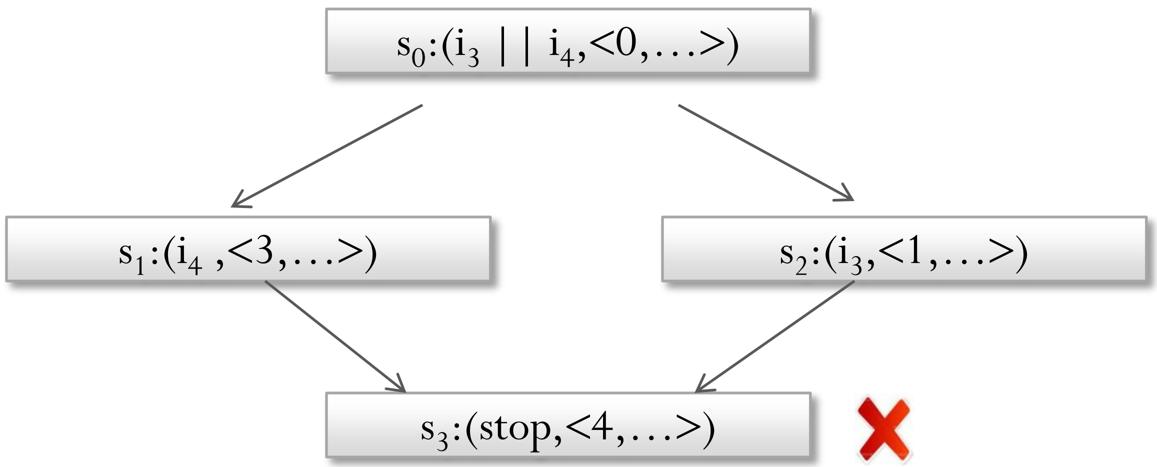


QoS Attribute	PBS	CBS	MS	SS
Response Time(ms)	1	2	3	1
Availability(%)	90	80	80	80
Cost($)	3	2	2	2

LTS Integrated with Response Time

- $s'.Q(r) = s.Q(r) + Responsetime(a)$ where $(s,a,s') \in \rightarrow$

e.g., $i_3 \,||\, i_4$ ✗ response time of i_3 and i_4 are 3s and 1s

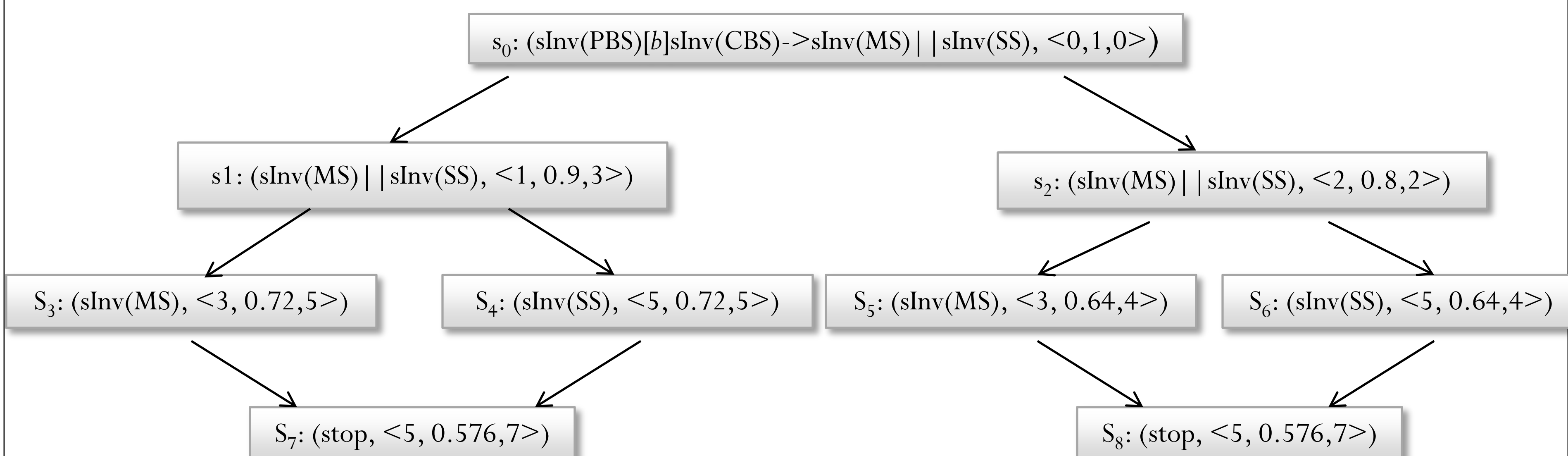


LTS Integrated with Response Time

- We propose the annotated approach for response time.

$$P_0' = [[[sInv(PBS)]^1 \triangleleft b \triangleright [sInv(CBS)]^2]^2 \rightarrow [[sInv(MS)]^5 || [sInv(SS)]^3]^5]^5$$

- The LTS of CPS is



Implementation in PAT

- Implemented in PAT
- PAT is available at http://www.patroot.com/
- 1M lines of C# code, 21 modules with 100+ build in examples
- Used as an educational tool in e.g. York Univ., Univ. of Auckland, NII (Japan), NUS
- 2000+ registered users from 400+ organizations in 52 countries and regions

Evaluation

- **Loan Service (LS):**

  provide users for applying loans

- **Travel Agency Service (TAS):**

  help users to arrange the flight, hotel and transport for a trip

- **Computer Purchasing Service (CPS)**

Services	Property	Result	#State	#Transition	Time(s)
CPS	(replyUser $\wedge$ (responseTime$>$5))	invalid	21	29	0.0087
	$\Box$ responseTime$\leq$5	valid	26	36	0.0089
	$\Box$ availability$>$0.6	valid	26	36	0.0083
LS	Reach (replyUser $\wedge$ (responseTime$>$6))	invalid	106	241	0.0584
	$\Box$ responseTime$\leq$6	valid	242	572	0.1866
TAS	Reach (replyUser $\wedge$ (responseTime$>$3))	invalid	128	287	0.0631
	$\Box$ responseTime$\leq$3	valid	264	622	0.0642
	Reach (replyUser $\wedge$ (availability$\leq$0.3))	invalid	128	287	0.0437

Conclusion and Future Work

- **Conlusion**
  - We propose an approach for verification on combined functional and non-functional properties for Web service directly based on its semantics directly.
  - Our approach has been implemented and evaluated on the real-world case studies, and this demonstrates the effectiveness of our method.
- **Future Work**
  - Apply state reduction techniques to improve our approach
  - Extend our work to other domains, e.g., sensor network

Q&A