Formal Analysis of Behavioral Exchangeability of Services

Jarungjit Parnjai
Supervisor : Prof. Dr. Wolfgang Reisig
in METRIK since : 2008-06-01

Motivation

A distributed system in disaster management consists of a collection of heterogeneous computational entities. These entities interact asynchronously and locally without centralized control, to achieve an emerging global system behavior. Such a system can be modeled as a network of multiple services, in which each entity is a service that represents a work unit in disaster management.

A service in such a context is subject to *changes* according to various scenarios. For instance, the behavior of a service shall be adjusted in order to respond to emerging tasks, or there is a requirement to accommodate a completely new service into the system due to unforeseen circumstances.

To respond promptly to various changes, the behavior of a service must be adaptable. To this end, the original service will exchanged with its behaviorally adapted version. It is crucial that such an update must neither disrupt the interaction with any other entity in the system nor produce an unfavorable effect to the entire system.

Problem statement

Consider a service S interacting deadlock-freely with its partner (potentially an environment of S). A *well-designed update* of S is a service S' that preserves every deadlock-free partner of S whereas an *ill-designed update* of S is a non *well-designed update* of S.

Two *Behavioral Exchangeability* notions, each of which guarantees a *well-designed update* of a service, have been considered :

1) *Behavioral Equivalence* - two services S and S' have exactly the same set of deadlock-free partners,
2) *Behavioral Accordance* - every deadlock-free partner of a service S is also a deadlock-free partner of a service S'.

Regarding these two notions, the following questions are investigated in this dissertation :

- Given S and S', is S' a *well-designed update* of S?
- Given S, how to correct an *ill-designed update* S' of S?
- Given S, how to synthesize a *well-designed update* S'' of S w.r.t. a given requirement?

The major goal of this dissertation is to develop a systematic approach to answer all questions listed above. This shall be helpful for a service designer, e.g. a domain expert, to rapidly and effectively administrate the changing behaviors of a service.

Approach

For each exchangeability notion, a finite representation S^*, which compactly represents the (possibly infinite) set of all *well-designed updates* of a given service S, shall be developed. With the finite structure S^*, one can do these followings:

- efficiently decide if a service S' is a *well-designed update* of S (decision algorithm),
- give the best optimal suggestion on how to correct an *ill-designed update* of S (correction algorithm), and
- synthesize a *well-designed update* of S from S^* (synthesis algorithm).

Current and Future work

Regarding *accordance*, S^* as well as its decision algorithm and correction algorithm have successfully been developed [1]. A synthesis algorithm is currently under development using an accordance-preserving transformation system, which consists of a canonical representative of S generated from S^* and two different sets of accordant-preserving transformation rules.

Regarding *equivalence*, S^* is also currently under development. The challenge thereby is to find necessary and sufficient conditions for behavioral equivalence, as the behavioral equivalence does not coincide with classical equivalence notions, such as trace equivalence or bisimulation. A decision algorithm, a correction algorithm, and a synthesis algorithm shall be developed as a future work. Nevertheless, the development is planned under a common framework as those regarding accordance.

REFERENCES

[1] Parnjai, J., Stahl, C., Wolf, K.: *A finite representation of all substitutable services and its applications*. In Proceedings of the 1st Central-European Workshop on Services and their Composition, ZEUS 2009, Stuttgart, Germany, March 2009.