

Report July 10th

PCD definition and Implementation

Report about the problems found in the current implementation, solutions proposed, and new version of the PCD including SLA.

1. Modifications in the implementation

Problem 1: More than one PCD were created to the same concrete service. **Solution:** after creating the PCDs, a new method is called to select PCDs for the same concrete service and create one single PCD. This new PCD has a set of abstract services on its coverage domain. **Result:** as expected just one PCD is created to a concrete service.

Problem 2: Null pointer exception while trying to rewrite the query. That happened because the old implementation considers that each PCD has only a single mapping between one abstract service in the concrete service definition and one abstract service in the query. **Solution:** a list of mappings called 'phi' was added in the PCD attributes. The entire code was traced to find places where the old mapping is used, and then changed to check all the list of mappings instead of only one single mapping. **Result:** The rewriting answers are OK, but the parameters are wrong.

Problem 3: more parameters than expected are in the rewriting. That happened because of the loops to read the list of mappings between variables added in the problem before. **Solution:** bug correction in the set query method, checking if one mapping a variable is found we do not have to search for more mappings. **Result:** now, the rewriting is OK.

2. Updating PCD definition

Based on this former PCD definition, the extension to incorporate SLA measures is a tuple $\langle S, SLA_s, Qpref, h, dep, \varphi, G, Def, has_opt \rangle$, where:

S is a concrete service;

SLA_s is a set of tuples $\langle measure, value, predicate \rangle$ associated to the concrete service S ;

h are mapping between terms in the head of the service definition and terms in the right side of the service definition;

dep is a set of abstract services in the definition of S that are inter-dependents;

φ are mapping between terms from the abstract composition to terms in the concrete service definition;

G is a set of abstract services names and quality constraints covered by S ;

Def is a set conditions that are not covered by S alone; and

has_opt is a boolean flag to indicate that some abstract service in the definition of S has been used in G and has an optional parameter.