Appendix A. Formal Model of Service-Oriented System Design - Definitions

The model captures the structure of SO system as a bi-directional graph with its sub-graphs representing constituent services. This graph is then expressed using standard set theory notation as a pair \( <E, R> \), where \( E \) symbolizes the set of elements of \( SOS \), and \( R \) is a binary relation on \( E (R \subseteq E \times E) \) representing the relationships between the elements of \( SOS \). The subset of the model used in the definitions of metrics is summarised in this appendix, the complete model can be found in [M. Perepletchikov, C. Ryan, K. Frampton, and H. Schmidt, "Formalising Service-Oriented Design," Journal of Software (JSW), vol. 3, pp. 1-14, 2008].

**Definition 1** (SO System (SOS))

\[
SYS = <SI, BPS, C, I, P, H, R>,
\]

where \( SI \) is a set of all service interfaces; \( BPS \) is a set of all business process scripts; \( C \) is a set of all OO classes; \( I \) is a set of all OO interfaces; \( P \) is a set of all procedural packages; \( H \) is a set of all package headers that exist in the system; and \( R \) is the set of all common and possible relationships between various design artefacts.

**Definition 2** (Service (s))

\[
s = <si_s, BPS_s, C_s, I_s, P_s, H_s, R_s>, \text{ is a service of } SOS \text{ if and only if } si_s \in SI \land (BPS_s \subseteq BPS \land C_s \subseteq C \land I_s \subseteq I \land P_s \subseteq P \land H_s \subseteq H) \land (BPS_s \cup C_s \cup I_s \cup P_s \cup H_s \cap s) \land R_s \subseteq R.
\]

Note that \( \cap s \) symbol represents service membership, where the allocation of elements to services is performed by considering the possible call paths in response to invocations of operations exposed in a service interface.

**Definition 3** (operations of an element)

For each element \( e \in SI \cup BPS \cup C \cup I \cup P \cup H \) let \( Op(e) \) be the set of operations \( op \) of element \( e \). In addition, operations can be defined individually to cover for the specific element types. For example, for each service interface \( si \in SI \) let \( SOp(si) \) be the set of service operations \( sop \) of service interface \( si \).

**Definition 3.1** (operation parameters, return type, and pre- and post-conditions)

i) Operations can have (optional) input parameters, which can be formally defined as: For each operation \( op \in Op(e) \) let \( Param(op) \) be the set of parameters \( par \) of \( op \); ii) Operations can have (optional) return type, which can be formally defined as: For each operation \( op \in Op(e) \) let \( returnType_{op} \) be the return type of \( op \); and iii) Operations can have (optional) pre- and post-conditions, which can be formally defined as: For each operation \( op \in Op(e) \) let \( Cond(op) \) be the set of pre- and/or post-conditions \( cond \) of \( op \).

Note that as was the case with the operations (Definition 3), input parameters, return types, and pre- and post-conditions can be defined to cover for the specific operation types.

**Definition 4** (attributes of an element)

For each element \( e \in SI \cup BPS \cup C \cup I \cup P \cup H \) let \( Atr(e) \) be the set of attributes \( atr \) of element \( e \).

**Definition 5** (collaboration relationships between service-oriented design entities)

A collaboration \( c_{op} \) captures elements that interact in order to achieve some desired functionality in response to all possible invocations of operation \( op \) belonging to some element \( e \).

Formally: \( c_{op} \in CO(e) = <Param(op \in Op(e)), CS> \), where \( CO(e) \) is a set of all collaborations of element \( e \); and \( CS \) is the set of collaboration sequences \( cs_{op} \in Op(e) \).

A collaboration sequence captures the set of interacting elements that achieve functionality exposed in operation \( o \) based on specific inputs (i.e. parameter values) and can be defined as: \( cs_{op \in Op(e)} \in CS(e) = <SI_{cs}, BPS_{cs}, C_{cs}, I_{cs}, P_{cs}, H_{cs}> \), where \( SI_{cs} \subseteq SI \), \( BPS_{cs} \subseteq BPS \), \( C_{cs} \subseteq C \), \( I_{cs} \subseteq I \), \( P_{cs} \subseteq P \), \( H_{cs} \subseteq H \). This represents the set of interacting elements that achieve functionality exposed in operation \( o \) based on specific inputs. In terms of graph theory notation, collaboration sequence \( cs_{op \in Op(e)} \) represents an open or closed walk starting at element \( e \).