

Specification Patterns from Research to Industry: A Case Study in Service-Based Applications

[Extended Abstract]

Domenico Bianculli¹, Carlo Ghezzi², Cesare Pautasso³, and Patrick Senti⁴

¹SnT Centre - University of Luxembourg, Luxembourg, Luxembourg
domenico.bianculli@uni.lu

²DEEPSE group - DEIB - Politecnico di Milano, Milano, Italy
carlo.ghezzi@polimi.it

³Faculty of Informatics - University of Lugano, Lugano, Switzerland
cesaire.pautasso@usi.ch

⁴formerly with Credit Suisse AG, Zürich, Switzerland
patrick.senti@gmail.com

Specification patterns [DAC98] have been proposed as a means to express recurring properties in a generalized form, allowing developers to state system requirements precisely and map them to specification languages like temporal logics. The majority of past work has focused on the use of specification patterns in the context of concurrent and real-time systems, and has been limited to a research setting. In this presentation we report the results of our study [BGPS12] on the use of specification patterns in the context of service-based applications (SBAs); the study focused on industrial SBAs in the banking domain. The study collected and classified the requirements specifications of two sets of case studies. One set consisted of 104 cases extracted from research articles in the area of specification, verification and validation of SBAs published between 2002 and 2010. The other set included 100 service specifications developed by our industrial partner for its service-oriented information system over a similar time period. During the study, each requirement specification was matched against a specification pattern; in total, we analyzed and classified 290 + 625 requirements specifications from research and industrial data, respectively. The requirements specifications were classified according to four classes of property specification patterns. Three of them correspond to the systems of specification patterns proposed by Dwyer et al. [DAC98], by Konrad and Cheng [KC05], and by Gruhn and Laue [GL06]; these patterns have been widely used for the specification and verification of concurrent and real-time systems. The fourth group includes patterns that are specific to service provisioning and have emerged during the study; they are:

Average response time (S1) is a variant of the bounded response pattern [KC05] that uses the average operator to aggregate the response time over a certain time window.

Counting the number of events (S2) is used to express common non-functional requirements such as reliability (e.g., “number of errors in a given time window”) and throughput (e.g., “number of requests that a client is allowed to submit in a given time window”).

Average number of events (S3) is a variant of the previous pattern that states the average number of events occurred in a certain time interval within a certain time window, as in “the average number of client requests per hour computed over the daily business hours”.

Maximum number of events (S4) is a variant of pattern S3 that aggregates events using the maximum operator.

Absolute time (S5) indicates events that should occur at a time that satisfies an absolute time constraint, as in “if the booking is done in May, a discount is given”.

Unbounded elapsed time (S6) indicates the time elapsed since the last occurrence of a certain event.

Data-awareness (S7) is a pattern denoting properties that refer to the actual data content of messages exchanged between services as in “every ID present in a message cannot appear in any future message”.

The study showed that: a) the majority of requirements specifications stated in industrial settings referred to specific aspects of service provisioning, which led to the definition of the new class of specification patterns; b) the specification patterns proposed in the research literature [DAC98, KC05, GL06] were barely used in industrial settings.

Furthermore, the new class of specification patterns led to the definition of a new specification language able to express them; the language, introduced in [BGS13], is called SOLOIST (*SpecificatiOn Language fOr service compoSitions inTeractions*) and is a many-sorted first-order metric temporal logic with new temporal modalities that support aggregate operations on events occurring in a given time window.

Acknowledgements. This work has been partially supported by the European Community under the IDEAS-ERC grant agreement no. 227977-SMScom and by the National Research Fund, Luxembourg (FNR/P10/03).

References

- [BGPS12] Domenico Bianculli, Carlo Ghezzi, Cesare Pautasso, and Patrick Senti. Specification Patterns from Research to Industry: a Case Study in Service-based Applications. In *Proc. of ICSE 2012*, pages 968–976. IEEE Computer Society, 2012.
- [BGS13] Domenico Bianculli, Carlo Ghezzi, and Pierluigi San Pietro. The Tale of SOLOIST: a Specification Language for Service Compositions Interactions. In *Proc. of FACS'12*, volume 7684 of *LNCS*, pages 55–72. Springer, 2013.
- [DAC98] Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Property specification patterns for finite-state verification. In *Proc. of FMSP '98*, pages 7–15. ACM, 1998.
- [GL06] Volker Gruhn and Ralf Laue. Patterns for Timed Property Specifications. *Electron. Notes Theor. Comput. Sci.*, 153(2):117–133, 2006.
- [KC05] Sascha Konrad and Betty H. C. Cheng. Real-time specification patterns. In *Proc. of ICSE '05*, pages 372–381. ACM, 2005.