Abstract—With the increasing complexity and stringent requirements of modern large-scale distributed systems, well-structured representations of software design knowledge arise as a promising approach to keep delivering high quality products in a timely and cost-effective way. Although domain-specific architecture styles and reference architectures help in conveying such design knowledge, the lack of systematic and structured representations make it hard to grasp design alternatives promptly and support well-informed trade-off analysis. This short paper presents DuSE-MT—a supporting tool for the DuSE approach to architectural design of self-adaptive systems. DuSE-MT implements: i) a generic metamodel for systematic representation of design spaces (DuSE), which enables automated architecture design and analysis; ii) a specific design space for the self-adaptive systems domain (SA:DuSE); iii) a set of metrics that capture quality attributes of resulting self-adaptive architectures; and iv) a multi-objective optimization approach to explicitly elicit a trade-off decision by finding out a set of Pareto-optimal candidate architectures. Our approach has been evaluated in a case study involving self-adaptive cloud-based services.

I. INTRODUCTION

Designing architectures for software-intensive systems that fulfill complex quality attributes and operate in highly unpredictable environments is a challenging task [1]. The well-orchestrated use of refined experience, good aesthetics, and right balance when making trade-off decisions is imperative if we are to design effective architectures for complex large-scale systems. Furthermore, developing systems for specific application domains ultimately yields a body of tacit design knowledge that would be of paramount value if systematically structured for routine reuse.

In particular, the design of effective architectures for self-adaptive systems requires architects become proficient in evaluating complex solution spaces [1]. Relevant approaches scattered over different research communities and solutions effective only in specific scenarios make it harder to understand generic architecture-centric approaches to self-adaptation.

In this paper, we present DuSE-MT¹—a supporting tool for the DuSE approach to automated architectural design and analysis of self-adaptive systems, presented in [2]. DuSE-MT fully implements: i) a generic metamodel for systematic representation of design spaces (DuSE), which enables automated architecture design and analysis; ii) a specific design space for the self-adaptive systems domain (SA:DuSE); iii) a set of metrics that capture quality attributes of resulting self-adaptive architectures; and iv) a multi-objective optimization approach to explicitly elicit a trade-off decision by finding out a set of Pareto-optimal candidate architectures.

¹http://duse.sf.net

A. Challenges when Designing Self-Adaptive Systems

Although a lot of current research have been devoted to the use of feedback control theory as the enabling mechanism for self-adaptation, it seems hard—even for experienced architects—to understand and evaluate trade-offs among currently available approaches. The variety of solutions available in many design dimensions rapidly opens up a huge space of candidate architectures for self-adaptive behavior. In particular, the following challenges are frequently faced when designing self-adaptive systems:

C1) Highly specialized domain and large design/solution space: deciding about a specific system modeling approach, control law, tuning method, and arrangements of feedback loops, heavily impacts not only system’s self-adaptation robustness and control overhead, but also common quality attributes such as scalability, flexibility and maintainability. Currently, self-adaptation is a mean to meet several different goals and a number of mechanisms for feedback control are available. It is very likely that novice architects adopt inferior architectures or be unaware of alternative solutions.

C2) Lack of support for early reasoning of quality properties: as self-adaptation capabilities become essential means to meet business goals, it is highly desirable that final product’s quality attributes can be evaluated/estimated at early development stages. Since architectural choices directly impact such attributes, it is reasonable that systematic representations of design knowledge for self-adaptive systems also include mechanisms for early quality attributes assessment. Such mechanisms would set the stage for well-informed trade-off decisions and help architects to easily grasp the consequences of choosing a particular candidate architecture.

C3) Lack of effective generic solutions: although a number of noteworthy mechanisms for self-adaptation have been proposed over the past years, they are usually tailored to very specific scenarios, which make it harder to reuse their solutions in other applications. Whilst optimization and configuration for the problem at hand are commonly required,
generic architecture-centric approaches may further leverage the adoption of self-adaptive systems in industry and the establishment of a more disciplined application domain.

II. DuSE-MT

Motivated by the aforementioned challenges when designing self-adaptive systems, we have designed DuSE – a search-based approach for domain-independent representation of design spaces and automated architecture design and analysis. A specific DuSE metamodel instance (SA:DuSE) systematically captures prominent degrees of freedom when designing self-adaptive systems. SA:DuSE enables the use of DuSE’s architecture optimization features to find out proper architectures for a specific self-adaptation problem at hand.

A. Applied Tactics in DuSE Design

In early stages of DuSE design, the following tactics were used to address some of the aforementioned challenges:

T1) Focus on explicit representation of feedback loops in architectural models described in standardized notations: our solution relies on the use of MOF, UML and XMI standards for architecture representation. The SA:DuSE design space instance defines a UML Profile that provides major stereotypes and tagged values for representing process components, controllers, sensors, actuators, and filters.

T2) Capture most prominent degrees of freedom as systematically structured design spaces: SA:DuSE specifies solutions (variation points) for the most prominent design dimensions for self-adaptive systems. Each variation point defines a set of changes that endow an initial non-adaptive system – provided as input to DuSE-MT – with such specific solution for self-adaptation. That mechanism set the stage for an automated approach to self-adaptive systems architecture design [3].

T3) Support the use of flexible architecture quality metrics: each candidate architecture resulting from the automated design space navigation is evaluated in terms of four quality metrics related to self-adaptation (control overhead, settling time, maximum overshoot and control adaptation).

T4) Enable automatic search for (near-)optimal solutions in huge design spaces: the design space representation and the quality metrics enable the use of multi-objective optimization approaches [4] to automatically find out (near-)optimal candidate architectures. DuSE-MT uses an elitist evolutionary approach to explicitly elicit a trade-off decision in terms of a set of Pareto-optimal architectures. It is up to the architect to apply some form of a posteriori preference articulation to pick out the ultimate architecture to be implemented.

B. Features and Architecture

Currently, DuSE-MT provides the following features: i) basic creation/manipulation of models described in MOF and UML languages, ii) creation of specific design spaces instances and quality metrics described in DuSE language, iii) run-time model manipulation via JavaScript, and iv) architecture optimization of input models with respect to a given design space instance. The DuSE’s multi-objective optimization approach was built on top of the NSGA-II evolutionary algorithm [4].

SA:DuSE is a specific design space instance for self-adaptive systems and also an accompanying UML Profile that defines basic annotations for input models. Such annotations provide the information required to identify the loci of architectural decisions in the input model, replicate design space’s dimensions for each applicable locus, and support manual as well as automated design space exploration. DuSE-MT defines a flexible internal architecture where new metamodels and design space instances are seamlessly accommodated as plug-ins. Furthermore, making architectural models available for manipulation via JavaScript facilitates the definition of new DuSE instances, devoted to other application domains. DuSE-MT has successfully been used in a case study involving self-adaptive cloud-based services, as described in [2].

III. Conclusion and Future Work

This short paper presents an overview of DuSE-MT: a supporting tool for automated architectural design in specific application domains. We have presented its major requirements, current available features, and some structural aspects of its architecture. Future work includes the implementation of a posteriori preference articulation in order to pick a single architecture out from those ones found in the Pareto-optimal set, sharing DuSE artifacts over a network, and mapping from design space navigation traces to higher level design theories.

REFERENCES


