A Semantic Approach to Deploying Product-Service Systems

Robert Andrei Buchmann, Business Informatics Research Center, Babeș-Bolyai University, Cluj Napoca, Romania
Dimitris Karagiannis, Knowledge Engineering Research Group, University of Vienna, Vienna, Austria

ABSTRACT
Conceptual modeling is commonly employed for two classes of goals: (1) as input for run-time functionality (e.g., code generation) and (2) as support for design-time analysis (e.g., in business process management). An inherent trade-off manifests between such goals, as different levels of abstraction and semantic detail is needed. This has led to a multitude of modeling languages that are conceptually redundant (i.e., they share significant parts of their metamodels) and a dilemma of selecting the most adequate language for each goal. This article advocates the substitution of the selection dilemma with an approach where the modeling method is agilely tailored for the semantic variability enabled by such a method is exposed to model-driven systems as RDF knowledge graphs, whereas the method evolution is managed with the Agile Modeling Method Engineering framework. The argument is grounded in the application area of Product-Service Systems, illustrated by a project-based modeling method.

KEYWORDS
Agile Modeling Method Engineering, Domain-Specific Modelling Method, Knowledge Graphs, Metamodeling, Model-Aware Application, Product-Service Modeling, Resource Description Framework, Semantic Queries

1. INTRODUCTION
The paper proposes a coupling between the Agile Modeling Method Engineering framework (Karagiannis, 2015) and the “modeling as knowledge representation” principle in order to achieve the flexibility required for addressing, with the same modeling language, both run-time and design-time concerns pertaining to the deployment of Product-Service Systems. In a more general sense, the proposal aims to defuse the traditional dichotomy between (1) modeling methods for software engineering (i.e., driven by “run-time concerns”) and (2) those tailored to support domain-specific business views, sense-making and communication among business stakeholders (i.e., addressing “design-time concerns”).

These classes of concerns typically require different levels of abstraction, detail or formal rigor – e.g., a modeling language designed to generate executable code may not fulfill the sense-making and communication requirements of stakeholders looking for high-level enterprise architecture analysis or process costing. The dichotomy stems from the fixed scope of most
modeling languages, as well as the assumption that modeling requirements are stable and a language that fulfills them can be selected from some established catalogue. Such a perspective is based on the blueprint thinking mentioned in (Loucopoulos & Kavakli, 2016), whereas the work at hand advocates modeling agility manifested not only at contents level, but also at method level (in language syntax, semantics, implementation). This type of agility, as enabled by the Agile Modeling Method Engineering (AMME) framework, ensures that a sufficiently comprehensive, “on-demand” semantic space is made available for heterogeneous modeling goals, covering both design-time and run-time requirements.

The application area of Product-Service Systems (PSS) will be used to ground the argumentation in project-based examples, tackling challenges that originated in the ComVantage EU research project (ComVantage, 2017; Open Models, 2017) and are currently being further developed in the successor project EnterKnow (EnterKnow, 2017). “Product-Service Systems” is a term associated with a hybrid business model aiming to sustain both consumption and production, thus jointly fulfilling customer needs through a mix of products and services (Halen et al., 2005). The key motivators of PSS are better targeted customer-orientation with improved responsiveness in manufacturing and a growing culture of virtualization that manifests as “product servitization” – i.e., products and services merging under the more abstract terms of functions or values. Satisfaction is achieved by selling the function of a product rather than the product itself, typically by enriching it with a suite of associated services and customizations that contribute to its value.

One key challenge for PSS modeling methods is to capture the heterogeneous and domain-specific semantic space that is necessary to satisfy the abovementioned classes of concerns: (1) run-time concerns, pertaining to the engineering of PSS features that are needed at run-time (for current operations); (2) design-time concerns, pertaining to system analysis and decision-making. The rationale for selecting PSS as an application domain for AMME is manifold, considering the following aspects: (1) the origin of the work in the mentioned project context, dealing with PSS modeling; (2) the lack of standard languages for PSS, commonly addressed by conveniently interpreting more abstract conceptualizations (from service science or enterprise architecture management); (3) domain-specificity considering product-service characteristics that might evolve and should be agilely incorporated in the modeling concepts.

While AMME is employed to design a common semantic space (and implement the corresponding modeling tool) a complementary mechanism must make the tailored semantics available to the model-driven functionality addressing the two categories of concerns. This is achieved with the help of the Resource Description Framework (RDF) (W3C, 2017a) employed as a knowledge representation and distribution medium. By coupling AMME and RDF, instead of having the modeling language impose what semantics will be available to model-driven systems (as in code generation scenarios), the flow of requirements and knowledge is reversed – model-driven systems (addressing both run-time and design-time concerns) may raise requirements that the modeling language should agilely fulfil.

The remainder of the paper is structured as follows: Section 2 will formulate the problem statement, will provide background both on the AMME methodologies and on related works. Section 3 will present an agile modeling method for Product-Service Systems developed within the AMME framework. A proof-of-concept implementation and the way it supports both run-time and design-time concerns are discussed in Section 4. Evaluation aspects are discussed in Section 5. The paper ends with conclusions and an outlook to future work.
Enforcing Modeling Guidelines in an ORDBMS-based UML-Repository
www.igi-global.com/chapter/enforcing-modeling-guidelines-ordbms-based/27833?camid=4v1a

Innovating Healthcare through Remote Monitoring: Effects and Business Model
www.igi-global.com/article/innovating-healthcare-through-remote-monitoring/144814?camid=4v1a