Cloud-Oriented Software Migration– A Model-Driven Approach[⋆]
(PhD Research Proposal)

Alexander Bergmayr

Business Informatics Group, Vienna University of Technology, Austria
bergmayr@big.tuwien.ac.at

Cloud computing offers new possibilities to improve the quality of software as resources, such as processing power and storage, platforms, and software, are viewed as commodities that are readily available at the scale of large data centers operated by cloud providers. To take full advantage of novel offerings (e.g., advanced scalable data persistence solutions) requires that the software is prepared for a cloud environment. This, in turn, implies that the software needs to comply with the peculiarities imposed by current cloud environments that might otherwise hinder its proper operation (e.g., stateful components in a highly scalable cloud environment). Still, migrating existing software to a cloud environment as a means of modernization appears to be valuable, thereby replacing the on-premise environment partly or even completely by a cloud environment. As the typical migration scenario requires to ”wire” both environments [1], the cloud provider and the cloud consumer perspective needs to be addressed [4].

Problem Statement and Relevance. Model-Driven Engineering (MDE) can play a major role in this respect not only to capture the necessary domain knowledge and to provide abstractions over diverse environments in terms of models, but also to support the shift from on-premise environments to cloud environments and between cloud environments through rigorous model transformation techniques [3]. However, to render such an MDE approach useful, three main challenges need to be overcome as depicted in Figure 1: *(C1) Code Injection & Extraction*, as apparently two heterogeneous technical spaces—programming and modeling—are involved, *(C2) Model Understanding*, as comprehensive model views tend to be orthogonal to the levels of abstraction, and *(C3) Model Modernization*, as high-level model views need to be anchored to the cloud computing domain and refined to a specific cloud environment.

Planned Contributions. In particular, three main contributions are expected by the proposed MDE approach that correspond to the identified challenges:

Metamodel Generation and Translators between Code and Model Level. A prerequisite for applying MDE techniques in the software migration is the availability of models that ideally conform to a metamodel. In a first step, legacy code is translated into Platform-Specific Models (PSMs) that comprise all specifics imposed by the platform the legacy software is built on. Clearly, the corresponding metamodels are required to facilitate model transformation techniques for coping with the understanding and modernization of models. The idea is to automatically generate metamodels from grammars by combining a translational approach with by-example techniques to reduce the manual effort and improve the quality compared to a sole translational approach.

[⋆] This work is co-funded by the European Commission under the ICT Policy Support Programme, grant no. 317859.

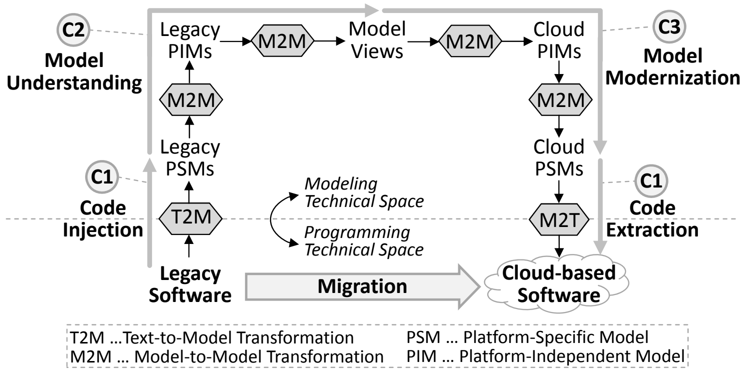

Fig. 1: Conceptual Overview of Cloud-Oriented Software Migration

Generic Transformations for Model Lifting and Model Slicing. To cover a wide range of migration scenarios requires to abstract from platform specifics. By lifting *Legacy PSMs* into higher-level Platform-Independent Models (PIMs), allows turning the focus on core software artifacts and hiding specifics of, e.g., the software runtime environments and data management capabilities. To facilitate good understanding of a selected part of the legacy software that is modernized, slicing techniques allow generating model views from *Legacy PIMs*. The idea is to employ UML to support such model views for software-, platform-, and infrastructure-related artifacts and provide generic model transformations that are customizable to gain dedicated model views.

Cloud Modeling Language and Cloud Optimization Patterns. Anchoring model views to the cloud computing domain requires modeling support and ideally guidance to refine them to cloud-specific models. The idea is to provide a UML-based model library that allows describing *Cloud PIMs* independent from cloud provider specifics. To keep such specifics separate from the model library, the idea is to develop UML Profiles for selected cloud providers. They are essential to gain *Cloud PSMs*, from which code is extracted. Regarding cloud-based optimizations, the idea is to capture them in terms of patterns that are operationalized by model transformations. They are defined on the model library and parameterized by stereotypes of the UML Profiles.

Current Status. Currently, the focus is on the development of the UML-based cloud modeling library and UML Profiles for selected cloud providers. First prototypes are available for generating metamodels from grammars and demonstrating the feasibility of an automated transformation chain for cloud-oriented software migration in a Java testbed. The presented research work is carried out in course of the ARTIST project [2] that provides real-world migration scenarios for evaluating the planned contributions.

References

1. Andrikopoulos, V., Binz, T., Leymann, F., Strauch, S.: How to Adapt Applications for the Cloud Environment. Computing 95(6), 493–535 (2013)
2. Bergmayr, A., Bruneliere, H., Cánovas Izquierdo, J.L., Gorroñogoitia, J., Kousiouris, G., Kyriazis, D., Langer, P., Menychtas, A., Orue-Echevarria Arrieta, L., Pezuela, C., Wimmer, M.: Migrating Legacy Software to the Cloud with ARTIST. In: CSMR. pp. 465–468 (2013)
3. France, R., Rumpe, B.: Modeling for the cloud. SoSyM 9(2), 139–140 (2010)
4. France, R., Rumpe, B.: The Evolution of Modeling Research Challenges. SoSyM 12(2), 223–225 (2013)