Case study: corresponding analysis

Purpose. We examined if its method fragments in our MLSAC metamodel are consistent and corresponded with real-world scenarios. This would ascertain if the theories, assumptions, and casual relations underlying the metamodel are valid and conform real-world reengineering process scenarios

Evaluation procedure. We used the tracing technique [1],[2] to identify the origin of activities that were performed in real-world scenarios to the corresponding ones in MLSAC metamodel method fragments. This indicates the extent to which the semantic of these scenarios are preserved by MLSAC. That is, method fragments in MLSAC, as the source model positioned at M2-level, are instantiated to express the activities that were defined in real-world reengineering processes, as the target model positioned at M1-level, which is a vertical model transformation according to the MOF framework. As each scenario has its own scope and focus, not all of MLSAC method fragments were required to use to represent the target reengineering process model. We thoroughly reviewed the available documents of real-world migration scenarios and attested how their activities can be derived from appropriate MLSAC's method fragments. In this thematic analysis, we used some leading questions to identify activities in each scenario to compare with related ones MLSAC, for example, deliverables were generated in the scenarios.

Accenture case: migrating full legacy application stack to cloud

We represented the reengineering process of Accenture which a leading global professional IT service provider company. The model includes five phases *Discovery*, *Assessment, Planning*, *Migration* and *Testing* [3]. The instantiation of a metamodel fragments at the M2-level creates a fragment object at M1-level (Figure 1). The highlighted elements in the figure are the instantiated fragments from the metamodel.

The instance of *Recover legacy system knowledge* in the *Plan* phase of MLSAC can be used to describe *Discover inventory applications* and *Capture metadata* in the Accenture. These activities identify stats such as the legacy application frequency usage, user profiles, architecture, workload, dependencies to other platforms, and programming languages used to implement components. The *Assessment* phase, centred on selecting a target cloud platform, is multiple inheritances of *Plan* and *Design* phases in MLSAC. Accenture defines a task named *Establish assessment framework* which identifies the most appropriate target cloud deployment options such as public, private or hybrid and an appropriate level of application remediation. This is an instance of the *Analyse context* in MLSAC. The *Assessment* phase in Accenture also defines activities *Determine target cloud vendor* and *Determine application disposition* regarding critical factors such as security and expected scalability. A comparison shows these activities are, respectively, producible by instantiating *Choose cloud provider* and *Define plan* in the *Plan and Design* phases of MLSAC. On the other hand, *Design cloud solution* in MLSAC is to identify application components appropriate for reengineering and their new distribution model on cloud servers. Instantiation of this method fragment generates *Determine cloud deployment and service* and *Design future architecture* performing in Accenture. Hence, as shown in Figure 2, there is an instantiation relationship between *Design cloud solution* in the MLSAC at M2-level to these two elements in the Accenture at M1-level. The *Migration planning* phase of Accenture activities *Design future architecture* and *Determine cloud deployment and service* are also producible through the instantiation of *Design cloud solution* in MLSAC.

The *Migration and Testing* phase includes activities for code refactoring, data migration, and acceptance testing of a legacy application in a new target cloud platform MLSAC supports these activities through *Document code changes and configurations*, *Migrate databases*, and *validate and accept*. More exactly, *Document code changes and Configurations* can be used to describe refactoring a legacy application source codes to be compatible with application programming interfaces of selected cloud services. Any incompatibilities between application and cloud platforms can be resolved with appropriate operation interfaces, messages, or data types. Indeed, *Document code changes and configurations* in Accenture is an instance of *Refactor codes* in the *Enable* phase of MLSAC.

Migrating database in Accenture is an instance of the *Migrate data* in MLSAC. *Validate and accept* is to test a reengineered legacy application in the cloud to confirm whether or not the actual reengineering has affected the original application functionality. In MLSAC, we distinguish between different tests activities as shown in the method fragment *Test system* in Figure 1. The relationships in Accenture method are producible using the relationships defined in MLSAC. For example, *Design future architecture* in Accenture uses information provided as a result of performing task *Determine target cloud vendor platform*. This relationship, i.e. *uses*, is an instance of the relationship between *Design cloud solution* and *Choose cloud provider* in MLSAC.

Revision to MLSAC. The *Migration planning* phase of Accenture has activities *Define migration roadmap* and *Conduct pilots*. In *Define migration roadmap* factors such as estimated reengineering cost, business calendars, assessment recommendations, and target architecture are also taken into account. Likewise, *Conduct pilots* is to build a prototype of the new solution architecture prior to its actual implementation in the new cloud platforms. To further support *Define migration roadmap* and *Conduct pilots* we added them to the *Plan* and *Design* phases of the metamodel.

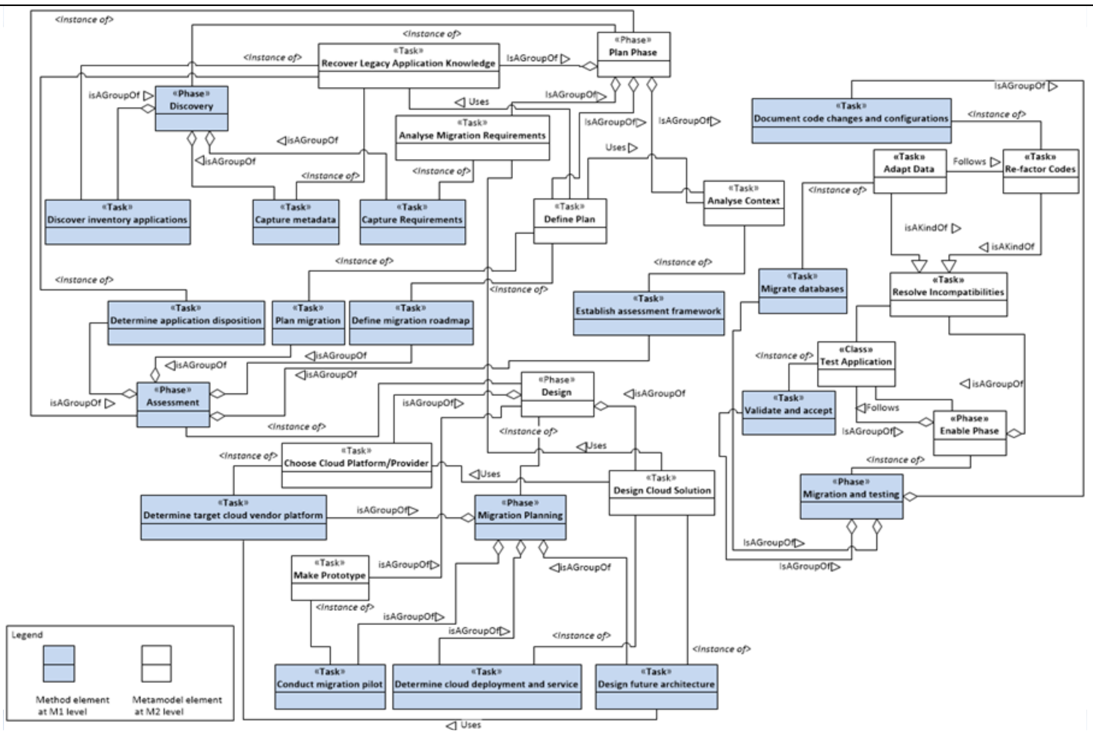

Fig 1. Conformance result of Accenture method fragments at M1-level to MLSAC metamodel at M2-level

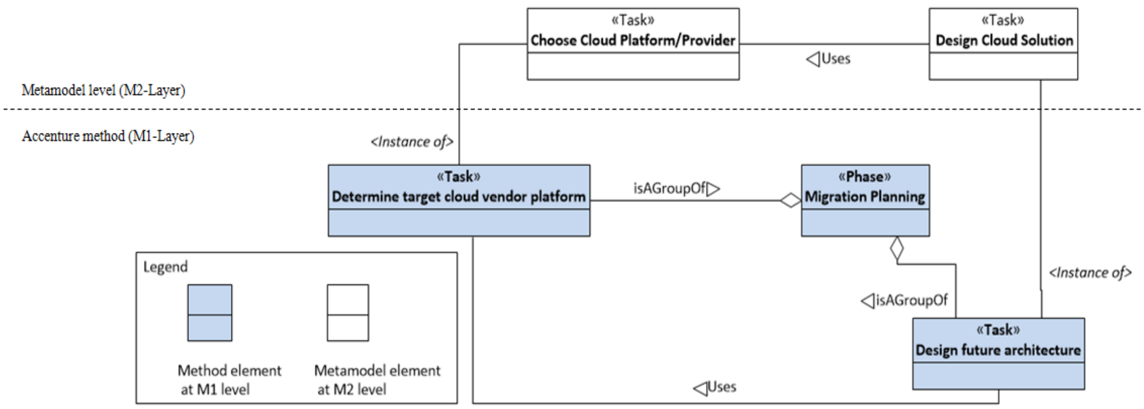

Fig 2. Instantiation of *Design cloud solution* and *Choose cloud provider* fragments and their relationships into design future architecture and determine target cloud vendor platform in Accenture

References

[1] M. Fahmideh, F. Daneshgar, F. Rabhi, and G. Beydon, "A generic cloud migration process model," *European Journal of Information Systems,* 2019.

[2] M. Antkiewicz, K. Czarnecki, and M. Stephan, "Engineering of framework-specific modeling languages," *Software Engineering, IEEE Transactions on,* vol. 35, no. 6, pp. 795-824, 2009.

[3] Accenture, "Accenture cloud application migration services," *Available at https://www.accenture.com/t20160728T233018__w__/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Technology_9/Accenture-Cloud-Application-Migration-Services.pdf (last access 2015),* 2014.