Technology-Aware Enterprise Modeling: Challenging the Model-Driven Architecture Paradigm

Irina Rychkova[1], Eddy Kiomba Kambilo[1], Nicolas Herbaut[1], Oscar Pastor[2], Rene Noel[2], and Carine Souveyet[1]

[1] University Paris 1 Pantheon-Sorbonne, Paris, France
{irina.rychkova,eddy.kiomba-kambilo,nicolas.herbaut,carine.souveyet}@univ-paris1.fr
[2] Universitat Politècnica de València, València, Spain
opastor@dsic.upv.es, rnoel@pros.upv.es

Abstract. We propose a holistic design approach that extends a traditional Model Driven Architecture paradigm with the bottom-up constraint analysis and propagation. We define a Unified Feature Model that represents the organizational goals and the technical properties of the solution, and use model-checking to reason about the constraints that arise from their interplay. In the illustrative example, we show how the technical decisions specified in a Feature Model can be formalized, validated, and translated into strategic limits of a Goal Model for further redesign of both business and technical enterprise models.

Keywords: MDA · Enterprise Modeling · Feature Model · model-checking.

1 Introduction

The Model-Driven Architecture (MDA) provides a fundamental framework for the design and development of enterprise information systems, prioritizing the top-down design process. It introduces three levels of model abstraction and a model transformation process, ensuring a traceability and alignment between the high-level business goals and developed technological components [10]. In this methodology, the selection of technological platforms and components is deferred to lower levels of the MDA design hierarchy. This approach is justifiable under the assumption that technological choices have primarily local effects. However, the emergence of disruptive technologies, such as blockchain challenges the established principles of MDA by exerting a broader influence on processes and goals of the enterprise as a whole.[9].

The intrinsic properties of blockchain not only create efficiencies locally, but also introduce strategic limits extending beyond the initially targeted areas of implementation. Consequently, the adoption of such technologies requires a holistic analysis of their potential impacts on the business processes and objectives enterprise-wide. Moreover, this analysis cannot be postponed to the later design stages, but accompany the whole design process, increasing the technology-awareness already at the early design stages.

We propose an approach for Technology-Aware Enterprise Modeling (TEAEM) that extends the MDA paradigm with the bottom-up constraint propagation and analysis. We depart from the idea that technical properties defined at the lower MDA abstraction levels can create implicit 'side effects' in the higher MDA levels. This position paper introduces TEAEM and presents the first results of its implementation. Some of the TEAEM steps are supported by the well known modeling tools. We developed several modules for model transformation, model unification and interpretation of results. The current version of TEAEM supports only goal - feature model unification. The component modeling will be integrated in the nearest future. We illustrate our approach on a theoretical example of an Organization seeking to implement a blockchain solution for its processes. The results of this analysis provide the ground for a technology-aware business (re)design and decision-making for the Organization.

In Section 2, we provide a brief reminder on the MDA and discuss its challenges; in Section 3, we introduce the TEAEM approach; in Section 4, we present our illustrative example; in Section 5, we provide a road-map for the future development of TEAEM and present our conclusions.

2   Background

Model Driven Architecture (MDA) is a software design approach that provides a set of guidelines for structuring specifications expressed as models, supporting traceability and business-IT alignment. MDA defines three primary types of models: *Computation Independent Model* (CIM) represents the system's requirements and business context without detailing the structure or processing; *Platform Independent Model* (PIM) specifies the system's structure and functionality but abstracts away the details of any specific implementation platform; *Platform Specific Model* (PSM) provides the technical details on system implementation using a particular technology or platform.

Methods and approaches for enterprise system design grounded on MDA are discussed in research literature for several decades. Alignment between enterprise models at different MDA abstraction levels is addressed in [17, 5, 21, 22]. Numerous works focus on validation and analysis of alignment between business processes and goals [4, 3, 1]. Formal methods are proposed for alignment verification in [18]. To acknowledge the constantly changing business environment, integration of organizational strategy and structure into MDA is addressed in [13]. In [14], a semi-automated *strategy-to-code* approach that integrates organizational, business process, and information system modeling is introduced. This approach is grounded on LiteStart modeling method [12] and ensures traceability across modeling levels.

The research presented in [9, 16, 11] acknowledges the rapid evolution and complexity of technological solutions and their impact on organizational strategy and processes. While technological solutions may initially excel in addressing specific business goals, their implementation can introduce strategic limits in other areas.

This necessitates the evolution of design approaches, including MDA, and motivates the TEAEM approach presented in this work.

3   Technology-Aware Enterprise Modeling

TEAEM is a holistic approach that extends the (top-down) MDA design paradigm with the bottom-up traceability and constraint propagation analysis.

3.1   TEAEM: Steps

Figure 1 illustrates the TEAEM approach. We use goal modeling [20], feature modeling [23] and component modeling [19] to represent an enterprise solution at the three MDA abstraction levels.

The top-down ***Model Design*** follows the MDA paradigm and consists of developing a Goal Model (CIM), a Feature Model (PIM) and a Component model (PSM) and their respective model transformations. The bottom-up ***Constraint Analysis*** represents our main theoretical contribution and consists of Model unification and Model checking/Impact analysis steps.
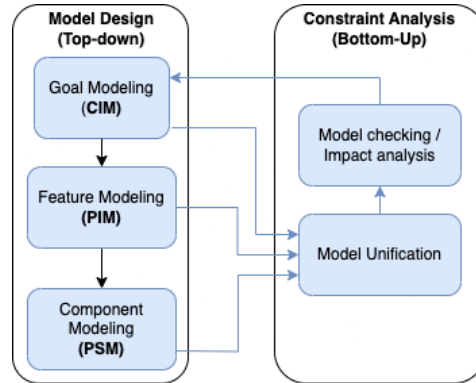

Fig. 1. TEAEM approach

Goal Modeling (CIM). A goal model representing the high-level business goals, requirements, and domain concepts is created in this step. This model correspond to CIM - the highest level of abstraction within MDA. It outlines the dependencies between goals and intended outcomes ensuring alignment with broader organizational context. These dependencies are explored in the further constraint analysis.

Feature modeling (PIM). A feature model is defined in this step. This model specifies the hierarchical structure of abstract technical functionalities (features) of the prospective solution and corresponds to PIM. The dependencies and

constraints between the features define alternative configurations for the technical solution.

Component modeling (PSM). A component model is created in this step. This model specifies technological components and platform-specific details necessary for implementation of the features defined in PIM. Technical constraints between the components provide the information about alternative implementations of the designed solution.

Model Unification. In this step, business goals, technical features, and components are specified within a single Unified Feature Model (UFM). For further impact analysis, we use logical expressions to formalize the mappings between the model elements defined at different MDA abstraction levels.

We define the following mapping types: *Goal to Feature Mapping* associates the business goals with the technical features satisfying these goals. This mapping reflects technical assumptions made by system engineers. *Feature to Goal Mapping* associates the technical features with the goals that can be possibly compromised or inhibited by implementation of these features. *Feature to Component Mapping* associates the features with the specific technological components implementing these features. *Component to Feature Mapping* explicitly defines possible restrictions or incompatibilities between the features and the components. *Component to Goal Mapping* indicates possible restrictions between the (soft) goals and the components such as quality, feasibility, performance issues etc. We combine these mappings with the constraints defined in the previous TEAEM steps to form the UFM.

Model checking / Impact analysis. We use model-checking to validate the UFM and to identify inconsistencies (if any) in its specification. A solver finds possible configurations of a solution and/or shows the conflicts between the elements of the unified model. We propose a domain-specific interpretation of these conflicts and recommendations for their resolution.

The Constraint Analysis can be performed at different stages of Model design and serve to: (a) identify and propagate the effect of design decisions specified in PIM on CIM; (b) identify and propagate the effect of design decisions specified in PSM on PIM and/or on CIM.

3.2 TEAEM: Implementation

In this work, we illustrate a semi-automated model unification and constraint analysis between CIM and PIM abstraction levels of MDA.

To implement the approach, the following three modules are developed[3]:

1. *Generation of the Unified Feature Model (UFM)*: The unified model comprises the goal model, features model, and constraints. *For the top-down model design,* we use the i* modeling language [2] and the PiStar tool [15] for Goal Modeling, and the feature modeling environment FeatureIDE [7]. The mapping for different MDA abstraction levels is added manually with an external

---

[3] https://github.com/Eddykams/TEAEM_develop

file using logical expression notation. We develop an automated Python module [transform.py] which takes as input the goal model, the feature model, and the provided constraints, and automatically transforms them into a unified model.

2. *Extraction of the constraints from the UFM:* We use FeatureIDE and its bundled Sat4j solver [8] to retrieve the constraints from the UFM. We generate an output in JSON that is further used for the constraint analysis step.

3. *Interpretation of the solution (constraints violation and their implication bottom-up)*: We develop a Python module [interpretation.py] that uses the .json list of constraint violations extracted from the UFM in FeatureIDE as an input and produces the domain-specific interpretations of these violations that can be further used for enterprise models redesign.

The current version of our tools is semi-automated. The goal model is designed with piStar tool, the feature model is designed with FeatureIDE. The mapping between goal model and feature model is formalized using logical expressions in an external text file. This file is used as an input to generate our Unified Model. The model-checking of the unified feature model is automatically applied by FeatureIDE. The model-checking results are extracted and interpreted by the developed module. We are working to provide a fully automated mapping process for the next version.

4   Illustrative Example

We illustrate the TEAEM approach on a simple example of the Organization that seeks to enhance transparency and auditability of its processes. The Organization considers blockchain technology as a platform for its enterprise solutions. In particular, the Organization needs to choose between two blockchain platforms (PublicBC_X and PrivateBC_Y). Further, in the design, it will configure the selected blockchain platform and its components. Since the Organization is also concerned with the GDPR compliance, more detailed analysis of the abstract capabilities and technical functionality of the prospective blockchain solution is required. We illustrate how TEAEM can support the analysis of technical constraints and their impact on the defined business goals of the Organization.

Goal Modeling (CIM): The goal model illustrated in Fig. 2.(a) defines the two goals of the Organization: 'GDPR Compliance' and 'Auditability' that have to be achieved. We specify this dependency with the i* And-refinement link that corresponds to the logical conjunction:

```
Business Goals => GDPR Compliance AND Auditability
```

Feature Modeling (PIM): The feature model illustrated in Fig 2.(b) specifies the abstract functionalities associated with blockchain technology. The prospective Technical Solution of the Organization (modeled as a root feature) will be grounded either on the functionalities of a 'PublicBC_X' or a 'PrivateBC_Y'. We specify this with the alternative feature group in FeatureIDE that corresponds to exclusive OR expression:

```
Technical Solution => PublicBC_X XOR PrivateBC_Y
```
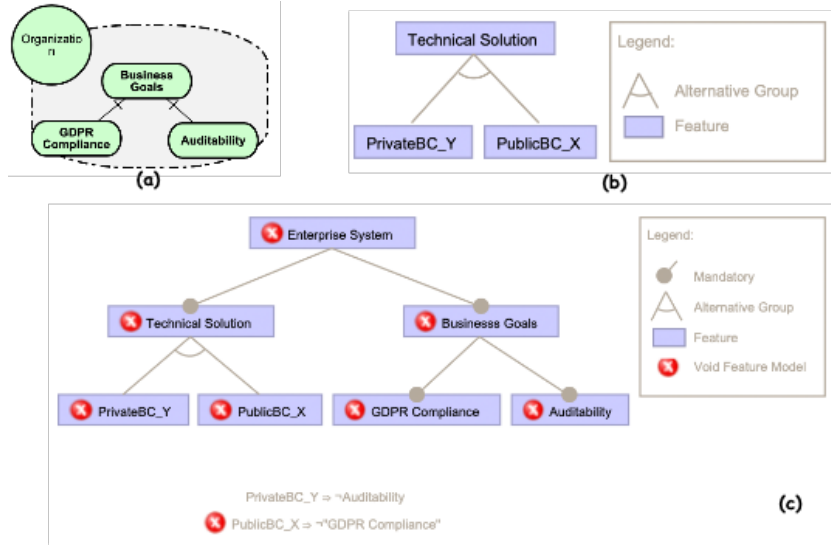

Fig. 2. (a): Goal Model of the Organization; (b): Feature Model with a proposition of alternative blockchain solutions; (c): Unified Feature Model that merges (a) and (b).

Once the goal model and solution proposition are formalized, we proceed with the (bottom-up) constraint propagation and analysis.

Model Unification. We create a unified feature model that specifies the 'Enterprise System' of the Organization (Fig. 2.(c)). Our UFM represents the goals (from CIM) and the features (from PIM) using the feature modeling formalism. Here we model 'Technical Solution' and 'Business Goals' as mandatory features of the 'Enterprise System' root feature.

To formalize the mapping between the created MDA models, We assume the following assumptions: we consider generic properties of Public blockchain X and Private blockchain Y. The Private blockchain Y limits the auditability of the blockchain because their nodes are controlled by a single (private) entity and the immutability is contingent upon the trustworthiness of this entity. The Public blockchain X does not face this immutability issue because any node can participate in the consensus and validation processes, ensuring that operations are conducted fairly and transparently. Nevertheless, this immutability of public blockchains prevents from implementation of a fundamental GDPR's 'right to be forgotten' - an alteration/deletion of data on demand. We map the technical features defined in PIM to the goals that can be compromised by implementation of these features using the following logical expressions:

```
PublicBC_X => NOT GDPR Compliance;
```

```
PrivateBC_Y => NOT Audit
```

Goal model, Feature model and Feature to Goal mapping presented above are documented as XML files and are used for semi-automated generation of the UFM.
Model checking/ Impact analysis. The FeatureIDE automatically executes the model-checking on the generated UFM. In our example, the UFM is invalid (i.e., the model is `void`), meaning that no Enterprise System satisfying actual constraints can be configured. We identify and trace the sources of conflicting constraints to the model elements defined at different MDA levels (PIM and CIM in our example). Using our developed Extraction and Solution Interpretation modules, we provide the domain-specific constraint interpretation, aiming to assist business users in redesign.

```
Logical constraints:
 ['Enterprise System ⇒ Technical Solution', 'Enterprise System ⇒ Businesss Goals', 'Techni
cal Solution ⇒ PrivateBC_Y v PublicBC_X', 'Businesss Goals ⇒ GDPR Compliance', 'Businesss
Goals ⇒ Auditability']
***********************************

Domain—specific constraint analysis & interpretation:
———————————————————————————————————————————————————————
To correct your UFM, you have the following options:
———————————————————————————————————————————————————————
Make constraint 'GDPR Compliance' non—mandatory, OR Remove constraint 'GDPR Compliance'.
Make constraint 'Auditability' non—mandatory, OR Remove constraint 'Auditability'.
```

Fig. 3. Domain-specific constraint analysis & interpretation.

In this example, the business expert faces the problem where any choice of blockchain solution compromises one of the business goals defined by the Organization. Along with modification or refinement of the prospective Technical Solution, this analysis calls for redefinition or re-prioritization of the Business Goals in response of the bottom-up constraint propagation.

5   Conclusion and Future Work

We presented TEAEM approach that contributes to design of enterprise systems within the MDA paradigm. While existing approaches support traceability and alignment between goals and technical solutions top-down, they often fall short in guiding stakeholders through the decision-making process and analysis of inconsistencies bottom-up. Our approach enables reconciliation between business goals, technical capabilities and specific solutions through formal analysis and propagation of constraints.

We outlined the TEAEM steps (illustrated in Fig.1). Model unification and Model checking / Impact analysis are the two steps that extend the MDA providing constraint propagation and analysis. We developed a technique for the semi-automated generation of the Unified Feature Model (UFM) and used the feature modeling environment FeatureIDE for automated model-checking.

In this article, we illustrated the propagation of constraints from PIM to CIM. Identification and propagation of the effect of design decisions specified in PSM on PIM and/or on CIM, and the automated mapping between the elements at different MDA levels of abstraction will be addressed in our future work.

We illustrated our approach on a short theoretical example. Developing a realistic example is an important next step. We plan to conduct a case study on the design and analysis of a Supply Chain Management solution based on blockchain technology. In this case study, propagation and impact of technical properties related to specific blockchain solutions on the organizational supply chain management process and strategic goals will be examined. In particular, we will focus on trustworthiness goals addressed in our previous work [6].

References

1. Cortes-Cornax, M., Matei, A., Dupuy-Chessa, S., Rieu, D., Mandran, N., Letier, E.: Using intentional fragments to bridge the gap between organizational and intentional levels. Information and Software Technology **58**, 1–19 (2015)
2. Franch, X., López, L., Cares, C., Colomer, D.: The i* framework for goal-oriented modeling. Domain-Specific Conceptual Modeling: Concepts, Methods and Tools pp. 485–506 (2016)
3. Gröner, G., Asadi, M., Mohabbati, B., Gašević, D., Bošković, M., Parreiras, F.S.: Validation of user intentions in process orchestration and choreography. Information Systems **43**, 83–99 (2014)
4. Guizzardi, R., Reis, A.N.: A method to align goals and business processes. In: International Conference on Conceptual Modeling. pp. 79–93. Springer (2015)
5. Horita, H., Honda, K., Sei, Y., Nakagawa, H., Tahara, Y., Ohsuga, A.: Transformation approach from kaos goal models to bpmn models using refinement patterns. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing. pp. 1023–1024 (2014)
6. Kambilo, E.K., Rychkova, I., Herbaut, N., Souveyet, C.: Addressing trust issues in supply-chain management systems through blockchain software patterns. In: International Conference on Research Challenges in Information Science. pp. 275–290. Springer (2023)
7. Kastner, C., Thum, T., Saake, G., Feigenspan, J., Leich, T., Wielgorz, F., Apel, S.: Featureide: A tool framework for feature-oriented software development. In: 2009 ieee 31st international conference on software engineering. pp. 611–614. IEEE (2009)
8. Le Berre, D., Parrain, A.: The sat4j library, release 2.2. Journal on Satisfiability, Boolean Modeling and Computation **7**(2-3), 59–64 (2010)
9. Marques, L., Pereira, R.: Creating value with blockchain for organizations. In: Digital Technologies and Transformation in Business, Industry and Organizations, pp. 17–41. Springer (2023)
10. Mellor, S.J.: MDA distilled: principles of model-driven architecture. Addison-Wesley Professional (2004)
11. Morkunas, V.J., Paschen, J., Boon, E.: How blockchain technologies impact your business model. Business Horizons **62**(3), 295–306 (2019)
12. Noel, R., Panach, I., Ruiz, M., Pastor, O.: The litestrat method: Towards strategic model-driven development. In: Proceedings of the 29th International Conference on Information Systems Development (2021)

13. Noel, R., Panach, J.I., Pastor, O.: Challenges for model-driven development of strategically aligned information systems. IEEE Access **10**, 38237–38253 (2022)
14. Pastor, O., Noel, R., Panach, I.: From strategy to code: Achieving strategical alignment in software development projects through conceptual modelling pp. 145–164 (2021)
15. Pimentel, J., Castro, J.: pistar tool–a pluggable online tool for goal modeling. In: 2018 IEEE 26th International Requirements Engineering Conference (RE). pp. 498–499. IEEE (2018)
16. Rajnak, V., Puschmann, T.: The impact of blockchain on business models in banking. Information Systems and e-Business Management **19**, 809–861 (2021)
17. Ruiz, M., Costal, D., España, S., Franch, X., Pastor, O.: Gobis: An integrated framework to analyse the goal and business process perspectives in information systems. Information Systems **53**, 330–345. https://doi.org/10.1016/j.is.2015.03.007
18. Rychkova, Irina: Formal semantics for refinement verification of entreprise models (2008). https://doi.org/10.5075/EPFL-THESIS-4210, http://infoscience.epfl.ch/record/126002
19. Trinidad, P., Cortés, A.R., Pena, J., Benavides, D.: Mapping feature models onto component models to build dynamic software product lines. In: SPLC (2). pp. 51–56 (2007)
20. Van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: Proceedings fifth ieee international symposium on requirements engineering. pp. 249–262. IEEE (2001)
21. de la Vara, J.L., Sánchez, J., Pastor, O.: On the use of goal models and business process models for elicitation of system requirements. In: International Workshop on Business Process Modeling, Development and Support. pp. 168–183. Springer (2013)
22. Wegmann, A., Regev, G., Rychkova, I., Lê, L.S., De La Cruz, J.D., Julia, P.: Business and it alignment with seam for enterprise architecture. In: 11th IEEE international enterprise distributed object computing conference (EDOC 2007). pp. 111–111. IEEE (2007)
23. Yu, Y., Lapouchnian, A., Liaskos, S., Mylopoulos, J., Leite, J.C.: From goals to high-variability software design. In: Foundations of Intelligent Systems: 17th International Symposium, ISMIS 2008 Toronto, Canada, May 20-23, 2008 Proceedings 17. pp. 1–16. Springer (2008)