



HAL
open science

Configuration of Software Product Lines Driven by the Softgoals: the TEAEM Approach

Eddy Kiomba Kambilo, Nicolas Herbaut, Irina Rychkova, Carine Souveyet

► **To cite this version:**

Eddy Kiomba Kambilo, Nicolas Herbaut, Irina Rychkova, Carine Souveyet. Configuration of Software Product Lines Driven by the Softgoals: the TEAEM Approach. The Practice of Enterprise Modeling. PoEM 2024., Dec 2024, Stockholm, Sweden. pp.262-278, 10.1007/978-3-031-77908-4_16 . hal-04813530

HAL Id: hal-04813530

<https://hal.science/hal-04813530v1>

Submitted on 2 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Configuration of Software Product Lines Driven by the Softgoals: the TEAEM Approach

Eddy Kiomba Kambilo¹, Nicolas Herbaut¹, Irina Rychkova¹, and Carine Souveyet¹

University Paris 1 Pantheon-Sorbonne, Paris, France
{eddy.kiomba-kambilo,nicolas.herbaut, irina.rychkova,
carine.souveyet}@univ-paris1.fr

Abstract. The Model-Driven Architecture (MDA) serves as an essential framework for designing enterprise information systems, emphasizing the alignment and traceability of goals across different modeling layers. However, MDA typically overlooks the integration of qualitative attributes, or softgoals, which are critical for enhancing user satisfaction. Our previous work introduced the Technology-Aware Enterprise Modeling (TEAEM) approach, which enhances MDA by integrating model checking, validation, and impact analysis.

This paper extends TEAEM to more effectively incorporate softgoals. We achieve this by integrating SysML component modeling for low-level, and softgoals into the high-level of the MDA. These advancements facilitate bottom-up constraint propagation and ensure that technological decisions are reflected consistently at all levels of abstraction, thereby optimizing the system to meet strategic business goals.

Additionally, we propose generating configurations in software product lines, driven by the fulfillment of softgoals, to apply the TEAEM approach.

Keywords: MDA · Enterprise Modeling · Softgoals · Software Product Lines

1 Introduction

The Model-Driven Architecture (MDA) provides a foundational framework for the design and development of enterprise information systems, emphasizing a top-down design approach with three levels of model abstraction: the Computational Independent Model (CIM) for capturing business goals, the Platform Independent Model (PIM) for representing system functionality independent of specific technologies, and the Platform Specific Model (PSM) for detailing the technological choices made [13]. This structured approach allows for deferred technological decisions, assuming that such choices primarily have localized impacts.

However, the MDA approach often overlooks the integration of softgoals due to their subjective and qualitative nature. Softgoals, unlike hardgoals, lack clear-cut achievement criteria and encompass aspects such as cost, security, response

time, and performance. These qualitative attributes are crucial for successful software production as they influence user satisfaction, feature prioritization, and development trade-offs.

Motivated by the need to better incorporate qualitative attributes and analyzing technological impacts into enterprise modeling, we previously introduced Technology Aware for Enterprise Modeling (TEAEM)[17] which extends MDA by integrating model checking, validation and impact analysis of technological choices before product generation. This approach enables the unification of CIM-PIM within a “Unified Model”, allowing the identification and resolution of inconsistencies across different MDA levels. If inconsistencies are detected, TEAEM reports constraint violations to the business user and assists in decision-making. TEAEM had some limitations. Firstly, it focused on the unification of CIM and PIM without extending support to the PSM. Additionally, the approach was centered on the use of hardgoals, with little attention given to softgoals.

This paper aims extends TEAEM with two major contributions: First, we integrate the MDA low-level PSM using SysML component modeling, achieving a unified CIM-PIM-PSM model that enables bottom-up constraint propagation and analysis. This integration ensures that technological choices made at the PSM level are consistently reflected across all abstraction levels, enhancing the overall coherence and bottom-up traceability within the enterprise modeling process. Second, we extend TEAEM by integrating softgoals into the CIM. By including softgoals in the early stages of modeling, we can better address qualitative attributes throughout the system development life-cycle. This approach allows for the generation of code (configuration) guided by the satisfaction of these softgoals, thereby optimizing the system for user satisfaction and strategic business goals.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of key concepts such as MDA, Software Product Lines, and Softgoals. Section 3 details the TEAEM approach and its extension to integrate softgoals. Section 4 discusses the implementation, mapping details, and an illustrative example used to validate our approach. Finally, Section 5 presents the discussion and conclusions of this work.

2 Background

2.1 MDA, Top-Down Design and Enterprise Architecture

The architecture of a system constitutes what is essential about that system considered in relation to its environment[8]. There are different frameworks used in enterprise architecture and software engineering.

Model Driven Architecture is a software design approach that provides a set of guidelines for structuring specifications, which are expressed as models. The core of MDA is its focus on three primary types of models; *Computation Independent Model* represents the system’s requirements and business context, without detailing the structure or processing. *Platform Independent Model* specifies the system’s structure and functionality but abstracts away the details of

any specific platform. *Platform Specific Model* combines the specifications in the PIM with the details of how the system uses a particular type of platform.

Many works exist in the literature that exploit the MDA concept to make goal alignment and traceability of goal. In [14], authors presents a strategy-to-code (S2C) methodological approach integrating organizational, business process, and information system modeling to support strategic alignment in software development. It emphasizes model-driven development and conceptual modeling for semi-automatic software generation and traceability across different modeling levels. By using a working example, it illustrates how strategic definitions can be transformed into specific software components through the integration of Lite*, Communication Analysis, and the OO-Method, demonstrating the feasibility of achieving strategic requirements, traceability, and automatic software generation.

Recent Enterprise Architecture frameworks exist that give advice on the processes and practices of Enterprise Architecture, and one of these, that the ArchiMate Specification is fully aligned with, is the TOGAF standard[8]. ArchiMate is focus on the the problem of aligning strategic business plans with the plans and goals for the development of Information Systems based on Information Technology (IS/IT).

2.2 Software Product Lines

MDA and SPL are complementary approaches that greatly benefit from the integration of softgoals. Software Product Line Engineering (SPLE) serves as a methodological framework for developing software families, characterized by significant overlap in functional and non-functional attributes [3].

A primary focus of SPLE is to identify and codify the common features of a cluster of software-intensive applications within a specific domain [1]. An SPL is generally composed of three main types of artifacts: *Problem Space*: this describes the features and functionalities desired by users across all potential products in the family. *Solution Space*: This includes the design and implementation specifics for all components across different products within the SPL and *The Mappings*: These establish the connections between features in the problem space and their respective components in the solution space. In this work, we choose SPL like application domain of TEAEM.

2.3 Softgoals and Utility Functions for Evaluating Softgoals

Softgoals serve as quality criteria that software systems or their components must meet during development. Softgoals are subjective, relative, diverse, and interactive, making them crucial yet challenging to model due to their qualitative nature [7].

The literature highlights the significance of softgoals in determining the success of software projects. For instance, consider the softgoals for a software system such as *performance* ("The system should respond quickly to user inputs")

and *security* ("The system should protect user data from unauthorized access"). These qualitative goals pose the question: how can we effectively evaluate them?

Significant research has been conducted on identifying softgoals within Information Systems (IS). For example, one study presented a comprehensive list of 114 softgoals identified in information systems [12], while another proposed a taxonomy categorizing these softgoals into four groups: Real and Web-based Systems, Web-based Systems, Real-time Systems, and Information Systems [5]. We aim to adapt some of these identified softgoals and quantify them.

To quantify softgoals, we propose writing a utility function for each softgoal, facilitating the evaluation of how well a system meets these goals. Softgoals like usability, reliability, and performance are inherently qualitative. By converting these into quantifiable metrics through a utility function, we enable more objective assessments and comparisons.

The following section will introduce our extended approach and explain how to generate configurations that meet these softgoals.

3 TEAEM Approach guiding configuration by softgoals

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

3.1 TEAEM Extension

We propose extending TEAEM with the following contributions: (1) Integration of PSM to illustrate technological choices using component modeling (SysML). (2) Integration of softgoals to guide business users in generating configurations within the SPL.

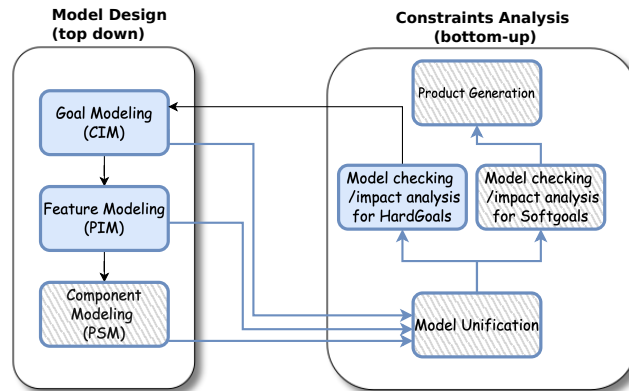


Fig. 1: TEAEM approach to meet Softgoals. Hashed blocks show TEAEM extension proposed in this paper

Figure 1 illustrates the extension of the TEAEM approach. The model is subdivided into two parts:

On the left hand side represents the top-down MDA design approach with three abstraction levels: CIM, PIM, and PSM.

- *CIM Layer*: We start by creating the goal model using the i* framework[4] with the PiStar tool[15]. This model includes both hardgoals (functional requirements) and softgoals (non-functional requirements).
- *PIM Layer*: We design the Feature Model using the FeatureIDE tool [10].
- *PSM Layer*: We create component models with SysML, which include technical metadata about alternative implementations of the designed solution.

On the right-hand side, we extend TEAEM by unifying the PIM, CIM, and PSM layers of MDA into a single model called the “Unified Feature Model (UFM)”. This unification is crucial for several reasons:

- Model checking and validation help validate the UFM to confirm it meets requirements, identify inconsistencies in the specification, and show conflicts between elements of the unified feature models.
- Impact Analysis: Helps identify how changes in one part of the model affect others by reporting constraint violations from low-level to high-level.
- Configuration Generation in SPL: Facilitates the generation of configurations that satisfy softgoals.

To achieve this unification, we define the following mapping:

- *Goal to Feature Mapping* (TEAEM): This mapping associates business goals with the technical features that fulfill those goals. It reflects the technical assumptions made by system engineers.
- *Feature to Goal Mapping* (TEAEM): This mapping associates technical features with the goals that could potentially be compromised or hindered by their implementation.
- *Feature to Component Mapping* (this paper): This new mapping associates features with the specific technological components that implement them.

The mappings are formalized using a symbolic mathematics library to structure the logical expressions. This library also develops utility functions to optimize the softgoals. Further details are provided in section 4.

After unifying the model, we use model-checking with a boolean solver to validate the UFM and identify inconsistencies causing conflicts between goals, features, and components.

If model checking returns false, indicating no feasible configuration, we provide a domain-specific interpretation of these conflicts and recommend solutions. If model checking returns true, indicating multiple feasible configurations, we integrate softgoals into the decision-making process. We use a utility function to quantify each softgoal and apply a Multiple-Criteria Decision Making (MCDM) algorithm to optimize softgoal satisfaction, determining the best configuration for the business user.

The main distinction between TEAEM and other frameworks like MDA or ArchiMate is its integration and analysis of inconsistencies across different abstract level (goal, feature, component) within the MDA. While ArchiMate offers a holistic view of enterprise architectures, it does not specifically focus on detecting and managing inconsistencies between different model types or guiding product generation by softgoals.

3.2 Modeling Softgoals with the Utility Function

As mentioned in section 2.3, we reuse some of the softgoals defined by [12].

In this context, Key Performance Indicators (KPIs) are commonly used to measure the performance of various aspects of a system. However, we use a utility function instead of KPIs because the utility function allows for a more comprehensive and integrative approach. While KPIs provide valuable insights into specific performance metrics, they often fail to account for the trade-offs between conflicting goals. The utility function, on the other hand, can encapsulate multiple softgoals into a single measure, allowing for the optimization of overall user satisfaction and system performance in a balanced manner.

The utility function quantifies values derived from component modeling (value of each component) and serves as input for the MCDM algorithm to generate configurations that maximize user satisfaction. We focus on the following softgoals for the implementation and test phases in sections 4 and 5.

Table 1: Description of Softgoals with Blockchain Components

Softgoals	Description	Refined-into	Notation
Cost	System should be cost-efficient.	Operates with minimal expenses and reduces the overall cost of ownership.	C
Response_Time	System should be fast.	Processes requests quickly and efficiently, providing timely feedback to users.	R
Throughput	System should have high throughput.	Processes incoming data quickly and sends new data without delay.	T

The overall goal is to minimize the total cost of the components, which is the sum of individual cost functions $C_i(x_i)$ for each component i . C is the total cost, and x_i are the decision variables adjusted to optimize the cost. R is the total response time, $R_i(x_i)$ is the response time function for each component i , and T is the total throughput that we aim to minimize, $\min_{0 \leq i \leq n} T_i$, the throughput function for each component i .

We define this objective function by integrating the different softgoals to optimize:

$$\text{Min } F = w_1C + w_2R - w_3T$$

Where F is the combined objective function, C is the total cost, R is the total response time, and T is the total throughput. w_1, w_2, w_3 are the weights assigned to cost, response time, and throughput, respectively, reflecting their relative importance for the business users.

The next section presents the different modules we developed.

4 Implementation

We implemented TEAEM, an automated process for model unification across different MDA abstraction levels, detecting constraint violations, performing impact analysis, and generating software product line configurations driven by softgoals.

We developed the following modules¹:

1. *Generation of the Unified Feature Model (UFM)*: The unified model includes the goal model, feature model, component models, and constraints. For top-down model design, we use the i* modeling language [4] and the PiStar tool for Goal Modeling, FeatureIDE for feature modeling, and SysML for component models. Mapping for CIM-PIM-PSM is achieved using SymPy with logical expressions. This module automates the generation of the unified feature model.
2. *Extraction of Constraints from the UFM*: This module performs model checking/validation. We use FeatureIDE and its bundled Sat4j solver [11] to retrieve constraints from the UFM, generating an output in JSON for constraint analysis when inconsistencies arise between the three MDA levels.
3. *Interpretation of Constraint Violations and Their Implications*: We developed a module that takes the outcomes.json list of constraint violations from FeatureIDE and produces domain-specific interpretations of these violations to guide business users in their decision-making.
4. *Generation of Configurations that Meet Softgoals*: This module calculates utility functions using SymPy, based on parameter values from SysML component models. The utility function's Min/Max values help identify the best configuration that matches the specified requirements.
5. *Optimization Using the Topsis Algorithm*: We implement the Topsis algorithm for Multiple-Criteria Decision Making, ensuring the optimization and evaluation of our Unified Model to achieve highly optimized solutions.

We make three types of mapping as specified on the section 3.1. For The mapping of Goal to Feature and Feature to goal, We use the mathematical logical expressions rules that we passed with Sympy in order to execute automatically the logic.

¹ https://github.com/Eddykams/TEAEM_develop

For the mapping PIM-PSM, We have two cases:

(1) Mapping One Feature - One Component:

The following bijection function modeling the mapping between PIM-PSM :

$$H_1 = F \implies C (A)$$

with, F: Features and C: Components.

(2) Mapping One Feature With Multiple Components: We use TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) is a multi-criteria decision analysis method that ranks alternatives based on their distance to an ideal solution and a negative-ideal solution. It helps in selecting the best option by comparing the relative closeness of each alternative to the ideal solution.

In the next section, we present an illustrative example where we apply TEAEM to configure products to meet soft goals.

5 Illustrative Example: Counterfeiting Drugs in Supply Chain

5.1 Motivation

The market of counterfeit drugs has become a 200-billion-dollar business annually, according to the World Health Organization (WHO). According to a WHO report, up to 10% of all sold drugs globally are fake, with significantly higher rates in parts of Africa and Asia. This business is very dangerous because of life-and-death implications for patients (thousands of deaths every year) and also for the pharmaceutical industry's reputation, which can lose a lot of money. The complexity of pharmaceutical supply chain operations is the primary reason for this issue. With a large number of handovers to different supply chain partners before drugs reach hospitals and pharmacies, there is a significant lack of traceability. Manufacturing is vulnerable to counterfeit raw materials or ingredients from unknown sources. Moreover, illicit producers can relabel fake products to infiltrate legitimate distribution channels. For the pharmaceutical sector, the European Union (EU) and the Drug Supply Chain Security Act (DSCSA) in the United States provide a deadline of 2023 for the industry to implement a traceability system.

Based on the work [9][6], we present the following illustrative example. Company X, based in Europe and Africa, must enter the pharmaceutical market in compliance with EU regulations and operate in Africa. The organization has many branches in different countries that require a system to manage transactions recorded across multiple locations. Additionally, there is a need for a system that enhances trust among various stakeholders.

The system must address the following needs: *Reduce drug counterfeiting (fraud detection)* by ensuring traceability and transparency at every step of the product transfer. Ensure *secure data management* by enhancing privacy through access controls and encryption, and by maintaining the integrity and immutability of data. *Manage recalls of defective products*; conditions of transporting drugs

must be verified through real-time audits using IoT, sensors, or RFID tags. The authenticity of products must be checked by log-audit.

5.2 TEAEM for configuration in SPL to meet softgoals

The main goal of Enterprise X is to determine if, with their goals, features, and components, they can generate a product or if there are any inconsistencies between them before generating the product. The organization proposes to integrate the following softgoals: maximal performance, minimal cost for blockchain components, better response time, and minimal cost for the electronic solutions (RFIDs, IoT sensors, or NFC tags).

Step1: We present a partial goal model of the enterprise, depicting different softgoals and the links between goals and softgoals in the Figure 2.

Step 2: To build the feature model, we made a literature review to identify different solutions proposed for combating drug counterfeiting in the supply chain illustrated in 2. According to some findings in the literature, [2][16] assert, for example, that the ideal anti-counterfeit technology in an enterprise system should have a high level of safety, faster product application, established standards, simple to check, have automatic authentication, be accessible to consumers, and comply with industry regulations.

Table 2: Requirements and Technologies Identified

Goals - Requirements	References	Blockchain	Cloud	ML
Transparent	[16]	✓	-	-
Traceability	[16][2][19]	✓	-	✓
Reliable	[18][16][2][19]	✓	✓	✓
High Level of Safety	[16]	✓	-	-
faster product application	[18]	-	✓	-
Simple to audit	[16]	✓	-	-
Automatic Process	[18][16][2]	✓	✓	-
Compliant with industry Regulations	[18][19]	-	✓	✓
Accountability	[18]	-	✓	-
Trust	[16]	✓	-	-
Decentralized	[16][2]	✓	-	-
High Security	[2]	✓	-	-
Flexibility	[18]	-	✓	-

We built the feature model based on the technologies identified in 2. Three prominent technologies were proposed: blockchain technology, for its features such as traceability, trust, and immutability; Cloud Computing [18], known for its flexibility and faster application production; and Machine Learning, which can identify counterfeit drugs through algorithmic data analysis. Additionally, associated electronic equipment for traceability is proposed, such as RFID tags, IoT sensors (pressure, geolocation), and NFC tags.

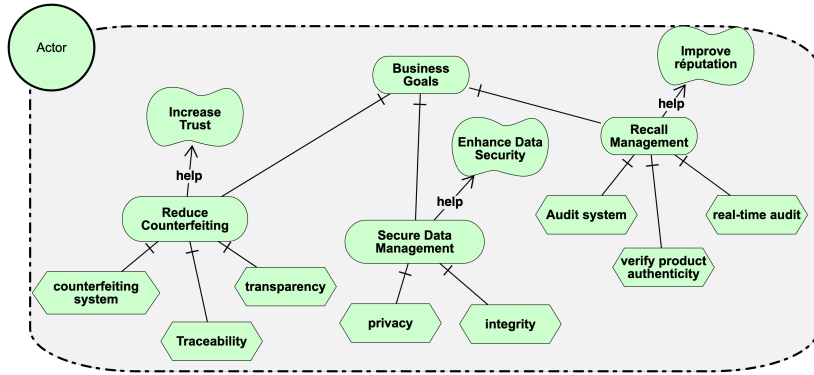


Fig. 2: Goal Modeling of Enterprise X

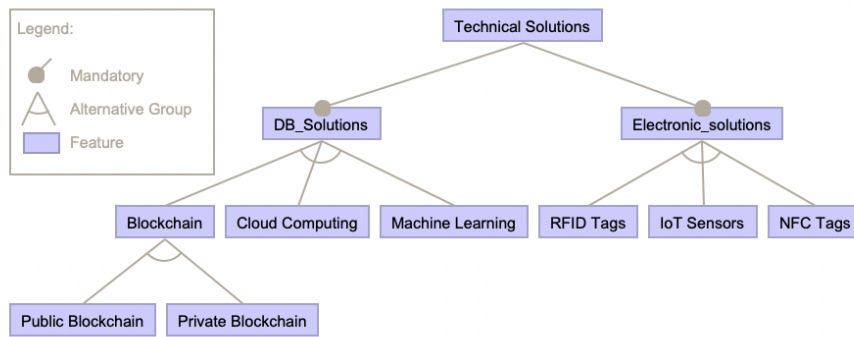


Fig. 3: Feature Model of Enterprise X

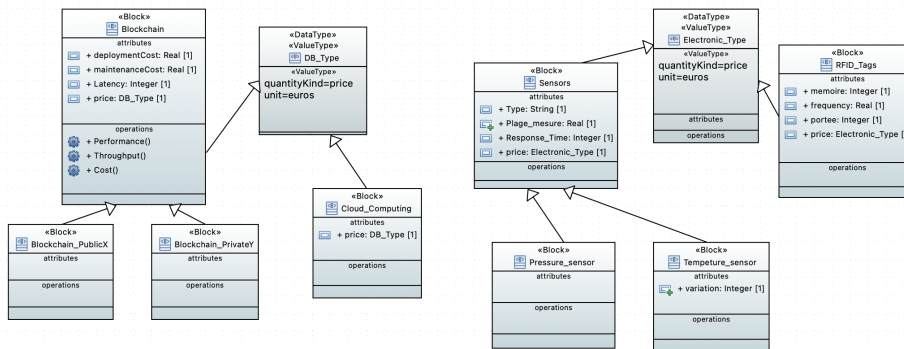


Fig. 4: Component Model of Enterprise X

Step 3: Based on the analysis presented in Table 2, we decided to use blockchain technology as a component to illustrate our example. Blockchain is a novel technology that has not yet been widely adopted by enterprises due to the lack of standardization and scarcity of engineers skilled in its implementation. The literature identifies two principal types of blockchains: public blockchain and private blockchain. Using SysML, we illustrate the component model for blockchain technology and other technologies. However, for the evaluation phase, we focus on the value of the blockchain component.

Step4: TEAEM - The unified model, as depicted in Fig.5, put within a single model the three abstract level of the MDA. In our study, the model checking results were positive, indicating that the unified model is valid. However, the next step involves generating configurations that satisfy the softgoals defined in the previous section.

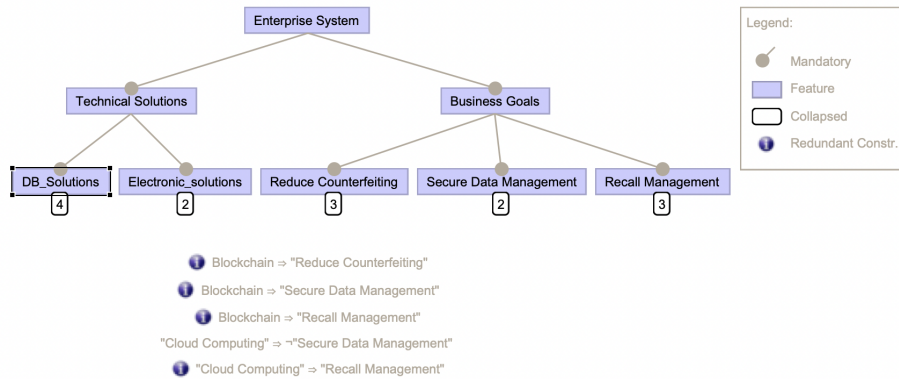


Fig. 5: Model Unification

Step 5: After model unification, we use model checking/validation to determine if a valid configuration exists for this unified model or if there are any inconsistencies. *The unified feature model generates 4 possible configurations*, indicating that there are no inconsistencies or constraint violations. Therefore, refining the model is unnecessary. In [17], we presented an example where constraint violations were detected. The question then arises: how do business users choose the "best" configuration that meets their softgoals?

Step 6: As outlined in the previous section, to generate configurations guided by the satisfaction of softgoals, we detail the softgoals discussed in Section 3.2, including their associated utility functions and the blockchain components employed. The values of component properties are inputted as parameters into the utility functions to select the optimal configuration. We use blockchain configurations and utilize the TOPSIS method for multi-objective optimization.

In the absence of a generalizable throughput/response time formula for all possible cases, we have chosen to formulate the following assumptions: Response Time is the minimal time for the detection of counterfeiting drugs and depends on the component properties: Transaction Processing Time, Block Creation Time, and Smart Contract Execution Time. The throughput is a function of block size, block execution time, and network bandwidth for the blockchain.

Table 3: Utility functions and blockchain component properties of softgoals

Softgoals	Utility function	Components properties
<i>Blockchain Cost</i>	$C = (\min \sum_{i=1}^n C_i(D_c, M_c))$	Deployment cost, Maintenance cost
<i>Throughput</i>	$T = T(B_s, B_t, N_b)$	Block Size, Block Time, Network Bandwidth.
<i>Response Time</i>	$R = R(Tx_i, B_c, SC_e)$	Tx Processing Time Block Creation Time SC Execution Time
<i>Electronic_Cost</i>	$EC = (\min \sum_{i=1}^n C_i(D_c, E_c, M_c))$	Deployment cost Execution cost Maintenance cost

5.3 Results

After integrating the blockchain component values as parameters in the utility function to generate products, we made the following tests:

(1) *We help to choose configurations that meet one softgoal.* For example, when the organization needs to minimize blockchain costs:

```

***** Components Mapping Completed. *****
Analyze of configuration: conf1.xml
Selected Feature: PublicBC_X
PublicBC_X - Ethereum: Price = 40.0
###
Analyze of configuration: conf0.xml
Selected Feature: PrivateBC_Y
PrivateBC_Y - Hyperledger: Price = 35.0
***** DECISION GUIDING CHOICE *****
The utility function: 35.0
To reduce Cost, The best Configuration is: conf0.xml
***** *****

```

Fig. 6: Configuration guiding by the Minimal Blockchain Cost

To minimize blockchain costs, the implemented module maps the value of the component across different configurations, calculates the value using the utility function implemented by SymPy, and returns this value for comparison with

other configurations. The configuration with the lowest cost is considered the winner. For one component, it is easy to generate a product that matches this softgoal.

(2) *Help to choose configurations that meet multiple criteria:* In this case, we have a multi-objective optimization. We need to choose the best configuration guided by multiple softgoal criteria.

Table 4: Softgoals Criteria with Heigh

Attributes	Blockchain Cost	Response_Time	Throughput	Electronic Cost
Conf0	300	15 seconds/bloc	18 Tx/seconds	140
Conf1	350	15 seconds/bloc	16 Tx/seconds	120
Conf2	200	2 seconds/bloc	100Tx/seconds	200
Conf3	250	3 seconds/bloc	200Tx/seconds	180

For the test phase, it is important to clarify that the values presented in Table 4 are provided for illustrative purposes only and do not represent calculations from a utility function. Instead, these values are intended to demonstrate the application of MCDM with TOPSIS.

The table 4 illustrates the four configurations found after the model checking/validation. Different values (Blockchain Cost, Response Time, Throughput, and Electronic Cost) were passed as inputs to our component model. In this case, we have a multi-objective optimization where we need to choose the best configuration guided by multiple softgoal criteria. Many optimization algorithms exist in the literature, and we implemented the TOPSIS method for Multiple-Criteria Decision Making.

For the evaluation phase, for example, if we have input data illustrated in Table 4, we have the following outcomes:

Scenario 1: We defined the same weight for all parameters [0.25,0.25,0.25,0.25] and identified Response Time, Blockchain Cost, and Electronic Cost as non-benefit parameters (functions to minimize) and Throughput as a benefit parameter (function to maximize). We have the following outcomes:

Table 5: Scenario 1 Outcomes Overview

Attributes	BC_Cost	R_Time	Throughput	E_Cost	S_POS	S_NEG	Score	Rank
Conf0	300	15	18	140	0.2215	0.0862	0.2801	3
Conf1	350	15	16	120	0.2349	0.0613	0.2069	4
Conf2	200	2	100	200	0.0689	0.2139	0.7561	2
Conf3	250	3	200	180	0.0532	0.2215	0.8061	1

To reduce the blockchain cost, the cost of the electronic equipment, and the response time, while increasing performance with throughput, the best configuration is Configuration 3 (Conf3), which achieves 80% satisfaction among business

users. The worst configuration is Configuration 1 (Conf1), with only 20% satisfaction among business users. This is clearly specified with different constraints and the values of blockchain components as parameters. The configuration that meets the specified softgoals is Configuration 3.

Scenario 2: Business users place significant emphasis on Response_Time compared to other criteria. We adjust the weight according to preferences, for example, 0.6 for Response Time and 0.13 Blockchain Cost, 0.13 Throughput and 0.13 Electronic Cost.([0.13, 0.61, 0.13, 0.13]).

Table 6: outcomes for Scenario 2

Attributes	BC_Cost	R_Time	Throughput	E_Cost	S_POS	S_NEG	Score	Rank
Conf0	300	15	18	140	0.3682	0.0344	0.0856	3
Conf1	350	15	16	120	0.3695	0.0245	0.0622	4
Conf2	200	2	100	200	0.0275	0.3675	0.9301	1
Conf3	250	3	200	180	0.0347	0.3416	0.9075	2

The best configuration has changed and is now Configuration 2 (Conf2), with 92% satisfaction among business users. The worst configuration is still Configuration 1 (Conf1), with only 6% satisfaction among business users. We can clearly see that a different softgoal weight will lead to different configurations.

6 Lesson Discussion, & Research Challenges

The TEAEM approach by integrating softgoals ensures that qualitative attributes such as user satisfaction, security, and performance are prioritized, resulting in configurations that align more closely with business goals. The unified CIM-PIM-PSM model enhances coherence and traceability across abstraction levels, leading to fewer errors and more complete configurations. This integration ensures that technological choices made at the PSM level are consistently reflected across all levels, reducing the risk of inconsistencies. TEAEM-generated configurations exhibit fewer errors and higher completeness, as validated by model checking.

The use of the TOPSIS method for MCDM in TEAEM facilitates the selection of optimal configurations based on multiple softgoal criteria. This approach allows business users to make informed decisions quickly, balancing trade-offs between competing goals. Empirical data show that TEAEM reduces the time required for configuration generation and decision-making compared to manual methods and other SPL approaches.

Compared to other approaches, TEAEM adds value in model-checking validation and constraints analysis propagation for bottom-up traceability. MDA focuses on top-down design, traceability, and alignment of goals from business goals to code generation. ArchiMate is designed to provide a comprehensive, integrated view of enterprise architecture, focusing on the relationships between

different domains (such as business, application, and technology). However, it does not inherently include specific mechanisms for detecting and managing inconsistencies between different model types. Additionally, while ArchiMate can represent various goals and requirements, it does not explicitly focus on guiding product generation through softgoals.

The implementation of TEAEM and the various evaluations carried out have allowed us to learn the following lesson:

- *Business users have the ability to express their preferences on the different softgoals, and expressing these preferences will have a technical impact. The weight or importance that a business user assigns to a softgoal during the configuration of a product in a Software Product Line (SPL) significantly influences the best configuration that can be proposed to them.*

TEAEM presents certain limitations, such as the complexity involved in formalizing utility functions for systems with non-linear attributes. This complexity can challenge the accurate quantification and optimization of softgoals. Also, managing the explosion of the feature model in large-scale projects remains a significant challenge, potentially complicating the optimization of solutions for specific product derivations.

Future work on the TEAEM approach should focus on developing formal guidelines for defining and applying utility functions, which will improve accuracy and make the approach more adaptable across various scenarios. Exploring advanced modularity techniques will be crucial for managing the complexity of feature models and addressing the feature model explosion problem, simplifying the process of configuration generation. Evaluating the methodology's practical impact through real-world applications is also critical to ensure its effectiveness. By tackling these areas, the TEAEM approach can be further enhanced to offer greater value in enterprise modeling.

7 Conclusion

In this paper, we extend the Technology-Aware Enterprise Modeling approach by incorporating softgoals to generate configurations within Software Product Lines. Our enhancement addresses the limitations of MDA methods in managing non-functional requirements, which are essential for user satisfaction.

Our primary contribution is the unification of the CIM-PIM-PSM into a single model, ensuring a traceable design process that connects business goals with technical implementations. We also integrate model checking and impact analysis to detect inconsistencies early, aiding decision-making and minimizing errors. The second contribution integrates softgoals into the TEAEM process, enabling SPL configurations to meet both goals and softgoals, ensuring the final products align with both user needs and business goals.

We demonstrate our approach using a counterfeit drug detection example in the supply chain, leveraging blockchain technology as a component. Future work will focus on managing feature models to address the challenge of feature model explosion.

References

1. Bagheri, E., Ensan, F., Gasevic, D., Boskovic, M.: Modular feature models: Representation and configuration. *Journal of Research and Practice in Information Technology* **43**(2), 109–140 (2011)
2. Bapatla, A.K., Mohanty, S.P., Kougianos, E., Puthal, D., Bapatla, A.: Pharmachain: A blockchain to ensure counterfeit-free pharmaceutical supply chain. *IET Networks* **12**(2), 53–76 (2023)
3. Bošković, M., Mussbacher, G., Bagheri, E., Amyot, D., Gašević, D., Hatala, M.: Aspect-oriented feature models. In: *Models in Software Engineering: Workshops and Symposia at MODELS 2010, Oslo, Norway, October 2-8, 2010, Reports and Revised Selected Papers* 13. pp. 110–124. Springer (2011)
4. 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)
5. Gazi, Y., Umar, M.S., Sadiq, M.: Classification of nfrs for information system. *International Journal of Computer Applications* **115**(22) (2015)
6. Hastig, G.M., Sodhi, M.S.: Blockchain for supply chain traceability: Business requirements and critical success factors. *Production and Operations Management* **29**(4), 935–954 (2020)
7. Hu, H., Ma, Q., Zhang, T., Tan, Y., Xiang, H., Fu, C., Feng, Y.: Semantic modelling and automated reasoning of non-functional requirement conflicts in the context of softgoal interdependencies. *IET Software* **9**(6), 145–156 (2015)
8. Josey, A.: *ArchiMate® 3.0. 1-A pocket guide*. Van Haren (2017)
9. Kambilo, E.K., Zghal, H.B., Guegan, C.G., Stankovski, V., Kochovski, P., Vodislav, D.: A blockchain-based framework for drug traceability: Chaindrugtrac. In: *Proceedings of the 37th ACM/SIGAPP symposium on applied computing*. pp. 1900–1907 (2022)
10. 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)
11. Le Berre, D., Parrain, A.: The sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation* **7**(2-3), 59–64 (2010)
12. Mairiza, D., Zowghi, D., Nurmaliani, N.: An investigation into the notion of non-functional requirements. In: *Proceedings of the 2010 ACM symposium on applied computing*. pp. 311–317 (2010)
13. Mellor, S.J.: *MDA distilled: principles of model-driven architecture*. Addison-Wesley Professional (2004)
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. Rajora, N.: Pharmaceutical drug traceability by blockchain and iot in enterprise systems. *Universal Journal of Pharmacy and Pharmacology* pp. 11–18 (2023)
17. Rychkova, I., Kiomba Kambilo, E., Herbaut, N., Pastor, O., Noel, R., Souveyet, C.: Technology-aware enterprise modeling: Challenging the model-driven architecture paradigm. In: van der Aa, H., Bork, D., Schmidt, R., Sturm, A. (eds.) *Enterprise*,

- Business-Process and Information Systems Modeling. pp. 388–396. Springer Nature Switzerland, Cham (2024)
18. Vijayaraj, N., Rajalakshmi, D., Immaculate, P., Sathianarayani, B., Rajeswari, S., Gomathi, S.: An innovative approach to improve the quality of pharmaceuticals approach using cloud computing. *EAI Endorsed Transactions on Pervasive Health and Technology* **10** (2024)
 19. Zhou, R., Wang, W., Padoan, A., Wang, Z., Feng, X., Han, Z., Chen, C., Liang, Y., Wang, T., Cui, W., et al.: Traceable machine learning real-time quality control based on patient data. *Clinical Chemistry and Laboratory Medicine (CCLM)* **60**(12), 1998–2004 (2022)