

MULTI 2025 - 12th International Workshop on Multi-Level Modeling

7 October 2025

MODELS 2025, Michigan, USA



Consistent Modeling of Agility: Applications of Flexible Multi-Level Modeling

Razieh Dehghani, Anne Koziolk | October 7th, 2025

1. Background and Motivation

- Software Process Modeling
- Modeling Agile Processes
- Challenges
- Opportunities

2. Proposed Modeling Approach

- Project-Package Example of Usage (1)
- Project-Package Example of Usage (2)
- KPI Improvement Example of Usage (1)
- KPI Improvement Example of Usage (2)
- Challenge-Solution Pair 1 (1):
Assumption Validity_Intrinsic Associations
- Challenge-Solution Pair 1 (2):
Assumption Validity_Intrinsic Associations
- Challenge-Solution Pair 2 (1):
Non-Fixed Roles_Intrinsic Operations
- Challenge-Solution Pair 2 (2):
Non-Fixed Roles_Intrinsic Operations
- Challenge-Solution Pair 3 (1):
Need for Realistic Progress Measurement_Delegation Relations
- Challenge-Solution Pair 3 (2):
Need for Realistic Progress Measurement_Delegation Relations

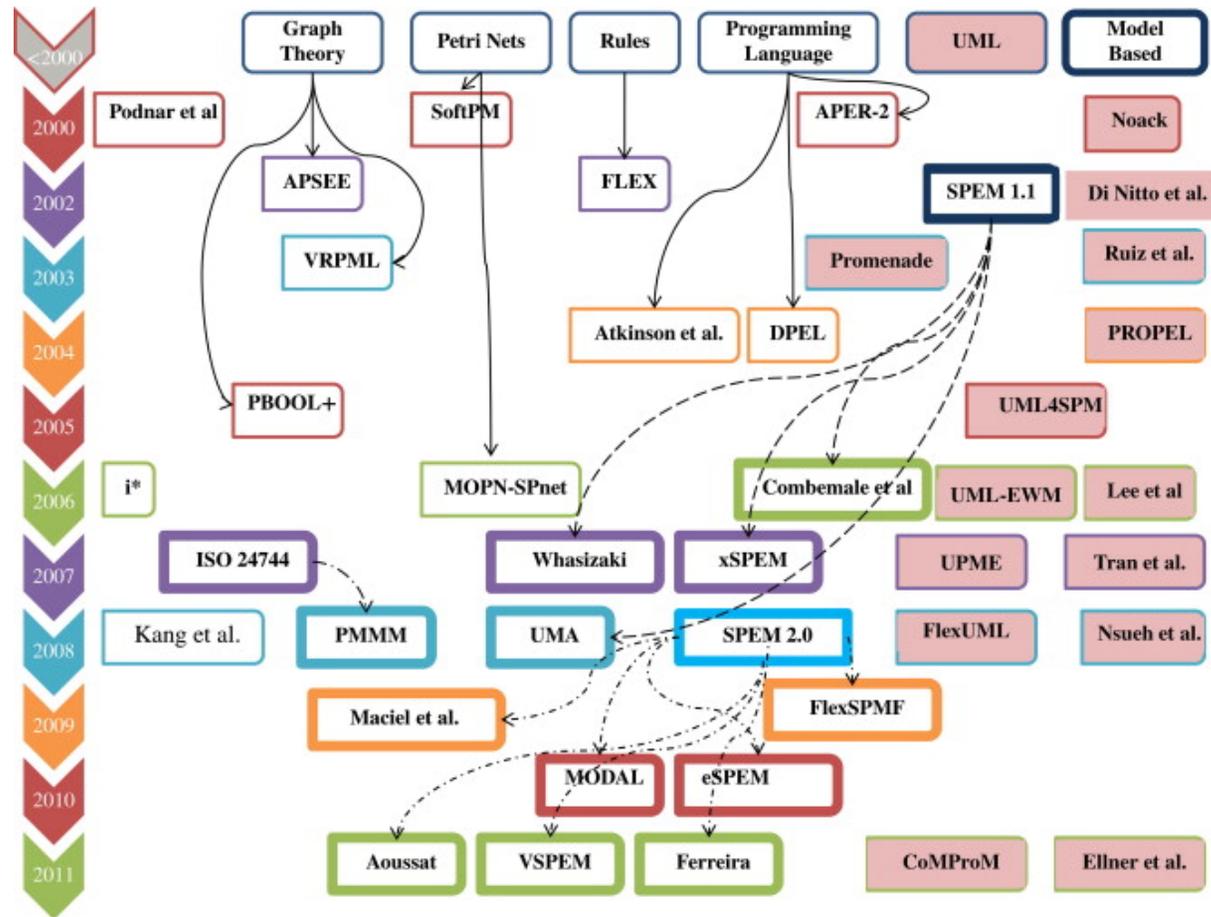
3. Comparative Analysis

- Flexible Modeling of Scrum Events
- Contextual Modeling – Scrum Meeting Room
- Delegation of Scrum Master Responsibility
- Modeling Dependent Associations in Scrum
- Illustration

4. Future Works



Software Process Modeling



derived from [2]

Background and Motivation

●○○○

Proposed Modeling Approach

○○○○○○○○○○

Comparative Analysis

○○○○

Future Works

○

References

Existence of various software process modeling languages [2]

Existence of open issues for future research [2]

"Need for verification of usefulness by measuring variability mechanisms and understandability in practice" [2]

Challenges

- Prerequisite for Adopting Agile Methodologies: Identifying the adaptation points and ensuring their alignment with overarching agile principles.
- Need to view agile methodologies within a broader framework to which all components of a project or organization must remain consistent.
- Traditional, non-multi-level modeling approaches fall short due to the need for variability in abstraction levels and the dependency of customizations on those levels. For instance, the roles in a Scrum project may need to be customized based on both the characteristics of the Scrum framework and the specific work package to which the project belongs.

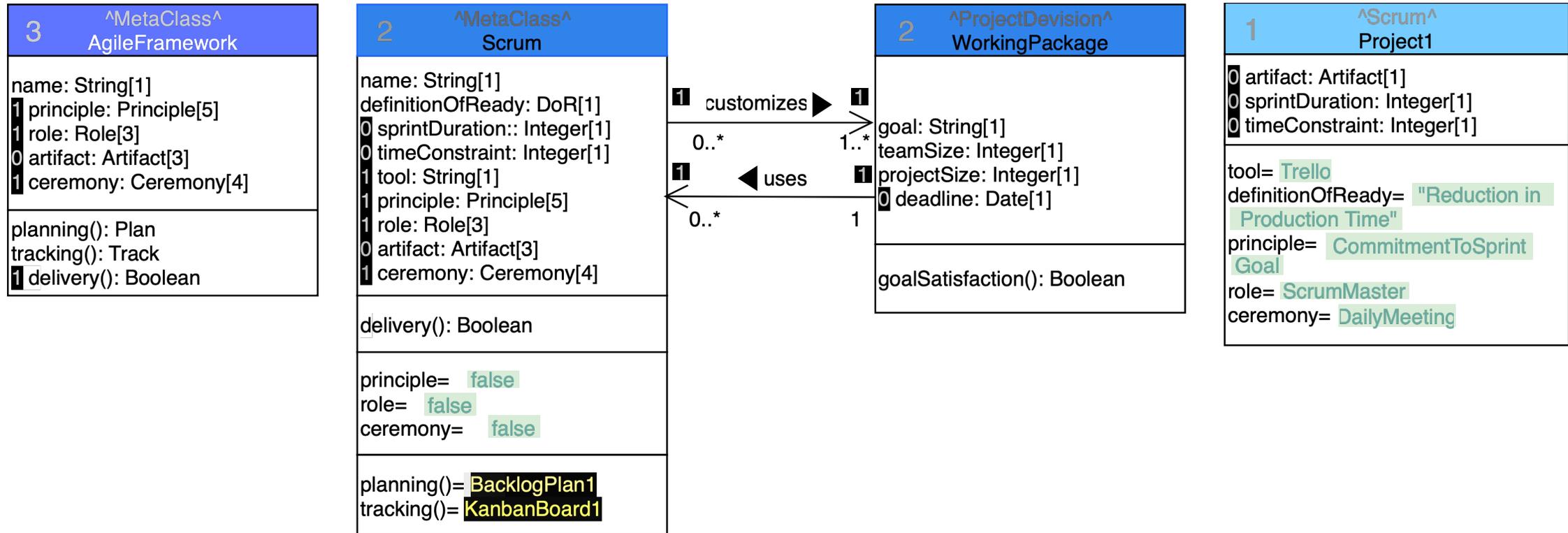
Opportunities

- Allows reuse of types across abstraction levels
- Various Supportive Tools/Approaches such as Melanee
- Need for Delegation due to involvement of dynamic roles and contexts in Agile Practices (Delegation allows context-specific behavior; e.g., role shifts, temporary ownership)
- Support for delegation by FMMLx

Project-Package Example of Usage (1)

- In large-scale projects, multiple working packages may exist, each consisting of distinct sub-projects.
- Every work package has its own objectives, team size, project scope, and deadline.
- The achievement of goals for each work package should be verifiable through its corresponding operations.
- The deadlines should be decidable at M0 level were a real project is in run.
- Depending the goals of a particular working package and the nature of its artifacts, different agile methodologies, such as XP, Scrum, or others, may be selected for individual sub-projects.

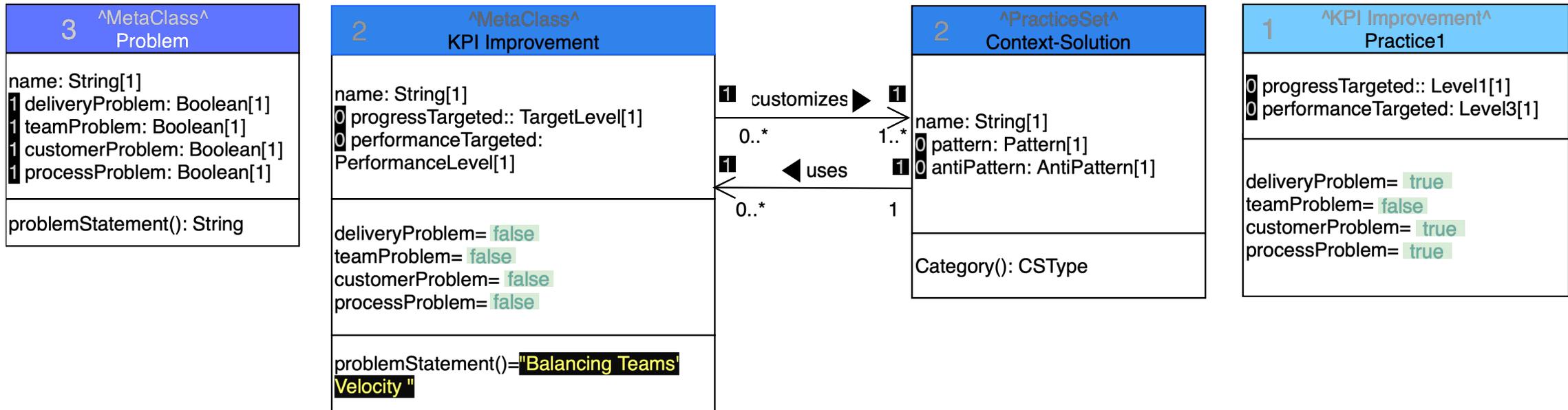
Project-Package Example of Usage (2)



KPI Improvement Example of Usage (1)

- How an agile practice can be modeled within a specific context to address a particular problem by improving a Key Performance Indicator (KPI)?
- A "Problem" can be categorized as delivery-related, team-related, customer-related, or process-related.
- "KPIs" aim to improve progress and performance levels.
- "Context-Solution" refers to a specific pattern or anti-pattern that can be applied in a given situation. It is characterized by a distinct name and category, and it should leverage KPIs to potentially link the identified improvement area to the proposed solution.

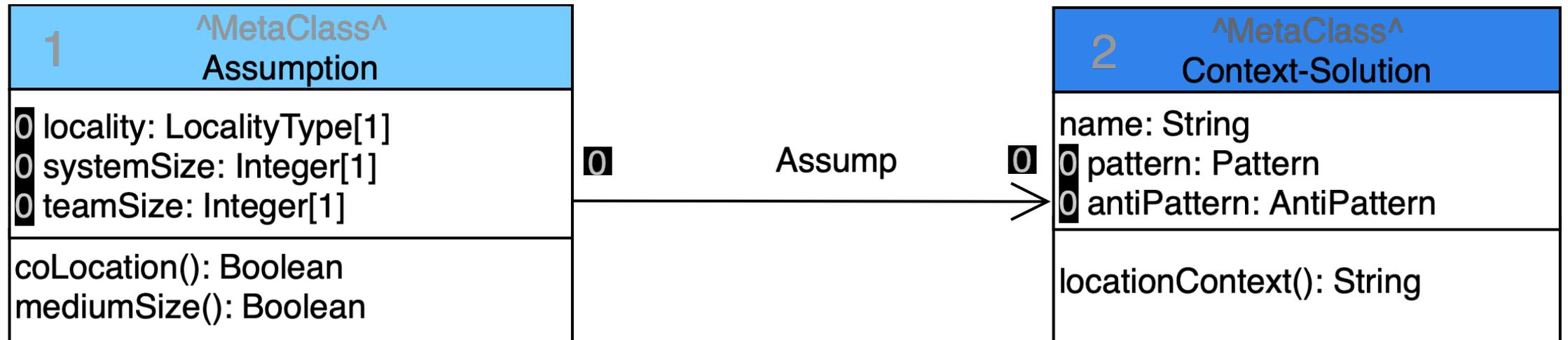
KPI Improvement Example of Usage (2)



Challenge-Solution Pair 1 (1): Assumption Validity_Intrinsic Associations

- Certain assumptions must be satisfied to effectively apply agile approaches. For example, it should be feasible for developers and customers to be co-located.
- FMMLx enables creation of association relationships across different levels of abstraction and allows control over the level at which instantiation occurs.
- Modeling assumptions as flexible associations. Assumptions about location and size will be considered at the lowest level where an appropriate pattern/anti-pattern should be used.

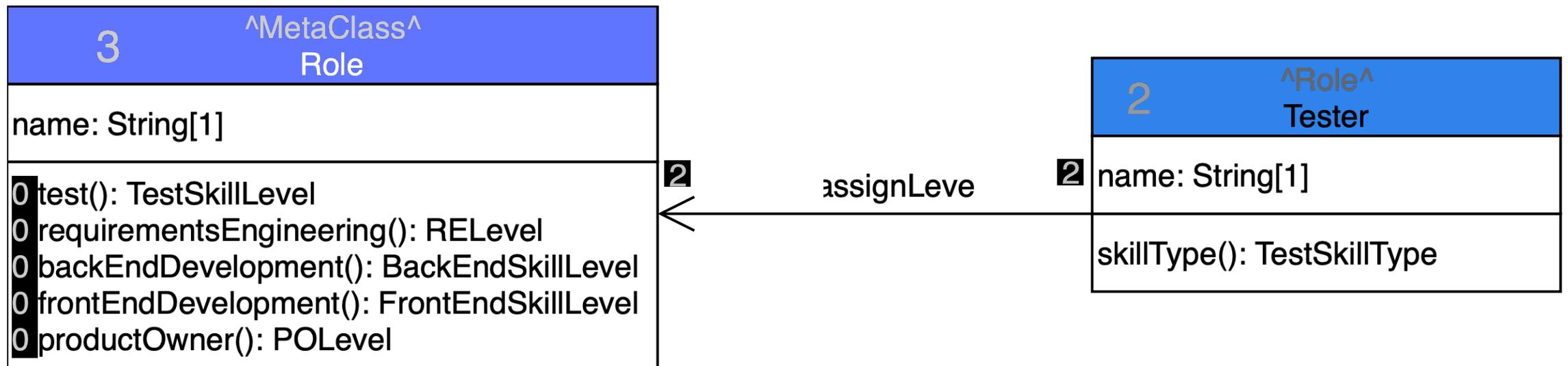
Challenge-Solution Pair 1 (2): Assumption Validity_Intrinsic Associations



Challenge-Solution Pair 2 (1): Non-Fixed Roles_Intrinsic Operations

- In Agile approaches, fixed roles with predefined responsibilities are replaced by Cross-Functional Teams (CFTs).
- Every team must be capable of developing a complete software component independently, from inception to deployment.
- This shift introduces flexibility, which can be modeled through intrinsic operations that are instantiated at specific levels.
- Operations of a Role (corresponding to various skills) could be instantiated at the lowest level to specify the skill level a person possesses in different domains.

Challenge-Solution Pair 2 (2): Non-Fixed Roles_Intrinsic Operations



Challenge-Solution Pair 3 (1): Need for Realistic Progress Measurement_Delegation Relations

- Measuring progress in an agile process is inherently more complex than in traditional methodologies.
- This complexity arises from factors such as various team velocities, the variability in how "Definition of Done" is defined, and the dynamics of cross-functional teams.
- It is often more effective to delegate progress measurement to appropriate stakeholders who are best positioned to assess specific aspects of the process.
- FMMLx supports using delegation for this purpose. Rather than relying on inheritance solely to gain the capability of progress measurement.

Background and Motivation
○○○○

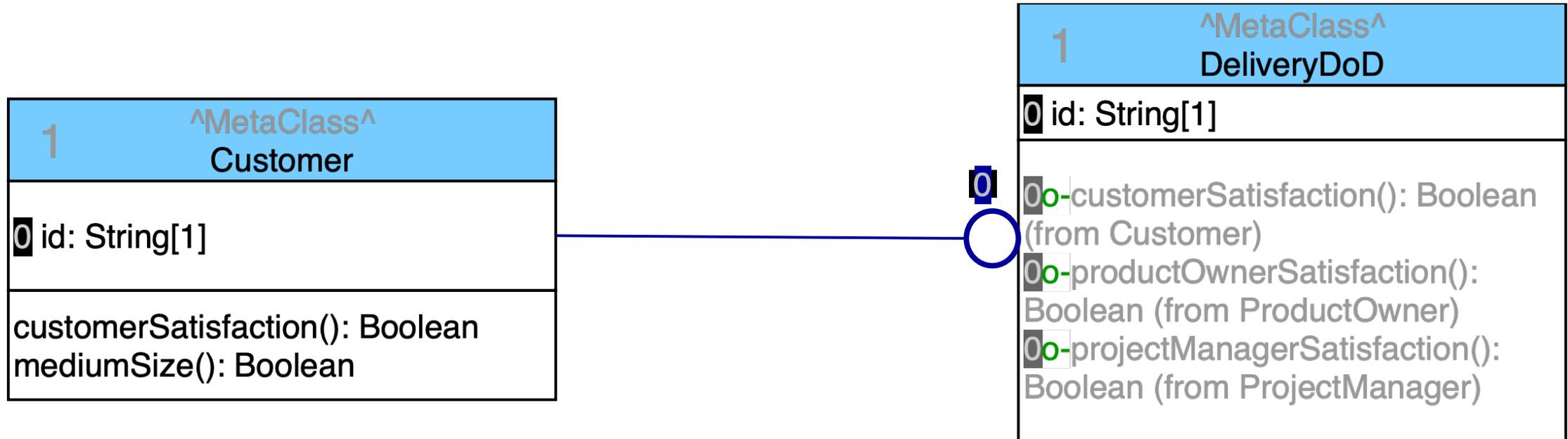
Proposed Modeling Approach
○○○○○○○○●○

Comparative Analysis
○○○○○

Future Works
○

References

Challenge-Solution Pair 3 (2): Need for Realistic Progress Measurement_Delegation Relations



Flexible Modeling of Scrum Events

■ Traditional (2-Level Modeling):

- Events as static classes.
- All events share same attributes (e.g., `Duration`), operations (e.g., `Schedule()`), and associations.
- Leads to inflexibility across event types.

■ FMMLx Approach:

- Attributes and operations defined as intrinsic types, instantiated only where needed.
- Example: `Schedule()` in *SprintPlanning*, `Duration` only in events like *Daily Stand-ups*.
- Enables context-sensitive, adaptive modeling.

Contextual Modeling – Scrum Meeting Room

■ Traditional Modeling Challenge:

- Project-specific data (e.g., meeting locations) clutter core models.

■ FMMLx Solution:

- Supports auxiliary classes like MeetingRoom.
- Keeps Scrum core concepts clean (roles, artifacts, activities).
- Allows flexible inclusion of contextual project information.

Delegation of Scrum Master Responsibility

■ Real-World Need:

- Scrum Master may delegate meeting facilitation to developers for training, self-organization, or absence.

■ FMMLx Capability:

- Supports delegation relationships.
- Preserves original role structure.
- Models dynamic responsibility shifts.

■ Example: Facilitation delegated to Scrum Master (main) and Developer (alternative).

Modeling Dependent Associations in Scrum

■ Scrum Principle:

- Product Owner owns the product backlog.
- Sprint backlog ownership depends on product backlog ownership.

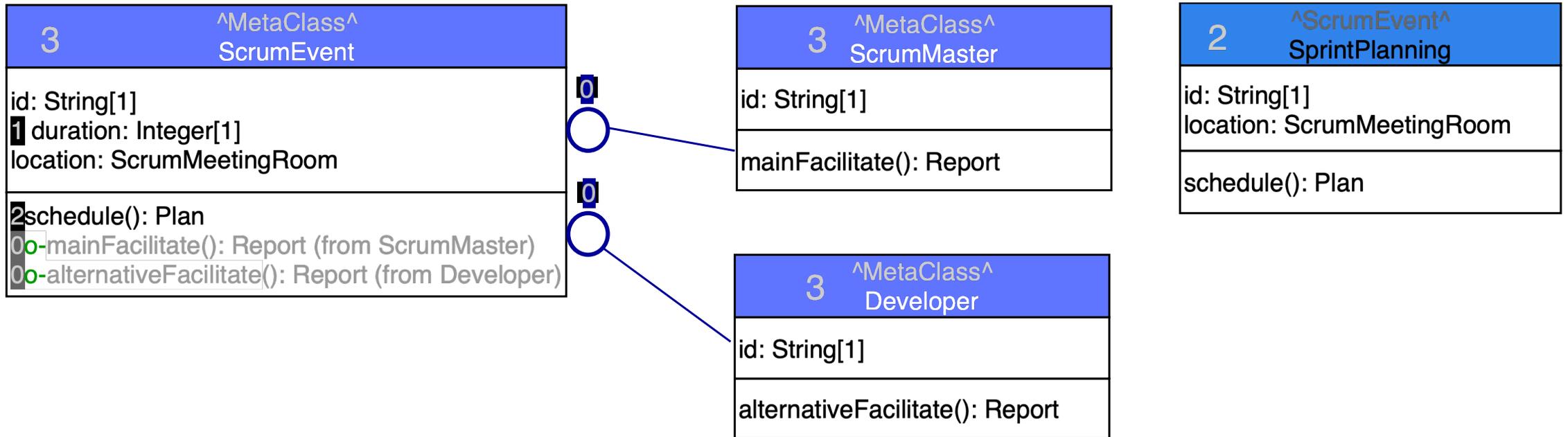
■ Limitations of Traditional Modeling:

- Can't easily express semantic dependencies between associations.

■ FMMLx Advantage:

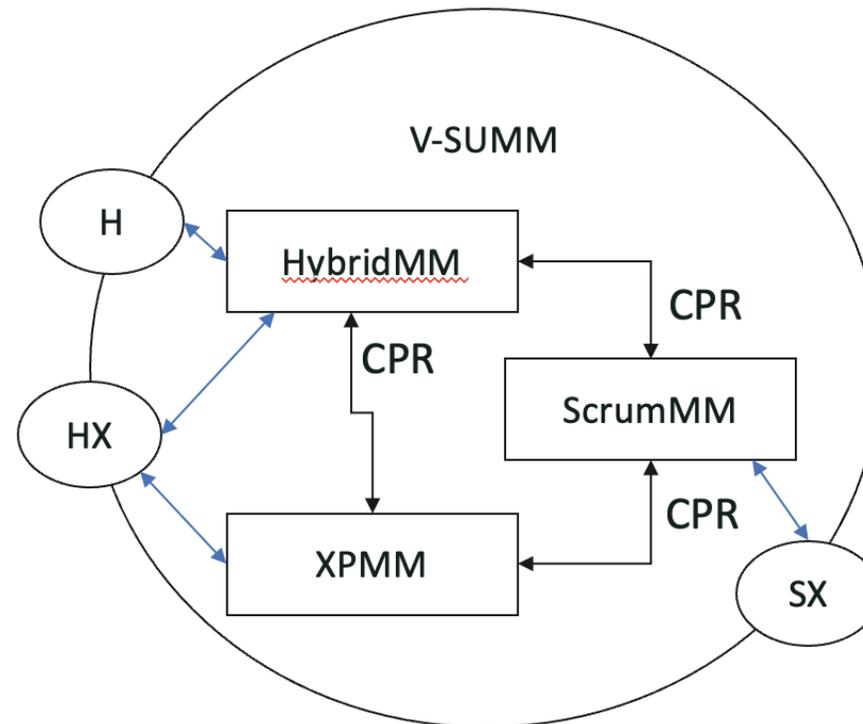
- Supports dependent associations with prerequisite relationships.
- Improves semantic clarity and expressiveness.

Illustration



Future Works

- Using Vitruvius: Defining consistency preservation rules across different metamodels of agile processes [3], [4]



References

- [1] Agile Alliance. *Subway Map to Agile Practices*. <https://agilealliance.org/wp-content/uploads/2025/07/agile-alliance-subway-map-to-agile-practices-2.pdf>. Accessed: 2025-10-07. 2025.
- [2] Laura García-Borgoñon et al. “Software process modeling languages: A systematic literature review”. In: *Information and Software Technology* 56.2 (2014), pp. 103–116.
- [3] Heiko Klare et al. “Enabling consistency in view-based system development—the vitruvius approach”. In: *Journal of Systems and Software* 171 (2021), p. 110815.
- [4] Thomas Weber et al. “Towards deep reactions in multi-level, multi-view modeling”. In: *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*. 2024, pp. 760–769.