Name : Isiram OUKAKI
Name of your level 1 : Morgane Gaspar
Tittle : Different Models for Model Matching, An analysis of approaches to support model differencing
Keywords specific : Model matching, Identity-based matching, Typed attribute graphs, Similarity-based matching, Feature similarity, Configuration, Custom algorithm, Signature-based approach, Accuracy, Effort, Trade-off, Practical experience, Experimentation, Modeling technologies, Infrastructure.

Different Models for Model Matching:
An analysis of approaches to support model differencing

Intro:
The increasing use of Model-Driven Engineering (MDE) highlights the need for effective versioning mechanisms to manage changes in model-based artifacts. However, existing document comparison algorithms, like those in CVS and SVN, often fall short in adequately capturing model differences. Recent research has focused on developing more robust model comparison techniques, particularly for UML diagrams and generalized metamodels. These techniques tackle the challenging task of calculating model differences, which relies on graph isomorphism—a computationally complex problem. In our analysis, we find that there's no one-size-fits-all solution for model matching. Instead, the best approach depends on factors like accuracy, efficiency, and domain independence. We review various model matching approaches, evaluate them against specific criteria, and discuss their applicability in real-world scenarios.

2 Model differencing in MDE

Determining model differences involves three phases: calculation, representation, and visualization.

- Calculation: Identifying differences between models using various methods.
- Representation: Converting differences into a usable format for analysis.
- Visualization: Presenting differences in a way that is easy to understand, from diagrams to textual data.

The paper focuses on difference calculation, particularly model matching, while noting the importance of representation for user readability

3 Current approaches to model matching

This approach assumes that each model element possesses a unique, persistent identifier assigned upon creation. Matching models relies on identifying elements based on these identifiers. Its advantages include requiring no user configuration and being fast. However, it's not suitable for models constructed independently or for representation technologies lacking support for unique identifiers.

3.2 Signature-Based Matching

In [17] **(see the annex on the document),** the authors acknowledge the limitations of static identity-based matching and propose signature-based matching instead. In this technique, each model element's identity is dynamically calculated from its feature values using a user-defined function specified in a model querying language. Unlike static identity matching, this method can compare independently constructed models. However, it requires developers to specify functions for calculating identities, adding configuration effort.

3.3 Similarity-Based Matching

Similarity-based matching treats models as typed attribute graphs and identifies matching elements based on feature similarity. However, not all features are equally important, so algorithms need configuration to specify feature weights. Examples include SiDiff, similarity flooding algorithm, and DSMDiff. These approaches generally yield more accurate results than identity-based matching, but fine-tuning feature weights is an empirical process and may not consider language semantics, which can enhance accuracy.

Conclusion : The selection of a model matching approach involves balancing accuracy with effort. When limited effort is available, a combination of identity and similarity-based algorithms like DSMDiff, SiDiff, or EMF Compare's built-in algorithm is a reasonable compromise. However, investing effort into configuring a similarity algorithm or implementing a custom signature-based approach can yield more accurate results. For cases where improved accuracy and performance justify significant effort, a custom matching algorithm based on infrastructure like EMF Compare or ECL is recommended.

These recommendations stem from practical experience with various matching algorithms and tools. However, directly comparable data is lacking due to differences in modeling technologies each algorithm/tool supports. For example, SiDiff is based on Fujaba and supports only UML 1.3 models, while DSMDiff is built on GME, and ECL/EMF Compare operate on EMF models.