Maria Lolita GONCALVES MELIM u21923049 (L1).

L0 : DEVARIEUX Lucas, BURLON Carla.

These articles explore various aspects of software engineering and artificial intelligence, with a particular focus on Intelligent Driver Model (IDM) model-driven engineering, which is a theory used in traffic engineering to model driver behavior in traffic. It aims to describe how drivers adjust their speed according to various factors such as distance from other vehicles, current speed and desired speed. This model is often used in traffic simulation and the design of traffic control systems. IDM is often formatted with megadata and AI for vector-borne disease prediction. The first article proposes a catalog of refactorings for model-to-model (M2M) transformations in IDM, highlighting the importance of maintainability in these operations. Indeed, direct communication between two devices without humans requires constant maintenance to automate processes and improve them over time.
Despite this, the lack of automation remains a major problem, even when predefined programs are applied.

The component model for model transformations, offering a reusable approach to their conceptualization and pipeline implementation. The article discusses the limitations of meta-models, which may require additional manual work. The method for adapting OCL constraints is therefore important. OCL, or Object Constraint Language, is a formal specification language used mainly in the field of model-driven engineering (MDE). It is used to define constraints and rules on models created using modeling languages such as UML (Unified Modeling Language). OCL can be used to specify conditions and invariants that must be respected by models to guarantee their consistency and conformity with system specifications. It is used to express static and dynamic constraints on model objects, as well as to define operations on these objects. OCL is often used in conjunction with other modeling languages to enrich the specification of software systems and facilitate the analysis and verification of their properties. They evolve from their meta-model within the IDM framework, aiming to maintain compliance and consistency of constraints throughout the evolution process. Examining the current state of AI in industries and proposing a "data ecosystem" framework to overcome data management and AI adoption challenges. Included are:
- Data collection.
- Data integration.
- Predictive modeling (disease risk prediction).
- Machine learning.
- Early warning system.
- Decision support tool.

The introduction of a DSL(designed to facilitate the extraction of different types of models from existing code bases, helping to understand the structure, behavior and dependencies of the software system.) would therefore be important, but is the subject of much debate. The functionalities and capabilities of the DSL, provides examples or case studies illustrating its application in real projects, and evaluates its efficiency and user-friendliness.

Finally, the last article explores the performance and scalability of EMF as a model comparison framework, highlighting the factors that influence its effectiveness in different scenarios. Here's a list of some scenarios:
- Model size (the more elements, the longer the processing time).
- Algorithm efficiency.
- Resource constraints (memory, CPU).
- Optimization techniques.
- Tool integration.
- User configuration and tuning.

In summary, these articles provide an in-depth overview of various areas of research and practice in software engineering and AI, covering topics ranging from model maintenance to vector disease prediction, data management and model comparison.