**Formalization of the Operation Recorder**

**based on**

**Graph Transformation Theory**

Sebastian Gabmeyer

---

**Introduction**

- **Model Driven Development (MDD)** places models at the heart of all software development processes.
- Methods to describe, control, and verify the evolution and manipulation of models are thus urgently required.
- The theory of **graph transformation** provides a rule-based approach to manipulate graphs.
- However, graph rewriting rules are **hard to get right** without adequate tool support.
- The **Operation Recorder** provides the required tool support, but lacks the required **formal background**.
- If an **alignment between the concepts** provided by the Operation Recorder and those provided by the theory of graph transformation is achieved, we’re one step closer to our goal!

---

**Running Example**

**Pull-Up Field Refactoring**

- Pull-up all fields of same name and type from the subclass level to the superclass.

---

**Graph Transformation Theory**

A graph rewriting rule is described by a left-hand side **L**, an interface graph **K**, and a right-hand side **R**.

Is this a valid rewriting rule?

Answer: Yes and no; it depends on the employed approach.

---

**Operation Recorder**

- A tool survey was conducted, which compared the features of ASG, EMF Tiger, Fujaba, and the Operation Recorder. Their modeling capabilities were tested with the Pull-Up field refactoring. The Operation Recorder won, Fujaba had its leaks.
- The Operation Recorder provides a **by-demonstration environment**, which allows the user to demonstrate the intended transformation.
- An operation is derived by comparing the initial and the revised model.
- Operations are described with pre- and postcondition templates.
- **Advanced templates:** optional, non-existence, and iteration templates as well as negative application conditions.

---

**Conceptual Alignment**

- **Aim:** Align the concepts provided by the Operation Recorder to those provided by the graph transformation theory.
- Establish a **measure of equivalence** to compare the concepts.
- Extend the theory to **template graphs**.

---

**Conclusion**

- A framework capable to transform software models, i.e., attributed graphs with inheritance, composition, and multiplicities, was gradually established.
- By embedding the Operation Recorder into this framework, standard, optional, and non-existence templates as well as negative application conditions were successfully aligned.
- The achieved alignment provides a formal foundation for the Operation Recorder allowing the verification of and reasoning about the modification it performs.

---


Contact: sg@myname.com