Domains specific modeling (DSM) raises the level of abstraction by specifying a metamodel that is aligned to a particular problem domain and constructing model interpreters that synthesize the domain models into software artifacts. In the presence of new stakeholder requirements, it is possible that a metamodel undergoes numerous changes during periods of evolution. Consequently, there is a fundamental problem in keeping the model interpreters up to date with these changes. This poster presents two research objectives to facilitate model interpreter evolution in the presence of metamodel changes:

- **Formalization of the interpreter implementation**
- **Evolution of the interpreter from model transformation specification**

### Key Challenges

**Challenge 1:** Lack of formally-written model interpreter
- Different developers may program interpreters in various ways
- Hard to maintain and evolve such subjective realizations of model interpreters

**Challenge 2:** Lack of formal specification for metamodel transformation
- Metamodel transformation specifications must include the entire knowledge for the underlying interpreter evolution
- \( \Delta_{\text{old}} \xrightarrow{\Delta_{\text{new}}} \Delta_1 \)

### Technical Approach: Interpreter Evolution from Model Transformation Specification

- Model transformation specification is used to define the transformation of the models that conform to two different metamodels. It is composed of pattern specification, replacement rule, and a set of constraints.
- The interpreter transformer analyzes the model transformation specifications, invokes the pattern matcher to locate the constituents to be transformed within the traversal strategy model, and eventually updates them according to the replacement rule specifications.
- User actions should remain intact because they embody the semantic intuition of a model interpreter.
- The transformed traversal strategy and user action code will be used to generate the new interpreter that can work under the new metamodel with the preserved semantics intuition.

### Case Study: Evolution through Specialization of Domain Concepts

**Old Domain**

```plaintext```
Model 1
```

**New Domain**

```plaintext```
Model 2
```