

## 0.1 Semantische Verarbeitung des AST

Für die korrekte Verwendung des ASTs benötigt das DMF auch eine Softwarekomponente, welche den AST durchläuft und ein semantisches Modell erstellt. Anhand dieses semantischen Modells werden semantische Regeln überprüft.

Diese Softwarekomponente ist die erste Komponente welche mithilfe von Golang implementiert wird.

### 0.1.1 Das semantische Modell

Das semantische Modell bildet alle Informationen, die aus dem AST entnommen werden können, ab. Dazu gehören Referenzen zum AST für Positionen in der Modelldatei, die verschiedenen PackageElemente und die NamensElemente.

### 0.1.2 Übersetzen des ASTs

Die Übersetzung des ASTs beginnt mit dem Erstellen eines SemanticContext. Dieser Kontext beinhaltet die erkannten Fehler, das bisherige Modell, den Text der Modelldatei und die TreeCursor. Mithilfe des TreeCursors kann der AST durchlaufen werden. Der Kontext durchläuft den AST in der PreOrder-Reihenfolge. Für jedes Element des ASTs enthält der Kontext eine Methode zum Parsen.

Sollte der AST Import Statements beinhalten so werden zunächst die referenzierten Modelle verarbeitet. Die importierten Packages werden nun in das Modell übernommen. Das Laden der verschiedenen Modelle wird mithilfe einer Callback Struktur außerhalb der Semantik-Komponente definiert. So können verschiedene Logiken genutzt werden.

Am Ende des Parsen ist Modell vollständig mit allen gültigen Elementen. Elemente die durch den Parser im AST als Fehlerhaft gekennzeichnet wurden, werden ignoriert.