

# Software Engineering: Tipps zu PlantUML

Miron Schmidt, Sommersemester 2025

PlantUML funktioniert normalerweise sehr flüssig, es gibt aber ein paar ärgerliche Probleme, die sehr viel Zeit kosten können. Die Verteilung der Entitäten im Diagramm kann PlantUML überfordern, vor allem in Verbindung mit Paketen oder komplexen Assoziationen. (Ich fasse hier alle Arten von Dingen, die miteinander in Bezug gesetzt werden können, als Entitäten zusammen, da es ja verschiedene UML-Diagramme gibt, in denen nicht nur Klassen vorkommen können.)

Hier sind ein paar Tipps, mit denen man den Frust im Zaum halten kann, wenn PlantUML nicht so will wie man selbst.

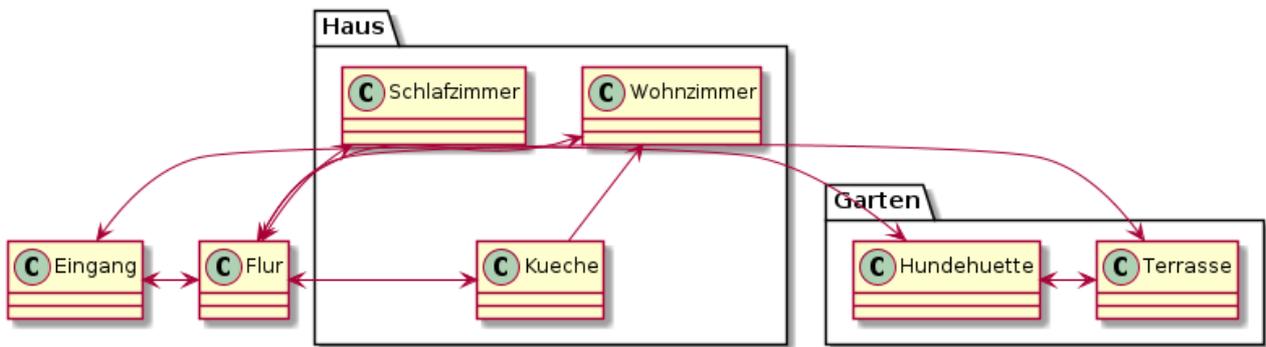
Zuerst ein allgemeiner Hinweis: Die *Form* einer Verbindung (mit/ohne Pfeil, Vererbung, gestrichelt usw.) ist beim Auftreten von Schwierigkeiten unerheblich. Es geht nur um das Bilden von Pfaden zwischen den Entitäten, das problematisch sein kann – daher spielt die *Richtung*, also die Seite(n), an der bzw. denen der Pfeil sitzt, sehr wohl eine Rolle. Wenn Entitäten, die miteinander in Bezug gesetzt werden müssen, aus organisatorischen Gründen weit auseinanderliegen, schlimmstenfalls mehrmals in einem Diagramm, ist es ebenfalls schwieriger, sinnvolle Verbindungen zu finden, als wenn sie geordnet nebeneinander zu finden sind.

Mit Paketen kommt PlantUML etwa prinzipiell gut zurecht, wenn die Pfade aber kompliziert sind, werden einzelne Entitäten mitunter falsch paketierte.

Ein Beispiel: Dieser Code –

```
class Eingang
Eingang <-> Flur
package Haus {
  class Wohnzimmer
  Wohnzimmer <-> Flur
  Wohnzimmer <-- Kueche
  class Kueche
  Kueche <-> Flur
  class Flur
  class Schlafzimmer
  Schlafzimmer <-> Flur
}
package Garten {
  class Hundehuette
  Hundehuette <-> Terrasse
  Hundehuette <-> Eingang
  class Terrasse
  Terrasse <-> Flur
}
```

- führt zu diesem Diagramm:

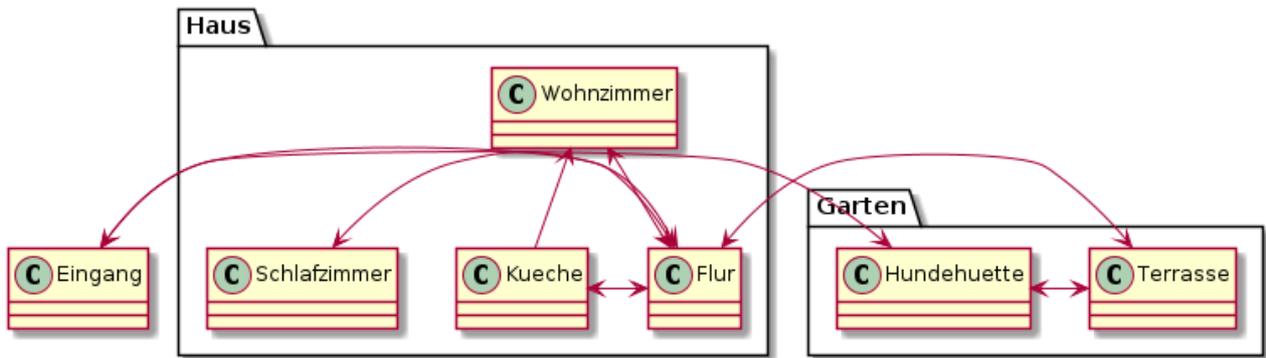


Man beachte, dass die Klasse *Flur* in der Darstellung aus dem Paket *Haus* herausgezogen wurde. (Die Tür zwischen Küche und Wohnzimmer ist in dieser Demonstration übrigens eine Klapptür, die nur aus der Küche ins Wohnzimmer durchlaufen werden kann.)

Der wichtigste Tipp ist daher: Alle Assoziationen sollten ans Ende der Datei gestellt werden. PlantUML ist besser darin, sie auf diese Weise nacheinander zu berücksichtigen, als wenn sie nahe bei den Entitäten platziert werden (was für menschliche Augen allerdings weniger strukturiert aussieht).

Das Beispiel sieht nun so aus:

```
class Eingang
package Haus {
    class Wohnzimmer
    class Kueche
    class Flur
    class Schlafzimmer
}
package Garten {
    class Hundehuette
    class Terrasse
}
Eingang <-> Flur
Wohnzimmer <-> Flur
Wohnzimmer <-- Kueche
Kueche <-> Flur
Schlafzimmer <-> Flur
Hundehuette <-> Terrasse
Hundehuette <-> Eingang
Terrasse <-> Flur
```



Die Pakete stimmen jetzt, aber die Assoziationen, die im ersten Versuch schon sehr wirr aussahen, sind eher noch unübersichtlicher als vorher.

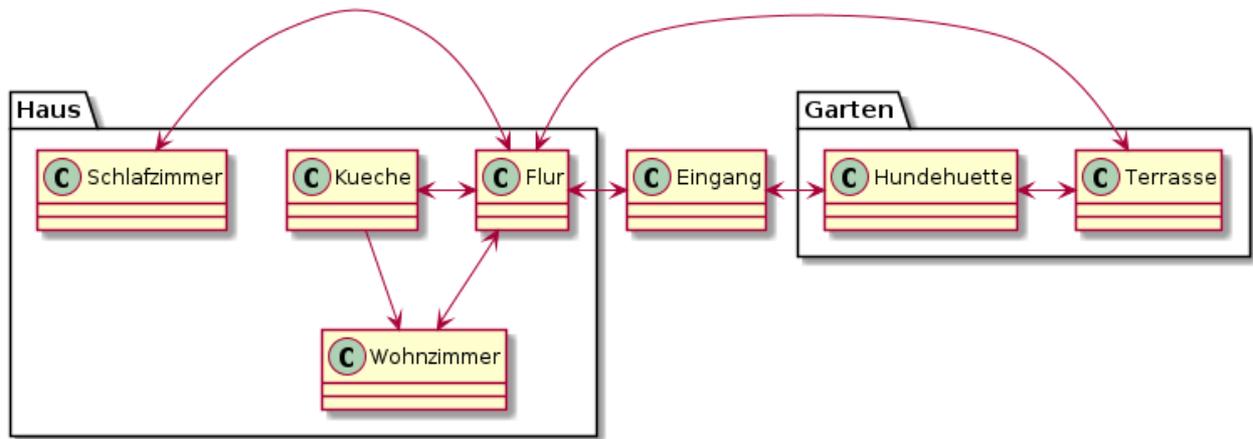
Entitäten, die von PlantUML an eine ungünstige Stelle gesetzt werden, und Assoziationen zwischen den Entitäten können im Code von Hand verschoben werden. Manchmal finden sich dann bessere Verbindungen als vorher. (Die Logik dahinter ist nicht sofort eingängig, aber durch Ausprobieren erahnt man nach und nach Regeln.)

Nach einigem Schieben kommen wir zu folgendem Code und dem daraus resultierenden Diagramm, mit dem wir uns absolut zufrieden geben können:

```

class Eingang
package Haus {
  class Flur
  class Wohnzimmer
  class Schlafzimmer
  class Kueche
}
package Garten {
  class Terrasse
  class Hundehuette
}
Flur <-> Eingang
Flur <-> Terrasse
Flur <-> Wohnzimmer
Kueche --> Wohnzimmer
Kueche <-> Flur
Schlafzimmer <-> Flur
Hundehuette <-> Terrasse
Hundehuette <-> Eingang

```



Hoffentlich klar ist: Eine Anpassung der fachlichen Ausarbeitung (Assoziationen, Pakete) sollte nur als allerletzter Ausweg in Betracht gezogen werden. Normalerweise reichen die zuvor genannten Tipps aus, um das Diagramm zu reparieren.

Und damit wünsche ich viel Spaß beim eigenen Ausprobieren dieser Hinweise! Der Beispielcode eignet sich übrigens gut zum Experimentieren, da ja schon einige Änderungen mit ihren Ergebnissen vorgeführt wurden.