Testautomatisierung von Co-Simulationsverbünden

Christoph Heidelbach, Markus Graube, Frederic Bloch

TraceTronic GmbH
Stuttgarter Str. 3
01189 Dresden
christoph.heidelbach@tracetronic.de
markus.graube@tracetronic.de
frederic.bloch@tracetronic.de

Abstract: Co-Simulationssysteme ermöglichen in bestimmten Anwendungsfällen eine frühere Integration von modularen Software-Projekten – auch über Domänengrenzen hinweg. So können Integrations- sowie Systemtests eher ausgeführt und detektierte Fehler schneller an die Entwicklung zurückgemeldet werden. Zur Ausführung der Tests wird in der Regel ein bewährtes Testautomatisierungstool (TAT) eingesetzt.

Der Beitrag erläutert die verschiedenen technischen Eigenschaften von Co-Simulationen, skizziert deren möglichen Lebenszyklus für den produktiven Einsatz im automobilen Umfeld und beschreibt wie die Integration von TATs in diese Art von Testsystemen zukünftig gelingen kann. Der Fokus liegt dabei insbesondere auf den Anbindungsarten (Kontrolle, Daten, Zeit, Konfiguration), welche durch die TATs unterstützt werden müssen und wie dies mit dem Schnitt des Co-Simulationssystems (High-Cut vs. Low-Cut) zusammenhängt.

1 Einleitung

Komplexe Fahrzeugfunktionen besitzen einen großen Eingaberaum, der hochgradig vernetzte Prozesse innerhalb des Fahrzeugs und die Umwelt einschließt. Funktionstests auf jeder Entwicklungsstufe sind daher elementar. Während Komponententests oftmals gut an Entwicklungstools angebunden werden können, sind für nachfolgende Integrationstests meist zusätzliche Werkzeuge nötig. Häufig sind diese jedoch nicht systemkompatibel, vor allem wenn entwicklerspezifische Domänen (z.B. Software, Modell) verlassen und reale Komponenten in die Testumgebung eingefügt werden (z.B. Hardware).

Mit Co-Simulationen ist der Übergang zwischen unterschiedlichen Domänen einfacher zu realisieren, da sie die einzelnen Systeme unter Beibehaltung der bestehenden Schnittstellen miteinander verbinden. So ist die Ausführung von Integrations- und Systemtests frühzeitig möglich.

2 Co-Simulation – Stand der Technik

Der Einsatz von Co-Simulationen erfolgt insbesondere beim Testen von Fahrfunktionen in Assistenzsystemen. Deren Tests benötigen ein komplexes Fahrzeug- und Umgebungsmodel, um den Eingabe- bzw. Ausgaberaum der Fahrassistenzfunktion möglichst umfangreich abdecken zu können. Die einzelnen Modelle werden durch unterschiedliche Simulationstools implementiert. Co-Simulationsframeworks lassen nun die Modelle verschiedener Domänen (vgl. Abbildung 1) miteinander kommunizieren und ermöglichen deren Verwendung in einer gemeinsamen Simulation, der Co-Simulation (mehr Details bei [GTB+17]). Die Teilnehmer an einer Co-Simulation müssen sich auf einige Grundmechanismen einigen, die üblicherweise durch das Co-Simulationsframework bereitgestellt werden:

- Datensynchronisation (Simulationsdaten): Die Hauptfunktionalität einer Co-Simulation ist der Austausch von Daten zwischen den Teilnehmern. Dabei werden verschiedene Arten sowie Strukturen von Daten übertragen und wieder richtig interpretiert. Die Übertragung sollte dabei möglichst performant stattfinden, um die Nachteile der notwendigen Datenübertragung klein gegenüber den Vorteilen der parallelisierten Berechnung zu halten. Weiterhin soll es möglich sein, die Ausgänge von Teilnehmern mit Eingängen von anderen Teilnehmern zu verschalten – im Idealfall ohne, dass diese sich vorher auf einen gemeinsamen Namen geeinigt haben. Das ermöglicht eine Wiederverwendung von Teilnehmern.
- Zeitsynchronisation (Simulationszeit): Eine weitere wichtige Eigenschaft von Co-Simulation ist die Synchronisierung der internen Zeiten der Teilnehmer. Die Zeitbasis der Teilnehmer darf nicht zu weit auseinanderlaufen, um die Simulationsergebnisse nicht zu stark zu verfälschen. Dies kann zum Beispiel durch unabhängige Uhren mit regelmäßiger Synchronisierung oder durch eine zentrale diskrete Uhr passieren. Dabei ist die Simulationszeit für SiL-Systeme üblicherweise unabhängig von der Realzeit (wall clock). HiL-Systeme erfordern hingegen eine harte Kopplung an die wall clock.
- Zustandsmaschine (Simulationszustand): Die Discovery (wer ist Teilnehmer der Co-Simulation), Konfiguration, Initialisierung und der Simulationsstart mehrerer Teilnehmer werden durch eine Zustandsmaschine synchronisiert. Die Bedeutung der einzelnen Zustände ist spezifiziert und das gültige Verhalten muss von allen Teilnehmern implementiert werden. Je nach Zustand kann bspw. ermittelt werden ob bereits Daten- oder Zeitsynchronisation bestehen sollte. Darüber ermöglicht die Zustandsmaschine erst eine wirkungsvolle Automatisierung von Co-Simulationsverbünden.

Die Sicherstellung oder zumindest Prüfung einer deterministischen Simulation in einem Co-Simulationsverbund erfordert eine Verknüpfung von Zeitsynchronisation mit Datensynchronisation.

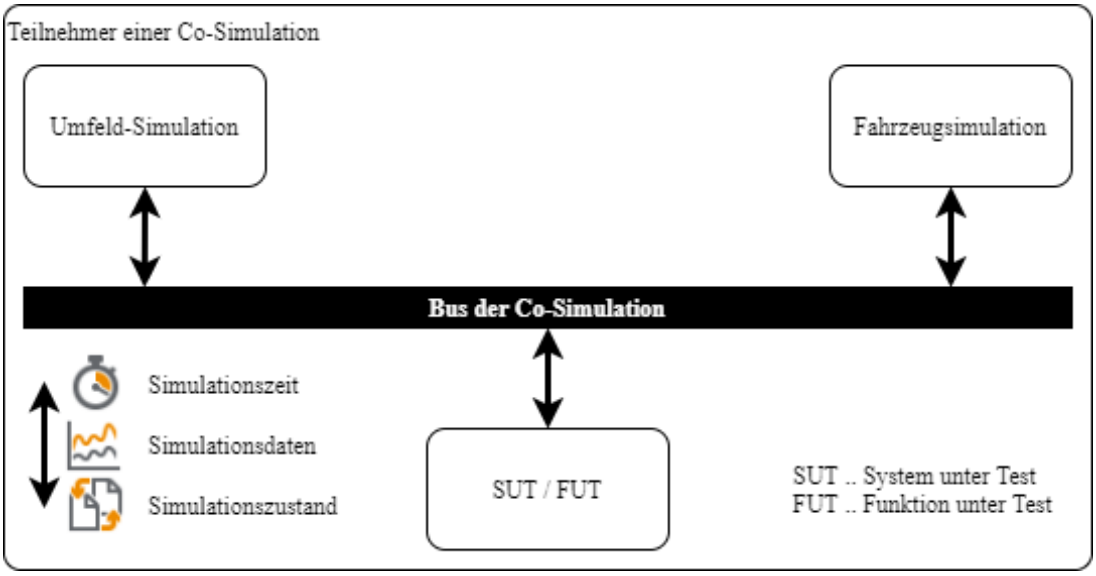

Abbildung 1: Teilnehmer eines Co-Simulationsverbundes

Wenn ein Simulationsteilnehmer einen neuen Zeitschritt berechnen soll, aber die entsprechenden Daten für diesen Zeitstempel auf Grund von Netzwerklatenz noch nicht eingetroffen sind, wird die Gesamtsimulation nicht-deterministisch. Dies muss entweder erkannt werden (in HiL-Systemen) oder der Start des nachfolgenden Simulationsschritt muss verzögert werden, bis die Daten eingetroffen sind (SiL-Systeme).

Anwendungsfeld

Im Fahrzeugumfeld sind Co-Simulationen insbesondere für hochkomplexe Anwendungsfälle, z. B. für automatisiertes Fahren, relevant [GGF19]. Unter Beachtung der Rückwirkung auf das Umfeld und der möglichen Kopplung mehrerer vECUs wird hierfür eine Umfeld-Simulation in die Fahrzeugsimulation eingespeist.

Im ersten Ansatz berechnet die Fahrzeugsimulation das physikalische Verhalten des Fahrzeugs und Fahrentscheidungen. Die Umfeld-Simulation berechnet alle äußeren Einflüsse auf das Fahrzeug und das Testobjekt. Das können weitere Verkehrsteilnehmer sein, unterschiedliche Wetterbedingungen oder andere physikalische Verhalten. Die Simulation des Testobjektes nutzt die Ergebnisse aus Umfeld- und Fahrzeugsimulationen und beeinflusst so das Verhalten der Fahrzeugsimulation.

Für die Simulation der Umgebung, des Fahrzeugs und des Testobjekts werden üblicherweise unterschiedliche Simulationswerkzeuge verwendet. Mithilfe der Co-Simulation lassen sich diese Tools jedoch einfach koppeln. Dadurch ist es später auch leichter, zusätzliche Sensoreinspeisungen hinzuzufügen, die das SUT nutzen kann.

Datenebene

Je nach Anwendungsfeld und verwendeten Co-Simulationsteilnehmern werden Daten in unterschiedlichen Abstraktionsgraden ausgetauscht. Bei der Kopplung von vECUs in höheren Leveln werden Nachrichten im spezifischen Busformat (Low-Cut) ausgetauscht, wohingegen bei niedrigen Leveln generische Signale ohne konkrete Berücksichtigung der Buscharakteristika (High-Cut) ausgetauscht werden [VDA710]. Ähnlich erfolgt der Austausch zwischen Umfeld- und Fahrzeugsimulation. Da es aber hier keinen Fahrzeugbus gibt, werden Nachrichten auf einer signalorientierten Ebene versendet. Damit besteht eine Kommunikation auf beiden Leveln, die sowohl die Co-Simulation als auch eine Testautomatisierung unterstützen muss.

3 Lebenszyklus einer Co-Simulation

Für einen erfolgreichen Einsatz von Co-Simulationen in einem produktiven Umfeld sind neben der eigentlichen Simulation weitere Schritte zu berücksichtigen. In Summe ergibt sich so ein abgeschlossener Prozessablauf, der als „Lebenszyklus Co-Simulation“ definiert wird (vgl. Abbildung 32).

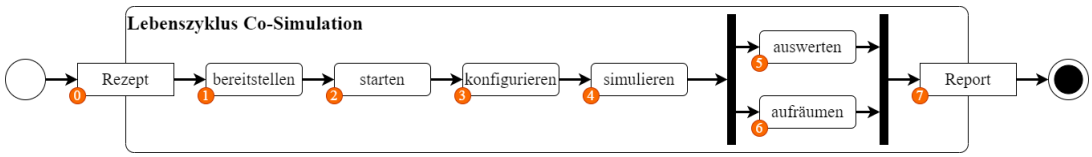

Abbildung 2: Lebenszyklus einer Co-Simulation

0. Der Lebenszyklus beginnt mit einem Rezept, der Aufbau und Konfiguration des Co-Simulationssystems spezifiziert. Dieser Bauplan ermöglicht die Automatisierung der Co-Simulation. Darin enthaltende Daten werden zu unterschiedlichen Zeitpunkten im Lebenszyklus abgerufen.
1. Softwarekomponenten, die in dem Rezept eingetragen sind, müssen auf den entsprechenden Ausführungsumgebungen installiert bzw. bereitgestellt werden. Dieser Vorgang wird direkt durch den Systemaufbau der Co-Simulation, bspw. der Zielplattform, beeinflusst.
2. Die einzelnen Softwarekomponenten werden anschließend gestartet und treten dem Co-Simulationsbus bei. Sie können nun über den Co-Simulationsstack kontrolliert werden. Das Verhalten von Software kann während des Starts, z.B. über Umgebungsvariablen oder Kommandozeilenargumente, parametrisiert werden. Diese Daten können im Rezept hinterlegt werden.
3. Co-Simulationsspezifische Attribute, wie Schrittweite, Zeitkonfiguration, Startwerte oder Umgebungsszenario, lassen sich direkt vor der eigentlichen Simulation konfigurieren. Damit sind verschiedene Parametervariationen umsetzbar. Diese Daten können im Rezept hinterlegt werden.
4. Die eigentliche Simulation wird ausgeführt. Das Co-Simulationsframework stellt die Daten- und Zeitsynchronisation zwischen den verschiedenen Softwarekomponenten sicher.

5. Die während der Simulation erzeugten Daten müssen bewertet werden. Dies zu automatisieren kann mitunter sehr kompliziert sein, da verschiedenste Komponenten den Verlauf der Simulation beeinflussen. Hier müssen sowohl Konfigurationsdaten aus dem Rezept, als auch die verwendeten Szenarien berücksichtigt werden. Die Auswertung der Simulationsdaten muss dabei nicht zwangsläufig auf der Simulationshardware erfolgen. Sie kann stattdessen ausgelagert werden. Wichtig ist auch die Sicherung der Simulationsergebnisse im Sinne der Traceability.
6. Parallel dazu wird damit begonnen, den Ursprungszustand auf der verwendeten Hardware wiederherzustellen. Dies kann je nach Systemaufbau unterschiedlich erfolgen. Ziel ist es, dass keine Softwareartefakte oder laufenden Prozesse zurückbleiben, die die nachfolgenden Co-Simulationsdurchläufe kompromittieren.
7. Der Lebenszyklus endet mit einem Report, in dem die Ergebnisse des Co-Simulation menschenlesbar aufbereitet wurden. Dieser Report enthält auch alle notwendigen Daten, um die Co-Simulation in der gleichen Konfiguration wiederholt ausführen zu können.

4 Co-Simulation und Testautomatisierung

Zur Ausführung der Tests auf der Co-Simulationsplattform wird in der Regel ein bewährtes Testautomatisierungstool eingesetzt. Dabei kann es ausreichend sein, dass das TAT nur einen kleinen Teil der prinzipiell möglichen Simulationskopplungen (*Zeit*, *Daten* und *Zustandsmaschine*) und Lebenszykluskopplungen (*bereitstellen*, *starten*, *konfigurieren*) an die Co-Simulation unterstützt.

Simulationskopplung

Die Anbindung einer Testautomatisierungslösung an ein Co-Simulationssystem kann entweder durch ein Interface zwischen TAT sowie Co-Simulationsbus und durch ein Interface zu den Simulationstools, die in das Co-Simulationssystem eingebunden sind, erfolgen (siehe Abbildung 3).

Da viele TAT bereits Anbindungen an übliche Simulationswerkzeuge aus dem Automotive-Bereich bieten, ist eine direkte Kopplung des TAT mit diesen Werkzeugen eine gute Möglichkeit (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**), mit der Co-Simulation zu interagieren. Je nach

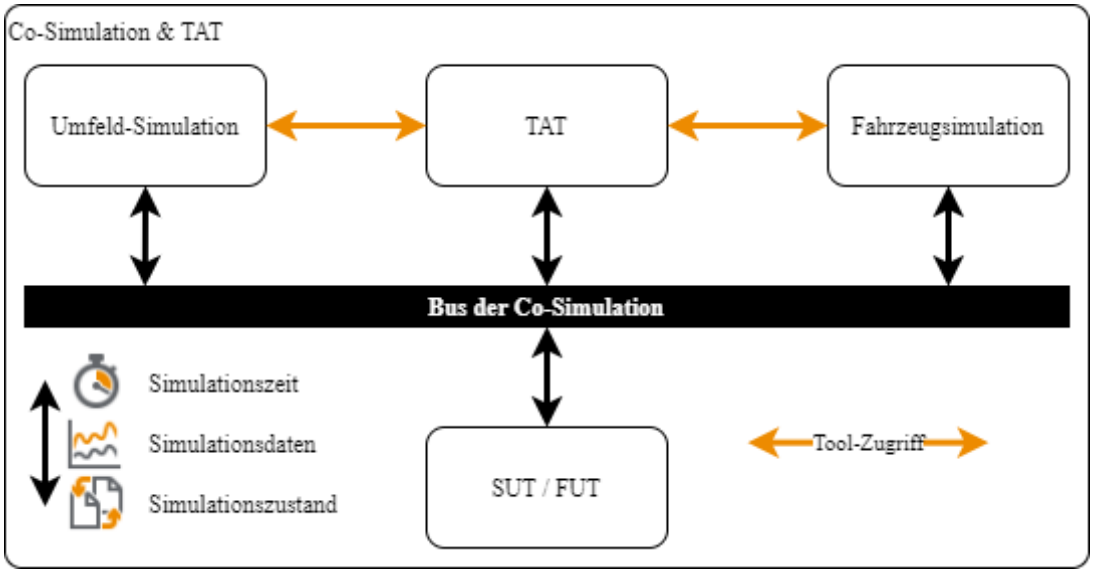

Abbildung 3: TAT im Co-Simulationsverbund

Simulationswerkzeug können sich jedoch Schwierigkeiten ergeben, das Werkzeug zeitgleich an das Testautomatisierungstool und ein Co-Simulationssystem zu koppeln.

Beide potentiellen Teilnehmer wollen nämlich Zugriff und Einflussmöglichkeiten auf Daten und Zeit des Simulationswerkzeugs. In diesem Fall, oder wenn es schlicht (noch) keine direkte Simulationswerkzeug-TAT-Kopplung gibt, ist eine Anbindung des TAT an den Co-Simulationsbus erforderlich. Weiterhin ermöglicht diese direkte Kopplung den Einsatz verschiedener TAT in unterschiedlichen Co-Simulationssystemen.

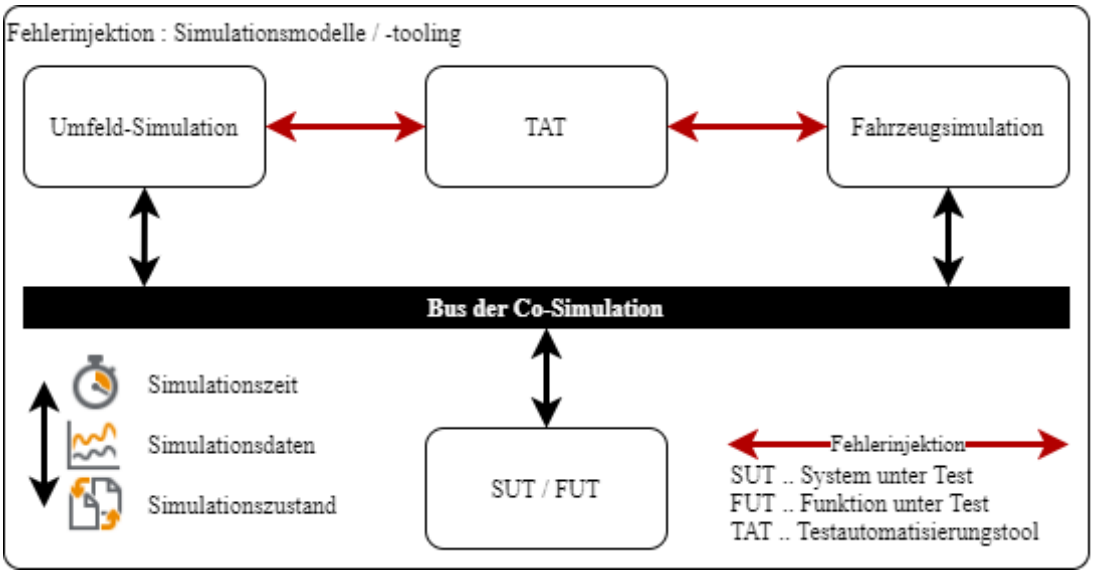

Abbildung 4: Fehlerinjektion über Toolschnittstellen

Je nach Anwendungsfall benötigt ein TAT, neben dem Tool-Zugriff doch direkten Zugriff auf den Co-Simulationsbus: z. B. wenn die Simulationswerkzeuge auf verteilten Hosts laufen und von mehreren Daten im Test benötigt werden (siehe Abbildung 5**Fehler! Verweisquelle konnte nicht gefunden werden.**). In diesem Fall

muss das TAT üblicherweise selbst Teilnehmer im Co-Simulationsverbund sein und somit auch die Zeitsynchronisierung und die entsprechende Zustandsmaschine implementieren. Dabei ergibt sich die Situation, dass das TAT einerseits die Zustandsmaschine für die anderen Teilnehmer steuert als auch selbst durch diese Zustandsmaschine gesteuert wird. Durch eine entsprechende Entkopplung in der

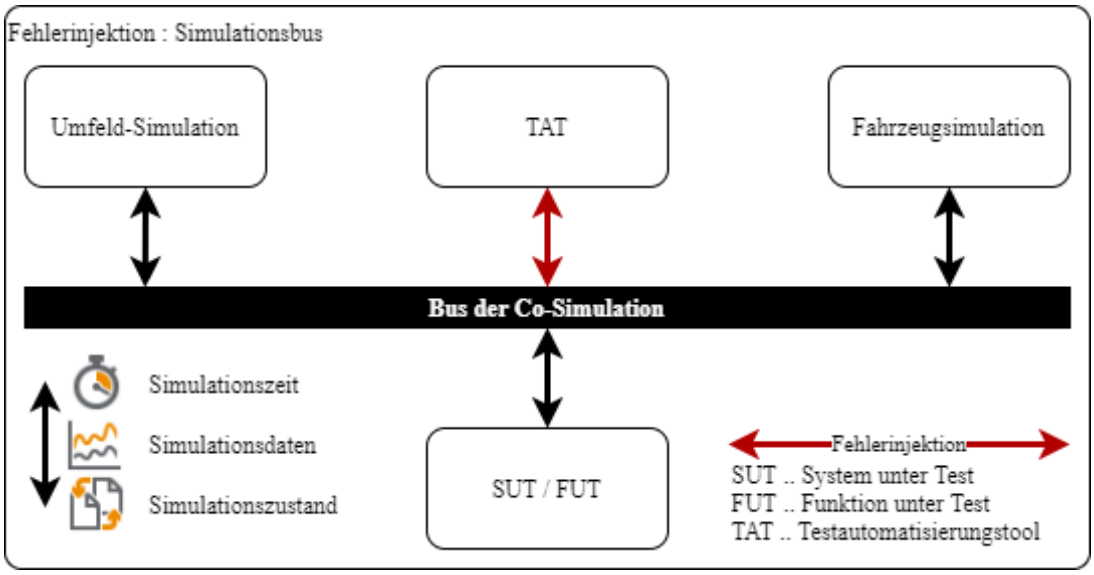

Abbildung 5: Fehlerinjektion über Co-Simulationsbus

Implementierung kann dies aber gelöst werden.

Fehlerinjektion

Durch Erweiterungen in den Simulationsmodellen kann eine Fehlerinjektion erreicht werden, die dann über das TAT über die tool-spezifische Schnittstelle angesteuert werden kann. Alternativ kann die TAT die „fehlerhaften“ Signale selbst auf dem Co-Simulationsbus injizieren, ohne dass Simulationsmodelle geändert werden müssen. Hierzu muss die Co-Simulation aber das dynamische Umschalten von Signalen unterstützen, das ebenfalls durch das TAT gesteuert werden können muss.

Orchestrierung von Co-Simulationen

Für das Testen von Fahrassistenzsystemen ist neben den Zugriff auf Daten und Zeit der Teilnehmer vor allem auch die Skalierbarkeit von Co-Simulationssystemen wichtig. Den nur mit einer hohen Skalierbarkeit, also die Möglichkeit, eine große Anzahl an Testfällen parallel und verteilt auf virtuellen Ressourcen ausführen zu lassen, können Co-Simulationen einen konkreten Mehrwehrt bei der Absicherung von Fahrassistenzsystemen leisten. Dadurch rücken die Lebenszyklusabschnitte *bereitstellen* und *konfigurieren* von Co-Simulationssystemen, sowie die Verteilung der notwendigen Ressourcen immer mehr in den Fokus. Nur durch die Orchestrierung

von Co-Simulationsverbünden und den darauf ausgeführten Funktionstests, ist eine Skalierung erst möglich.

Die dafür notwendige Automatisierung ist ein weitestgehend offenes Feld. Etablierte TAT unterstützen diese Funktionalität nur zu einem kleinen Teil. Zusätzlich bieten auch die meisten Co-Simulationsplattformen wenige Mechanismen, um mit dieser Problematik effizient umzugehen. Je nachdem, wer diese übergeordnete Orchestrierung vornimmt (TAT, Co-Simulationsplattform, Testmanagement), ergeben sich Auswirkungen auf die Anbindung des TAT und der Definition des Testsystems.

Wenn die Orchestrierung von außen geschieht, muss das TAT bereits Teil des Rezepts sein. Das *Starten* der Teilnehmer erfolgt dann ebenfalls außerhalb des TAT. Das TAT bekommt bei seinem Start von außen die Testfälle übergeben. Nun muss eine geordnete Übergabe der Kontrolle von der äußeren Orchestrierung an das TAT erfolgen. Insgesamt ist damit die Skalierung und verteilte Parametervariation leichter zu bewerkstelligen.

Ist das Rezept als Testfall im TAT implementiert, benötigt das TAT die Mechanismen für die Bereitstellung und den Start der Teilnehmer. Dafür ermöglicht dieser Ansatz einen leichteren Start für die Entwicklung von Testfällen, da dies alles aus der TAT heraus passieren kann. Andererseits ist dadurch das Zusammenführen der Testresultate bei Parametervariation schwieriger zu bewerkstelligen.

5 Zusammenfassung und Ausblick

Die Frage, welche Aufgaben ein „Testautomatisierungstool für Co-Simulation“ übernimmt, kann nicht abschließend beantwortet werden. Es hängt stark davon ab, welches Co-Simulationsframework eingesetzt wird, welche Funktionen davon genutzt werden und welcher Usecase umgesetzt wird.

Klar ist jedoch, dass die Orchestrierung von Testsystemen auf Basis mittels Co-Simulation komplex ist und durch ein geeignetes Tooling unterstützt werden muss. In bestimmten Fällen kann dies durch ein TAT geschehen.

6 Literaturverzeichnis

[GTB+17] Gomes, C., Thule, C., Broman, D., Larsen, P. G., & Vangheluwe, H. (2017). Co-simulation: State of the art. arXiv preprint arXiv:1702.00686

[GGF19] Gebauer, M., Günther, F. & Fruth, M. (2019) Testing Challenges for Autonomous Driving: From ASAM OpenX to the evaluation of millions of simulations. ASAM International Conference 2019.

[VDA710] VDA-710: VDA Empfehlung “Software-in-the-Loop (SiL) Standardisierung”. Juli 2022