CARL VON OSSIETZKY universität OLDENBURG

Bewertung von Simulationszuständen für eine gezielte Analyse risikoreicher Systeme

Von der Carl von Ossietzky Universität Oldenburg
- Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften -
zur Erlangung des Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von **Herrn Volker Gollücke, M.Sc.**
geboren am 14.05.1983 in Bremerhaven

Gutachter:
Prof. Dr.-Ing. Axel Hahn
Prof. Dr. Martin Fränzle

Tag der Disputation:
28.11.2016

für meine wundervollen Kinder und meine bezaubernde Frau

Danksagung

An dieser Stelle möchte ich nochmal ein besonderes Dankeschön an meinen Doktorvater Herrn Prof. Dr.-Ing. Axel Hahn aussprechen, der mir diese Promotion ermöglicht und mich methodisch und fachlich unterstützt hat. Des Weiteren möchte ich mich bei meinem Zweitgutachter Herrn Prof. Dr. Martin Fränzle bedanken, der mir mit fachlichen und methodischen Ratschlägen Unterstützung bei der Erstellung der vorliegenden Arbeit gegeben hat.

Ebenfalls möchte ich meinen Kollegen in der Abteilung Systemanalyse und -optimierung der Carl von Ossietzky Universität sowie der Abteilung Kooperierende Mobile Systeme am OFFIS e.V. - Institut für Informatik danken. In den Jahren der Zusammenarbeit konnte ich viele Gespräche und Diskussionen führen, die bei der Entstehung dieser Arbeit sehr geholfen haben. Besonders Erwähnen möchte ich hier Sören Schweigert, Rainer Droste und Jan Pinkowski für die tolle Zusammenarbeit sowie meine wissenschaftliche Hilfskraft Florian Klein für seine Unterstützung bei der Implementierung.

Zu guter Letzt möchte ich meiner Familie Danke sagen. Ohne euch hätte ich diese Arbeit nicht fertigstellen können. Danke für euren Zuspruch, euer Verständnis und eure Liebe.

Oldenburg, den 02. März.2017

Zusammenfassung

Die Identifikation und das Vermeiden potenzieller Risiken ist eine wichtige Aufgabe in jedem System Design Prozess. Die Wahrscheinlichkeit ein Risiko zu unterschätzen hängt dabei stark von der Komplexität des betrachteten Systems ab.

Hierbei ist es hilfreich Mittel und Wege zu kennen, die bei der Überprüfung eines initialen Risikobildes, also den vermuteten Risiken, helfen. Aktuelle Sicherheitsbestimmungen, zum Beispiel in der Domäne von Offshore Operationen, benötigen eine Beschreibung aller involvierten Risiken. Um diese Beschreibung erstellen zu können, kann ein von einem Systemexperten modelliertes System, inklusive einer Verhaltensspezifikation, analysiert werden. Innerhalb dieses Systems müssen Risiken gesucht und Gründe für deren Auftreten analysiert werden.

Der in dieser Arbeit betrachtete Ansatz beschreibt eine Methodik, um die Distanz zu einem Risiko bei einer simulativen Analyse zu berechnen. Dabei beschreibt die Distanz keine räumliche Entfernung, sondern die Nähe eines Systemzustands zu einem zu vermeidenden Systemzustand; der risikoreichen Situation. Der Ansatz verwendet Konzepte aus dem Bereich der Rare Event Simulation, welche genutzt werden um das simulative Erreichen einer risikoreichen Situation zu beschleunigen. Es werden Daten gesammelt über den Weg, der zur der zu vermeidenden Situation geführt hat. Diese Daten können in einer manuellen Risikoanalyse weiterverwendet und zur Anpassung von Vorgaben und Prozessen hinsichtlich ihrer Sicherheit genutzt werden.

In der vorgestellten Arbeit wird beschrieben, wie Distanzfunktionen erstellt und genutzt werden können, um Systemzustände hinsichtlich ihrer Nähe zu risikoreichen Situationen zu bewerten. Zusätzlich wird die Frage beantwortet wie die Ergebnisse der Distanzfunktionen genutzt werden können, um Co-Simulationen zu den untersuchten risikoreichen Situationen zu führen.

Abstract

The identification and avoidance of potential risks is an important task in any system design process. The probability of underestimating a risk depends strongly on the complexity of the system under consideration.

It is helpful to know the ways and means, which help reviewing the suspected risks. Current safety regulations, for example in the domain of offshore operations, require a description of all the risks involved. In order to create this description a system including a behavioral specification modeled by a system expert can be analyzed. Within this system, risks have to be searched and reasons for their occurrence have to be analyzed.

The approach presented in this work, describes a method to calculate the distance to a risk at a simulative analysis. The distance does not describe a distance in space, but the proximity of a system state to an avoidable system state, the risk situation. The approach uses concepts from the field of Rare Event Simulations, which are used to accelerate the simulative reach of a risky situation. During the simulation runs data is collected about the way, which was taken to reach the avoidable situation. These data can be further used in a manual risk analysis and for matching rules and processes in terms of safety.

The presented work describes how distance functions can be created and used to assess system states in terms of their proximity to risky situations. In addition, the question of how the results of the distance functions can be used to guide co-simulations to the examined risky situations is answered.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

„Risiko ist die Bugwelle des Erfolges“
- Carl Amery, Schriftsteller

Dieses kurze Zitat sagt in wenigen Worten aus, wie Risiken grundsätzlich eingeordnet werden können. Meistens gehen Risiken dem Erfolg voraus. Keine Entdeckung, keine Erschließung, keine Erfindung ohne dass damit Risiken verbunden sind. Wo immer es Ungewissheiten gibt existieren auch Risiken. Ungewissheiten sind dort vorhanden, wo die Möglichkeit besteht, dass das Ergebnis eines Ereignisses von dem erwarteten Ergebnis abweicht (vgl. [BGKW08]).

Auch wenn Risiken eingegangen werden müssen, um Ziele zu erreichen, so bleibt ein Risiko immer etwas, das so gering und berechenbar wie möglich gehalten werden sollte.

Die Identifikation und das Vermeiden potenzieller Risiken ist eine wichtige Aufgabe in jedem System Design Prozess. Die Wahrscheinlichkeit ein Risiko zu unterschätzen, hängt dabei immer sehr stark von der Komplexität des betrachteten Systems ab. Hierbei ist es gut Mittel und Wege zu kennen, die bei der Überprüfung eines initialen Risikobildes, demzufolge den vermuteten Risiken, helfen. Aktuelle Sicherheitsbestimmungen, zum Beispiel in der Domäne von Offshore Operationen, benötigen eine Beschreibung aller involvierten Risiken (vgl. [Rene13, Thom14]). Um diese erstellen zu können, kann ein von einem Systemexperten modelliertes System inklusive einer Verhaltensspezifikation analysiert werden. Innerhalb dieses Systems müssen Risiken gesucht und die Gründe für deren Auftreten analysiert werden.

Systeme, die auf ihre Risiken untersucht werden sollen, sind teilweise sehr komplex, so dass eine Untersuchung mittels klassischer Risikoanalyse- und Bewertungstechniken zwar durchführbar ist, sich aber zusätzliche Fragen ergeben. So stellen sich bei der Anwendung klassischer Methoden, wie der Fehlerbaumanalyse, Fragen nach der Vollständigkeit der betrachteten Informationen, zum Beispiel, ob alle Ursachen für die identifizierten Fehler erkannt wurden. Hierbei bietet sich der Einsatz von Simulation als Untersuchungsmethode an, welche es erlaubt Experimente unter verschiedenen Konfigurationen des modellierten Systems durchzuführen und dabei Informationen über das Auftreten von Risiken zu sammeln.

Der in dieser Arbeit betrachtete Ansatz beschreibt eine Methodik, mit der die Distanz zu einem Risiko bei einer simulativen Analyse mittels einer so genannten Risikodistanzfunktion definiert und berechnet werden kann. Dabei beschreibt die Distanz

keine räumliche Entfernung, sondern die Nähe eines Systemzustands zu einem vermeidenden Systemzustand - der risikoreichen Situation. Eine theoretische Grundlage und Inspiration hierfür liefert unter anderem das aus dem Bauwesen bekannte semiprobabilistische Sicherheitskonzept (vgl. [Din03, Din12]).

Die Daten, die über den Weg welcher zur vermeidenden Situation geführt hat gesammelt wurden können in einer nachgelagerten Risikoanalyse weiterverwendet und zur Anpassung von Vorgaben und Prozessen hinsichtlich der Sicherheit genutzt werden.

Zusätzlich erfolgt der Einsatz von Konzepten aus der Rare Event Simulation, welche genutzt werden, um das simulative Erreichen einer risikoreichen Situation zu beschleunigen. Der Ansatz unterstützt dabei einen Teil der Importance Splitting Technik aus dem Gebiet der Rare Event Simulation (vgl. [JeLS13, JuSh06, MoPG10]), der genutzt wird, um Black-Box-Simulationen zu risikoreichen Situationen zu führen. Dabei wird der Annahme gefolgt, dass ein Simulationszustand, der näher an einer Gefahr ist, diese schneller erreicht als ein weiter entfernter. Die ermittelte Risikodistanzfunktion wird in diesem Fall als Importance Function genutzt, über die in der Importance Splitting Technik die Bewertung der Simulationszustände getätigt wird.

Um Simulationslauf übergreifend Informationen über den Verlauf dieser zu sammeln, können in den Simulationsläufen erreichte Simulationszustände zusätzlich in einer Datenbank persistiert und um Metainformationen angereichert werden. Diese beinhalten dabei die Bewertung des Zustands mittels der Risikodistanzfunktion, wie oft der persistierte Zustand erreicht wurde, wie oft aus diesem Zustand exploriert wurde und zusätzlich die minimale Risikodistanz, der aus diesem Zustand erreichten Zustände aus vorherigen Simulationsläufen. All diese Informationen können ebenfalls mit in die Steuerung der Simulation einfließen hinsichtlich einer schnelleren Erreichung risikoreicher Zustände. Zur Umsetzung dieses Ansatzes werden Beschreibungsmethoden für Situationen und Ähnlichkeitsmaße benötigt, mit denen verschiedene Situationen beschrieben und miteinander verglichen werden können. Um diese aufstellen und anwenden zu können werden Techniken aus der Domäne des Information Retrieval eingesetzt, die den Vergleich zweier Systemzustände hinsichtlich ihrer Ähnlichkeit in angemessener Zeit durchführen können (vgl. [BüCC10]).

1.1 Motivation

Schäden und Unfälle in der maritimen Branche sind trotz hoher Sicherheitsstandards ein nicht zu verkennendes Problem (s. Abbildung 1-1). Die Statistik zeigt die Entwicklung der weltweiten Versicherungsschäden im Zeitraum von 2006 bis 2014 herbeigeführt durch

maritime Katastrophen. Laut der Statistik handelt es sich bei den betrachteten Schäden um Sach- und Betriebsunterbrechungsschäden, ohne Haftpflicht- und Lebensversicherungsschäden. Unter anderem wurden Passagierschiffs-, Tanker- und Frachter Unglücke sowie Unfälle auf Bohrinseln oder Ölplattformen aufgenommen. Allein im Jahr 2014 beliefen sich die versicherten Schäden, die aus weltweiten maritimen Katastrophen resultierten, auf eine Summe von rund 783 Millionen US-Dollar. Zwei Jahre zuvor (2012) betrug die Summe der versicherten Schäden sogar 2,2 Milliarden US-Dollar.

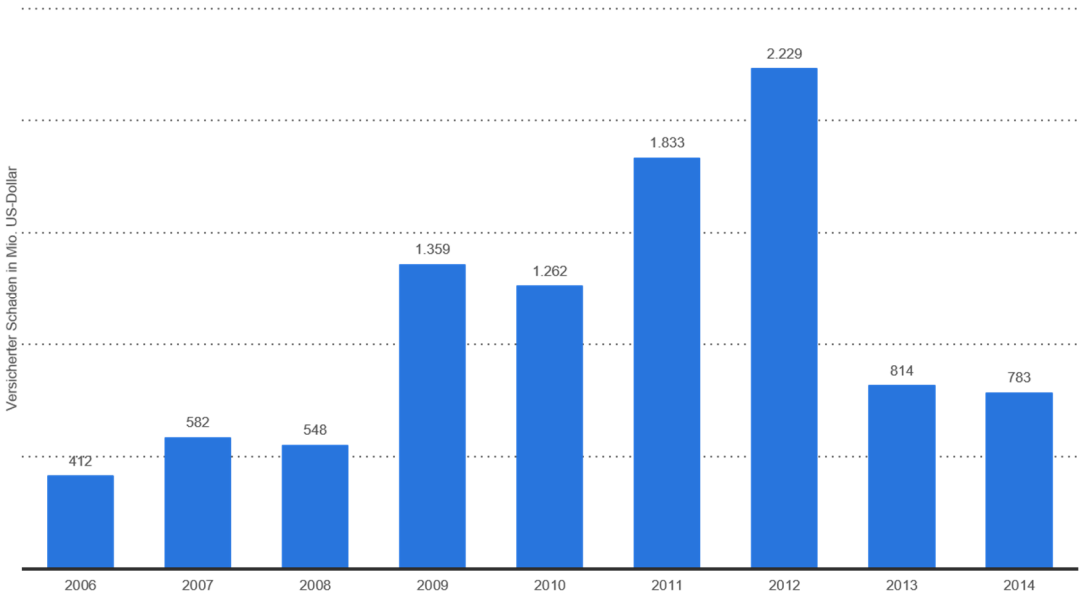


Abbildung 1-1 Weltweite Versicherungsschäden verursacht durch maritime Katastrophen von 2006 bis 2014 (in Millionen US-Dollar) (vgl. [Swis15]).

Die in dieser Arbeit näher betrachtete maritime Domäne erhielt in den letzten Jahren viel Aufmerksamkeit. Gerade dem Bereich der Offshore-Windparks wird von Seiten der Forschung und Politik viel Potenzial zugeordnet, da es sich um eine neuere Art der sauberen Energiegewinnung handelt. Weniger allerdings liest man über die Risiken, die mit dem Aufbau verbunden sind. So mussten neben vielen Verletzten auch mehrere Tote bei der Arbeit in und an den Windparks beklagt werden (vgl. [Dapd12, Pres13, Wagn12]).

Jedoch sind nicht nur bei Offshore-Arbeiten Sachschäden, Verletzte und Tote zu beklagen, sondern auch generell sind die oft vernachlässigten Gefahren der maritimen Domäne präsenter und vielfältiger denn je und reichen vom Mann über Bord, über Kollisionen zu Feuer an Bord von Schiffen und Bohrinseln (s. Abbildung 1-2). Die Art, wie man mit diesen Risiken umgeht, unterscheidet sich jedoch von Domäne zu Domäne sehr stark. Das

Umgehen mit Risiken bedeutet, dass man sich im Vorhinein Gedanken über die möglicherweise auftretenden Gefahren machen muss, bevor die eigentliche Unternehmung startet.

Häufig liegt das Auftreten von Gefahren an einem falschen Umgang mit diesen zur Planungszeit. Dabei ist fehlende Zeit ein großes Problem, das dazu führt, dass als selten eingestufte Gefahren nicht weiter betrachtet werden, was wiederum zu einer Vernachlässigung dieser bei den vorangehenden Analysen führt (vgl. [Aven14, MeDD05]).

Abbildung 1-2 Verschiedene Unfallarten in der maritimen Domäne.

Für Arbeiten in der Offshore-Domäne werden sogenannte HSE (Health, Safety, Environmental) Pläne von Systemexperten erstellt. Diese werden genutzt, um sich einen Überblick über die auszuführende Arbeit zu verschaffen und finden dann unter anderem Verwendung zur Versicherung dieser (vgl. [Thom14]).

Jedoch ist die Aufstellung dieser Pläne ein anstrengender und zeitintensiver Prozess, welcher leider auch oft nicht alle Risiken und Ursachen beschreibt. Gerade Risiken, deren Auftreten als sehr selten angesehen ist, werden oft ignoriert - unter anderem auch da zu wenige Informationen über diese vorhanden sind.

Die zu analysierenden Arbeitsabläufe werden von Experten entwickelt und beschrieben. Die Erwartungen daran sind, dass sich keine bis wenig Fehler in diesen Beschreibungen befinden. Durch die Analyse sollen mögliche Risiken aufgedeckt und im Falle der Entdeckung neuer Risiken durch eine Anpassung des Sicherheitskonzepts abgefangen werden. Um aussagekräftige Informationen über Risiken und deren Auftreten zu sammeln, ist es notwendig viele unabhängige Informationen über das Auftreten der zu untersuchenden Risiken zusammen zu tragen. Ein Beispiel für einen zu untersuchenden Arbeitsablauf ist der Aufbau von Windkraftanlagen auf der offenen See. Ein Teil dieser Arbeit ist der Transport von Ladung vom Schiff zu den aufzubauenden Anlagen.

Soll dieser Arbeitsablauf hinsichtlich des Auftretens angegebener Risiken überprüft werden, können mathematische Ansätze verfolgt oder eine simulative Analyse angewendet werden. Durch die oft vorhandene hohe Komplexität der betrachteten Systeme und möglicherweise nicht ausreichender Daten ist es jedoch sehr schwer analytische Modelle für formale bzw. mathematische Analysen zu erstellen. Aus diesem Grund bietet sich die Nutzung einer simulativen Analyse an, für die die realen Abläufe sehr gut nachgebildet werden können. Des Weiteren lassen sich durch eine Simulation des Systems komplexe Zusammenhänge betrachten. Dazu gehören physikalische Effekte, wie das Nachschwingen eines Kranseiles, Umweltfaktoren wie Wind, Regen, Blendungen durch die Sonne, Nebel und weitere vom betrachteten Arbeitsablauf abhängige Umwelteinflüsse und deren Auswirkungen auf die physikalischen Effekte und vorhandenen Akteure. Zudem bietet es sich bei komplexen Berechnungen innerhalb einer Simulation an hierfür eigenständige Simulatoren zu verwenden, die auf verschiedenen Systemen ausgeführt werden (Co-Simulation).

Bei der Anwendung von Simulation existieren auch ein paar Nachteile, wie in der folgenden Auflistung zu sehen (vgl. [Mari97]):

1. Simulationsmodelle sind meistens stochastisch, was dazu führt das Ergebnisse nur geschätzt werden können

2. Hoher Datenbedarf für Simulationsmodelle der nicht immer befriedigt werden kann. Ein Modell kann nicht besser als der schwächste Punkt sein. Es werden zwar detaillierte Modelle erstellt, die benötigten Parameter aber nur auf Grund von sehr wenigen Beobachtungen geschätzt oder nur auf nicht validierten Annahmen.

3. Simulationsmodelle sind sehr aufwändig in der Entwicklung und liefern in manchen Fällen große Datenmengen die nicht detailliert genug analysiert werden können. Daher werden Ergebnisse oft falsch interpretiert.

Neben den genannten Nachteilen bietet der Einsatz von Simulation jedoch auch die folgenden weiteren Vorteile (vgl. [Mari97]):

1. Experimente können einfach, kostengünstig und gefahrlos durchgeführt werden. Durch die Beeinflussbarkeit aller Faktoren können Experimente unter fast beliebigen Bedingungen durchgeführt werden.
2. Identifikation von Aktionen, die benötigt werden, um einen vorgegebenen Prozess oder eine Tätigkeit zu verbessern.
3. Evaluation der Auswirkungen von Veränderungen vor der eigentlichen Umsetzung.
4. Systeme können in unterschiedlich langen Zeitintervallen beobachtet werden, wenn die Zeit innerhalb der Simulation entsprechend gestaucht oder gestreckt wird.

Die zu untersuchenden Gefahren sind oft selten auftretende Ereignisse, für die das Anwenden von naiven Simulationsmethoden sehr zeitaufwändig ist. Es müssen sehr viele Simulationsläufe betrachtet werden, in denen keine Risiken auftreten und auf die - natürlich nur bei der Simulation und Analyse - am liebsten verzichtet werden würde. Dies kann unter anderem an der Betrachtung von kontinuierlichen Räumen liegen oder der Einbeziehung von menschlichem Verhalten welches nicht deterministisch ist.

Um die Zeit zu verkürzen, existieren Optimierungsmethoden für Simulationen zu denen unter anderem auch der in dieser Arbeit betrachtete Bereich der Rare Event Simulation gehört.

1.2 Zieldefinition und Beitrag der Arbeit

Die Ziele dieser Arbeit bestehen in der Nutzung einer Bewertung von Systemzuständen hinsichtlich der Risikonähe, um eine gezielte Analyse risikoreicher Systeme durchzuführen. Die wissenschaftliche Fragestellung, die in den weiteren Kapiteln dieser Arbeit beantwortet wird, lässt sich folgendermaßen definieren.

1. Wie kann die Distanz zu risikoreichen Situationen in den Läufen einer Co-Simulation definiert und ermittelt werden?

2. Wie können die ermittelten Distanzen zur beschleunigten Erreichung der zu analysierenden risikoreichen Situationen genutzt werden?

Der in dieser Arbeit vorgestellte Ansatz beschreibt zum einen, wie Distanzfunktionen definiert und ermittelt werden können und wie die ermittelten Distanzfunktionen verwendet werden, um Situationen hinsichtlich ihrer Nähe zu kritischen Situationen zu bewerten. Des

Weiteren wird beschrieben, wie die ermittelten Distanzen zur beschleunigten Erreichung der risikoreichen Situationen genutzt werden können. Dies basiert dabei auf der, aus der Rare Event Simulation bekannten, Importance Splitting-Technik, welche genutzt wird, um Black-Box Simulationen in Richtung risikoreicher Situationen zu führen und dabei eine sinnvolle Reduktion der Anzahl von Simulationsläufen zu erzielen. Des Weiteren werden Techniken aus dem Bereich des Information Retrieval angewandt, um zwei Simulationssituationen in angemessener Zeit bzgl. ihrer Ähnlichkeit zu bewerten, um weitere Entscheidungsgrundlagen für die Steuerung des Simulationsverlaufs zu erhalten.

Die zu erreichenden Ziele, lassen sich dabei in die folgenden vier Bereiche unterteilen:

1. Definition und Ermittlung der Distanzbeschreibung
2. Integration der Distanzbeschreibung in Co-Simulationsläufe zur Bewertung von Systemzuständen
3. Steuerung einer Co-Simulation zur gezielten Untersuchung risikoreicher Situationen
4. Entwicklung einer Toolchain zur Anwendung und Evaluation der Methodik und Integration der Distanzbeschreibung in eine Co-Simulation

Der Beitrag dieser Arbeit wird dabei ausgehend von den Zielen, den betrachteten Grundlagen und dem Stand der Technik und Wissenschaft, eine Methodik beschreiben, um die simulative Nutzung von Risikodistanzen zu ermöglichen. Dabei muss dargestellt werden, wie die Distanzfunktionen zu definieren sind, um die Nähe zu Risiken zu berechnen. Zusätzlich wird definiert, welche Anforderungen an die Distanzfunktionen, die Distanzermittlung und die verwendeten Simulatoren gestellt werden müssen, damit diese in der entwickelten Methodik verwendet werden können. Des Weiteren wird in dieser Arbeit aufgezeigt, wie die Distanzberechnung in eine Co-Simulationsumgebung integriert werden kann. Der letzte Beitrag der Arbeit ist die Beantwortung der Frage, wie durch die ermittelte Distanzermittlung eine Rare Event Simulation unterstützt werden kann. Dabei wird vorgestellt, welche Anforderungen an Simulatoren sowie Steuerungskomponente erfüllt sein müssen, damit Techniken aus dem Bereich der Rare Event Simulation zusammen mit der Risikodistanzbewertung angewendet werden können.

Um die Ziele zu erreichen, stehen verschiedene Informationen zur Verfügung. Dazu gehört die Beschreibung des Systems und eine Spezifikation des Verhaltens, über die ebenfalls die formale Beschreibung der Arbeitsabläufe erfolgt und an der die Risiken als mögliche Gefahren und deren Ursachen definiert sind.

1.3 Begriffsbildung

In diesem Abschnitt werden die Begriffe, die im Rahmen dieser Arbeit gehäuft Verwendung finden, im Kontext dieser Arbeit geklärt und beschrieben. Es werden die Begriffe aufgeführt, für die unterschiedliche Definitionen vorhanden sind oder die für diese Arbeit neu eingeführt wurden.

Systemmodell: Das Systemmodell beschreibt den Aufbau des zu untersuchenden Systems mit verschiedenen Systemelementen wie bestimmten Personen und Objekten plus der zu analysierenden Umwelt desselben inklusive einer Verhaltensbeschreibung.

Eigenschaft: Eine Eigenschaft steht in dieser Arbeit für einen charakterisierenden Wert eines Systemelements. Über verschiedene Ausprägungen, also unterschiedlich zugewiesene Werte, lassen sich Systemelemente beschreiben und unterscheiden. Die eindeutige Bezeichnung, die Position und die Orientierung einer Person sind Beispiele für eine Eigenschaft.

Parameter: stehen für spezialisierte Eigenschaften, die als Stellgröße dienen. Über Parameter lässt sich das Verhalten von Systemelementen beeinflussen. Die maximale Rotationsgeschwindigkeit eines Krans ist ein Beispiel für einen Parameter.

Systemmodellinstanz: Die Systemmodellinstanz stellt das mit Werten belegte Systemmodell dar.

Situation: Ist der Zustand des untersuchten Systems zu einem bestimmten Zeitpunkt. Dieser wird dabei über die gesetzten Eigenschaften einer Systembeschreibung (der Systemmodellinstanz) und einem zusätzlichen Zeitstempel beschrieben.

Risikosituation: Der Begriff Risikosituation steht stellvertretend für Situationen, die vermieden werden sollten. Beispiel für eine Risikosituation ist die eingetretene Kollision zwischen zwei bestimmten Objekten.

Risikodistanz: Im Rahmen dieser Arbeit wird bei der Verwendung des Begriffs Risikodistanz von der Nähe zum Auftreten einer Risikosituation gesprochen und beschreibt dabei nicht ausschließlich eine räumliche Distanz. Diese wird mit einem Wert zwischen 0 (die Risikosituation ist aufgetreten) und 1 (keine relevante Nähe zum Auftreten der Risikosituation) angegeben.

Ursache: Als Ursache wird eine, die Risikosituation begünstigende Sammlung von Ausprägungen von Eigenschaften benannt. Als Beispiel für eine Ursache kann eine geringe

Sichtweite von wenigen Metern angeführt werden, welche das Risiko einer potentiellen Kollision erhöht.

Indikator: Als Indikatoren werden synonym die eine Ursache begünstigenden Ausprägungen von Eigenschaften bezeichnet. Diese bestehen aus einem Eintritts- und Relevanzraum.

Eintrittsraum: Eintrittsräume beschreiben die Grenzen einzelner die Ursache beschreibende Indikatoren, in denen der Indikator als eingetreten gilt.

Relevanzraum: Relevanzräume beschreiben die Grenzen einzelner die Ursache beschreibende Indikatoren, in denen die Entfernung zum Indikator als relevant und außerhalb dessen als nicht relevant zu betrachten ist.

1.4 Forschungsmethodik und Aufbau der Arbeit

Bei der Erstellung dieser Arbeit wurde ein ingenieursmäßiges Vorgehen verfolgt (s. Abbildung 1-3). Ausgehend von einer Motivation und Problemstellung wurde die wissenschaftliche Fragestellung und zu dieser eine Zielsetzung aufgestellt. Diese betrachtend wurde der Handlungsbedarf und die Anforderungen an den eigenen Ansatz ermittelt in dem der aktuelle Stand der Technik und Wissenschaft betrachtet wurde. Ausgehend von diesen wurde der eigene Ansatz entwickelt, welcher in einem prototypischen Framework umgesetzt und abschließend evaluiert wurde.

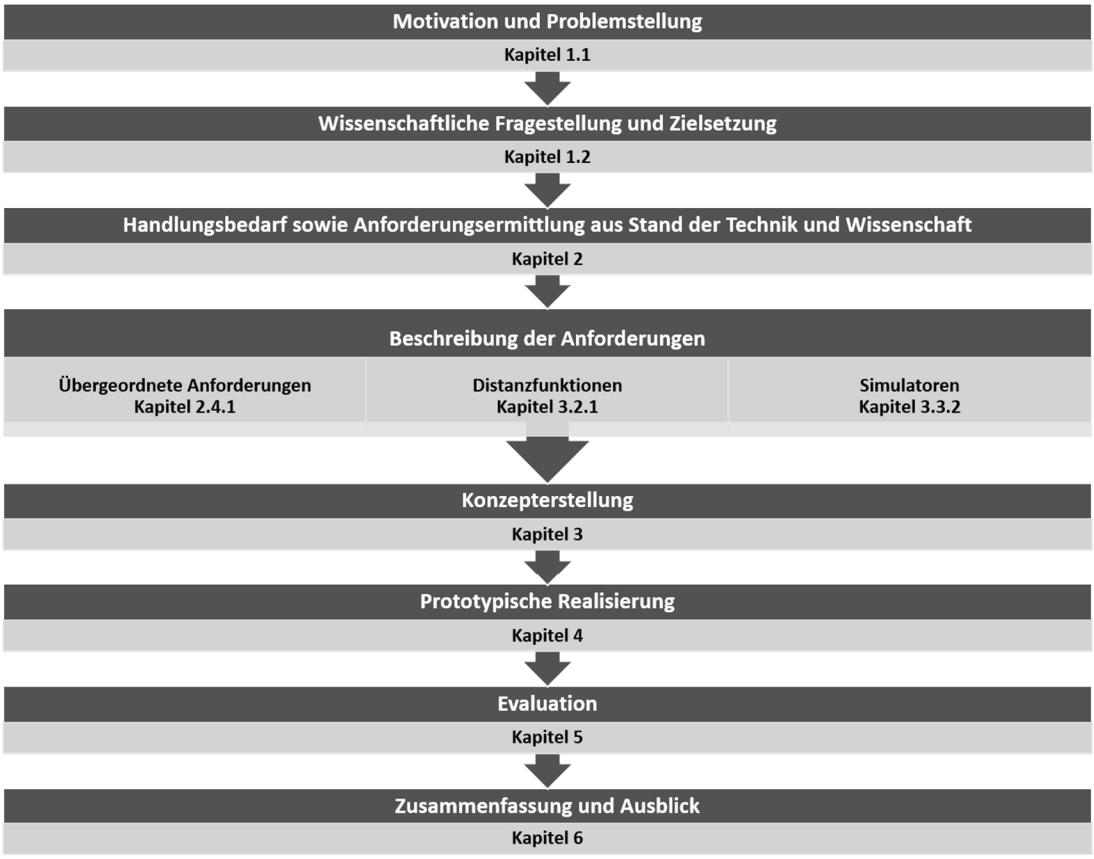


Abbildung 1-3 Vorgehen inklusive der dementsprechenden Einordnung in die Gliederung dieser Arbeit.

Die Arbeit ist dabei wie folgt strukturiert. Zunächst wurde die Einleitung inklusive der Motivation für eine simulative Analyse risikoreicher Systeme vorgestellt. Dazu wurde auf die aktuelle Situation verwiesen und die Problemstellung abgeleitet. Danach wurde die wissenschaftliche Fragestellung vorgestellt und die daraus abgeleiteten Ziele für diese Arbeit. Zusätzlich wurde eine erste Begriffsbildung in Kapitel 1 „*Einleitung*“ gegeben.

Im folgenden Kapitel „*Stand der Wissenschaft und Technik zur Risikoanalyse durch Simulation*“ (2) werden aktuell verfügbare Methoden und Techniken sowie Software zur simulativen Risikoanalyse, Simulationsoptimierung und Simulationsframeworks vorgestellt, die ähnliche Ziele wie die dieser Arbeit umsetzen oder aber Teilziele davon. Zum besseren Verständnis werden die hierfür notwendigen Grundlagen erläutert. Die vorgestellten Grundlagen unterteilen sich in eine Betrachtung von Methoden aus dem Feld der probabilistischen Risikoanalyse, Modellierung und Simulation, dem allgemeinen Bereich der Simulationsoptimierung und spezieller der Rare Event Simulation. Das Kapitel schließt mit dem ermittelten Handlungsbedarf inklusive der für den eigenen Ansatz aufgestellten übergeordneten Anforderungen.

Im anschließenden Kapitel „*Bewertung von Simulationszuständen in Co-Simulationen zur beschleunigten simulativen Analyse*“ (3) wird der entwickelte eigene Ansatz vorgestellt. Dieser setzt dabei auf die ermittelten Anforderungen auf und beschreibt die drei betrachteten Aspekte: Modellierung, Risikodistanzbeschreibung und simulative Analyse. In diesem Kapitel werden zusätzlich die ermittelten Anforderungen an die Risikodistanzfunktionen und die verwendeten Simulatoren vorgestellt.

In Kapitel 4 wird die genutzte Definition der eingesetzten Co-Simulation und der verwendeten Simulatoren vorgestellt, welche das entwickelte Framework und der eigene Ansatz für die Anwendung voraussetzen.

Um Methodik und Techniken anzuwenden sowie die ermittelten Risikodistanzfunktionen zu ermitteln und in Co-Simulationen verwenden zu können, wurde als eigener Ansatz ein der Zielstellung und den ermittelten Anforderungen folgendes Framework konzipiert. Das prototypisch umgesetzte Framework wird dabei zur Konfiguration, Analyse und Steuerung von verteilten Simulationen genutzt (*DistriCT – Distributed Controlling Toolkit*). Dieses wird ebenfalls in Kapitel 4 vorgestellt und auf Realisierungsdetails eingegangen.

Im Evaluations-Kapitel 5 „*Evaluation der entwickelten Methodik und Simulationsunterstützung*“ findet die Überprüfung des eigenen Ansatzes mit dem entwickelten Framework statt. Hierfür wurden repräsentative Szenarien aus der maritimen Domäne gewählt, um die Funktionsweise zu evaluieren und die Machbarkeit und Anwendung des Ansatzes zu demonstrieren.

Abschließend wird in Kapitel 6 „*Zusammenfassung und Schlussfolgerungen für den eigenen Beitrag*“ eine Zusammenfassung der Arbeit vorgestellt und ein Ausblick gegeben, der weitere Forschungs- und Einsatzmöglichkeiten aufzeigt.

2 Stand der Wissenschaft und Technik zur Risikoanalyse durch Simulation

Risikoanalyse ist die systematische Untersuchung von Unsicherheiten und Risiken, die in bestimmten Bereichen auftreten können. Risikoanalysten versuchen die Risiken von Institutionen und Operationen zu identifizieren und zu verstehen wie und wann sie auftreten. Zusätzlich wird versucht die Auswirkung in finanziellen oder in Bereichen der Sicherheit zu schätzen. Risikomanager starten dabei mit einer Risikoanalyse und versuchen Aktionen festzulegen, die die erkannten Risiken verhindern oder mindern. Viele Unternehmen müssen Risiken jeden Tag aufs Neue betrachten und bewerten, so dass die Risikoanalyse und das Risikomanagement eine zentrale und sehr wichtige Aufgabe für diese ist.

Bei der quantitativen Risikoanalyse wird ein mathematisches Modell eines Projektes oder Prozesses erstellt, welches unbestimmte Parameter, die nicht kontrolliert werden können als auch Parameter, die kontrolliert werden können inkludiert. Ein quantitatives Risikomodell berechnet den Einfluss unbestimmter Parameter auf verschiedene Ausgabekriterien, wie zum Beispiel Konsequenzen für die Umwelt oder die Effizienz eines Prozesses. Ein Modell dieser Art kann den Entscheidungsträgern dann dabei helfen den Einfluss und die Konsequenzen eines eingeschlagenen Weges besser zu bewerten.

Ein anderer Weg, um mit Unsicherheiten und Risiken umzugehen ist die Durchführung eines Experiments. Dies ist jedoch oft zu gefährlich oder zu teuer, um es in der realen Welt anzuordnen. Daher bietet es sich an ein vereinfachtes Modell der realen Welt zu beschreiben und das Experiment mittels des auch in dieser Arbeit betrachteten Ansatzes der Simulation durchzuführen.

In den folgenden Abschnitten dieses Kapitels werden aktuell verfügbare Methoden und Techniken inklusive Software zur simulativen Risikoanalyse, Simulationsoptimierung und Simulationsframeworks vorgestellt, die ähnliche Ziele wie die dieser Arbeit umsetzen bzw. verfolgen oder aber Teilziele davon. Zum besseren Verständnis werden hierfür die notwendigen Grundlagen erläutert. Die vorgestellten Grundlagen unterteilen sich in eine Betrachtung von Methoden aus dem Feld der probabilistischen Risikoanalyse, Modellierung und Simulation, dem allgemeinen Bereich der Simulationsoptimierung und spezieller der Rare Event Simulation mit der in dieser Arbeit genauer betrachteten Importance Splitting

Methode (vgl. [JeLS13, JuSh06, KaOl07, MoPG10]). Das Kapitel schließt mit einer Zusammenfassung und dem ermittelten Handlungsbedarf sowie den daraus ermittelten übergeordneten Anforderungen an den eigenen Ansatz.

2.1 Probabilistische Risikoanalyse

In diesem Abschnitt soll ein kurzer Überblick über verschiedene klassische Ansätze zur probabilistischen Risikoanalyse gegeben werden. Zu den betrachteten Techniken gehören die Fehlermöglichkeits- und -einflussanalyse (engl. Failure Mode and Effect Analysis, FMEA), die Ereignisbaumanalyse (engl. Event Tree Analysis, ETA) und die Fehlerbaumanalyse (engl. Fault Tree Analysis, FTA), welche für die Strukturierung von Fehlern und Ursachen für die Risikodistanzfunktion mittels Fehlerbäumen genutzt wird.

Fehlermöglichkeits- und einflussanalyse

Die Fehlermöglichkeits- und einflussanalyse wird genutzt um Produktfehler mit einer Kennzahl zu bewerten betreffs der Bedeutung und der Auftretens- und Entdeckungswahrscheinlichkeit. Die FMEA gehört dabei zu den analytischen Methoden der Zuverlässigkeitstechnik und wird in der Entwicklungsphase von Projekten angesiedelt. Die verfolgten Ziele sind dabei eine höhere technische Zuverlässigkeit sowie eine Fehlervermeidung von vornherein. Die FMEA hat ihr hauptsächliches Einsatzgebiet im Sicherheits- bzw. Qualitätsmanagement und ist weit verbreitet im Automobilbereich und in der Luft- und Raumfahrtdomäne.

Häufig wird zur Anwendung der FMEA ein Ursache-Wirkungs-Diagramm genutzt. Der Ablauf sieht dabei so aus, dass zunächst ein Fehlerort lokalisiert wird, danach wird die Fehlerart bestimmt, die Fehlerfolge beschrieben und zum Abschluss die Fehlerursache ermittelt.

Zur Risikobeurteilung werden die Kennzahlen als Grundlage verwendet. Diese sind ganzzahlige Werte zwischen eins und zehn die mittels Bewertungskatalogen vergeben werden. Durch die Multiplikation der Kennzahlen wird eine sogenannte Risiko-Prioritätszahl (RZP) berechnet dessen Anspruch es ist eine Aussage über die Rangfolge der Risiken zu erstellen. Dies ist jedoch nach der DIN EN 60812 (vgl. [Din06]) allgemein nicht gegeben. [KaBu09, Stam03, Stän11]

Ereignisbaumanalyse

In der Ereignisbaumanalyse wird eine mögliche Auswirkung eines Ereignisses auf das Gesamtsystem überprüft. Die Anwendung der Ereignisbaumanalyse kann nur richtig angewendet werden, wenn die Systemanforderungen bekannt sind.

Das untersuchte Ereignis wird dabei im Ereignisbaum als Startereignis genutzt. Die Wirkungen auf das System werden als Form verschiedener Verzweigungen im Baum, die die Funktion oder den Ausfall der Systemkomponenten als Reaktion auf das Startereignis zeigen, graphisch angegeben.

Die einzelnen Pfade, welche von einem Startereignis zu einem definierten Endzustand gehen, sind die möglichen Unfallsequenzen. Dabei ist jede Verzweigung mit einer bestimmten Ausfallwahrscheinlichkeit verbunden. Durch die Berechnung des Produktes der Wahrscheinlichkeiten vom Startereignis und verschiedenen Abzweigungen auf dem Pfad ergibt sich die Wahrscheinlichkeit einer bestimmten Unfallsequenz. Die Gesamtunfallwahrscheinlichkeit wird durch die einfache Addition aller Pfad-Wahrscheinlichkeiten berechnet sofern die Komponenten der unterschiedlichen Pfade Unabhängig voneinander sind.

Die Ereignisbaumanalyse nutzt zusätzlich die nachfolgend beschriebene Fehlerbaumanalyse, wenn keine empirischen Werte für die Wahrscheinlichkeiten der Verzweigungen und Ereignisse vorlegen um diese damit zu ermitteln. [Eric05, Köni13, Stän11]

Fehlerbaumanalyse

Die Fehlerbaumanalyse basiert auf der booleschen Algebra welche zur Bestimmung der Wahrscheinlichkeit eines Ausfalls eines Gesamtsystems genutzt wird (vgl. DIN25424 [Din81, Din90]). Bei einer Fehlerbaumanalyse werden die logischen Kombinationen von Teilsystemausfällen gesucht die zu einem Gesamtsystemausfall führen.

Hierfür wird das Gesamtsystem unterteilt in die verschiedenen Ereigniskombinationen die in einem Gesamtausfall münden können. Die Menge dieser Ereigniskombinationen kann je nach betrachtetem System eine sehr große Anzahl erreichen.

Bei der quantitativen Fehlerbaumanalyse wird nach dem Aufbau des Fehlerbaums jedem Basic-Event (Blattelement des Fehlerbaums) eine bestimmte Eintrittswahrscheinlichkeit für den Ausfall zugewiesen.

Die Wahrscheinlichkeiten für einen Ausfall werden bei der Fehlerbaumanalyse mit den jeweiligen verwendeten logischen Gattern in Beziehung gesetzt. Ein „Und“-Gatter verknüpft statistisch unabhängige Ereignisse an seinen Eingängen und gibt die

Wahrscheinlichkeit für den Ausfall der verknüpften eingehenden Systeme an seinem Ausgang an. Ein „Oder“ bildet die Wahrscheinlichkeit, wenn eines, mehrere oder alle Basiskomponenten ausgefallen sind.[Stän11, VDFM02]

Die Verwendung von Fehlerbäumen eignet sich für eine Strukturierung der potentiellen Risiken und einer Verknüpfung der möglichen Ursachen. Jedoch wird durch die Untersuchung großer immer komplexer werdender Systeme, mit der erwähnten riesigen Anzahl an Ereigniskombinationen, die Verwendung von Simulation ein immer wichtiger werdender Aspekt bei der Risikoanalyse. Ein Indiz hierfür sind aktuelle Projekte wie ENABLE-S3 (vgl. [Paul15]) und PEGASUS (vgl. [Bmwi16, S.30+31]) die für die Risikoanalyse auch auf eine simulative Vorgehensweise setzen.

2.2 Simulative Risikoanalyse

Der Bereich der Simulation umfasst die Hauptbereiche „Modellierung“ – Die Erstellung einer Repräsentation von Etwas und „Simulation“ – Das Nutzen eines Tools wie einen Computer, um die dynamischen Charakteristiken eines real existierenden Systems zu imitieren (vgl. [Aust04]).

Unter Modellierung wird der Prozess des Erstellens eines Modells verstanden. Ein Modell repräsentiert ein System mit all seinen Einflussfaktoren, möglichen Zuständen, Wechselwirkungen und seinem Verhalten. Ein Modell ist dabei ähnlich aber simpler als das nachgebildete System.

Ein Ziel bei der Erstellung eines Modells ist es einem Analysten zu ermöglichen die Effekte durch Änderungen am System vorherzusagen. Es ermöglicht also festzustellen, welche Reaktion das zugrundeliegende System auf gewisse äußere Einflüsse zeigt (vgl. [Boss04]).

Auf der einen Seite, sollte ein Modell eine große Annäherung an das reale System darstellen und dessen typische Merkmale beinhalten. Auf der anderen Seite darf es nicht so komplex werden, dass es unmöglich ist es zu verstehen und Experimente daran durchzuführen. Ein gutes Modell ist ein überlegtes Abwägen zwischen Realismus und Einfachheit. Meistens wird zunächst ein einfaches Modell erstellt dessen Komplexität nach und nach erhöht wird.

Ein wichtiger Punkt bei der Modellierung ist die Validität des Modells. Techniken zur Modellvalidierung beinhalten die Simulation des Modells unter unbekannten Eingabewerten und dem Vergleich des Modellzustands mit dem Systemzustand. Im Allgemeinen ist ein Modell, welches für Simulationsstudien verwendet wird, ein mathematisches Modell welches mit der Hilfe von Simulationssoftware entwickelt wurde. Die Klassifikation

mathematischer Modelle beinhaltet deterministische Modelle bei denen die Abhängigkeiten fix sind, stochastische Modelle bei denen zu mindestens eine der Ein- oder Ausgabevariablen probabilistisch ist, statische Modelle bei der die Zeit nicht betrachtet wird oder dynamische Modelle bei denen über die Zeit variierende Interaktionen zwischen Variablen betrachtet werden.

Eine Simulation eines Systems ist die Ausführung des Modells des Systems. Eine Simulation erfolgt meistens am Computer, da der Aufwand gegenüber physikalischen und realen Analogien signifikant kleiner ist. Zusätzlich lassen sich zeitliche Prozesse beschleunigen oder, für eine Betrachtung sehr schneller Vorgänge, verlangsamen. Ein generelles Simulationsmodell umfasst *n* Eingabevariablen und *m* Ausgabevariablen (s. Abbildung 2-1).

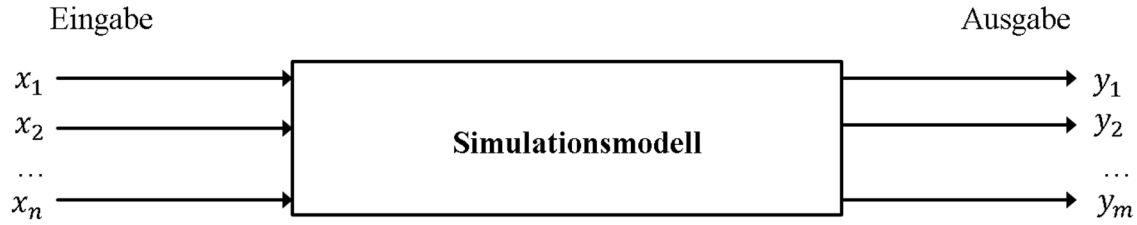


Abbildung 2-1 Ein Simulationsmodell (angelehnt an [CaMa97]).

Die Konfiguration des Modells kann angepasst werden und es können Experimente durchgeführt werden. Normalerweise ist dies unmöglich, zu teuer oder unpraktisch mit dem realen System, welches durch ein Systemmodell repräsentiert wird, durchzuführen. Weitestgehend ist Simulation ein Tool zur Evaluation der Performanz von existierenden oder zukünftigen System unter verschiedenen Konfigurationen die von Interesse sind und über einen langen Zeitraum (Echtzeit) getestet wurden (vgl. [Mari97]).

Simulationen unter anderem zu sicherheitskritischen Systemen haben oftmals den Zweck, physikalische Belastungsgrenzen zu bestimmen. Diese führen bei Überschreitung zur Zerstörung des Modells, was bei der Verwendung von Computern jedoch keine Konsequenzen hat (vgl. [Boss04]). Für die richtigen Rückschlüsse aus derartigen Experimenten muss jedoch eine hinreichend korrekte Abbildung zwischen Original und Modell gesichert sein (vgl. [PLHH13]).

Durch eine Änderung der Variablen innerhalb der Simulation können Vorhersagen über das Systemverhalten gemacht werden. Es ist ein Tool zur virtuellen Untersuchung des Verhaltens des untersuchten Systems (vgl. [Bank10]).

Computer Simulationen werden oft als Ergänzung bei der Analyse modellierter Systeme verwendet bei denen eine analytische Lösung nicht möglich ist. Es existieren viele verschiedene Arten von Computer Simulationen. Eine Eigenschaft teilen sich aber alle, den Versuch repräsentative Szenarios für ein Modell zu generieren für die ein Testen aller möglichen Zustände nahezu unmöglich wäre.

Ein bekannter Simulationstyp ist die Monte Carlo Simulation, welche nach der Stadt in Monaco benannt wurde, die bekannt ist für ihre Casinos und Glücksspiele. Bei ihr handelt es sich um eine mächtige mathematische Methode, um eine quantitative Risikoanalyse durchzuführen. Die Monte Carlo Methode basiert auf dem Random Sampling Verfahren, was als Gegenstück zum Münzwurf angesehen werden kann. Die Ergebnisse vom Random Sampling werden an das mathematische Modell weitergegeben und benutzt, um die Ausgaben der Simulation zu berechnen. Dieser Prozess wird viele Male wiederholt und dann mit Hilfe von Tools ausgewertet, um Statistiken über die Auswirkungen verschiedener Parameterkonstellationen zu erhalten. [Rayc08]

2.2.1 Methoden und Techniken zur simulativen Risikoanalyse

Im Bereich der simulativen Risikoanalyse existiert bereits eine große Anzahl an vorhandenen Programmen. Ein kleiner aber repräsentativer Teil dieser soll in den folgenden Abschnitten vorgestellt werden. Die enthaltenen Informationen basieren dabei auf Literaturrecherchen und wenn nicht anders vorliegend auf Angaben der Hersteller. Am Ende dieses Abschnitts wird eine Tabelle mit einer Zusammenfassung präsentiert (s. Tabelle 2-1).

IWrap[1]

IWRAP (IALA Waterway Risk Assessment Program) ist ein simulatives Analysetool (vgl. [Iala10,S.11]) welches für eine Risikobewertung in der maritimen Domäne genutzt werden kann. Mittels IWRAP kann die Frequenz von Kollisionen und Strandungen in einem gegebenen Wasserweg gemessen werden ausgehend von modellierten Informationen über das Verkehrsaufkommen und der Routengeometrie. Dabei sind die betrachteten Risiken vorgegeben und können nicht ohne größeren

[1] IWRAP:http://www.gatehouse.dk/en-US/Fields-of-Expertise/Maritime/Products/IWRAP-Risk-analysis.aspx [zuletzt abgerufen am 03.03.2017]

Aufwand um neue Risikobewertungen erweitert werden.

RENO[2]

Bei RENO handelt es sich um eine Simulationssoftware für eine probabilistische Ereignis- und Risikoanalyse. Das Tool erlaubt es dem Nutzer, mit Hilfe von Flussdiagrammen komplexe Risikoanalysen durchzuführen. RENO setzt dabei auf eine visuelle Erstellung der Modelle, mit der probabilistische oder deterministische Szenarien gebaut werden können. Die Software bietet dabei eine Reihe von Definitionen und Konstrukte, die es erlauben, die zu untersuchende Situation schneller zu erstellen. Zudem gibt es eine integrierte Auto-Vervollständigen und Debugger-Funktionen, die bei der Validierung des Modells helfen. Nach Erstellung des Modells können mittels Simulation die Ergebnisse für die Risikoanalyse abgeschätzt werden.

@Risk[3]

Das Programm @Risk erstellt mittels Monte-Carlo Simulation Risikoanalysen. Dabei wird auf eine vom Nutzer anzulegende Kalkulationstabelle zugegriffen. @Risk ermittelt die möglichen Ergebnisse und deren Wahrscheinlichkeiten. Durch eine Verwendung von genetischen Algorithmen oder der Anbindung von OptQuest ermöglicht @Risk zusätzlich die optimale Belegung von Variablen zu ermitteln, so dass die bestmögliche Zuweisung von Ressourcen oder der effizienteste Ablaufplan gefunden werden kann. @Risk setzt dabei auf eine vollständige Integration in die Microsoft Excel Umgebung was eine Anwendung für Simulationen in denen ebenfalls komplexe physikalische Prozesse betrachtet werden nahezu unmöglich macht.

Risk Solver Pro[4]

Bei Risk Solver Pro handelt es sich ebenfalls um ein Risikoanalyse Tool für vorhandene Microsoft Excel-Modelle mittels Monte-Carlo-Simulation. Es besitzt analytische Wahrscheinlichkeitsverteilungen, Statistik- und Risikomaße und die Möglichkeit mehrere parallele Simulationsläufe zu parametrisieren. Zusätzlich zu einer Vielzahl von Darstellungsmöglichkeiten durch Diagramme bietet Risk Solver Pro Methoden zur Sensitivitätsanalyse und „Was wäre wenn?“-Analysen. Bei diesem

[2] RENO: http://www.reliasoft.com/reno/ [zuletzt abgerufen am 03.03.2017]
[3] @RISK: http://www.palisade.com/risk/de/ [zuletzt abgerufen am 03.03.2017]
[4] Risk Solver Pro: http://www.solver.com/risk-solver-pro [zuletzt abgerufen am 03.03.2017]

Tool stellt es sich jedoch ebenfalls als Schwierigkeit heraus komplexe physikalische Prozesse zu betrachten durch die Verwendung von Microsoft Excel als einzige Möglichkeit zur Modellierung.

GoldSim[5]

GoldSim ist eine probabilistische Simulationssoftware zur Risikoanalyse und Entscheidungsunterstützung. Die GoldSim Plattform erlaubt die Visualisierung und dynamische Simulation vieler verschiedenartiger Systeme (Physikalische Systeme, Finanz Systeme, usw.). Der Nutzer baut ein Model seines Systems in Form eines Einfluss-Diagramms in dem mittels Graphen Daten und Gleichungen erstellt und manipuliert werden können. Neben der Möglichkeit verschiedene Wahrscheinlichkeitsverteilungen anzuwenden und Risiko- und Verlässlichkeitsanalysen durchzuführen unterstützt GoldSim auch ein Distributed Processing Modul, welches die Simulationsausführung auf verschiedenen Computern im Netzwerk erlaubt, wobei jedoch die Anbindung externer Simulatoren nicht unterstützt wird.

TUTS

Der TUTS (Threshold Uncertainty Tree Search) Algorithmus ist ein auf Simulation basierender Ansatz um seltenes aber kritisches Fahrerverhalten in Zusammenarbeit mit einem Assistenzsystem zu entdecken. TUTS setzt dabei auf der Monte-Carlo Simulation auf, in dem Sinne, dass wiederholt ein Szenario aus einem initialen Zustand gestartet wird, dabei aber verschiedene auf Wahrscheinlichkeiten basierende Wahlen trifft. Um seltene Situationen zu entdecken benötigt der TUTS Algorithmus eine benutzerdefinierte Funktion, welche die Kritikalität für jeden Simulationslauf misst. Damit die möglichen Wahlen zur Verfügung stehen, müssen eingesetzte Simulatoren ihren internen Zustand für den TUTS Ansatz freigeben. [POFC12, PWFP13, PWFP13]

[5] GoldSim: http://www.goldsim.com [zuletzt abgerufen am 03.03.2017]

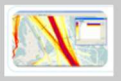
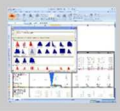
Name	Beschreibung	Pro	Contra
IWrap 	Software zur Risikobewertung von Schiffsverkehr bestimmter geografischer Gebiete	- Geprobter Einsatz in der maritimen Domäne	- Vorgegebene nicht einfach erweiterbare Risikofaktoren - Hohes Abstraktionslevel
RENO 	Simulationssoftware für eine probabilistische Ereignis- und Risikoanalyse	- Graphische Modellierung	- Flussdiagramme zur Modellierung der Simulation
@Risk 	Risikoanalyse mittels Monte-Carlo Simulation	- Anbindung verschiedener Optimierungsstrategien	- Vollständige Integration in MS Excel Umgebung
Risk Solver Pro 	Risikoanalyse mittels Monte-Carlo Simulation	- Parallelisierung von Simulationsläufen - Methoden zur Sensitivitätsanalyse	- Aufbauend auf mittels MS Excel erstellten Modellen
GoldSim 	Probabilistische Simulationssoftware zur Risikoanalyse	- Dynamische Simulation verschiedenartiger Systeme - Verteilte Simulationsausführung möglich	- Einbindung externer Simulatoren nicht möglich
TUTS 	Simulativer Ansatz zur Entdeckung seltenen aber kritischen Fahrerverhaltens	- Auf Distanzen basierende Bewertung - Entscheidungsfindung auf Basis des internen Zustands	- Keine Unterstützung von Black-Box Simulatoren

Tabelle 2-1 Übersicht über die in dieser Arbeit betrachteten Methoden und Techniken zur simulativen Risikoanalyse.

2.3 Simulationsoptimierung

Simulationsoptimierung ist ein Bereich, der die Aufmerksamkeit vieler Forscher, auch im Bereich der Risikoanalyse, angezogen hat. So wird dort nach Einstellungen für die Eingabeparameter gesucht, die ein Risiko, eine Risikosituation begünstigen. Die Simulationsoptimierung hat dabei den Vorteil viel effizienter eine sehr viel größere Anzahl von Szenarien zu analysieren als traditionelle Ansätze zur Optimierung (Szenario Optimierung, Robust Optimization) (vgl. [BGKW08]).

Das Ziel der Simulationsoptimierung ist die Minimierung der eingesetzten Ressourcen, weil gleichzeitig eine Maximierung der ermittelten Evaluationen in Simulationsexperimenten erreicht wird (vgl. [CaMa97]).

Die Simulationsoptimierung versucht die optimale Einstellung für gegebene Eingabeparameter zu finden (s. Abbildung 2-1: x_1 - x_n), welche die Ausgabewerte (s. Abbildung 2-1: y_1 - y_n) optimieren ohne jedoch explizit jede mögliche Konstellation evaluieren zu müssen. Probleme dieser Art sind regelmäßiger Bestandteil des ingenieurmäßigen Vorgehens. Ein Modell zur Simulationsoptimierung ist in Abbildung 2-2 zu sehen.

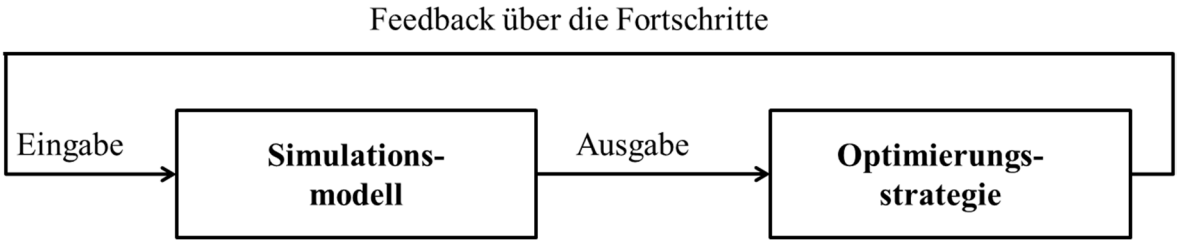


Abbildung 2-2 Ein Simulationsoptimierungsmodell (angelehnt an [CaMa97]).

Das Ergebnis des Simulationsmodells wird von einer Optimierungsstrategie verwendet, um Feedback zur weiteren Suche der optimalen Lösung zu bekommen. Dies wiederum liefert die nächsten Einstellungen der Eingabeparameter für das Simulationsmodell.

Optimierungsalgorithmen sind in der angewandten Mathematik dem Gebiet der Optimierung zugeordnet. Diese werden eingesetzt, wenn für eine nichtlineare Abbildung das globale Optimum und die für dieses benötigte Variablenwerte gefunden werden sollen.

Bei nichtlinearen Abbildungen gilt das Prinzip der Superposition nicht. Aus mehreren vorgegebenen Variablenwert Kombinationen kann nicht auf ein unbekanntes Ergebnis zu gegebenen Variablenwerten geschlossen werden (vgl. [Benk03]). Eine vollständige Überprüfung aller Permutationen der Variablenwerte ist zwar möglich, jedoch sehr ineffizient und nur bei sehr kleinen Suchräumen anwendbar (vgl. [KBKM12]). Durch die wiederholte Anwendung eines bestimmten Systems versuchen Optimierungsalgorithmen lokale Minima oder Maxima zu finden (vgl. [HeGt10]).

Eine Möglichkeit zum Finden von Minima und Maxima ist zum Beispiel der Einsatz von evolutionären Algorithmen. Aus Sicht der Mathematik bedeutet dies, dass der Algorithmus eine Permutation der zuletzt verwendeten Variablenwerte erzeugt, das Ergebnis berechnet und den Vergleich zu vorangegangenen Ergebnissen zieht. Diese Methode wiederholt sich, bis die Abbruchbedingungen erfüllt wurden. Ein Optimierungsalgorithmus erzeugt folglich die Konfigurationswerte für die Simulation und erhält nach Abschluss der Simulation die Ergebniswerte.

Viele Optimierungsverfahren sind auf die Effizienz der darunterliegenden Simulation angewiesen. Um diese möglichst effizient zu machen existieren verschiedene Techniken, wie die, im nächsten Abschnitt vorgestellte, Rare Event Simulation, welche genutzt wird um einen häufigeren Auftritt seltener Ereignisse in Simulationen zu erreichen.

2.3.1 Rare Event Simulation

Seltene Ereignisse in Simulationen, sind Ereignisse mit Auftrittswahrscheinlichkeiten ab 10^{-9}. Meist handelt es sich bei den seltenen Ereignissen um Ausnahmesituationen die ohnehin so selten wie möglich auftreten sollen. Dennoch sind meist gerade diese seltenen Ereignisse bei Simulationen von Interesse, da gerade Ausnahmesituationen zu Fehlern führen welche einen großen Schaden hervorrufen. Um verlässliche Aussagen über das Ereignis treffen zu können, werden viele unabhängige Auftritte des Ereignisses benötigt. Wie erwähnt dauert es zu lange ein, geschweige denn hunderte solcher Auftritte von seltenen Ereignisse zu beobachten, wenn man eine naive Simulation verwendet. Deswegen werden hier Rare Event Simulationstechniken benötigt, um schneller an verlässliche Aussagen zu kommen [JeLS13, JuSh06, KaOl07, MoPG10, Flur90, Glas93].

Rare Event Simulationstechniken sind statistische Methoden zu denen Techniken aus dem Bereich der Bewertungs- und Auswahlmethoden gehören.

Importance Sampling wurde effektiv eingesetzt und kann signifikante Geschwindigkeitssteigerungen erreichen bei der Suche nach seltenen Ereignissen [Shah95].

Die Grundidee von Importance Sampling ist ein System zum Beispiel mit verschiedenen Wahrscheinlichkeitsverteilungen zu simulieren. Dies erhöht wiederum die Wahrscheinlichkeit der Auswahl typischer Pfade, die das seltene Ereignis von Interesse betreffen. Für jeden Pfad der während der Simulation betrachtet wird, wird die geschätzte Messung mit einem Korrekturfaktor multipliziert, um eine nicht verzerrte Schätzung der Messung im Originalsystem zu erhalten. Die größte Schwierigkeit beim Importance Sampling ist das Bestimmen einer angemessenen Änderung der Messungen [BiAz13, CrCS11]. Innerhalb der Importance Sampling Methode existieren verschiedene Unterarten, welche wiederum unterschiedliche Eigenschaften garantieren. So gibt es neben anderen noch das Balanced Failure Biasing, welches garantiert, dass der Fehlerwert innerhalb eines angegebenen Bereiches gesucht wird (vgl. [NiSN01]).

Eine der bekanntesten Methoden aus dem Bereich der Rare Event Simulation ist das Importance Splitting. Beim Importance Splitting wird der Simulationsprozess an Zwischenzuständen zu seltenen Ereignissen aufgeteilt, um die Menge auftretender ausnahmefreier Zustände zu minimieren (s. Abbildung 2-3). Dabei verfolgt man die iterative Approximation an die seltenen Ereignisse, wobei wenig aussichtsreiche Trajektorien, also Simulationsläufe, die sich vom seltenen Ereignis entfernen, wieder verwirft (vgl. [Heid95]).

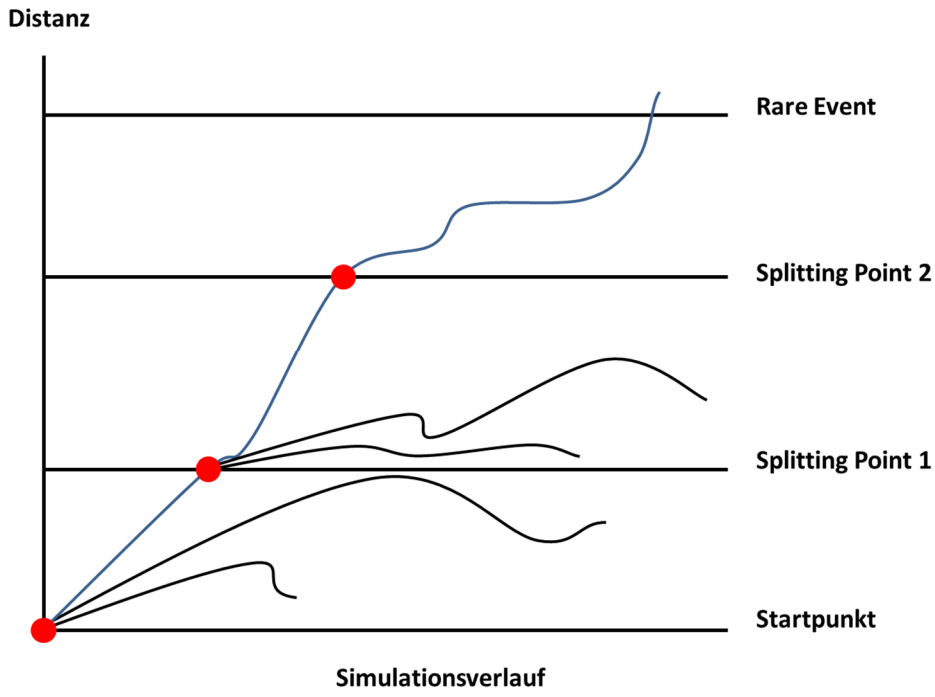


Abbildung 2-3 Beispiel für optimierte Simulationstrajektorien durch den Einsatz einer Importance Splitting Technik.

Bei einer naiven Simulation werden meistens die Gebiete von Zuständen simuliert, die weit entfernt sind vom zu beobachtenden seltenen Ereignis. Ein Beispiel hierfür ist die Simulation von Pufferüberläufen in einer Netzwerkqueue. Ist die Servicerate der Pakete um einiges größer als die Ankunftsrate in der betrachteten Warteschlange ist der Puffer meistens wenig befüllt oder sogar leer. Dies bedeutet, dass viel Rechenzeit verschwendet wird, da der Systemzustand weit entfernt von einer gefüllten Netzwerkqueue ist (Abbildung 2-3: schwarze Pfade). Ein großes Interesse besteht jedoch daran das System n-mal zu simulieren und dabei aussichtsreiche Simulationsläufe weiter zu verfolgen (Abbildung 2-3: blauer Pfad) und die anderen zu verwerfen (vgl. [Heid95]).

Eine Erweiterung des Importance Splitting ist die RESTART Methode. RESTART steht dabei für **RE**petitive **S**imulation **T**rials **A**fter **R**eaching **T**hresholds. Ein wichtiger Vorteil der RESTART Methode ist das die Anwendung unabhängig von der Charakteristik des simulierten Systems angewendet werden kann. Die RESTART Methode die mittels einem Threshold Wert arbeitet wurde erstmals 1991 von Villén-Altamirano vorgestellt. Die Idee dahinter ist die folgende: wenn ein seltenes Ereignis A gegeben ist von dem die Wahrscheinlichkeit bestimmt werden soll wird ein zusätzliches Ereignis C definiert. Hierfür gilt $C \supseteq A$ und $p\{A\} \ll p\{C\} \ll 1$, so dass sich $p\{A\} = p\{C\} \cdot p\{A \,/\, C\}$ ergibt. In einer naiven Simulation lässt sich $p\{C\}$ besser schätzen als $p\{A \,/\, C\}$, da $p\{C\}$ von der ganzen Simulation geschätzt wird wohingegen $p\{A \,/\, C\}$ nur von dem kleineren Teil der Simulation geschätzt wird in dem C auftritt. In einer RESTART Simulation wird die Schätzung der $p\{A \,/\, C\}$ verbessert in dem mehrere Simulationsläufe wiederholt durchgeführt werden in denen das Ereignis C auftritt. [GöSc96, ViVi94]

Der Ansatz sich dem risikoreichen Zustand zu nähern und aus einem näheren Zustand neue Simulationsläufe zu starten ist es auch welcher in dieser Arbeit näher betrachtet wird um die Risikodistanz und die Steuerung der Simulationsläufe zu vereinen (s. Abschnitt 3.3.5 - Nutzung der Risikodistanzen in einer Co-Simulation).

Durch Importance Sampling und Importance Splitting existieren zwei verschiedene Verfahren um die benötigten Informationen über seltene Ereignisse zu ermitteln. Bei dem Importance Sampling Verfahren muss genau bekannt sein wie das System aufgebaut ist und funktioniert um eine neue, effektivere Wahrscheinlichkeitsverteilung zu finden, welche eine höhere Wahrscheinlichkeit und kleinere Varianz für das seltene Ereignis aufzeigt (vgl. [RBSJ12]). Beim Importance Splitting Verfahren dagegen muss, damit der Algorithmus implementiert werden kann, nur eine Importance Function gefunden werden. Diese kann bei komplexeren Systemen auch mehrdimensional sein (vgl. [JeLS13]).

Die Anwendung der vorgestellten Rare Event Methoden ist jedoch nicht leicht möglich da einige Anpassungen erfolgen müssen bis eine Verwendung erfolgen kann (vgl. [HoKl79]).

2.3.2 Methoden und Techniken zur Simulationsoptimierung

Es befindet sich eine Vielzahl von Programmen auf dem Markt, welche zu Simulations- und Optimierungszwecken eingesetzt werden. In den folgenden Abschnitten wird eine Selektion der betrachteten Frameworks und Software beschrieben. Die enthaltenen Informationen basieren dabei auf Literaturrecherchen und wenn nicht anders vorliegend auf Angaben der Hersteller. Am Ende dieses Abschnitts wird eine Tabelle mit einer Zusammenfassung präsentiert (s. Tabelle 2-2).

OptQuest

Das erste vorgestellte System namens OptQuest ist eine Engine, die zur Optimierung von komplexen Systemen, wie Simulationsmodellen, eingesetzt werden kann. Hierfür werden Algorithmen wie die Scatter- und Tabu-Suche (vgl. [GlLa13, MaLG06]), sowie künstliche neuronale Netze verwendet. Dabei ist OptQuest kein eigenständiges Programm, sondern eine Programmbibliothek, die Funktionen zur Optimierung von externen Simulationen zur Verfügung stellt.

Bei OptQuest ist es die Aufgabe des Nutzers eine Verbindung zwischen den einzelnen Komponenten zu implementieren. Es existieren verschiedene kommerzielle Anwendungen zur Ausführung einer Simulation, wie Crystal Ball[6], Arena[7], ProModel[8] und SIMUL8[9], welche eine OptQuest-Integration integriert haben. Die Aufgabe des Hauptteils dieser Software dient jedoch der Simulation von Geschäftsprozessen, dem Ressourceneinsatz und der Planung innerhalb von Unternehmen (vgl. [Optq11]).

[6] Crystal Ball: http://www.oracle.com/de/products/applications/crystalball [zuletzt abgerufen am 03.03.2017]
[7] Arena: https://www.arenasimulation.com [zuletzt abgerufen am 03.03.2017]
[8] ProModel: https://webdev1.promodel.com [zuletzt abgerufen am 03.03.2017]
[9] Simul8: http://www.simul8.com [zuletzt abgerufen am 03.03.2017]

Guido

Bei Guido handelt es sich um eine hybride Software zur Verifikation von Schaltkreisen. Die Software verwendet dafür formale Techniken zur Verifikation um eine Simulation hinsichtlich eines Verifikationsziels zu leiten. Die Führung basiert dabei auf Distanzfunktionen, welche aus der Struktur der Schaltkreise abgeleitet werden. Ein so genannter Trace Sequence Controller überwacht und steuert dabei die Simulation in dem eine ausgeglichene Auswahl zwischen zufälligen Werten und kontrollierten Hill-Climbing stattfindet. Vom Standpunkt einer dynamischen Simulation kann Guido als eine Technik betrachtet werden um den Explorationspfad einer zufälligen Simulation zu steuern, um zu einem Verifikationsziel zu gelangen. Während jedes Schrittes der Simulation wird eine Anzahl an potentiellen nächsten Zuständen in Betracht gezogen und der Zustand gewählt welcher näher an das gewählte Ziel führt [ShBe06].

ASTRO

Der Vorteil der RESTART Methode unabhängig von der Charakteristik des simulierten Systems zu sein wurde genutzt, um es in einer Simulationsbibliothek namens ASTRO (Advanced Simulation Tool with Restart Optimization) im Jahr 1994 von der Firma Telefónica I+D zu implementieren. ASTRO stellt eine Umgebung zur Verfügung die eine Nutzung der RESTART Funktionen ermöglicht wie eine Beobachtung der Zustandsvariablen, Speicherverwaltung und die Umplanung von Ereignissen (Event Rescheduling) [ViVi94]. Dabei wurde ASTRO für SUN Workstations als SET von C Funktionen umgesetzt (vgl. [VMGF94]). Die Verwendung in Co-Simulationen wird in den verfügbaren Quellen nicht betrachtet wo durch eine Nutzung in dem entwickelten Framework als nicht sinnvoll erachtet wurde.

MonteQueue

Die MonteQueue Software unterstützt ebenfalls die Importance Sampling Technik ist dabei aber auf die Simulation spezieller Systeme ausgelegt. Im Fall von MonteQueue für Product-Form Multiclass Queueing Netzwerke. Dabei wird ein Softwarepaket zur Verfügung gestellt welches die schnelle Überprüfung von großen und kleinen Netzwerken erlaubt. MonteQueue erhält dabei die geschätzten Werte über die Performanz in dem Importance Sampling Techniken angewandt werden. [RoWa97]

Plant Simulation

Das Tool Plant Simulation ist ein Simulationswerkzeug in dem hauptsächlich logistische Systeme durch Modelle dargestellt werden können. Durch die Erstellung dieser Modelle können Simulationen ausgeführt werden, um Eigenschaften eines Systems zu untersuchen und Optimierungen und mögliche Schwachstellen zu erkennen. Hierfür werden von Plant Simulation genetische Algorithmen eingesetzt, die durch ein vorgegebenes Ergebnis die zugehörigen Eingabewerte ermitteln. Für unterschiedliche Szenarien können durch den Einsatz von Analysewerkzeugen Statistiken und Diagramme generiert werden. Dieses Programm wird dabei hauptsächlich für eine Optimierung der Produktivität, der Personalplanung und weiterer betriebswirtschaftlicher Faktoren eingesetzt [Plan15].

EMSO

EMSO (Environment for Modeling, Simulation and Optimization) ist eine Simulationssoftware in der Modelle für Simulationen erstellt, ausgeführt und optimiert werden können. Diese Software wurde im Rahmen einer Masterarbeit an der Bundesuniversität von Rio Grande do Sul entworfen [Emso04]. Danach wurde EMSO vom ALSOC-Projekt weiterentwickelt, welches diese Software kostenlos zur freien Verfügung stellt (vgl. [Also00]). EMSO erlaubt die Erstellung von Simulationsmodellen mittels einer grafischen Oberfläche. Für die erstellten Modelle wird neben einem eigenen Optimierungsverfahren eine Schnittstelle angeboten, um externe Algorithmen zu integrieren. Simulationen lassen sich dabei jedoch nicht direkt von außen anbinden, sondern müssen über den vorhandenen internen Model-Editor angelegt werden (vgl. [Emso04]).

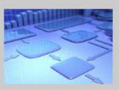
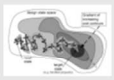
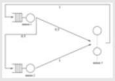
Name	Beschreibung	Pro	Contra
OptQuest 	Engine zur Optimierung von komplexen Systemen	- Bibliothek - Unterstützung verschiedener Optimierungsverfahren	- Festgelegtes Einsatzgebiet: Geschäftsprozesse, Planung innerhalb von Unternehmen
Guido 	Hybride Software zur simulativen Verifikation von Schaltkreisen	- Auf Distanzen basierende Bewertung	- Vorgegebene Optimierungsstrategie - Feste Domäne
ASTRO 	Umgebung zur Nutzung der RESTART Funktionalität	- Bibliothek - Beobachtung von Zustandsvariablen	- Keine Verwendung in Co-Simulationen betrachtet - Relativ hohes Alter (1994)
MonteQueue 	Software zur Anwendung der Importance Sampling Technik	- 4 Importance Sampling Techniken integriert	- Ausgelegt auf spezielle Systeme
Plant Simulation 	Simulationswerkzeug zur Untersuchung logistischer Systeme	- Komplette Design- und Runtimelösung	- Hauptsächlich zur Untersuchung betriebswirtschaftlicher Faktoren
EMSO 	Simulationssoftware	- Komplette Design- und Runtimelösung - Schnittstelle zum einbinden eigener Optimierungsverfahren	- Keine Möglichkeit externe Simulatoren anzubinden

Tabelle 2-2 Übersicht über die in dieser Arbeit betrachteten Methoden und Techniken zur Simulationsoptimierung.

2.4 Zusammenfassung und Vorstellung des Handlungsbedarfs

Die vorangegangenen Abschnitte zeigen, dass verschiedene Software zur simulationsbasierten Risikoanalyse vorhanden ist. Allerdings gibt es wenig bekannte Programme und Bibliotheken, die die geforderte Flexibilität bei der Integration aller Prozessschritte, von der Darstellung der Ergebnisse bis zur Konfiguration, bieten.

Die gezeigten Produkte sind auf einen bestimmten Einsatzbereich der Simulationen spezialisiert oder bieten nur eine Komponente, um bestehende Programme um Optimierungsfunktionen zu erweitern. Zusätzlich war die Betrachtung von verteilten Simulatoren (insbesondere externen Simulatoren) bei keiner betrachteten Software ein Schwerpunkt.

Ein weiterer wichtiger Handlungsbedarf zeigte sich in der Beschreibung von Funktionen die die Nähe zu einer risikoreichen Situation berechnen. Hierbei fiel auf, dass in den untersuchten Programmen entweder nur bestimmte vorgegebene Risiken bewertet werden konnten oder aber allgemeine Funktionen angegeben werden konnten. Eine strukturierte Führung hinsichtlich der Erstellung von Risikodistanzfunktionen war in den untersuchten Methoden und Tools nicht vorhanden. Auch die Betrachtung der Unterstützung von Black-Box Simulatoren und die Verwendung von simulationslaufübergreifenden Informationen wurden wenig bis gar nicht betrachtet.

Bei der weiteren Betrachtung der vorhandenen Software und der Untersuchung hinsichtlich der Verwendbarkeit ergab sich, dass das eigene Konzept nicht in diese integrierbar oder mit diesen umsetzbar ist. Aus diesen Gründen wurde innerhalb dieser Arbeit die Entwicklung eines eigenen Frameworks als sinnvoll erachtet.

Des Weiteren wurden in diesem Kapitel auf die Optimierung von Simulationen durch Rare Event Simulationstechniken eingegangen. Auch im Bereich der Rare Event Techniken wurde Software vorgestellt, welche sowohl die Importance Sampling und Importance Splitting Technik unterstützen. Von einer Verwendung der vorgestellten Programme musste jedoch abgesehen werden da diese fest in ihre jeweiligen Simulationsumgebungen eingebettet sind.

Aus dem Ziel der Bewertung von Simulationszuständen für eine gezielte Analyse risikoreicher Systeme und der Betrachtung des aktuellen Stands der Wissenschaft und Technik konnten dabei übergeordnete Anforderungen an den eigenen Ansatz und die zu

analysierende Co-Simulation abgeleitet werden. Diese werden im folgenden abschließenden Abschnitt vorgestellt.

2.4.1 Übergeordnete Anforderungen

[A_A1] **Methodisch unterstützte Erstellung von Risikodistanzfunktionen** Um eine einfache Erstellung von Distanzfunktionen zur Berechnung der Nähe zur einer risikoreichen Situation zu ermöglichen, muss es möglich sein diese geführt herzuleiten.

[A_A2] **Eintritt des Risikos und Risikodistanz korrelieren** Damit eine korrekte Analyse möglich ist, muss gewährleistet sein, dass eine Bewertung durch die Risikodistanzfunktion innerhalb einer laufenden Simulation und die Näherung an die risikoreiche Situation korrelieren.

[A_A3] **Verwendbarkeit der Risikodistanzen zur Steuerung eines Simulationsverlaufs** Die berechneten Risikodistanzen können verwendet werden, um den Simulationsverlauf zu steuern und damit ggf. risikoreiche Situationen schneller zu erreichen.

[A_A4] **Unterstützung von Black-Box-Simulatoren** Oft werden auch Simulatoren, bei denen es keine Möglichkeit gibt den internen Zustand zu kennen, innerhalb von Co-Simulationen eingesetzt. Daher ist eine weitere Anforderung, dass auch diese Black-Box-Simulatoren mit dem entwickelten eigenen Ansatz verwendet werden können.

[A_A5] **Definierbarkeit von Risikodistanzen** Um eine korrekte Berechnung von Distanzen zu risikoreichen Situationen zu gewährleisten, muss das beobachtete System alle vom zuständigem Sicherheitsexperten als relevant angegebenen Eigenschaften beinhalten. Des Weiteren muss garantiert sein das diese zu jedem Simulationszeitpunkt verfügbar und aktuell sind.

[A_A6] **Unterstützung von Parameterexplorationen** Da zur Analyse eines Systems und dessen Risiken auch eine Aussage über vorhandene Grenzen von Attributen ermittelbar sein sollen ist eine weitere Anforderung, dass der entwickelte Ansatz eine Beschreibung und Durchführung von Parameterexplorationen unterstützt.

Neben den sechs übergeordneten Anforderungen an den eigenen Ansatz lassen sich zusätzlich auch die drei nachfolgenden übergeordneten Anforderungen an die zu analysierende Co-Simulation aufstellen.

[A_C1] **Beobachtbarkeit der Co-Simulation** Um eine Analyse des Systems durchzuführen, muss der Zugriff auf den Zustand einer Co-Simulation zu einem beliebigen Zeitpunkt des Simulationslaufs erfolgen können. Der Zugriff, also das Auslesen, darf dabei keine Auswirkung auf den Simulationsverlauf haben, weder Werte ändern noch eine zeitliche Verzögerung hervorrufen.

[A_C2] **Rückverfolgbarkeit der kommunizierten Daten** Damit zunächst eine manuelle Analyse über erstellte Logs und Traces und gegebenenfalls in fortführenden Arbeiten eine (semi-)automatische Überprüfung von Simulationsläufen möglich ist muss bekannt sein welcher Simulator wann bestimmte Daten kommuniziert hat.

[A_C3] **Steuerbarkeit der Co-Simulation** Um die Co-Simulation gezielt in die Richtung einer zu untersuchenden risikoreichen Situation zu führen, muss eine Steuerung der Co-Simulation vorgenommen werden können. Dazu gehört zum einen das Steuern der angeschlossenen Simulatoren und Komponenten hinsichtlich des Starten und Stoppen sowie dem Speichern und Laden des Co-Simulationszustands, aber auch das Setzen von Parametern, um eine Exploration dieser durchführen zu können.

.

3 Bewertung von Simulationszuständen in Co-Simulationen zur beschleunigten simulativen Analyse

Ausgehend von der wissenschaftlichen Fragestellung, den Zielen, dem ermittelten Handlungsbedarfs im Stand der Wissenschaft und Technik und den daraus abgeleiteten übergeordneten Anforderungen wurde ein Ansatz zur Bewertung von Risiken und der Nutzung dieser zur beschleunigten Erreichung aufgestellt.

In diesem Kapitel erfolgt die Beschreibung der Methodik zur Nutzung von Distanzen um eine Bewertung von Situationen hinsichtlich ihrer Distanz zu bestimmten Risiken in Co-Simulationen durchzuführen. Eine Distanz stellt dabei nicht zwangsläufig eine räumliche Nähe, sondern eine abstrakte Bewertung einer Situation, hinsichtlich der Nähe zu einem Risiko, dar. Das Risiko ist dabei definiert als eine Menge von kritischen Systemzuständen zu denen die Nähe einer aktuell untersuchten Situation bestimmt werden kann. Des Weiteren wird in den entsprechenden Abschnitten zum eigenen Ansatz auf Methoden eingegangen, die in dieser Arbeit zur Bewertung von Simulationszuständen (Situationen) genutzt werden. Zu diesen gehören Techniken aus dem Information Retrieval (vgl. [BüCC10, Ferb03, ThMC96]) welche ihren Einsatz zum Vergleich von Simulationszuständen finden, Distanzfunktionen im Allgemeinen (vgl. [PaLZ04, ScBG99, WuSA02, RoTa60]) und das aus der Baudomäne bekannte semiprobabilistische Sicherheitskonzept (vgl. [Din03, Klug07, Drpe06]).

Die entwickelte Methodik besteht aus den in der folgenden Abbildung 3-1 gezeigten drei Aspekten und ihrer vorgegebenen Reihenfolge. Der erste Aspekt ist die Modellierung des zu analysierenden Systems aus denen eine System-, Verhaltens- und Gefahrenbeschreibung hervorgeht. Der erste Aspekt wird durch die Nutzung der Arbeiten und entwickelten Tools von Droste (vgl. [Dros16]) und Pinkowski (vgl. [Pink15]) erfüllt. Der zweite Aspekt ist die Risikodistanzbeschreibung aus der Risikodistanzfunktionen entstehen, der letzte Aspekt ist die simulative Analyse, in der die zuvor ermittelten Beschreibungen eingehen, um den Simulationsablauf zu beschreiben.

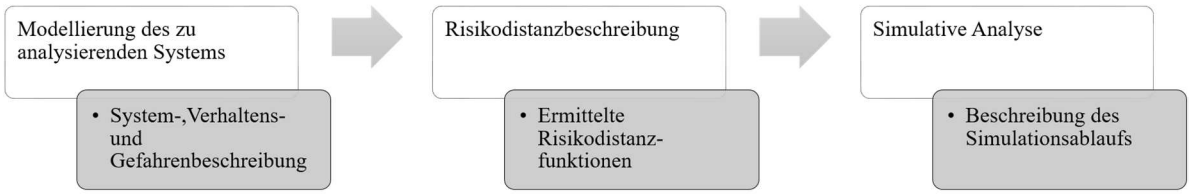


Abbildung 3-1 Die drei Aspekte der entwickelten Methodik (weiße Kästen) und ihre jeweiligen Ergebnisse (graue Kästen).

Im folgenden einleitenden Abschnitt werden die drei betrachteten Aspekte und ihre jeweiligen Bestandteile in Kurzform vorgestellt, während in den darauffolgenden Abschnitten detaillierter auf diese eingegangen wird.

Der erste Aspekt der Methodik ist die System-, Verhaltens- und Gefahrenbeschreibung (s. Abbildung 3-2. Hierbei wird zunächst ein System beschrieben (s. Abbildung 3-2:a). Dies setzt sich aus der Systembeschreibung inklusive einer Verhaltensspezifikation zusammen. Die Systembeschreibung wiederum besteht aus den betrachteten physikalischen Objekten und Umweltbedingungen.

Es lassen sich für das System relevante physikalische Objekte und Akteure auswählen und in einer Umgebung positionieren und Eigenschaften und Parameter wie z.B. die Masse einer physikalischen Ressource einstellen (s. Abbildung 3-3).

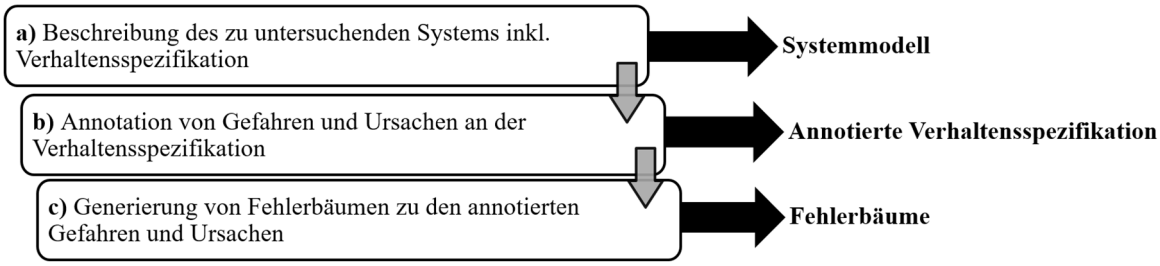


Abbildung 3-2 Vorgehen bei der Beschreibung des Systems, des Verhaltens und der möglichen Gefahren und Ursachen.

Um das Verhalten des Systems zu beschreiben, werden die Arbeitsabläufe aller Akteure, die Verwendung von Ressourcen und die Kommunikation zwischen Akteuren als Prozess modelliert. Zusätzlich ermittelt ein Systemexperte potentielle Gefahren und ordnet diese der Verhaltensspezifikation zu (s. Abbildung 3-2:b). Die annotierten Gefahren werden

wiederum genutzt, um Fehlerbäume zu generieren (s. Abbildung 3-2:c). Diese Basis nutzend setzt die entwickelte Methodik auf, welche die Ermittlung der Risikodistanzen und die Nutzung innerhalb einer Co-Simulation beschreibt.

Dabei verwenden diese Modelle als Grundlage ein gemeinsames Datenmodell (vgl. [DiHS14, SGHB14]). Das gemeinsame Datenmodell ist notwendig, um die Konzepte und Datentypen für eine formal beschriebene Semantik darzustellen und um die Kommunikation zwischen den Komponenten der Co-Simulation zu ermöglichen. Durch das gemeinsame Datenmodell wird zusätzlich ein gemeinsames Verständnis der Komponenten für ihre Umwelt aufgebaut.

Um die entwickelte Methodik nicht nur mit den Simulatoren verwenden zu können, die das gemeinsame Datenmodell verwenden, wird nur ein auf der MOF Spezifikation (Meta Object Facility) (vgl. [Spec07]) aufbauendes Modell benötigt, um das System zu beschreiben. Diese erlaubt den Zugriff auf die Eigenschaften des betrachteten Systems und eine Navigierbarkeit innerhalb der Objekt-Instanzen. Diese ist dabei nicht nur unidirektional möglich wie innerhalb der UML Sprache, sondern erlaubt eine bidirektionale Navigierbarkeit. Im Rahmen dieser Arbeit wurde eine Unterstützung mittels der Verwendung des Essential MOF (EMOF) (vgl. [Spec07]) umgesetzt, das eine Teilmenge der MOF ist. Durch die Verwendung dieses Meta-Modells des gemeinsamen Datenmodells ergibt sich zusätzlich der Vorteil, dass die Arbeit unabhängig von der Version des verwendeten gemeinsamen Datenmodells ist.

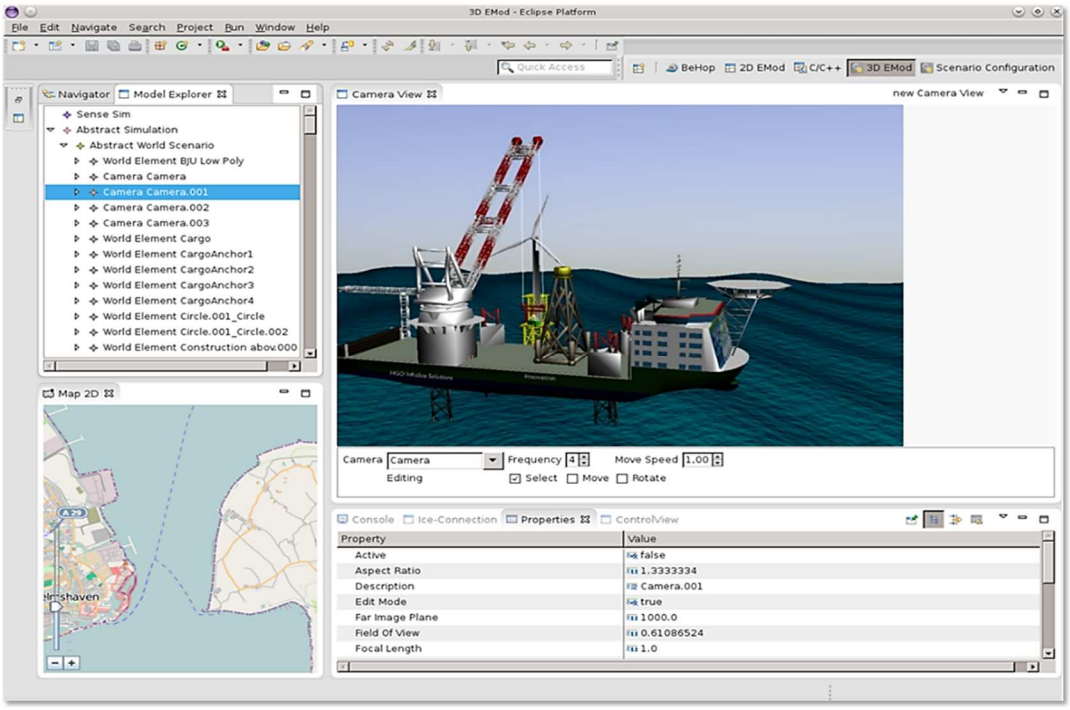


Abbildung 3-3 Screenshot eines Editors zur Konfiguration des Systemmodells.

Der zweite Aspekt beschreibt das generelle Vorgehen zur Ermittlung der Distanzbeschreibung, welches in dieser Arbeit entwickelt wurde (s. Abbildung 3-4) und bezieht sich damit direkt auf die Beantwortung der wissenschaftlichen Fragestellung und Erfüllung der Zielsetzungen aus Abschnitt 1.2 „*Zieldefinition und Beitrag der Arbeit*“.

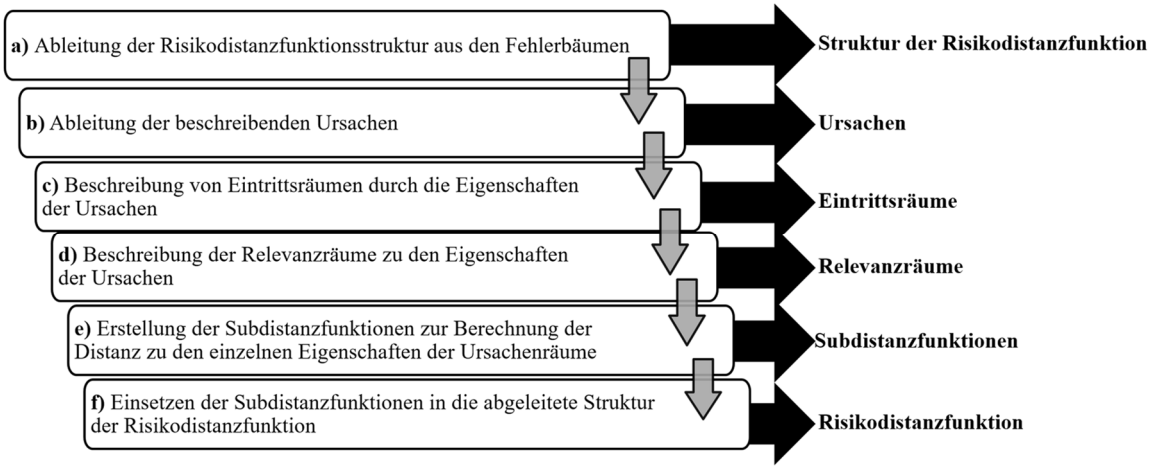


Abbildung 3-4 Vorgehen bei der Ermittlung der Distanzbeschreibung, die jeweilige Wertschöpfung der einzelnen Schritte ist durch die ausgehenden schwarzen Pfeile angegeben.

Ist die Definition des Systems abgeschlossen, wird vom Experten unter Zuhilfenahme der Fehlerbäume die Struktur der Risikodistanzfunktionen abgeleitet (s. Abbildung 3-4:a). Die Risikodistanzfunktion ist eine Bewertungsfunktion, welche die Nähe zu einer abstrakten, untersuchten, risikoreichen Situation bewertet. Die Angabe erfolgt dabei über verknüpfte mathematische Funktionen. Hierbei finden die im Vorfeld generierten oder erstellten Fehlerbäume Anwendung, indem ausgehend von dem im Fehlerbaum hinterlegtem hierarchischem und logischem Aufbau des Risikos und der Ursachen, Rückschlüsse für die Struktur der Distanzfunktion gezogen werden.

Im nächsten Schritt erfolgt die Ableitung der Ursachen aus dem Fehlerbaum (s. Abbildung 3-4:b), diese ergeben sich dabei aus den Basis-Events (den Blattelementen) der Fehlerbäume. Zu jeder Ursache wird vom Systemexperten ein Ursachenraum erstellt (s. Abbildung 3-4:c). Dieser besteht aus den vom Systemexperten ermittelten Eigenschaften, welche die Ursache beschreiben. Eigenschaften können sich dabei direkt auf die verbotene Zustandsmenge beziehen. Beim Risiko einer Kollision zwischen zwei Schiffen wäre ein Beispiel hierfür die Nähe zweier Objekte. Zum anderen können es auch das Risiko fördernde Eigenschaften sein, wie zum Beispiel die Übermüdung der Crew, die Verkehrsdichte oder die Sichtweite. Angegeben werden diese Ursachenräume durch eine minimale und/oder maximale Grenze. Daraufhin werden die Relevanzräume erstellt (s. Abbildung 3-4:d), die beschreiben, ab welchem minimalen und/oder maximalen Wert einer Eigenschaft eine Ursache relevant für die Risikoberechnung wird. Die Ursachen- und Relevanzräume nutzend, werden so genannte Subdistanzfunktionen erstellt (s. Abbildung 3-4:e), welche die Berechnung der Distanz zu den einzelnen Eigenschaften der Ursachenräume übernehmen. Die Kombinationen dieser Subdistanzfunktionen für jeden Ursachenraum ergeben die Distanz zu den einzelnen Ursachen des Risikos. Diese werden in die abgeleitete Struktur der Risikodistanzfunktion eingesetzt (s. Abbildung 3-4:f), wodurch die Funktion zur Berechnung der Distanz zu einem Risiko vervollständigt ist.

Der letzte Aspekt beschreibt die simulative Analyse. An dieser Stelle fließen die erstellten Modelle und ermittelten Risikodistanzfunktionen zur Beschreibung des Simulationsablaufs ein, um Regeln für den Verlauf einer Simulation zu beschreiben (s. Abbildung 3-5).

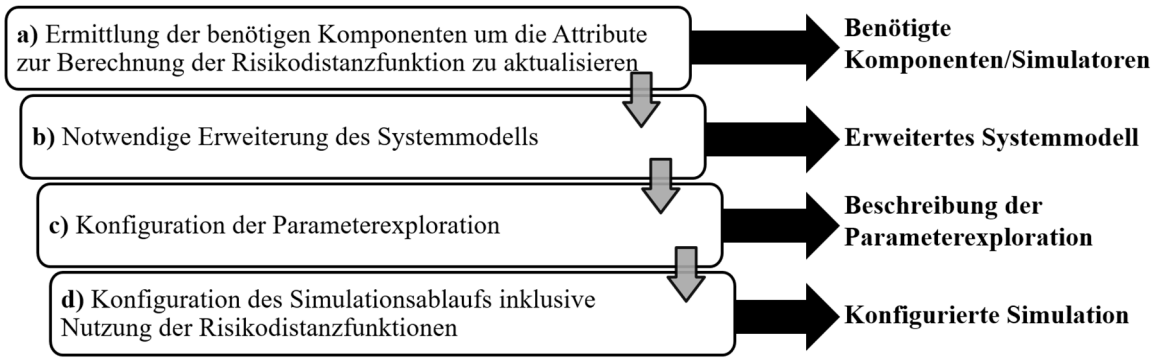


Abbildung 3-5 Vorgehen bei der Konfiguration des Verlaufs einer Simulation.

Im ersten Schritt werden die Komponenten und Simulatoren, der Co-Simulation, ermittelt, die zur Ausführung des Systems inklusive Verhaltensspezifikation benötigt werden (s. Abbildung 3-5:a). Im nächsten Schritt werden notwendige Erweiterungen am Systemmodell durchgeführt, die sich möglicherweise aus den zur Risikodistanzermittlung genutzten Komponenten ergeben (s. Abbildung 3-5:b).

Da zur Analyse des Systems oft verschiedene Parametrisierungen angewandt werden müssen, um deren Einfluss auf die Risiken mit betrachten zu können, wird in einem Explorationsmodell die Parameterexploration innerhalb der auszuführenden Simulation beschrieben, in der der Systemexperte festlegt, welche Systemeigenschaften auf welche Art exploriert und simuliert werden (s. Abbildung 3-5:c). Die einzelnen zu testenden Parametereinstellungen können dann in verschiedenen Simulationsläufen evaluiert werden.

Abschließend beschreibt der Systemexperte die Simulationssteuerung (s. Abbildung 3-5:d), hierfür werden die ermittelten Risikodistanzfunktionen verwendet, um den Verlauf der Simulation im Sinne einer Steuerung der Simulatoren zu beschreiben. So kann im Fall der Anwendung einer Importance Splitting-Technik das Ergebnis der Risikodistanzfunktionen genutzt werden, um das Speichern der Zustände der Simulatoren auszulösen oder gespeicherte Zustände von den Simulatoren laden zu lassen und diese mit einem neuen Parametersetting zu simulieren.

Die genaue Beschreibung der einzelnen Schritte der Methodik, der Systembeschreibung und der simulativen Analyse, wird in den nächsten Abschnitten an dem folgenden Beispiel aus der Offshore-Domäne präsentiert.

BEISPIEL 1: ZU ANALYSIERENDES SZENARIO

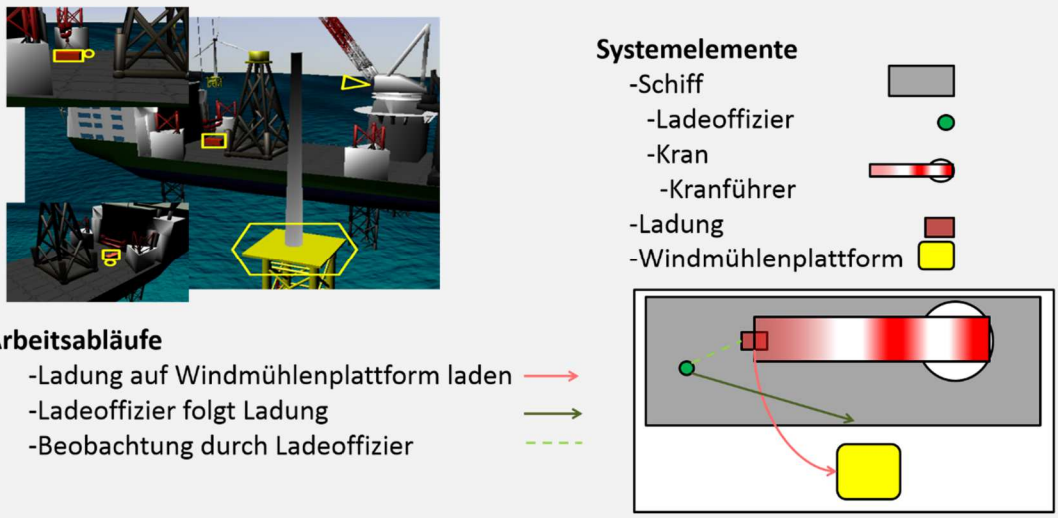


Abbildung 3-6 Übersicht über das zu analysierende Beispielszenario inklusive der zugehörigen schematischen Verhaltens- und Systembeschreibung.

Um die simulationsbasierte Risikobewertung zu erläutern, wurde ein maritimer Ladevorgang auf einem Jack-Up-Vessel[10], ein Spezialschiff für Offshore-Aktivitäten, gewählt. Ein Kranführer (s. Abbildung 3-6- Dreieck) transportiert eine Ladung (s. Abbildung 3-6-Rechteck) mit neuen Materialien zu einer Offshore-Plattform (s. Abbildung 3-6-Sechseck). Es gibt viele Faktoren, wie schlechte Sicht oder Kommunikationsprobleme, durch die möglicherweise Fehler auftreten, was zu Unfällen führen könnte. Aufgrund dieser Problematik wird während des Ladevorgangs eine Person (Ladeoffizier s. Abbildung 3-6-Kreis) benötigt, die das Laden überwacht und den Kranführer auf diese Probleme hinweisen kann. Daher folgt der Ladeoffizier der transportierten Fracht und versucht, sowohl den Kranführer als auch die Ladung in seinem Sichtfeld zu halten.

[10] Ein Jack-Up-Vessel/Errichterschiff ist ein spezielles für den Aufbau von Offshore-Windkraftanlagen entwickeltes Schiff mit Schwerlastkran. Es besitzt Hubvorrichtungen, um unabhängig vom Seegang arbeiten zu können.

3.1 Erstellung der System-, Verhaltens- und Gefahrenbeschreibung

Dieser Abschnitt erläutert die Arbeiten, die im Vorfeld der hier vorgestellten Methodik zur Risikodistanzermittlung durchgeführt werden müssen. Er bezieht sich dabei auf die Dissertationen von Droste und Pinkowski, die ein Modell zur Beschreibung von maritimen Arbeitsabläufen sowie deren formale Auswertung mithilfe von Fehlerbäumen, erstellt haben [Dros16, Pink15]. Dabei wird beschrieben, wie das simulierte System beschrieben werden kann, mittels eines System-, Verhaltens- und Fehlerbaummodells.

Als Startpunkt für die Untersuchung wird ein Modell des zu analysierenden Systems benötigt. Dieses setzt sich zusammen aus einer Systembeschreibung inklusive einer Verhaltensspezifikation. Zur Verhaltensspezifikation wird ein Prozess Modell verwendet (vgl. [DLSB12]), das auf Konzepten der Geschäftsprozessmodell und –notationsprachen basiert, wie der Business Process Model Notation (BPMN) und dem Aktivitäts- Diagramm der Unified Modelling Language (UML). Der Prozess beschreibt dabei Akteure, Aufgaben und dazugehörige Interaktionen. Dieser Modelltyp wurde im Speziellen für die Anwendung in der maritimen Domäne, spezieller für Offshore Operationen entwickelt kann aber auch auf weitere Domänen angewendet werden (vgl. [Dros16]). Die graphischen Elemente des Modells beschreiben dabei Aufgabensequenzen und Interaktionen. Weiterhin ist es möglich, diese mit dazugehörigen Gefahren sowie ihren Ursachen zu annotieren. Eine Gefahr beschreibt dabei ein mögliches Risiko, wie zum Beispiel die Verletzung oder den Tod einer Person oder einen Maschinenschaden. Eine Ursache beschreibt einen möglichen Auslöser für eine Gefahr. Bei der Annotation am Prozessmodell wird eine formale Spezifikation für Gefahren und Ursachen verwendet (vgl. [LäBP12]).

Mittels des Systemmodells lassen sich die betrachteten physikalischen Objekte und Umweltbedingungen beschreiben. Diese sind ebenfalls für die Verwendung im Prozessmodell nutzbar, um Akteure zu einem spezifischen Avatar (ein physisches Objekt des Systems) in der simulierten Umgebung zuzuordnen. Zum Beispiel kann ein Ladeoffizier zu einer Ladeoffizier-Ressource der simulierten Umgebung zugeordnet werden, die eine Position, maximale Geschwindigkeit und weitere für die Simulation relevante Attribute besitzt.

BEISPIEL 2: SYSTEMBESCHREIBUNG

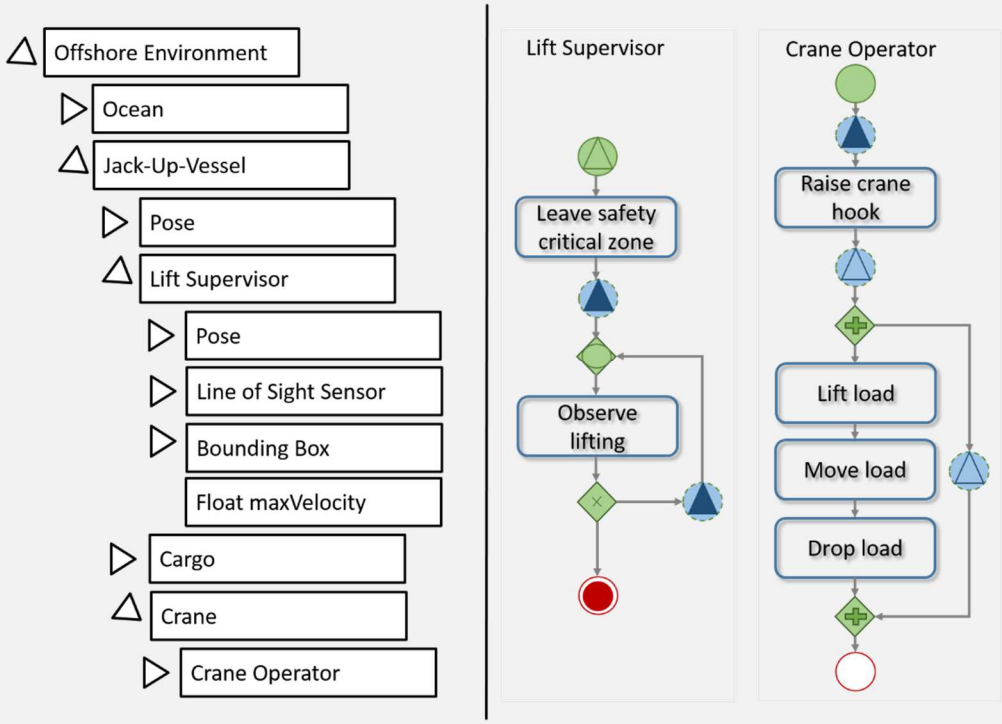


Abbildung 3-7 Beschreibung des Systems zum vorgestellten Beispielszenario.

In der Abbildung 3-7 sind die für das erwähnte Beispielszenario erstellten Teile des Systemmodells zu sehen. Das Prozessmodell (s. Abbildung 3-7: rechts im Bild) strukturiert dabei das Beispielszenario und damit die Aufgaben der beteiligten Akteure in eine BPMN ähnliche, graphische Repräsentation.

Das Szenario beginnt mit dem Start des Hebevorgangs. Der Kranführer signalisiert das beabsichtigte Anheben der Fracht an den Ladeoffizier. Der Ladeoffizier verlässt die sicherheitskritische Zone und signalisiert die Genehmigung des Hebevorgangs an den Kranführer. Wenn der Kranführer das OK erhält, wird die Ladung angehoben, zur Offshore-Plattform bewegt und dort positioniert. Der Ladeoffizier informiert den Kranführer kontinuierlich über den aktuellen Status, während der Kranführer den Kran dementsprechend steuert. Dabei benötigt der Ladeoffizier eine klare Sicht auf die Ladung, den Kranführer und potenzielle Hindernisse. Das Szenario endet mit dem erfolgreichen Platzieren der Fracht auf der Plattform.

Ein Teil der statischen Systembeschreibung (s. Abbildung 3-7: links im Bild) zu dem Beispiel aus der Offshore-Domäne beschreibt die genutzten Akteure, physikalischen Objekte und deren Eigenschaften. Zusätzlich wird die Zugehörigkeit der verwendeten Elemente ihren jeweiligen Referenzelementen gegenüber mittels der Systembeschreibung ermöglicht. So ist der Kran als ein Aufbau des Schiffes diesem über die Hierarchie der

Systembeschreibung zugeordnet. Die physischen Objekte bestehen aus dem Jack-Up-Vessel, dem darauf positionierten Kran des Schiffes und der transportierten Ladung. Die Akteure umfassen den Kranführer sowie den Ladeoffizier. Die physikalischen Objekte, sowie die Akteure haben Posen (Position und Orientierung) und unter anderem eine zugewiesene Masse als Eigenschaft, damit die Simulationsphysik realistisch auf sie wirken kann.

3.1.1 Fehlerbaumbeschreibung

Ein bekannter Weg, um eine Charakterisierung der involvierten Risiken durchzuführen, sind Fehlerbäume (vgl. [VDFM02]). Fehlerbäume geben einen Überblick über die potenziellen Risiken und die Verknüpfung ihrer möglichen Ursachen. Zusätzlich teilen Fehlerbäume das Risiko in verschiedene Ereignisse auf die in ihrer Kombination, vermutlich zum unerwünschten Risiko führen.

An der Verhaltensspezifikation werden von einem Systemexperten Gefahren annotiert, die während der Durchführung auftreten könnten. Zusätzlich werden von einem Systemexperten einzelnen Elementen der Verhaltensspezifikation (Akteur oder Aufgabe) mögliche auftretende Ursachen für die annotierten Gefahren zugeordnet.

Für jede annotierte Gefahr, wird ein Fehlerbaum erstellt, um die jeweiligen Ursachen, welche die Gefahr möglicherweise verursachen, logisch zu strukturieren. Da jede Ursache in Beziehung zu einem Element des Prozesses steht, ist der Fehlerbaum ebenfalls mit diesem Element verbunden. Umgekehrt, ist jeder Top-Event eines Fehlerbaums (Wurzelknoten) mit einer Gefahr verknüpft. Zum Beispiel können einem Aufgaben-Element zugewiesene Ursachen mit mehreren Fehlerbäumen verbunden sein, die die Gefahren und verknüpften Ursachen grafisch repräsentieren. Im Gegensatz zu anderen Methoden zur Fehlerbaum-Generierung, die zum Beispiel UML-Diagramme als Quelle zur Generierung nutzen (vgl. [LaGP11, PaDu02]) sind im vorgestellten Prozess Modell alle benötigten Informationen vorhanden. Um Fehler zu vermeiden sind die spezifischen, dem Prozess zugeordneten Gefahren und Ursachen dabei von einem erfahrenen Systemexperten zu modellieren [LPGD14,Pink15].

BEISPIEL 3: FEHLERBAUMBESCHREIBUNG

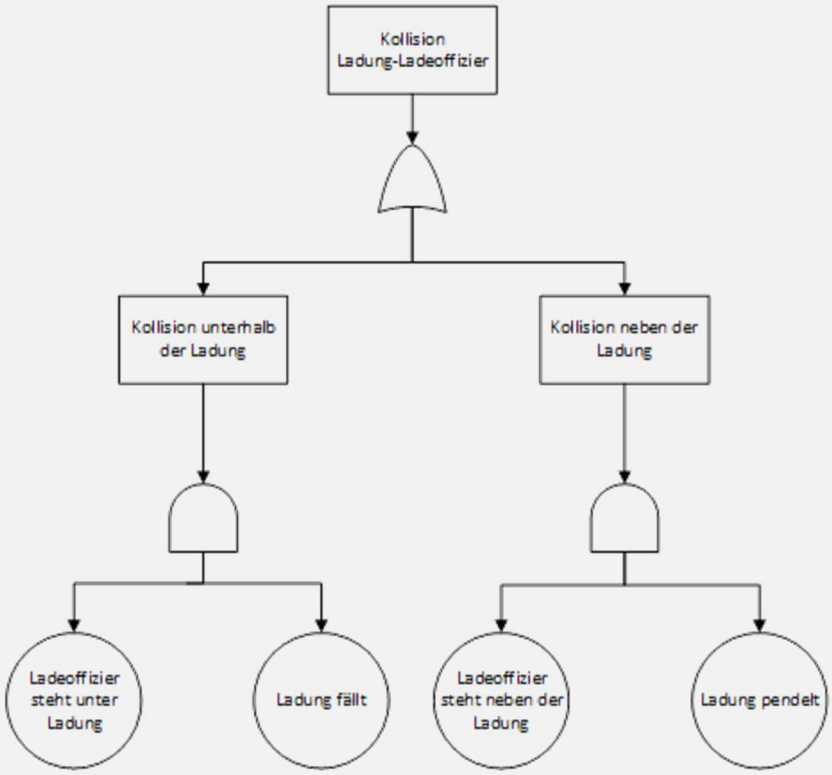


Abbildung 3-8 Übersicht über den generierten Fehlerbaum zum Beispielszenario.

Der Fehlerbaum, der in Abbildung 3-8 dargestellt ist, repräsentiert eine Auswahl von möglichen Ursachen und deren logische Verknüpfungen, die für eine Kollision der Ladung mit dem Ladeoffizier verantwortlich sind. Bei der vorgestellten Operation, wurde die potentielle Gefahr ”Kollision Ladung-Ladeoffizier” und entsprechende Ursachen, die die Gefahr hervorrufen können, annotiert. Diese Elemente werden in einem synthetisierten Fehlerbaum dargestellt. Die Gefahr kann demnach zum einen auftreten, wenn sich der Ladeoffizier unter der Ladung befindet aber auch wenn er sich neben der Ladung aufhält. Die Kollision unterhalb der Ladung muss dabei mehrere Ursachen haben, wie ein unkontrolliertes Herablassen oder Fallenlassen der Ladung durch den Kranführer und die Tatsache, dass der Ladeoffizier sich unterhalb der Ladung aufhält. Damit eine Kollision neben der Ladung auftreten kann, hat der Sicherheitsexperte die Gründe Ladeoffizier steht neben der Ladung und die Ladung pendelt am Prozessmodell annotiert.

Dies ist ein vereinfachter Fehlerbaum, der der Erklärung des Beispiels dient. Normalerweise wird bei einer drohenden Kollision eine Notbremsung vom Kranführer eingeleitet. Dass diese nicht ausgelöst wird, während die Ladung fällt, kann mehrere Ursachen haben, wie beispielsweise technische Mängel, die in weitere Teil-Ursachen aufgeteilt werden könnten.

3.2 Risikodistanzbeschreibung

Zur Bestimmung der Nähe zweier Dokumente oder wie der in dieser Arbeit betrachteten Systemzustände werden Distanzfunktionen verwendet. Bei Distanzfunktionen auch Metrik genannt handelt es sich um eine Abbildung die zwei Punkte eines unterliegenden Raumes auf deren (skalar wertige) Distanz zwischen einander abbildet.

Unter anderem finden Distanzfunktionen Einsatz im Bereich der Klassifikation. Hierbei ist die Annahme, dass Muster derselben Klasse ähnlich und Muster unterschiedlicher Klassen unähnlich sind. In diesem Fall ist die Ähnlichkeit der Muster durch deren Abstand im Merkmalsraum definiert, wobei eine große Ähnlichkeit durch einen geringen Abstand angezeigt wird. Die einzige Vorrausetzung für die Anwendbarkeit ist, dass eine Distanzfunktion $d(x, y)$ für zwei beliebige Muster x und y vorhanden sein muss [PaLZ04, ScBG99, WuSA02].

Je kleiner die Distanz zweier Objekte desto größer ist ihre Ähnlichkeit. Dies kann in der Regel genau quantifiziert werden. Ein Ähnlichkeitsmaß s mit dem Wertebereich $0 \leq s \leq 1$ ist zum Beispiel durch $d = 1 - s$ in das Distanzmaß d umwandelbar. Für die Güte eines Klassifikationsergebnisses ist die Wahl eines geeigneten Distanzmaßes von hoher Bedeutung [RoTa60].

In diesem Abschnitt wird die Methodik vorgestellt, die zur Ermittlung der Distanzbeschreibung zu einem Risiko angewendet wird (vgl. Anforderung [A_A1]). In Abbildung 3-9 wird gezeigt wie der Aufbau der Risikodistanzfunktion mit dem eigenen Ansatz erfolgt der in den folgenden Abschnitten genauer beschrieben wird.

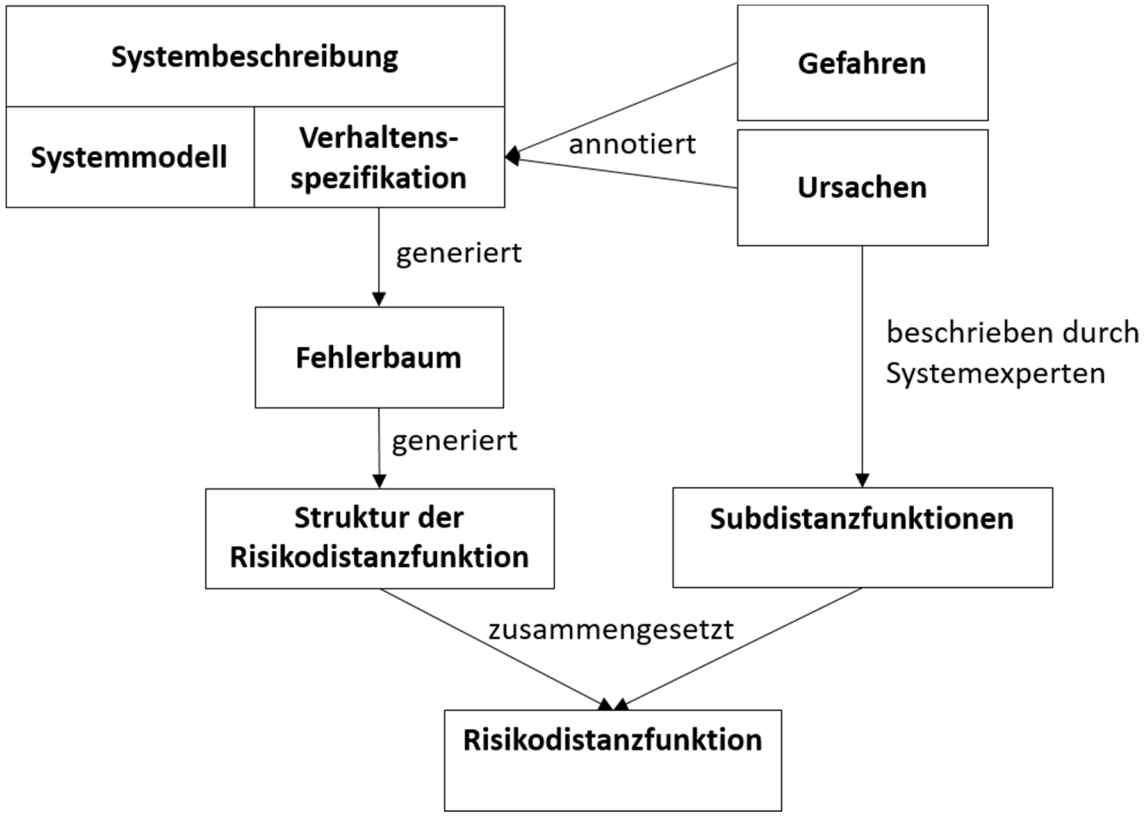


Abbildung 3-9 Übersicht über die Erstellung der Risikodistanzfunktion. Für die Erstellung findet eine Verwendung der System- und Verhaltensbeschreibung inklusive der annotierten Gefahren und Ursachen statt.

Diese setzt dabei auf der in den vorherigen Abschnitten beschriebenen Basis zur Beschreibung des Systems inklusive der vom Systemexperten erwarteten Gefahren und Ursachen auf. Das Ergebnis der Distanzbeschreibung ist eine Risikodistanzfunktion, welche eine mathematische Bewertungsfunktion darstellt, die die Nähe zu einer abstrakten zu untersuchenden risikoreichen Situation bewertet. Je geringer die Entfernung zum Risiko ist, desto niedriger das Ergebnis der Risikodistanzfunktion. Das Ergebnis einer Risikodistanzfunktion ist immer ein Wert zwischen 0 und 1, wobei 0 anzeigt, dass die Risikosituation eingetreten ist während 1 bedeutet, dass die aktuell überprüfte Situation keine relevante Nähe zu der untersuchten Risikosituation hat.

Die Eingabewerte, die für den Aufbau der Subdistanzfunktion notwendig sind, liegen dabei die in der DIN EN 1990:2010-12 vorgestellten Werten des Sicherheitskonzepts (vgl. [Klug07]) zu Grunde. Der Aufbau und das Konzept der Subdistanzfunktionen ist dabei inspiriert durch die Methode der Teilsicherheitsbeiwerte in einzelnen Grenzzuständen des semiprobabilistischen Sicherheitskonzepts.

Die Veröffentlichung der DIN 1055-100 (vgl. [Din03]) stellte ein Sicherheitskonzept für die Bewertung der Tragfähigkeit, Gebrauchstauglichkeit und Dauerhaftigkeit für Ingenieurbauwerke vor. Bei diesem Konzept werden Sicherheitsfaktoren angewendet um das Risiko des Versagens einer Tragstruktur mit den für die Berechnung verknüpften Annahmen des Modells so gering wie möglich zu halten.

Das vorgestellte Konzept, welches hinsichtlich der Bewertung der Tragfähigkeit teilweise auch in die Bewertung von Risiken in dieser Dissertation eingeflossen ist, basiert auf dem System der Teilsicherheitsbeiwerte, welche auf der Einwirkungs- (Last) und auf der Widerstandsseite (Tragfähigkeit) Anwendung findet. Teilsicherheitsbeiwerte werden zur Bestimmung des Bemessungswertes von Einwirkungen, Beanspruchungen oder Tragwiderständen aus den repräsentativen bzw. charakteristischen Werten genutzt.

Von charakteristischen Werten bei Einwirkungen wird von den wichtigsten repräsentativen Werten einer Einwirkung gesprochen. Von diesen wird vermutet, dass sie mit einer vorgegebenen Wahrscheinlichkeit unter Bezugnahme der Dauer der Nutzung des Tragwerks und der entsprechenden Bemessungssituation nicht über- oder unterschritten werden. Die entsprechenden Bemessungssituationen geben dabei die Bedingungen des Tragwerks vor inklusive der Einwirkungen für die der Planer die Einhaltung der Grenzzustände nachweist. Hierbei findet eine Unterscheidung zwischen ständigen, vorübergehenden, und außergewöhnlichen Bemessungssituationen statt.

Zu den wichtigsten Einflussfaktoren gehören dabei Einwirkungen wie Schnee, Wind, Temperaturen aber auch Baustoffeigenschaften wie Festigkeit und Steifigkeit und geometrische Größen wie Abmessungen und Geometrien. Dabei werden all diese Einflussfaktoren als Zufallsgrößen betrachtet die einer statistischen Streuung unterliegen.

Die Bewertung von Tragwerken und deren Bestandteilen wird dabei anhand von Bemessungskriterien durchgeführt, um zum einen den Grenzzustand der Tragfähigkeit und zum anderen den Grenzzustand der Gebrauchstauglichkeit zu ermitteln. Bemessungskriterien beschreiben dabei die für das Einhalten der Grenzzustände zu erfüllenden Bedingungen. Grenzzustände im Allgemeinen beschreiben den Zustand des Tragwerks, die bei einer Überschreitung anzeigen, dass die der Tragwerksplanung aufgestellten Anforderungen nicht mehr als erfüllt gelten können. Der spezielle Fall des Grenzzustands der Tragfähigkeit gibt dabei an, dass bei einer Überschreitung geradewegs eine Form des Versagens des Tragwerks, wie ein rechnerischer Einsturz, zu erwarten ist. Dieser Grenzzustand ergibt sich dabei aus dem größten rechnerischen Tragwiderstand. Innerhalb der Bemessungsnormen, welche bauartspezifisch sind wird der Bemessungswert des Tragwiderstandes festgelegt. [Klug07]

Die Einwirkungen auf das Tragwerk, wie einwirkende Kraft- oder Verformungsgrößen (z.B. Lasten im Falle der Tragwerke), werden in ständige und veränderliche Einwirkungen unterteilt. Für die Einteilung wird die zeitliche Veränderlichkeit der Einwirkungen betrachtet (vgl. [Din03]).

Eine Sicherstellung der vereinbarten Nutzungsbedingungen und Gebrauchseigenschaften (Bildung von Rissen und Bauteilverformungen) wird über die Nachweise in den Grenzzuständen der Gebrauchstauglichkeit gezeigt (vgl. [Klug07]).

Die Dauerhaftigkeit des Tragwerkes ist dabei gegeben, wenn die Einhaltung bestimmter konstruktiver Regeln gewährleistet ist. So muss unter anderem gewährleistet sein das eine regelgerechte Bemessung in den Grenzzuständen der Tragfähigkeit und Gebrauchstauglichkeit durchgeführt wurde und keine Probleme aufgefallen sind. Durch den Vergleich von Tragwiderstand und Einwirkung, bei dem immer nachgewiesen werden muss das der Bemessungswert des Widerstandes gleich oder größer ist als der Bemessungswert der Einwirkungen, wird der rechnerische Nachweis in den definierten Grenzzuständen durchgeführt. [Drpe06]

In Abbildung 3-10 ist der Zusammenhang für die Einwirkung E und die Tragfähigkeit/dem Widerstand R eines Bauteils grafisch dargestellt. Beide Zufallskenngrößen zeigen dabei einen streuenden Charakter. Ein Versagen der Tragfähigkeit lässt sich in der Darstellung durch den Zusammenhang $\mathrm{R} - \mathrm{E} < 0$ definieren. Für den Fall $R - E = 0$, wird gerade der Grenzzustand erreicht. Da für die beiden Verteilungsfunktionen, speziell an den Verteilungsenden, normalerweise ungenügende empirische Kenntnisse vorhanden sind, wird dafür Sorge getragen, dass zwischen den definierten Werten ein ausreichender Sicherheitsabstand gewährleistet ist.

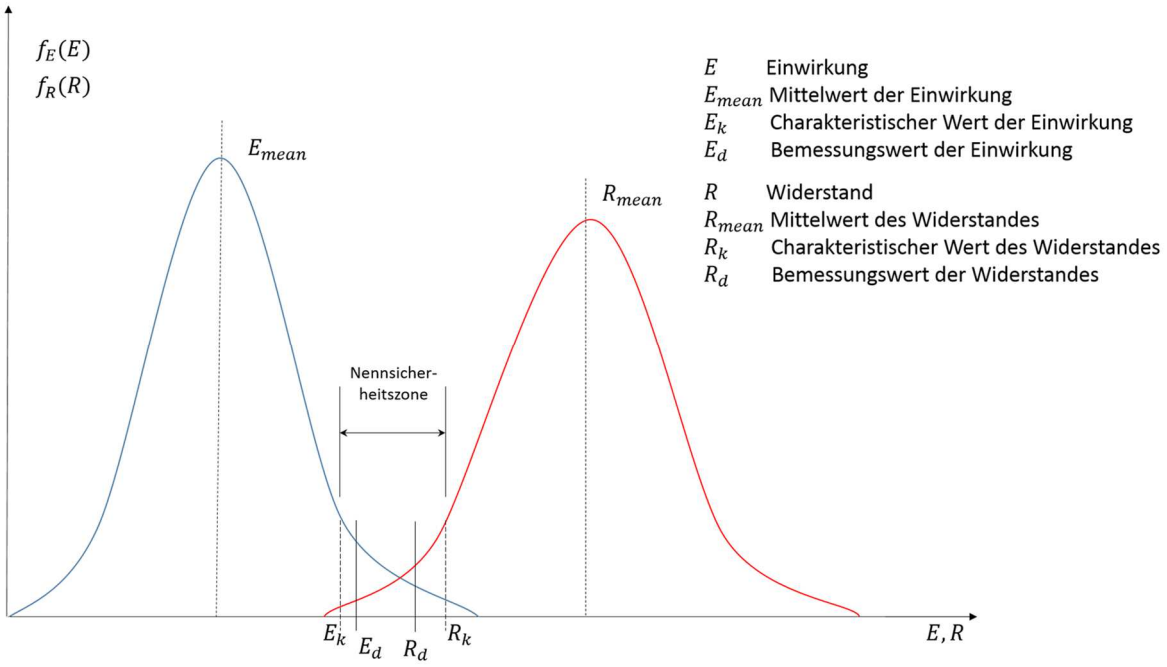


Abbildung 3-10 Zusammenhang zwischen einer Einwirkung E und einem Widerstand R eines Bauteils.

Die Analogie zwischen dem Konzept der Teilsicherheitsbeiwerte aus der Baustatik Domäne und dem in dieser Arbeit vorgestellten Ansatz zur Risikobewertung soll anhand der Abbildung 3-11 erläutert werden. So wie ein Tragwerk, wie zum Beispiel eine Brücke aus verschiedenen Bauteilen besteht (Stützpfeiler, Streben, Drahtseilen, Aufhängung, …), wird auch ein Risiko in dieser Arbeit als Zusammensetzung verschiedener Teilrisiken betrachtet. Beim Tragwerk sind die verschiedenen Bauteile unterschiedlichen Einwirkungen ausgesetzt zum Beispiel Windkräften, Schnee-, Personen oder Fahrzeuglasten bei dem von jedem Bauteil ein bekannter Grenzwiderstand für die jeweilige Einwirkung angenommen wird. Dieses findet sich bei der Risikobewertung wieder bei der jedes Teilrisiko aus bestimmten Einwirkungen und deren angegebenen Grenzwiderständen besteht.

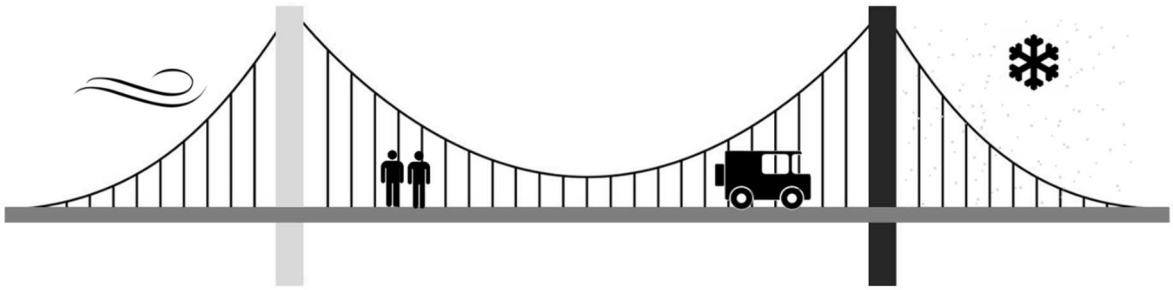


Abbildung 3-11 Eine Hängebrücke als Symbolbild für ein Tragwerk das aus verschiedenen Bauteilen, die unterschiedlichen Einwirkungen ausgesetzt sind, besteht.

In der DIN EN 1990:2010-12 (vgl. [Din12, Dinn12]) wird die Bewertung der Sicherheit eines Bauwerks anhand der Beanspruchung der jeweiligen Bauteile bewertet und wird auf konzeptioneller Ebene auf die Problemstellung dieser Arbeit angewendet. Im Falle des vorgestellten Ansatzes entspricht dies der Sicherheit bzgl. eines Risikos, wie der im Beispiel aufgeführten Kollision zwischen Ladeoffizier und Ladung. Die jeweiligen Bauteile und deren Beanspruchung werden im eigenen Ansatz durch die Subdistanzfunktionen beschrieben. Diese berechnen die Einwirkung der Risikoursachen auf das Gesamtrisiko. Jedes Bauteil besitzt Grenzwiderstände hinsichtlich unterschiedlicher Einwirkungen, wie z.B. der tragenden Last, über die ermittelt werden kann, ab wann ein Bauteil zerbricht und somit die Einwirkung den Widerstand des Bauteils überschreitet. Ein ähnlicher Ansatz wurde ebenfalls innerhalb der Subdistanzfunktionen umgesetzt, in dem angegeben werden muss ab welchem Wert genauer ab welcher Einwirkung ein Teilrisiko überschritten ist und somit als eingetreten gelten muss, was dem zu beschreibenden Eintrittsraum gleichzusetzen ist.

3.2.1 Anforderungen an Risiko- und Subdistanzfunktionen

In diesem Abschnitt werden die Anforderungen an die Risiko- und Subdistanzfunktionen vorgestellt. Die Erfüllung dieser ist notwendig, um eine Gegenüberstellung der Risikodistanzen zu ermöglichen und die stetige Verfügbarkeit der genutzten Eigenschaften und deren Ausprägung des Systemmodells zu gewährleisten.

[A_D1] **Vergleichbare Risikodistanzwerte** Während einer Simulation können verschiedene Risiken hinsichtlich ihrer Distanz parallel untersucht werden. Um eine Rangfolge der untersuchten Risiken erhalten zu können bzw. diese in Relation zu setzen bzgl. der nächsten Distanzen, müssen die

Risikodistanzwerte miteinander vergleichbar sein. Erst durch die Erfüllung dieser Anforderung kann die Frage, ob eine Simulationssituation einem Risiko A näher ist als einem Risiko B beantwortet werden.

[A_D2] **Stetigkeit der Risikodistanzfunktion** Die Stetigkeit wird gefordert, um zum einen Eigenschaften wie die Vermeidung von Sprungstellen im Risikodistanzverlauf zu erreichen. Hierdurch soll verhindert werden das kleine Änderungen des Systemzustands zu großen Unterschieden in der Risikobewertung führen.

[A_D3] **Verwendung kontinuierlicher Eingangswerte** Die Betrachtung kontinuierlicher Werte ist wichtig, um die vorherige Anforderung der Stetigkeit zu erfüllen. Zusätzlich wird durch diese Anforderung vermieden, dass Eigenschaften in den Subdistanzfunktionen genutzt werden, über die keine Annäherung beschrieben werden kann. Ein Beispiel hierfür wäre ein boolescher Wert, der anzeigt, ob ein Tank gefüllt ist oder nicht, anstatt einer Eigenschaft, die den Füllstand angibt.

[A_D4] **Die Distanz einer risikoreichen Situation evaluiert zu 0** Als letzte Anforderung sei noch erwähnt, dass die Risikodistanzfunktion zu einem Wert von 0 evaluiert, sobald ein Zustand aus der zuvor definierten risikoreichen Zustandsmenge erreicht wird. Mit der Erfüllung dieser Anforderung soll erreicht werden, dass die Risikodistanzfunktion auch immer die Frage nach dem Eintritt und Nichteintritt einer Gefahrensituation beantworten kann.

Neben den zu erfüllenden Anforderungen, die dazu führen müssen, dass die entwickelte Methodik angewendet werden kann existiert noch eine zusätzliche schwache Anforderung, die eine Weiterverarbeitung der Distanzergebnisse verlangt. Damit andere Komponenten und Simulatoren innerhalb einer Co-Simulation weitere Entscheidungen und Berechnungen auf Grund der ermittelten Risikodistanzen tätigen können, soll eine Weiterverarbeitung der Distanzergebnisse innerhalb der Co-Simulation gewährleistet sein. Daher sollen unabhängig von einer bekannten Nutzung die ermittelten Distanzen über das gemeinsame Datenmodell und die verwendete Kommunikationsinfrastruktur gepflegt werden können.

Die Angabe einer Risikodistanzfunktion aufbauend auf einem Fehlerbaum sollte dabei immer unter der Prämisse betrachtet werden, dass zum Zeitpunkt der Gefahrenannotation und der Fehlerbaumerstellung nicht alle das Risiko begünstigenden Faktoren bekannt sind. Daher muss durch den Sicherheitsexperten ein abwägen beim Detailgrad der

Risikodistanzfunktion erfolgen, um eine übermäßige Anpassung der Risikodistanzfunktion an eine sehr spezielle risikoreiche Situation zu vermeiden. Je spezifischer die Risikodistanzfunktion desto genauer die Bewertung bzgl. der angenommenen Indikatoren aber desto geringer die Möglichkeit nicht betrachtete Risiken zu entdecken.

3.2.2 Ableitung der Struktur der Risikodistanzfunktion

Um die Struktur einer Risikodistanzfunktion zu erhalten, werden die im zugehörigen Fehlerbaum angegebenen logischen Verknüpfungen („Und“/„Oder“-Beziehungen) der Basic und Intermediate Events (Unterelemente des Top-Events, die keine Blattelemente sind) des Fehlerbaums verwendet. Dabei wird in dieser Arbeit von einem direkt verwendbaren Fehlerbaum ausgegangen, der nicht auf Korrektheit überprüft werden muss. In Abbildung 3-12 ist die Verknüpfung zwischen Fehlerbaum und Struktur der Risikodistanzfunktion an einem Beispiel zu sehen. Das Top-Level Element ist der Namensgeber für die zu erstellende Risikodistanzfunktion (D_A). Durch „Und“ verknüpfte Elemente werden mittels Multiplikation berechnet. Da nicht mit Wahrscheinlichkeiten, sondern mit Distanzen gerechnet wird werden die eingehenden, vorher berechneten Distanzen invertiert, dann multipliziert und wieder invertiert um eine Distanz als Ergebnis zu erhalten. Die Formel, die sich daher für durch „Und“ verknüpfte Elemente ergibt ist $D_A = 1 - ((1 - D_B) * (1 - D_C))$. Mehrere mit „Oder“ verknüpfte Elemente werden durch die Berechnung der minimalen Distanz bestimmt ($D_B = \min(D_D, D_E)$). Die Berechnung der Blattelemente erfolgt, durch die Berechnung der Produkte der Distanzen zu den verschiedenen zu definierenden Eigenschaften der Ursachen, die in dieser Arbeit als Subdistanzfunktionen beschrieben werden ($D_C = 1 - \prod_{i \in 1..n} \left(1 - d_{c_i}\right)$). Dabei wurde bei der Auswahl von Multiplikation und Minimum auch die Anforderung [A_D2] der Stetigkeit beachtet, da die gewählten Operatoren diese nicht aufheben.

.

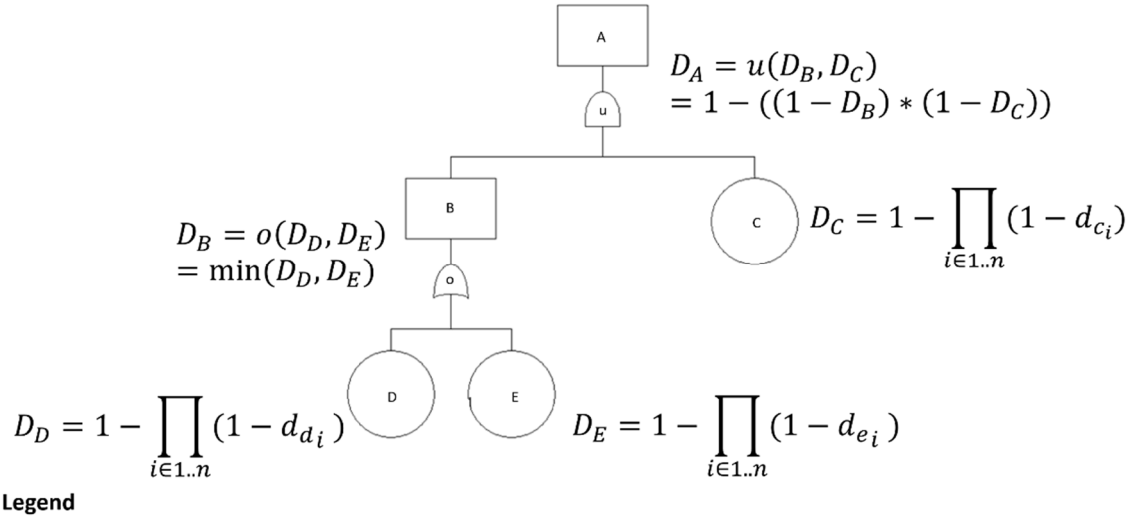
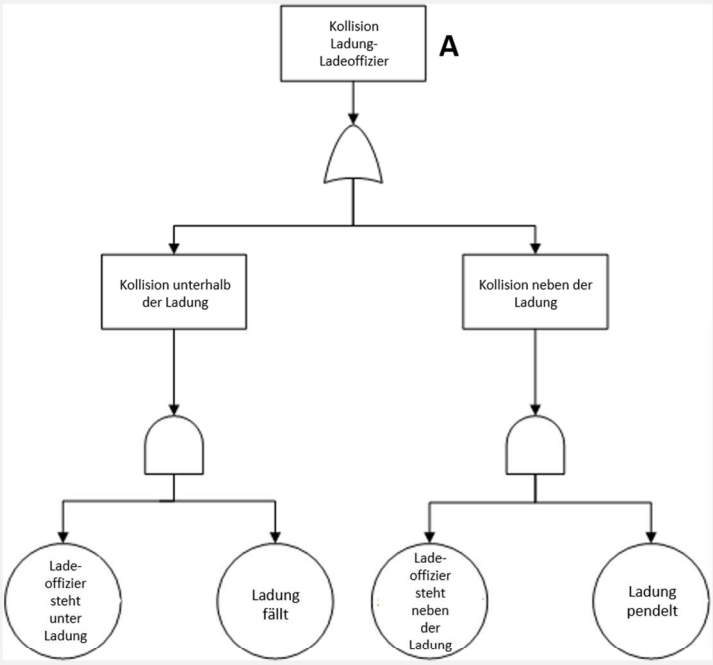


Abbildung 3-12 Beziehung zwischen Fehlerbaum und Struktur der Risikodistanzfunktion. D_X ist eine Distanzfunktion, die das Fehlerbaum Element X beschreibt, während d_Y die im folgenden Schritt zu ergänzenden Subdistanzfunktionen darstellt, die das Blattelement Y des Fehlerbaums beschreiben.

Die sich ergebende Struktur der Risikodistanzfunktion zum Fehlerbaum aus Abbildung 3-12 ist in Formel (1) angegeben. Damit eine Berechnung erfolgen kann, werden in einem folgenden Schritt die Subdistanzfunktionen hergeleitet, um die Risikodistanzfunktion zu vervollständigen.

$$D^A = 1 - \left(\left(1 - \min \left(1 - \prod_{i \in 1..n} 1 - d_{d_i}, 1 - \prod_{i \in 1..n} 1 - d_{e_i} \right) \right) * \left(1 - \left(1 - \prod_{i \in 1..n} 1 - d_{c_i} \right) \right) \right) \quad \textbf{(1)}$$

BEISPIEL 4: STRUKTUR DER RISIKODISTANZFUNKTION



$$D_A = \min\left(\begin{array}{l}\left(\left(1-\left(1-\prod_{i\in 1..k} 1-d_{B_i}\right)\right)*\left(1-\left(1-\prod_{i\in 1..k} 1-d_{C_i}\right)\right)\right), \\ \left(\left(1-\left(1-\prod_{i\in 1..k} 1-d_{D_i}\right)\right)*\left(1-\left(1-\prod_{i\in 1..k} 1-d_{E_i}\right)\right)\right)\end{array}\right)$$

Abbildung 3-13 Struktur der Risikodistanzfunktion für den generierten Fehlerbaum zum Beispielszenario.

Für den Fehlerbaum, der für das Beispielszenario abgeleitet wurde (s. Abschnitt 3.1.1), ergibt sich die in Abbildung 3-13 zu sehende Struktur für die Risikodistanzfunktion. Dabei wurden die logischen Verknüpfungen in die vorgegebenen mathematischen Funktionen umgewandelt. Dies bedeutet, dass die Produkte der invertierten Ergebnisse der Subdistanzfunktionen für Blattelement B und C („Und"-Operator „Kollision unterhalb der Ladung") sowie D und E („Und"-Operator „Kollision neben der Ladung") berechnet werden. Der „Oder"-Operator für das Gesamtrisiko berechnet den minimalen Risikodistanzwert aus den Ergebnissen der beiden Produkte.

3.2.3 Herleitung der Subdistanzfunktionen

In Abbildung 3-14 ist die Verknüpfung zwischen einem Fehlerbaum Basic-Event und dem Weg zu den benötigten Subdistanzfunktionen zu sehen.

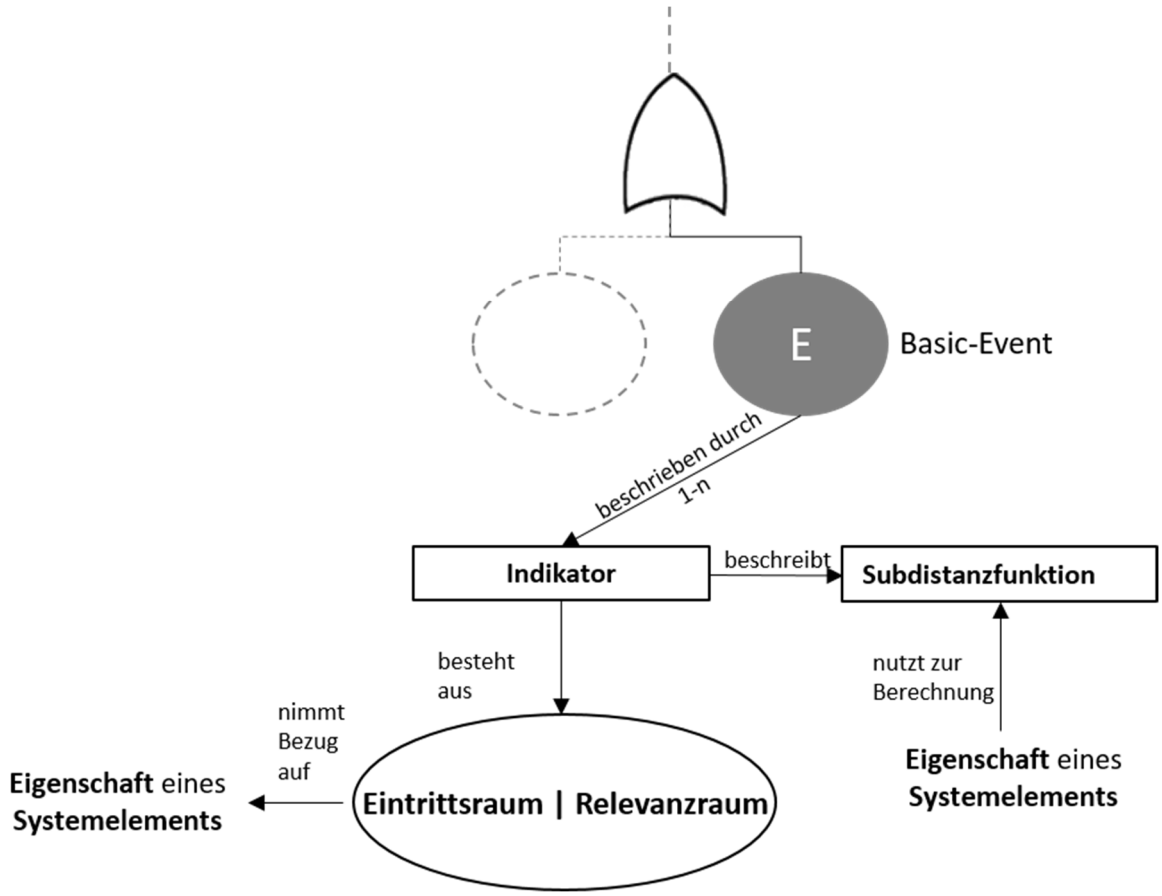


Abbildung 3-14 Beschreibung der Basic-Events für die Risikodistanzfunktion.

Im ersten Schritt werden vom Systemexperten so genannte Indikatoren ermittelt, die den Raum beschreiben, in dem die im Basic-Event beschriebene Ursache als eingetreten gilt. Für das Basic-Event E des Fehlerbaums aus Abbildung 3-13 wurde $1\text{-}\prod_{i\in 1..n}(1-d_{E_i})$ als Formel zur Berechnung der Ursachendistanz ermittelt. Das Produkt wird berechnet, da die verschiedenen Indikatoren als gemeinsam eintretende Ereignisse angegeben werden, die aufgetreten sein müssen, damit die im Basic-Event angegebene Ursache als eingetreten gelten darf. Die Kombination der Indikatoren durch mathematische Operatoren erfolgt also auf demselben Wege wie bei den durch das logische „Und“ verknüpften Elementen im Fehlerbaum.

Dabei stellt d_{E_i} die Subdistanzfunktion für den i-ten Indikator des Blattknotens E dar. Jeder Indikator bezieht sich auf genau eine Eigenschaft eines Elements des Systems mit kontinuierlichen Werten (vgl. Anforderung [A_D3]) und wird definiert durch einen Eintrittsraum (E_E) und Relevanzraum (R_E) (s. Abbildung 3-15). Ausgehend vom betrachteten Indikator kann der jeweilige Eintrittsraum und Relevanzraum durch eine untere $E_{E_{min}}/R_{E_{min}}$ und/oder obere Grenze $E_{E_{max}}$ respektive $R_{E_{max}}$ beschrieben werden. Der Eintrittsraum beschreibt den Wertebereich der Eigenschaft, der die im Basic-Event beschriebene Ursache begünstigt. Der Relevanzraum beschreibt den Wertebereich der Eigenschaft, der für eine Annäherung an das untersuchte Risiko relevant ist und ab welchem Wert der Eigenschaft keine Relevanz mehr für das untersuchte Risiko besteht.

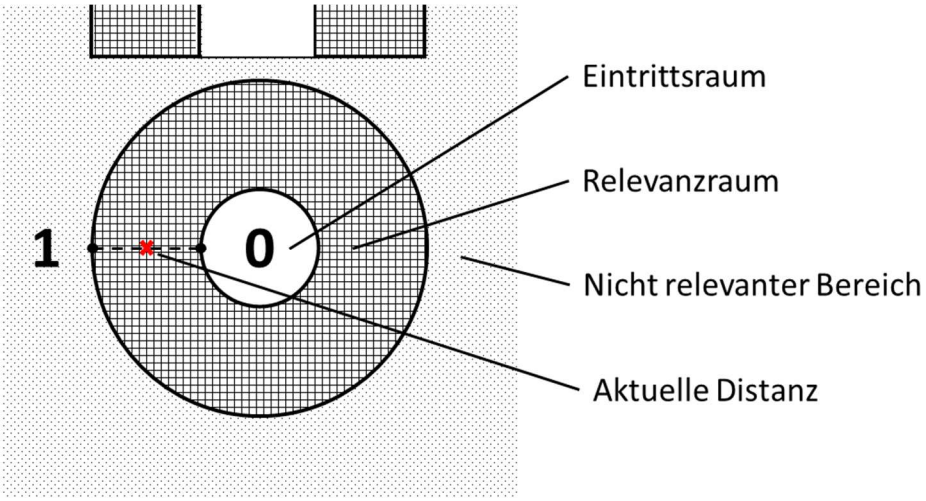


Abbildung 3-15 Übersicht über die zu beschreibenden Räume eines Indikators und der sich daraus ergebenden Werte und Bereiche.

Um dem Systemexperten bei der Ermittlung der beeinflussenden Eigenschaften des Systems zu unterstützen, wird mit Hilfe der verknüpften Modelle (s. Abbildung 3-16) eine Vorschlagsliste mit Eigenschaften von Elementen des Systemmodells erstellt. Die rot gepunkteten Linien zeigen dabei die verknüpften Elemente zwischen Fehlerbaum und Verhaltensspezifikation während die schwarz gestrichelten Linien die Nutzung von Ressourcen der Systembeschreibung in der Verhaltensspezifikation anzeigen.

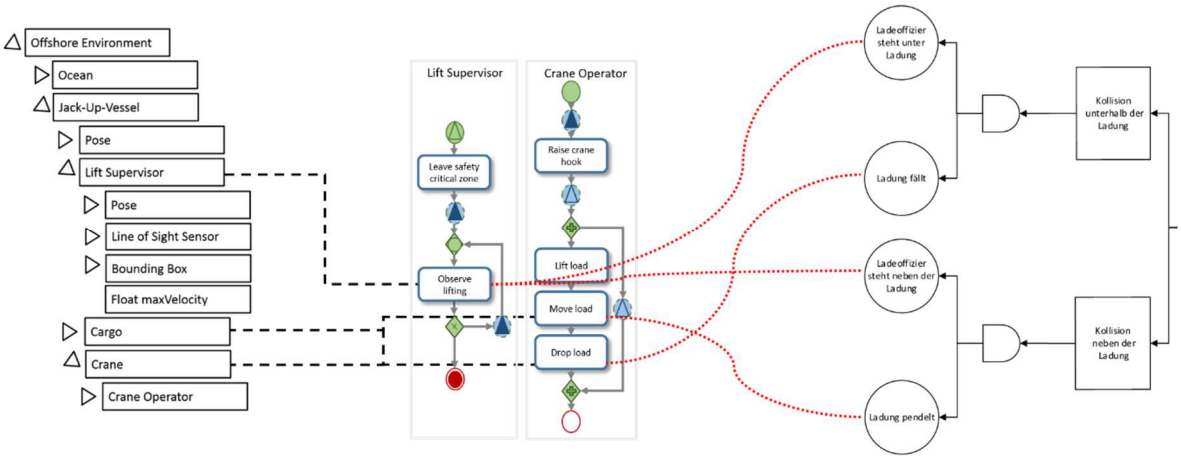


Abbildung 3-16 Darstellung der verknüpften Elemente der System-, Verhaltens- und Fehlerbaumbeschreibung.

Die vorgeschlagenen Eigenschaften des Systems basieren dabei auf den mit den Basic-Events des Fehlerbaums verknüpften Aufgaben-Elementen der Verhaltensspezifikation. Da innerhalb der Aufgaben-Elemente genutzte Ressourcen des Systemmodells angegeben sind, wird der Annahme nachgegangen, dass diese unter anderem mit der auftretenden Ursache eines Risikos zusammenhängen können. Da die Angaben jedoch weder vollständig noch fehlerfrei sein müssen, handelt es sich bei den ausgegebenen Eigenschaften des Systems nur um eine vorsortierte gewichtete Liste, die dem Sicherheitsexperten eine Unterstützung bei der Beschreibung der Risikodistanzfunktion geben soll. Hierbei ergibt sich die Gewichtung der einzelnen Elemente der Liste über die Entfernung zum mit einer Ursache annotiertem Aufgaben-Element. Eine genutzte Ressource aus dem annotierten Aufgaben-Element ist also höher gewichtet als eine Ressource aus einem Vorgänger Aufgaben-Element.

Neben den direkt nutzbaren Eigenschaften des Systemmodells lassen sich auch weiterverarbeitete Daten nutzen. Hierbei findet eine Unterteilung in einfach und komplex weiterverarbeitete Daten statt. Die Grenze für die einfach und komplex weiterverarbeiteten Daten wird dabei zwischen einfachen mathematischen Operationen (z.B. Berechnung einer Differenz) sowie Typ-Umwandelungen (z.B. Umrechnung einer GPS-Koordinate zu einer 3D-Koordinate) und komplexen Algorithmen deren Ergebnisse Werte mit einer neuen Semantik generieren gezogen. Ein Beispiel für eine einfache Eigenschaft ist das *X* Attribut der Position eines Systemelementes. Ein weiterverarbeitetes Element ist die Differenz zwischen zwei *X* Attributen der Position zweier unterschiedlicher Systemelemente, also deren Entfernung zueinander auf der *X*-Achse. Zusätzlich lässt sich durch die einfache Weiterverarbeitung eine Konvertierung von GPS-Koordinaten zu Koordinaten im euklidischen Raum vornehmen, um dann wiederrum die Distanz zu bestimmen.

Für die Weiterverarbeitung kann der Systemexperte auch komplexere Algorithmen bzw. Komponenten entwickeln. Ein Beispiel für so eine komplexe Komponente ist die Überschneidung zweier Bewegungsräume, der auf die Kollisionsgefahr hin zu untersuchenden Objekte (s. Beispiel 5). Die entwickelte Komponente kann in dem im nächsten Kapitel vorgestellten Simulations- und Bewertungsframework als Indikator in die Risikodistanzberechnung mit eingebunden werden. Die Anforderung bleibt auch an diese erstellten Komponenten, dass die berechnete Teildistanz einen Wert, der zwischen 0 (Risiko eingetreten) und 1 (Risiko nicht relevant) normalisiert ist, zurückliefern muss.

BEISPIEL 5: KOMPLEXE BERECHNUNG VON INDIKATOREN

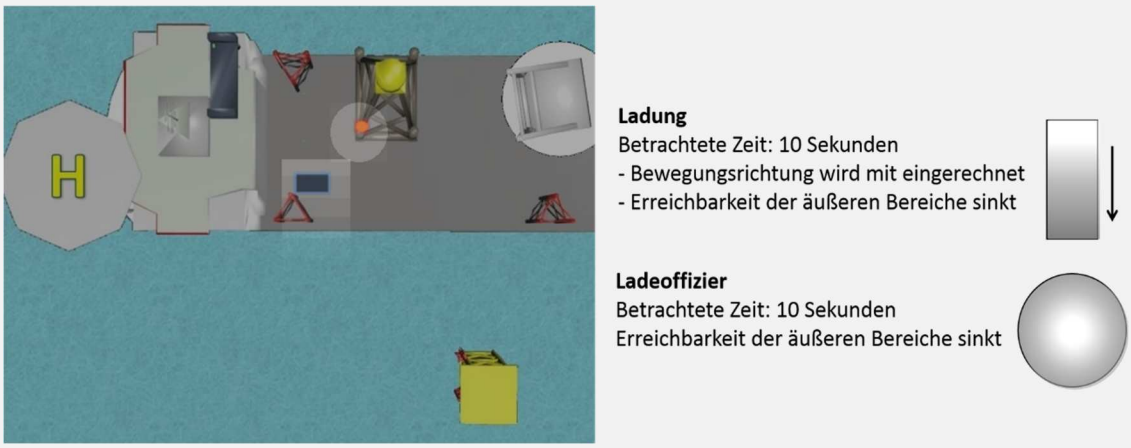


Abbildung 3-17 Beispiel für einen komplexen Indikator, der Bewegungsräume zur Berechnung einer Teilrisikodistanz nutzt.

Ein Beispiel für eine komplexe Berechnung von Indikatoren kann die Distanz zu einer bevorstehenden Kollision durch die Berechnung der Überschneidung von definierten Skalarfeldern sein (s. Abbildung 3-17). Diese werden beschrieben, in dem ein Bewegungsprofil für eine bestimmte maximale Zeit angegeben wird. In diesem Beispiel wird dabei betrachtet, wo der Ladeoffizier und die Ladung innerhalb der nächsten 10 Sekunden sein können. Das Bewegungsprofil des Ladeoffiziers wird dabei als Kreis beschrieben, da dieser menschspezifisch sehr schnell eine Richtungsänderung durchführen kann. Im Gegensatz dazu erhält die Ladung einen rechteckigen Raum, der über die Geschwindigkeit in die momentane Bewegungsrichtung ausgedehnt wird. Bei keiner Überschneidung der Objekte wird 1 als Distanz ausgegeben, da dies bedeutet, dass der Relevanzraum nicht erreicht wurde. Sobald eine Überschneidung der Bewegungsräume vorhanden ist, wird die Teildistanz über die prozentuale Überschneidung der Bewegungsräume unter Berücksichtigung der Zeit in der diese erreichbar sind, bestimmt. Normalisiert wird dabei über den minimalen

Überschneidungswert und die maximale Überschneidung zum ehesten Zeitpunkt (s. Abbildung 3-17 - Zentrum des Kreises beim Ladeoffizier und oberer Teil des Rechtecks bei der Ladung). Das Ergebnis dieses komplexen Indikators kann direkt innerhalb der Risikodistanzfunktion angewendet werden.

Die Angabe der Eintritts- und Relevanzräume erfolgt dabei über die folgenden Formeln (2),(3),(4) und (5), wobei A, B, C und D vom Systemexperten festzulegende Konstanten sind, die sich auf die Eigenschaften beziehen, mit denen ein Indikator beschrieben wird, während Y einen vorher definierten Wert der betrachteten Eigenschaft des Systems darstellen kann (zum Beispiel den Mittelpunkt des Hüllkörpers[11] der Ladung oder eine vorgegebene Geschwindigkeit).

$$E_{min}(Y) = Y - A \quad \textbf{(2)}$$

$$E_{max}(Y) = Y + B \quad \textbf{(3)}$$

$$R_{min}(Y) = Y - C \quad \textbf{(4)}$$

$$R_{max}(Y) = Y + D \quad \textbf{(5)}$$

BEISPIEL 6: BESTIMMUNG DER INDIKATOREN INKLUSIVE EINTRITTS- UND RELEVANZRÄUMEN

Für das Beispiel der Verladeoperation wurde der Fehlerbaum aus Abbildung 3-8 ausgewertet. Die daraus abgeleitete Risikobeschreibung, bestehend aus den Ursachen und deren neu ermittelten Indikatoren, ist in der folgenden Beschreibung aufgelistet (s. Tabelle 3-1). Die Ursachen wurden den Blattelementen des Fehlerbaums entnommen.

Ursachen	**Indikatoren**
1) Ladeoffizier steht unter Ladung	a) Abstand zu Ladungsmittelpunkt X zu gering b) Abstand zu Ladungsmittelpunkt Y zu gering

[11] Ein Hüllkörper (engl. *bounding volume*) beschreibt einen simplen geometrischen Körper, der ein komplexes dreidimensionales Objekt oder einen komplexen Körper umschließt.

2) Ladung fällt	a) Z-Distanz des Ladeoffiziers zur Ladung zu gering b) Z-Geschwindigkeit der Ladung zu hoch
3) Ladeoffizier steht neben der Ladung	a) Abstand zum Ladungsrand X zu gering b) Abstand zum Ladungsrand Y zu gering
4) Ladung pendelt	a) Geschwindigkeit der Ladung in x, y Richtung zu hoch b) Zu hohe Windstärke

Tabelle 3-1 Ermittelte Indikatoren zu den abgeleiteten Ursachen des vorgestellten Fehlerbaums aus Abbildung 3-8.

Im nächsten Schritt sind die Eintritts- und Relevanzräume zu den Indikatoren vom Systemexperten definiert worden (s. Tabelle 3-2). So wird beispielsweise für den Indikator *1) Ladeoffizier steht unter Ladung* als ein Eintrittsraum das Zentrum des Hüllkörpers auf der X-Achse – 2 Meter (Minimum) und + 2 Meter (Maximum) angegeben während der entsprechende Relevanzraum dem Zentrum des Hüllkörpers auf der X-Achse – 10 Meter (Minimum) und + 10 Meter (Maximum) entspricht.

Indikatoren	E_D	R_D
1) Lade-offizier steht unter Ladung	$E_D 1a_{min} = \text{Cargo.BoundingBox.Center.X} - 2\text{m}$ $E_D 1a_{max} = \text{Cargo.BoundingBox.Center.X} + 2\text{m}$	$R_D 1a_{min} = \text{Cargo.BoundingBox.Center.X} - 10\text{m}$ $R_D 1a_{max} = \text{Cargo.BoundingBox.Center.X} + 10\text{m}$
	$E_D 1b_{min} = \text{Cargo.BoundingBox.Center.Y} - 3\text{m}$ $E_D 1b_{max} = \text{Cargo.BoundingBox.Center.Y} + 3\text{m}$	$R_D 1b_{min} = Cargo.\text{BoundingBox.Center.Y} - 10\text{m}$ $R_D 1b_{max} = \text{Cargo.BoundingBox.Center.Y} + 10\text{m}$

2) Ladung fällt	$E_D 2a_{min}$ $= \text{Cargo.BoundingBox.Pos.Z} - 1\text{m}$	$R_D 2a_{min}$ $= \text{Cargo.BoundingBox.Pos.Z} - 3\text{m}$	
	$E_D 2b_{min} = 10\text{km/h}$	$R_D 2a_{min} = 2\text{km/h}$	
3) Lade-offizier steht neben der Ladung	$E_D 3a_{min}$ $= \text{Cargo.BoundingBox.minX} - 2\text{m}$ $E_D 3a_{max}$ $= \text{Cargo.BoundingBox.maxX} + 2\text{m}$	$R_D 3a_{min}$ $= \text{Cargo.BoundingBox.minX} - 10\text{m}$ $R_D 3a_{max}$ $= \text{Cargo.BoundingBox.minX} + 10\text{m}$	
	$E_D 3b_{min}$ $= \text{Cargo.BoundingBox.minY} - 3\text{m}$ $E_D 3b_{max}$ $= \text{Cargo.BoundingBox.maxY} + 3\text{m}$	$R_D 3b_{min}$ $= \text{Cargo.BoundingBox.minY} - 10\text{m}$ $R_D 3b_{max}$ $= \text{Cargo.BoundingBox.maxY} + 10\text{m}$	
4)Ladung pendelt	$E_D 4a_{min} = 20\text{km/h}$	$R_D 4a_{min} = 5\text{km/h}$	
	$E_D 4b_{min} = 62\text{km/h}$	$R_D 4b_{min} = 29\text{km/h}$	

Tabelle 3-2 Zuordnung der beschriebenen Eintritts- und Relevanzräume zu den jeweiligen ermittelten Indikatoren.

Ist der Indikator durch Eintritts- und Relevanzraum beschrieben, erfolgt die Erstellung der Subdistanzfunktion für den jeweiligen Indikator (s. Formel (6)).

Diese setzt sich dabei aus den ermittelten Grenzen des Eintritts- und Relevanzraumes sowie der betrachteten Eigenschaft X des Systems zusammen.

$$d_{D_i} = \begin{cases} 0, wenn\ X \geq E_{D_{min}}(Y) \wedge X \leq E_{D_{max}}(Y) \\ 1, wenn\ X \leq R_{D_{min}}(Y) \vee X \geq R_{D_{max}}(Y) \\ \min\left(\dfrac{\left(X - E_{D_{min}}(Y)\right)}{(R_{D_{min}}(Y) - E_{D_{min}}(Y))}, \dfrac{\left(X + E_{D_{max}}(Y)\right)}{(R_{D_{max}}(Y) - E_{D_{max}}(Y))} \right), sonst \end{cases} \quad \textbf{(6)}$$

Dabei wird zunächst überprüft, ob die betrachtete Eigenschaft sich bereits im Eintrittsraum befindet. Ist dies der Fall, ist das Ergebnis der betrachteten Subdistanzfunktion 0. Die nächste Überprüfung bezieht sich auf den Relevanzraum. Befindet sich die Eigenschaft außerhalb des Relevanzraums, wird 1 als Ergebnis der Subdistanzfunktion ausgegeben. Trifft keines der beiden Annahmen zu, wird das Ergebnis der Subdistanzfunktion über die Distanz der Systemelementeigenschaft zum minimalen und maximalen Eintrittsraum bestimmt. Diese Distanzen werden dann über die Distanz zwischen minimalem respektive maximalem Ereignisraum und Relevanzraum normalisiert (vgl. Anforderung [A_D1]). Durch die normalisierten Werte der Subdistanzfunktionen, die unabhängig von den verwendeten Eigenschaften, alle eine Teildistanz beschreiben, wird eine Vergleichbarkeit dieser erreicht. Aus den normalisierten Distanzen wird dann das Minimum ausgewählt und als Ergebnis der Subdistanzfunktion ausgegeben. Existiert nur eine minimale oder maximale Grenze des Eintrittsraums, gilt die Distanz zu dieser als Ergebnis. Die Auswahl des Minimums muss dann also nicht erfolgen. Bei der Definition der Subdistanzfunktion wurde darauf geachtet, dass diese die Anforderung der Stetigkeit (vgl. Anforderung [A_D2]) erfüllt. Des Weiteren konnte durch die Definition sichergestellt werden, dass eine untersuchte Situation deren Eigenschaftenwerte im Eintrittsraum liegen und somit einer risikoreichen Situation entsprechen auch zu einer Distanz von 0 evaluieren (vgl. Anforderung [A_D4]).

BEISPIEL 7: BESTIMMUNG DER SUBDISTANZFUNKTIONEN ZU DEN BESCHRIEBENEN INDIKATOREN

In diesem Beispiel wird anhand eines Indikators die Erstellung der dazugehörigen Subdistanzfunktionen erklärt. Dafür wird der Indikator „Ladeoffizier steht unter Ladung“ betrachtet (s. Tabelle 3-3).

1) Lade-offizier steht unter Ladung	$E_D 1a_{min}$ $= \text{Cargo.BoundingBox.Center.X} - 2\text{m}$ $E_D 1a_{max}$ $= \text{Cargo.BoundingBox.Center.X} + 2\text{m}$	$R_D 1a_{min}$ $= \text{Cargo.BoundingBox.Center.X} - 10\text{m}$ $R_D 1a_{max}$ $= \text{Cargo.BoundingBox.Center.X} + 10\text{m}$
	$E_D 1b_{min}$ $= \text{Cargo.BoundingBox.Center.Y} - 3\text{m}$ $E_D 1b_{max}$ $= \text{Cargo.BoundingBox.Center.Y} + 3\text{m}$	$R_D 1b_{min}$ $= \text{Cargo.BoundingBox.Center.Y} - 10\text{m}$ $R_D 1b_{max}$ $= \text{Cargo.BoundingBox.Center.Y} + 10\text{m}$

Tabelle 3-3 Indikator „Ladeoffizier steht unter Ladung“.

Bei den Indikatoren wurde zunächst vom zu schützenden Objekt der Eintritts- und Relevanzraum betrachtet der über die Verwendung der Position des Ladeoffiziers (LSV) in die Subdistanzfunktionen einfließt.

Im Folgenden werden die beiden Subdistanzfunktionen für die beiden Indikatoren dargestellt. Die erste beschreibt dabei den Abstand zum Ladungsmittelpunkt *X* während die zweite den Abstand zum Ladungsmittelpunkt *Y* beschreibt.

$$d_{D_{1a}} = \begin{cases} 0, wenn\ LSV.Position.X \geq E_D 1a_{min} \wedge LSV.Position.X \leq E_D 1a_{max} \\ 1, wenn\ LSV.Position.X \leq R_D 1a_{min} \vee LSV.Position.X \geq R_D 1a_{max} \\ \min\left(\frac{(LSV.Position.X - E_D 1a_{min})}{(R_D 1a_{min} - E_D 1a^D_{min})}, \frac{(LSV.Position.X + E_D 1a_{max})}{(R_D 1a_{max} - E_D 1a_{max})}\right), sonst \end{cases}$$

$$d_{D_{1b}} = \begin{cases} 0, wenn\ LSV.Position.Y \geq E_D 1b_{min} \wedge LSV.Position.Y \leq E_D 1b_{max} \\ 1, wenn\ LSV.Position.Y \leq R_D 1b_{min} \vee LSV.Position.Y \geq R_D 1b_{max} \\ \min\left(\frac{(LSV.Position.Y - E_D 1b_{min})}{(R_D 1b_{min} - E_D 1b_{min})}, \frac{(LSV.Position.Y + E_D 1b_{max})}{(R_D 1b_{max} - E_D 1b_{max})}\right), sonst \end{cases}$$

Die ermittelten Subdistanzfunktionen bzw. deren Ergebnisse werden dann in die abgeleitete Struktur der Risikodistanzfunktion eingesetzt.

3.3 Simulative Analyse

In diesem Abschnitt wird beschrieben, wie die Simulative Analyse mittels des beschriebenen Systems und der ermittelten Risikodistanzfunktionen erfolgt. Dabei wird auf die genutzte Co-Simulationsumgebung eingegangen, die Anforderungen an die einzubindenden Simulatoren und die Verwendung der Distanzfunktionen zur Steuerung der Simulation hinsichtlich einer Reduzierung von Simulationsläufen durch die Anwendung einer Importance Splitting Technik, welche es ermöglicht mit Black-Box Simulatoren arbeiten zu können (vgl. Anforderung [A_A4]).

3.3.1 Ermittlung der benötigten Simulatoren

Bei der Ermittlung der benötigten Komponenten/Simulatoren, erfolgt die Überprüfung der Systembeschreibung mit besonderer Beachtung der noch nicht gesetzten/aktualisierten Werte. Befinden sich Eigenschaften im Systemmodell, die noch nicht gesetzt werden, muss vom zuständigen Systemexperten entschieden werden, ob diese einen statischen Wert zugewiesen bekommen oder aber weitere Simulatoren diese dynamisch setzen.

Sind alle Eigenschaften mit Werten belegt, muss überprüft werden, ob die neu hinzugekommenen Komponenten/Simulatoren weitere Anforderungen in Bezug auf Eigenschaften an das Systemmodell haben. Ist dies der Fall, erfolgt eine erneute Erweiterung des Systemmodells mit wiederum nachfolgender Ermittlung der noch nicht gesetzten Werte und der Zuweisung von weiteren Komponenten. Als Ergebnis dieses Schrittes ergibt sich eine vervollständigte Liste der genutzten Komponenten/Simulatoren für die Co-Simulation.

3.3.2 Anforderungen an die eingesetzten Komponenten/Simulatoren

Die Anforderungen an die eingesetzten Komponenten/Simulatoren sind folgendermaßen angegeben:

[A_S1] **Unterstützung der genutzten Infrastruktur** Damit die einzubindenden Komponenten/Simulatoren innerhalb der Co-Simulation gesteuert werden können, müssen diese die Unterstützung der genutzten Co-Simulationsinfrastruktur gewährleisten.

[A_S2] **Bereitstellung der Distanzattribute** Alle Komponenten/Simulatoren, die integriert werden, um die Berechnung der Subdistanzfunktionen durchzuführen,

müssen die verwendeten Eigenschaften über die genutzte Infrastruktur innerhalb der Co-Simulation bereitstellen.

[A_S3] **Kontrolle über den eigenen Zustand** Um Techniken aus dem Bereich der Rare Event Simulation wie Importance Splitting nutzen zu können, müssen die verwendeten deterministischen Simulatoren, die einen internen Zustand pflegen, der nicht über das gemeinsame Datenmodell kommuniziert wird, eine Zustandskontrolle gewährleisten. Dies bedeutet, dass sie auf Anfrage (vgl. Anforderung [A_S4] Simulatorkontrolle) eines Controllers der Co-Simulation ihre aktuelle Situation speichern und laden können müssen.

[A_S4] **Simulatorkontrolle** Um eine Kontrolle der Co-Simulation hinsichtlich einer Rare Event Simulation zu unterstützen, müssen diese verschiedene Kontrollfunktionen (s. Tabelle 3-4) unterstützen. Insbesondere zu erwähnen ist dabei das Speichern der aktuellen Situation der eigenen Komponente/Simulation und der Start aus einer gespeicherten Situation. Die Unterstützung der Kontrollfunktionen ist notwendig, damit die Komponenten eine Kontrolle hinsichtlich der erwähnten Importance Splitting-Technologie erlauben, die ein Führen der Co-Simulation in Richtung der zu analysierenden, risikoreichen Situationen erlaubt.

[A_S5] **Konfigurierbarkeit der Eigenschaften von Systemelementen** Um eine Konfigurierbarkeit der Simulatoren zu unterstützen, müssen diese über neu gesetzte Parameter informiert werden können, die von den hinterlegten Standardparametern abweichen und von einer Explorationskomponente vorgegeben werden können (vgl. Anforderung [A_A6]).

Name	**Beschreibung**
Start (Zustands-ID)	Startet eine Komponente aus einem vorgegebenen Zustand, angegeben über eine ID. Wird eine leere ID übergeben wird kein Zustand geladen.
Stop	Stoppt die angeschlossene Komponente. Ein weiterer Start der Komponente ist möglich ohne ein neues Starten des Simulator-Programms.
Shutdown	Stoppt die angeschlossene Komponente und terminiert das Simulatoren-Programm.

Save (Zustands-ID)	Die Komponenten speichern ihren aktuellen Zustand unter der angegebenen ID ab.
Pause	Die angeschlossenen Komponenten werden pausiert.
Resume	Führt die angeschlossenen, pausierten Komponenten weiter aus.
Done	Komponenten, die einen Aufruf einer Kontrollfunktion erhalten haben, geben hierüber an, ob der Kontrollaufruf erfolgreich ausgeführt werden konnte.

Tabelle 3-4 Vorgegebene Simulatorfunktionen und ihre Beschreibung zur Simulator- und Situationskontrolle.

3.3.3 Notwendige Erweiterung der Systembeschreibung

Das aktuelle Systemmodell, das auf dem erwähnten gemeinsamen Datenmodell der Co-Simulation aufsetzt, wird zum Situationsvergleich mittels der jeweiligen Systemmodellinstanz zum aktuellen Zeitpunkt der Simulation aber auch zur Bewertung des abstrakten Risikos durch die Risikodistanzfunktion genutzt. Die genutzte Systembeschreibung muss beim Hinzufügen von Risikodistanzfunktionen auf Vollständigkeit überprüft werden. Sind nicht alle zur Beschreibung und Berechnung der Distanz nötigen Eigenschaften hinterlegt, muss das Systemmodell um diese erweitert werden. Die Erweiterbarkeit ist insbesondere notwendig, um die in Anforderung [A_A5] aufgestellte “Definierbarkeit von Risikodistanzen“ zu erfüllen.

3.3.4 Explorationsbeschreibung

Da zur Analyse des Systems und deren Risiken auch eine Aussage über mögliche Attributgrenzen ermittelt werden können sollen – z.B. ab welcher Windgeschwindigkeit sich das Risiko einer Kollision zwischen Ladung und Ladeoffizier erhöht – wurde ein Explorationsmodell entwickelt (vgl. Anforderung [A_A6]). Dieses beschreibt die zu explorierenden Parameter eines Systems, und setzt sich dabei aus den zwei folgenden Einstellungsmöglichkeiten zusammen:

1. Konfiguration der Parametergrenzen,
2. Konfiguration der Parameterexplorationstechnik

Ausgehend vom präsentierten Systemmodell, welches die genutzten Ressourcen und eine Verhaltensspezifikation beschreibt, kann ein zu konfigurierendes Explorationsmodell generiert werden, das weiter angepasst werden kann. Dies ist notwendig, um zu überprüfen, ob sich die Nähe zu Risiken z.B. bei anderen Umgebungseinflüssen ändert. Diese Anpassungen erlauben das Konfigurieren der Parametergrenzen (z.B. eine minimale und maximale Rotationsgeschwindigkeit eines Krans) und der Parameterexplorationstechnik (z.B. eine lineare Exploration, die zusätzliche Informationen über die Schrittweite erhält).

BEISPIEL 8: EXPLORATIONSBESCHREIBUNG

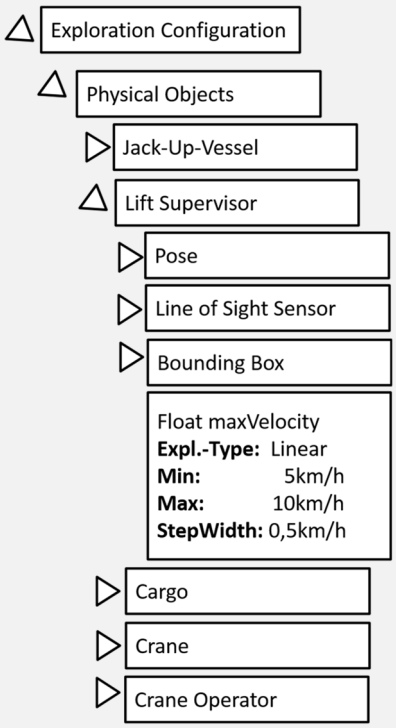


Abbildung 3-18 Ausschnitt der Explorationsbeschreibung zum vorgestellten Beispielszenario.

Die Systembeschreibung nutzend, wird eine erste Ausprägung der Explorationsbeschreibung erstellt (s. Abbildung 3-18). Das Jack-Up-Vessel, der Kran, die transportierte Ladung sowie der Ladeoffizier und Kranführer werden in Elemente des Explorationsmodells konvertiert, damit die zur Parameterexploration notwendigen weiteren Daten (z.B. Explorationstyp, Minimum und Maximum Wert) angegeben und angepasst werden können. Im angegebenen Beispiel wird die maximale Geschwindigkeit

des Ladeoffiziers eingestellt. Für diese wurde eine lineare Exploration als Explorationstyp angegeben mit einem minimalem Wert von 5km/h und einem maximalem Wert von 10km/h. Die Schrittweite beträgt dabei 0,5km/h was einer Gesamtanzahl von elf zu explorierenden Werten für das hier gezeigte einfache Beispiel entspricht.

3.3.5 Nutzung der Risikodistanzen in einer Co-Simulation

Dieser Abschnitt über die Co-Simulationsintegration beschreibt, wie die ermittelten Werte der Methodik zur Bewertung der Simulationszustände in einer Co-Simulation genutzt werden können. Dabei wird beschrieben wie die Einbindung der Risikodistanzfunktionen und die Verwendung dieser zur beschleunigten Erreichung der zu analysierenden Situationen innerhalb einer verteilten Simulationsumgebung erfolgt.

3.3.5.1 Einbindung der Risikodistanzfunktion

In Abbildung 3-19 ist die Einbindung der in der Methodik ermittelten Werte, im Kontext der Co-Simulation zu sehen. Dabei ist eine gepflegte Systemmodellinstanz mit den angeschlossenen Simulatoren verknüpft und erhält von diesen die durch die Simulatoren aktualisierten Werte (vgl. Anforderung [A_C1]). Durch die Systemmodellinstanz, werden die für die Integration einer Distanzbeschreibung notwendigen Komponenten „Risikodistanzberechnung“ und „Situationsvergleich“ über die Werte einer neuen Simulationssituation informiert (vgl. Anforderung [A_S2]). Die Risikodistanzberechnung erhält dabei die in der Methodik ermittelte Risikodistanzfunktion übermittelt, um die Risikoentfernung zu einem abstrakten Risiko, wie z.B. ”Kollision mit Ladung” oder ”Feuer an Bord” mit Hilfe der Werte der Systemmodellinstanz zu berechnen.

Der ermittelte Risikowert wird an den Situationsvergleich weitergeleitet, um eine Entscheidung bezüglich der weiteren Überprüfung der Situation zu vollziehen (vgl. Anforderung [A_A3]). Ist der Risikowert unterhalb eines angegebenen Schwellwertes, ermittelt der Situationsvergleich, ob die gleiche oder eine sehr ähnliche Situation während der einzelnen vorrangegangenen Läufe der Simulation in der Situationsdatenbank hinterlegt wurde.

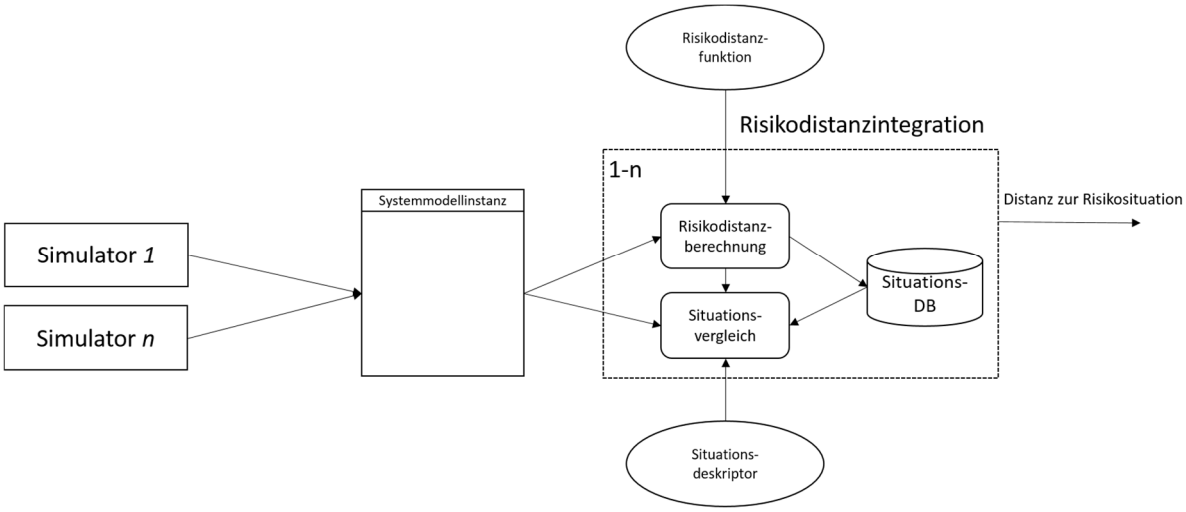


Abbildung 3-19 Übersicht über die Nutzung der Risikodistanzbewertung in einer Co-Simulationsumgebung.

Die Situationsdatenbank basiert dabei auf Techniken des Information Retrieval. Der Bereich der Informationsrückgewinnung (Information Retrieval) beschäftigt sich mit der computergestützten Suche nach komplexen Inhalten. Dabei wird dieser in die Bereiche Informatik, Informationswissenschaft und Computerlinguistik eingegliedert. Oft sind komplexe und in Datenbanken gespeicherte Inhalte wie Bilddaten zunächst nicht zugänglich. Beim Information Retrieval wird versucht bestehende Informationen aufzufinden jedoch nicht neue Strukturen zu entdecken, wie es beim Data-Mining und Text-Mining der Fall ist.

Automatisierte Information Retrieval Systeme werden verwendet, um das sogenannte „Information Overload“ zu reduzieren. Viele Universitäten und öffentliche Bibliotheken verwenden Information Retrieval Systeme, um einen Zugang zu Büchern und anderen Dokumenten zur Verfügung zu stellen. Die wohl sichtbarsten Information Retrieval Anwendungen sind Suchmaschinen (Google, Bing) innerhalb des Internets (vgl. [BüCC10]).

Durch die immer größer werdende Masse an Informationen verschwinden viele wertvolle Informationen ungenutzt da keine effektiven Methoden zur Informationsrückgewinnung existieren. Das Information Retrieval Problem wurde viele Jahre untersucht und die Lösung als ein Weg beschrieben, um relevante Informationen von irrelevanten zu trennen hinsichtlich eines bestimmten Informationsbedarfs. Diese Fragestellung beantwortende Systeme werden Information Retrieval Systeme genannt. Es existieren verschiedene Modelle von Information Retrieval Systemen. Die bekanntesten sind das Boolesche-, das Vektorraum- und das Probabilistische-Modell (vgl. [ThMC96]).

Ein Deskriptor beschreibt im Information Retrieval einen Term, der die Essenz des Themas eines Dokuments beschreibt. Sie werden als Schlüsselwörter verwendet werden, um Dokumente in einem Informationssystem abrufen, zum Beispiel in einem Katalog oder einer

Suchmaschine. Bekannte Formen von Schlüsselwörtern sind Web Tags welche direkt sichtbar sind und auch von nicht-Experten zugewiesen werden können. Index Terme können aus einem Wort, einer Phrase oder einem alphanumerischen Term bestehen. Sie werden erstellt in dem das zu indexierende Dokument analysiert wird wobei dies manuell oder automatisch passieren kann (vgl. [Ferb03]).

Die gefundenen Dokumente werden in die Datenbank kopiert. Dafür werden zwei Dateien erzeugt zum einen eine Dokumentendatei, zum anderen eine invertierte Version dieser Datei. In der invertierten Datei sind Informationen über die Position von Phrasen oder Wörtern im Dokument sowie Strukturinformationen zu finden. Die Phrasen und Wörter werden in der richtigen Reihenfolge abgelegt, als auch rückwärts persistiert. Dies erlaubt eine offene Linkstrukturierung. Das Speichern der invertierten Datei wird in einem Datenbankindex durchgeführt.

In dieser Arbeit wird der Bereich des Information Retrieval betrachtet, der für die Klassifizierung von Informationen verwendet wird. Dabei wird ein Deskriptor automatisch durch das zu Grunde liegende Systemmodell und dessen Ausprägung, der Systemmodellinstanz, beschrieben.

Sie erlaubt die Anfrage nach der Ähnlichkeit zu bereits erreichten Situationen. Dafür nutzt die vergleichende Komponente, einen Situationsdeskriptor, der auf den genutzten Elementen des beschriebenen Systemmodells und den Daten der Systemmodellinstanz basiert. Der Situationsdeskriptor ist dabei ein Vektor dessen Elemente die jeweiligen Eigenschaften der Systemelemente wiederspiegeln (s. Abbildung 3-20).

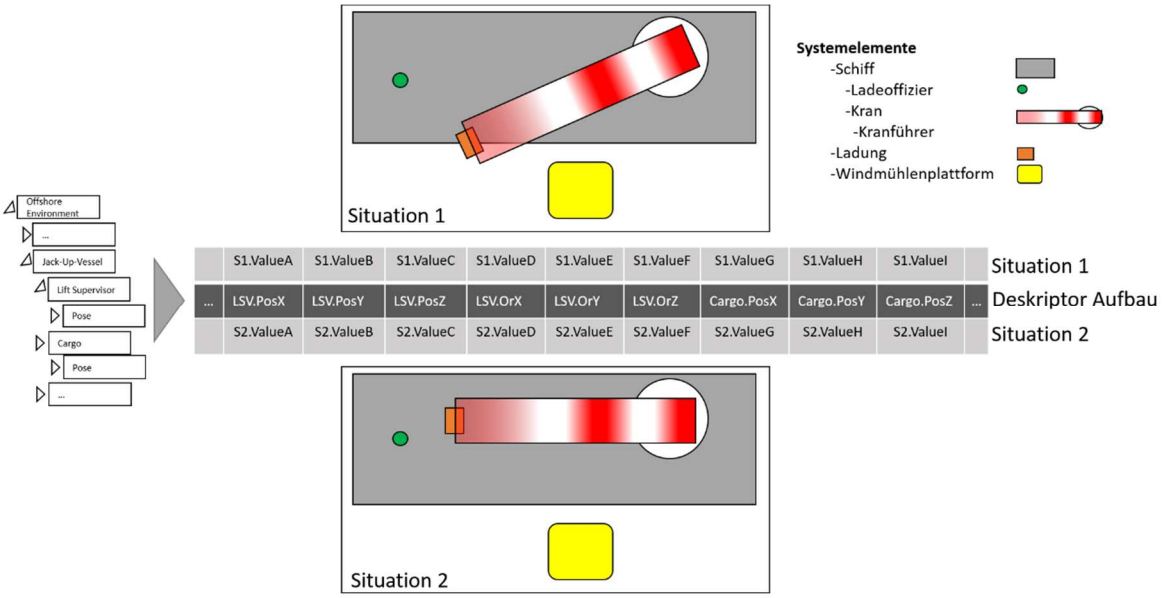


Abbildung 3-20 Beispiel für den Aufbau eines Situationsdeskriptors. Situation 1 und Situation 2 unterscheiden sich nur leicht durch die Orientierung des Kranarms und der Position der Ladung. Diese Eigenschaften weisen dementsprechend unterschiedliche Werte in ihrem jeweiligen Deskriptor auf.

Die Reihenfolge der Elementeigenschaften im Vektor ist fest vorgegeben, damit der Vergleich nicht verfälscht wird. Die Größe bzw. die Anzahl der Elemente der Deskriptoren für ein Simulationsszenario ist durch den Aufbau des Systemmodells bekannt. Der Vergleich erfolgt dabei auf Grundlage eines Distanzmaßes. Dabei können statische Eigenschaften für die Berechnung der Ähnlichkeit ignoriert werden. Eigenschaften mit numerischen Werten können direkt genutzt werden während boolesche Werte als False (Wert 0) und True (Wert 1) und textuelle Werte im Vergleich über ihre Gleichheit oder Ungleichheit berücksichtigt werden. Für die Berechnung der Distanzen lässt sich eine größere Anzahl an Maßen in der Literatur finden. So existieren unter anderem binäre Distanzmaße die nur die Übereinstimmung der einzelnen Vektorwerte überprüfen und bewerten, wie zum Beispiel die Tanimoto-Distanz [Nakh13] und geometrische Distanzmaße wie die Manhattan (L1) oder Euklidische Distanz (L2). Im Rahmen dieser Arbeit wurde sich für die Verwendung der euklidischen Distanz (L2), als bekanntes universell einsetzbares Distanzmaß entschieden.

Durch den Einsatz von für die Information Retrieval Technik entwickelten Datenbanken (z.B. Apache Lucene[12]) lässt sich die Ähnlichkeit durch das vorgegebene Distanzmaß, trotz

[12] Apache Lucene: http://lucene.apache.org/ [zuletzt abgerufen am 03.03.2017]

Aufrufen mathematischer Operationen zur Berechnung der Distanz, in angemessener Zeit durchführen.

Existiert keine ähnliche Situation in der Datenbank, wird die Datenbank um diesen Datensatz ergänzt und nach einem Simulationslauf um die maximale Risikoannäherung angereichert. Ist eine ähnliche Situation in der Datenbank persistiert, wird überprüft und aktualisiert, wie viele Simulationsläufe aus dieser bereits durchgeführt wurden und wie die maximale Annäherung an die abstrakte Risikosituation war.

Sind noch nicht ausreichend viele Simulationsläufe in die gleiche Situation exploriert oder die maximale Annäherung an die risikoreiche Situation vielversprechend, werden die Ergebnisse der Situationsbewertung ausgegeben und können genutzt werden, um z.B. die aktuelle Situation speichern zu lassen oder den aktuellen Simulationslauf abzubrechen.

In Abbildung 3-21 ist die Nutzung und der Aufbau der Situationsdatenbank zur Erkennung der Erreichbarkeit von Situationen innerhalb der Läufe einer Co-Simulation zu sehen. Die einzelnen Bilder beschreiben dabei von links nach rechts den Aufbau der Situationsdatenbank während eines beispielhaften Simulationsverlaufs. Die einzelnen Kreise stellen dabei Situationen/Systemzustände dar, die eine Bewertung erhalten haben. Von grün nach rot ist dabei die Nähe zum untersuchten Risiko dargestellt, während die verbindenden Pfeile die Erreichbarkeit von Zuständen darstellt. Die Vorteile der Situationsdatenbank sind dabei, dass oft besuchte ähnliche Situationen und benutzte Übergänge zwischen Situationen erkannt werden können. Dabei können über vorgegebene Schwellwerte Systemzustände innerhalb der Situationsdatenbank konsolidiert werden.

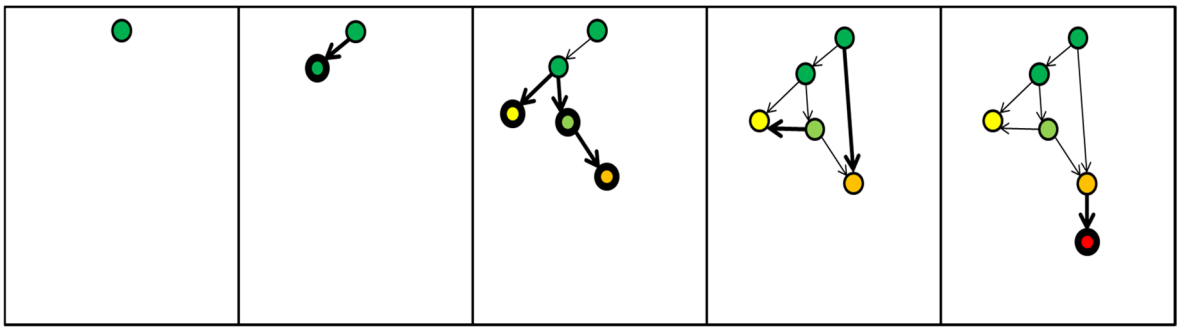


Abbildung 3-21 Nutzung der Situationsdatenbank zur Erkennung der Erreichbarkeit von Situationen innerhalb der Co-Simulation. Die Farben von grün (weit entfernt vom untersuchten Risiko) nach rot (Risiko eingetreten) geben dabei die Bewertung des Zustands an. Die Pfeile zeigen an, aus welcher Situation welche Situationen erreicht wurden.

Durch diese Informationen lässt sich, die auf dem Importance Splitting Verfahren aufsetzende Technik Simulationslauf übergreifend zur gezielten Steuerung der Simulation in die Richtung einer risikoreichen Situation, umsetzen.

In Abbildung 3-22 ist ein Beispiel für optimierte Simulationstrajektorien zu sehen wobei die Splitting Points durch die Distanz zu risikoreichen Situationen vorgegeben wurden.

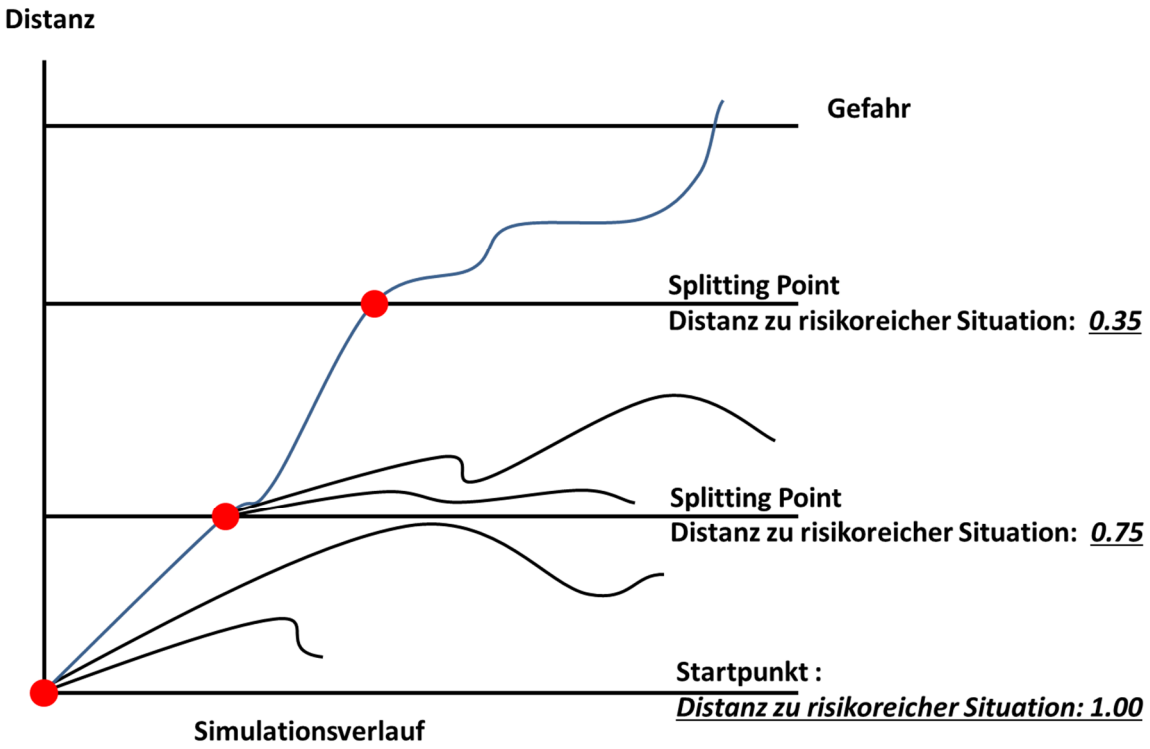


Abbildung 3-22 Beispiel für optimierte Simulationstrajektorien bei der Nutzung von Risikodistanzen zur Generierung der Splitting Points.

BEISPIEL 9: EINBINDUNG DER RISIKODISTANZFUNKTION IN EINE CO-SIMULATIONSUMGEBUNG

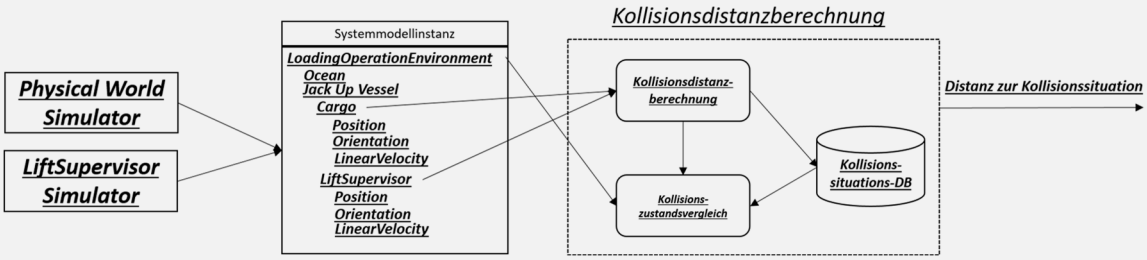


Abbildung 3-23 Übersicht über die Co-Simulations-Integration im Beispiel Szenario.

In Abbildung 3-23 ist die Einbindung der ermittelten Werte in die Co-Simulationsumgebung zu sehen. Als Simulatoren für das Szenario wurde ein Physical World Simulator (PWS) identifiziert, der die Attribute der physikalischen Objekte und Umwelteinflüsse simuliert und ein weiterer Simulator (LiftSupervisor Simulator), der die Steuerung des Ladeoffiziers übernimmt. Die endgültige Systembeschreibung ist in der Systemmodellinstanz zu sehen, welches alle benötigten Attribute zur Distanzberechnung beinhaltet. In dem verwendeten Beispiel die Ladung (Cargo) und den Ladeoffizier (LiftSupervisor) mit den für die Berechnung der Subdistanzfunktionen nötigen Eigenschaften. Die ermittelte Distanzbeschreibung zum Kollisionsrisiko, wird an die Risikodistanzberechnung übergeben und der ermittelte Risikosituationsdeskriptor an den Situationsvergleich. Zusätzlich wurde eine Datenbank für das Kollisionsrisiko hinzugefügt, in der die geringste Entfernung aus ähnlichen Zuständen beschrieben wird.

3.3.5.2 Exploration zur risikoreichen Situation

Wie in Abschnitt 3.3.5.1 (Einbindung der Risikodistanzfunktion) beschrieben, kann durch das Einbinden der Risikodistanzbewertung innerhalb der Simulation auf die Distanz der jeweiligen Situation zur untersuchten Situation reagiert werden. Ein Vorteil der daraus gezogen werden kann ist die Beschreibung einer gesteuerten Co-Simulation, auf Grundlage der berechneten Distanzen. In Abbildung 3-24 wird gezeigt, wie diese verwendet werden können. Wie aus dem vorherigen Abschnitt bekannt, wird die Funktionalität der Risikodistanzberechnung (Risikodistanzintegration) über eine Anbindung an die Systemmodellinstanz erreicht. Dabei werden die in der Systemmodellinstanz hinterlegten aktuellen Werte an die Distanzberechnung weitergeleitet, die Berechnung der Distanz findet statt und kann zur weiteren Verarbeitung in einer Simulationsablauflogik verwendet werden. In dieser werden alle Entscheidungen bzgl. des Simulationsverlaufs festgelegt. Sie enthält damit die Möglichkeit, die Distanzberechnungen hinsichtlich eines Schwellwertes zu testen

und auf dieser Grundlage den weiteren Verlauf der Simulation zu bestimmen (vgl. Anforderung [A_C3]).

Um den Verlauf definieren zu können, werden die Simulationskontrollfunktionalitäten eingesetzt, die im Abschnitt 3.3.1 „*Ermittlung der benötigten Simulatoren*“ vorgestellt wurden. Dies bedeutet, dass die Situation der Co-Simulation, nachdem sich laut der Berechnung der Distanzfunktion weiter der risikoreichen Situation genähert wurde, abgespeichert werden kann und aus dieser weitere Simulationsläufe gestartet werden können.

Um diese Läufe weiter zu konfigurieren, bietet eine Komponente zur Parameterexploration, die auf der erstellten Explorationskonfiguration basiert die Möglichkeit, Anfragen bzgl. der nächsten Parametereinstellungen zu stellen. Die angefragten nächsten Parameterwerte werden dann wiederum über die Systemmodellinstanz, also mittels der Kommunikationsinfrastruktur, oder in Datei Form (abhängig von den verwendeten Simulatoren) an die Simulatoren der Co-Simulation weitergeleitet. Die Simulationsablauflogik startet die nächsten Simulationsläufe aus der neu gespeicherten Situation und exploriert die Parameter laut der im Explorationsmodell angegebenen Grenzen, Schrittweiten und Techniken. Dabei kann innerhalb der Explorationskomponente angegeben werden, ob mittels einer Tiefensuche oder Breitensuche die nächste Parameterauswahl getroffen werden soll. Die Parameterexploration kann dabei auf verschiedenen Auslöser-Ebenen stattfinden. So kann die Explorationskomponente die nächsten zu explorierenden Werte am Start eines Simulationslaufs liefern aber auch auf andere Ereignisse, wie zum Beispiel eine bestimmte Simulationszeit, einen vorher definierten Simulationsschritt oder eine bestimmte Risikobewertung, reagieren.

Die abstrakt gehaltene Logik zur Beschreibung des Simulationsablaufs kann dabei beliebig komplex ausfallen und auch externe, über die Kommunikationsinfrastruktur angebundene Komponenten nutzen. Weitere notwendige Beschreibungen des Simulationsablaufs sind zum Beispiel die Definition des Endes der Simulation oder eines Simulationslaufes, welche zum Beispiel auf einer maximalen Anzahl von Simulationsläufen (Simulationsende) oder einer maximalen Simulationszeit zu der ein Simulationslauf abgebrochen wird basiert.

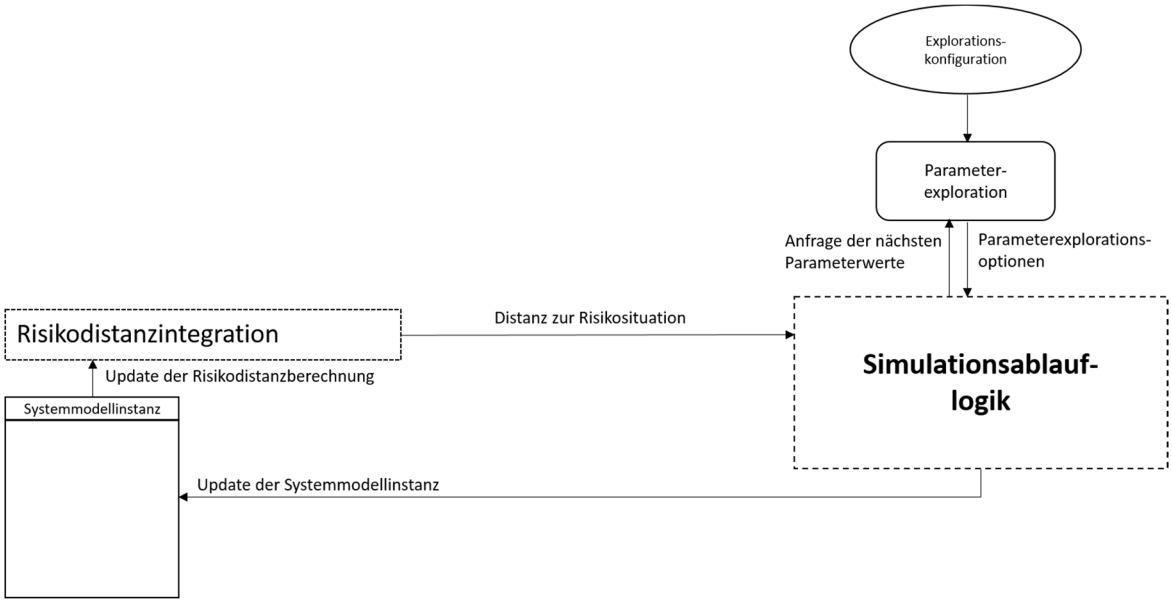


Abbildung 3-24 Übersicht über die Nutzung der Risikodistanzfunktionalität in einer Co-Simulation. Diese wird zur Exploration der Simulation in die Richtung der zu untersuchenden risikoreichen Situationen verwendet.

BEISPIEL 10: EXPLORATION ZUR RISIKOREICHEN SITUATION

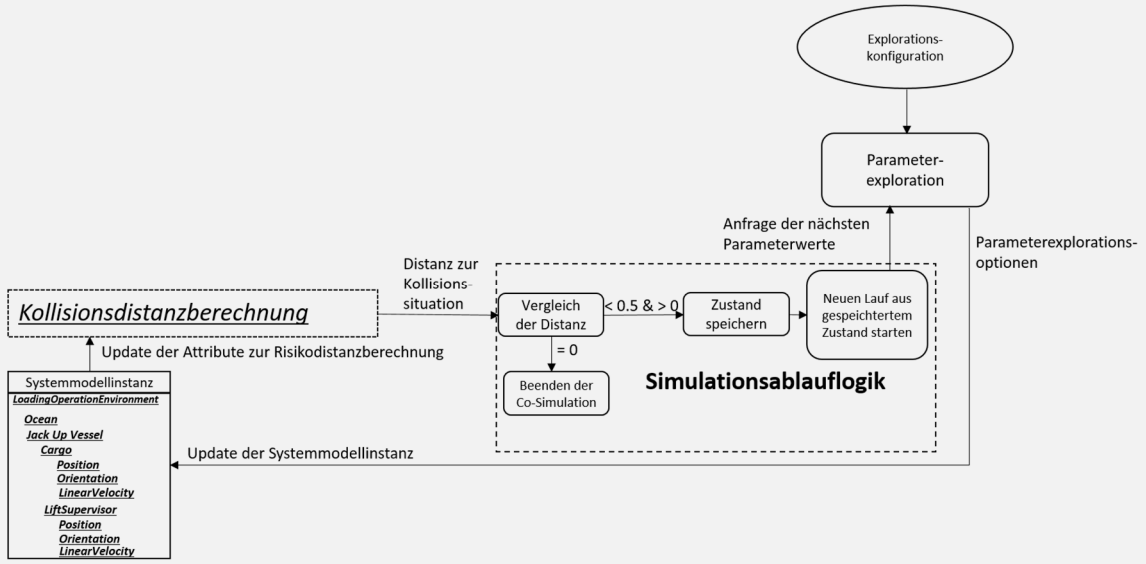


Abbildung 3-25 Übersicht über eine beispielhafte Co-Simulations-Integration, um zu einer risikoreichen Situation zu explorieren.

In Abbildung 3-25 ist die Einbindung der im vorherigen Abschnitt gezeigten, Kollisionsdistanzberechnung zu sehen. Diese wird im beschriebenen Beispiel genutzt, um den weiteren Verlauf der Simulation zu bestimmen. Dafür wird zunächst die Distanz durch einen Vergleichsoperator mit einem dynamisch definierbaren Schwellwert, der zunächst

auf 0,5 eingestellt ist, verglichen. Ist die Distanz kleiner als 0,5 und größer als 0, wird den Simulatoren mittels der Kontrollfunktionen mitgeteilt, ihre aktuelle Situation zu speichern. Im folgenden Schritt wird ein nächster Simulationslauf aus der gespeicherten Situation geladen. Sind die Simulatoren bereit, wird die Parameterexplorationskomponente angefragt, um die nächsten zu simulierenden Parameter für den neu geladenen Simulationslauf zu erhalten. Die erhaltenen Parameter werden in diesem Beispiel über eine Aktualisierung der Systemmodellinstanz an die Simulatoren gesendet und der neue Simulationslauf gestartet. Ist die Distanz gleich 0, also die risikoreiche Situation eingetreten, wird in diesem Beispiel die Co-Simulation beendet. Zusätzlich wurde das Simulationsende in diesem Fall dadurch definiert, dass ein Auslöser auf die aktuelle Anzahl von ausgeführten Simulationsläufen horcht und diese mit einer vorkonfigurierten maximalen Anzahl vergleicht. Ein weiterer Auslöser horcht auf die vergangene Simulationszeit des Simulationslaufs und beendet den Simulationslauf nach einem Maximum von 30 Minuten (die vom Systemexperten vorgegebene, maximale Dauer der erfolgreichen Verladeoperation). Die ermittelten Distanzen und die übertragenen Werte in der Systemmodellinstanz werden parallel bei der Ausführung mitgeloggt, um den Verlauf dieser im Nachhinein zu analysieren (vgl. Anforderung [A_C2]).

3.4 Zusammenfassung

In diesem Kapitel wurde die entwickelte Methodik zur Erstellung von Risikodistanzen vorgestellt. Dabei fand eine Unterteilung in die Basis und Vorarbeit, also das Vorgehen zur Beschreibung des zu untersuchenden Systems, die Erläuterung der entwickelten Methodik und die Integration in eine Co-Simulation statt. Die Teilaspekte wurden zusätzlich anhand von Beispielen zu einem durchgängigen Szenario aus der Offshore-Domäne präsentiert.

Die Basis des eigenen Ansatzes wurde aufgezeigt, die aus der Erstellung einer Systemmodellinstanz inklusive Verhaltensbeschreibung und annotierter Gefahren sowie Ursachen und dem daraus generierten Fehlerbaum besteht.

In der Erläuterung der entwickelten Methodik wurde gezeigt, wie aus den in der Vorarbeit präsentierten Modellen, insbesondere des Fehlerbaummodells, die Risikodistanzfunktion erstellt wird. Dabei wurde die Ermittlung der Eintritts- und Relevanzräume und die Erstellung der in die Risikobeschreibung einfließenden Subdistanzfunktionen näher betrachtet. Hierbei bleibt zu erwähnen, dass unglücklich formulierte Distanzfunktionen dazu führen können, dass sich die Distanz entfernt obwohl das Risiko/die Gefahr näher rückt. Jedoch wird die Risikoanalyse, die immer abhängig von Experten und ihrem Wissen ist,

durch die Verwendung und den Einsatz der Risikodistanzfunktionen nicht verschlechtert sondern durch die Möglichkeit übersehene Ursachen simulativ ausfindig zu machen verbessert. Um eine Unterstützung bei der Auswahl von Eigenschaften die für eine Risikodistanz relevant sein können zu geben, werden Vorschläge für den nutzenden Experten ausgehend von den Verknüpfungen des Fehlerbaums und der damit verknüpften Elemente, generiert.

Abschließend wurde die Einbindung in eine Co-Simulation aufgezeigt. Es wurde erklärt, wann und wie die Erweiterung der Systembeschreibung und die Ergänzung von Simulatoren/Komponenten zu erfolgen hat. Bei der Vorstellung der Methodik wurden ebenfalls die Anforderungen aufgezeigt, die an die Risikobeschreibung und die genutzten Simulatoren/Komponenten gestellt werden. Zusätzlich wurde das Explorationsmodell präsentiert, welches die Konfiguration einer systematischen Parameterexploration ermöglicht. Bei der Nutzung der Risikodistanzfunktionen fand eine Unterteilung zwischen der eigentlichen Integration der in der Methodik ermittelten Risikobeschreibung und der Verwendung dieser zur beschleunigten Erreichung der zu analysierenden, risikoreichen Situationen statt.

4 DistriCT - Ein Framework zur Konfiguration, Kontrolle und Analyse von Co-Simulationen

In diesem Kapitel wird die Realisierung des entstandenen Frameworks zur Konfiguration, Steuerung und Analyse von Co-Simulationen vorgestellt. Dabei wird beschrieben, wie die ermittelten Anforderungen zur Anwendung des eigenen Ansatzes im Konzept berücksichtigt worden sind, um Simulatoren und Simulationen zu beschreiben und um eine Steuer- und Analysierbarkeit dieser zu erreichen [GPLG14].

DistriCT (Distributed Controlling Toolkit) ist dabei ein Bestandteil des am OFFIS und der Universität Oldenburg entwickelten HAGGIS Systems [HGBS15, SGHB14]. Das HAGGIS System unterstützt bei der Entwicklung und Evaluierung neuer e-Maritime und e-Navigation Technologien. HAGGIS kann in die drei Bestandteile Modellierung (Szenario Definition, Experiment Definition), Simulation, Analyse und Observierung unterteilt werden (s. Abbildung 4-1).

Als Beispiel existieren Modellierungstools zur Beschreibung der statischen Objekte eines Szenarios und von Umweltbedingungen und ihrer Veränderung über die Zeit. Simulatoren wie zum Beispiel die Environment Simulation berechnen die Umwelteinflüsse nach den modellierten Angaben um und leiten diese über die Kommunikationsinfrastruktur an die anderen Simulatoren der Co-Simulation weiter. Die kommunizierten Daten basieren hierbei auf dem S-100 Standard. Die Aufgaben von DistriCT lassen sich in die Bereiche Experiment Definition und Analyse einordnen.

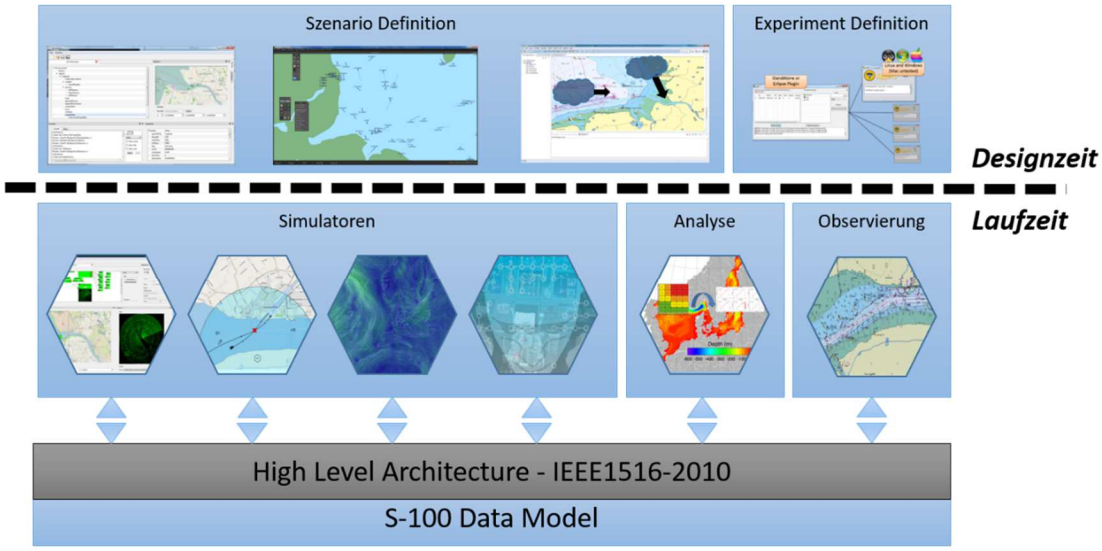


Abbildung 4-1 Gesamtarchitektur des HAGGIS Frameworks (vgl. [HGBS15]).

Das entwickelte Konzept für die DistriCT Komponente unterteilt sich dabei in die folgenden drei Bereiche, welche aus den ermittelten Anforderungen für den eigenen Ansatz abgeleitet wurden.

Analyse der Co-Simulation

Der Bereich der Analyse der Co-Simulation beschreibt, wie auf Änderungen von Eigenschaften des Systems reagiert werden kann und wie durch das Auslesen der Eigenschaftsänderungen die Berechnung der Risikodistanzen ermöglicht wird. Zusätzlich wird in diesem Bereich beschrieben, wie die kommunizierten Daten innerhalb einer Simulation und eines Simulationslaufs für eine weiterführende Offline-Analyse protokolliert werden.

Steuerung der Co-Simulation

Der Bereich der Steuerung beschreibt, wie die Steuerung der Co-Simulation erfolgen kann. In diesem wird beschrieben, wie Parameteränderungen gesetzt werden, wie Abbruch und Zielbedingungen zum Stoppen einer Co-Simulation genutzt werden und wie das Laden und Speichern des Co-Simulationszustandes erfolgt.

Konfiguration der Co-Simulation

Der Bereich der Konfiguration beschreibt, wie die Techniken zur Analyse und Steuerung der Co-Simulationsumgebung genutzt werden, um die Parameterexploration und den Verlauf der Co-Simulation zu beschreiben. Dies wird durch eine Beschreibung der Simulatoren und ggf. benötigter Middleware auf Programmebene, sowie einer Beschreibung der Systeme, die für die Ausführung der

Programme zur Verfügung stehen, ermöglicht. Die Beschreibung des Verlaufs erfolgt dabei über einen Simulationsplan. In diesem kann der Sicherheitsexperte mittels vordefinierter Komponenten den Simulationsverlauf zur Designzeit konfigurieren.

In Abbildung 4-2 ist eine vereinfachte Übersicht über die DistriCT Verwendung zu sehen. Es existiert eine zentrale Kommunikationskomponente, über welche die Simulatoren die benötigten Daten untereinander austauschen. Die Kommunikationskomponente pflegt eine Instanz des Systemmodells in der alle aktuell kommunizierten Daten der Simulatoren hinterlegt sind. Dieses ist über das DistriCT Framework nutzbar, um Analysekomponenten wie die Risikodistanzberechnung über den aktuellen Zustand der Co-Simulation zu informieren, um eine Berechnung der Distanzen durchzuführen. Des Weiteren existieren Kontrollkomponenten, welche die Steuerung der Simulatoren auf Programm- und Simulator-Ebene ermöglichen. Die Steuerung auf Programmebene erlaubt beispielsweise das Starten und Stoppen der Simulator-Programme auf unterschiedlichen Systemen. Die Steuerung auf Simulator-Ebene meint zum Beispiel das Laden und Speichern eines Zustands oder aber das Setzen von neuen Eigenschaftswerten in der Systemmodellinstanz, um Simulatoren über eine neue Parameterkonfiguration zu informieren. Der Ablauf der Simulation kann dabei in einem Graphen (Simulationsplan) mittels Analyse- und Kontrollkomponenten konfiguriert werden, um auf eingehende Ergebnisse der Analysekomponenten zu reagieren und eine Steuerung der Co-Simulation ausgehend von den neuen Analyseergebnissen zu gestalten.

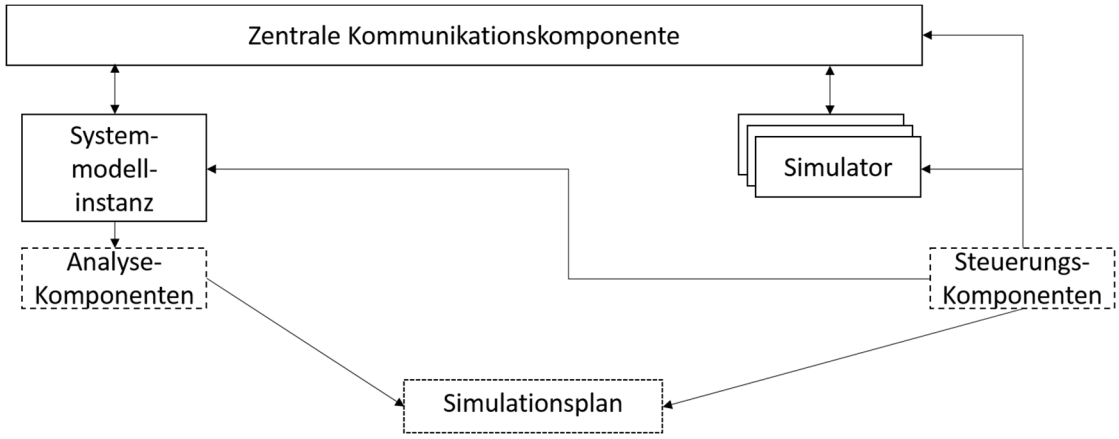


Abbildung 4-2 Vereinfachte Übersicht über die DistriCT-Architektur und Verwendung.

4.1 Definition der eingesetzten Co-Simulation und deren Simulatoren

Damit die benötigte Kontrolle und Steuerung der Co-Simulation erreicht werden kann, muss eine Spezifikation der Infrastruktur der Co-Simulation und der beinhalteten Komponenten erfolgen, die in diesem Abschnitt vorgestellt wird. Durch diese kann mittels des entwickelten DistriCT-Frameworks eine Verteilung der Simulationsprogramme auf verschiedene Systeme im Netzwerk, das Starten, Stoppen dieser wie auch eine Steuerung der Co-Simulation durchgeführt werden.

4.1.1 Genutzte Infrastruktur der Co-Simulation

In der verwendeten Co-Simulation wurde sich für die High Level Architecture (HLA) Infrastruktur entschieden [LäGH13]. HLA ist ein Standard für die Implementierung einer Simulationsumgebung, die den Austausch von Objektinstanzen und Interaktions-Nachrichten (HLAInteractions) erlaubt. Die Kommunikation findet meistens[13] über eine dedizierte zentrale Komponente statt, die Runtime Infrastructure (RTI). Alle Simulatoren, genannt Federates im HLA Vokabular, registrieren sich bei der RTI. Sie kommunizieren, welche Objekte und Interaktionen sie aktualisieren bzw. senden möchten und für die sie über Updates informiert werden möchten. Um dies zu konfigurieren, existiert eine gemeinsame Definition aller Objektklassen und Interaktionen. Die Basis dafür heißt Object Model Template (OMT) und wird als Federation Object Model (FOM) von der RTI und als System Object Model (SOM) von allen Federates verwendet. Eine Simulationsumgebung, die aus mehreren Federates und einer RTI besteht, heißt Federation im HLA Vokabular. Jeder Federate in der Föderation besitzt eine eigene SOM. Die SOM legt für die einzelnen Federates fest für welche Objekte und Attribut-Änderungen Benachrichtigungen erhalten und Updates gesendet werden können. Die RTI erhält eine aggregierte Fassung aller SOMs, die FOM. Durch die FOM kennt die RTI alle kommunizierten Daten und kann somit die Kommunikation kontrollieren und eine Validierung dieser Daten durchführen.

[13] Es existieren Implementierungen des Standards, wie zum Beispiel Portico (http://www.porticoproject.org [zuletzt abgerufen am 03.03.2017]), in denen keine dedizierte zentrale RTI genutzt wird. Hierbei übernimmt ein Federate diese Aufgabe.

Da es sich bei der HLA nur um eine Spezifikation handelt, gibt es keine Referenzimplementierung. Allerdings existieren mehrere Implementierungen, sowohl kommerzielle als auch nicht-kommerzielle, die den Standard als Grundlage verwenden.

Drei Versionen der Definition werden verwendet (Version 1.3, [Usde98], Version 1516-2000, [Ieee00], Version 1516-2010, [Ieee10]), die nicht kompatibel untereinander sind. Tatsächlich, ist auch die Version 1516-2000 unter unterschiedlichen Implementierungen nicht vereinbar, da der Standard defekt ist und zwei verschiedene Interpretationen davon bestehen (vgl. [Gran04]).

Um eine Verwendung externer Simulatoren mit unterschiedlichen HLA Implementierungen zu garantieren wurde eine Hilfsbibliothek umgesetzt. Diese ermöglicht, die intern verwendete HLA Umsetzung zu wechseln ohne eine Änderung der Schnittstelle zu den Simulatoren durchzuführen. Nur die Bibliothek selbst muss angepasst werden, um mehrere Implementierungen zu unterstützen, die auf einfache Art und Weise ausgetauscht werden können (vgl. [LäGH13]).

4.1.2 Aufbau der verwendeten Co-Simulation

In Abbildung 4-3 ist der Aufbau der verwendeten Co-Simulationsinfrastruktur (*CoSimulationConfiguration*) als vereinfachtes Klassendiagramm zu sehen. Diese besteht aus einer vorgegebenen Infrastruktur über die der Aufbau und die Anforderungen definiert sind welche während der Designzeit zur Validierung genutzt werden. Dabei wird beispielsweise sichergestellt, dass eine bestimmte Middleware für die Kommunikation vorhanden ist oder benötigte Dateien für die Ausführung der Middleware und Simulatoren generiert werden. Die Co-Simulation verwendet dabei *Requirement*-Objekte, um zum Beispiel als zentrale Kommunikationskomponente das HLA Runtime Infrastructure Gateway (RTIG) zu fordern (*MiddlewareRQ*). Zusätzlich werden über *SimulationComponentRQ* Anforderungen an die Middleware und die Simulatoren angegeben.

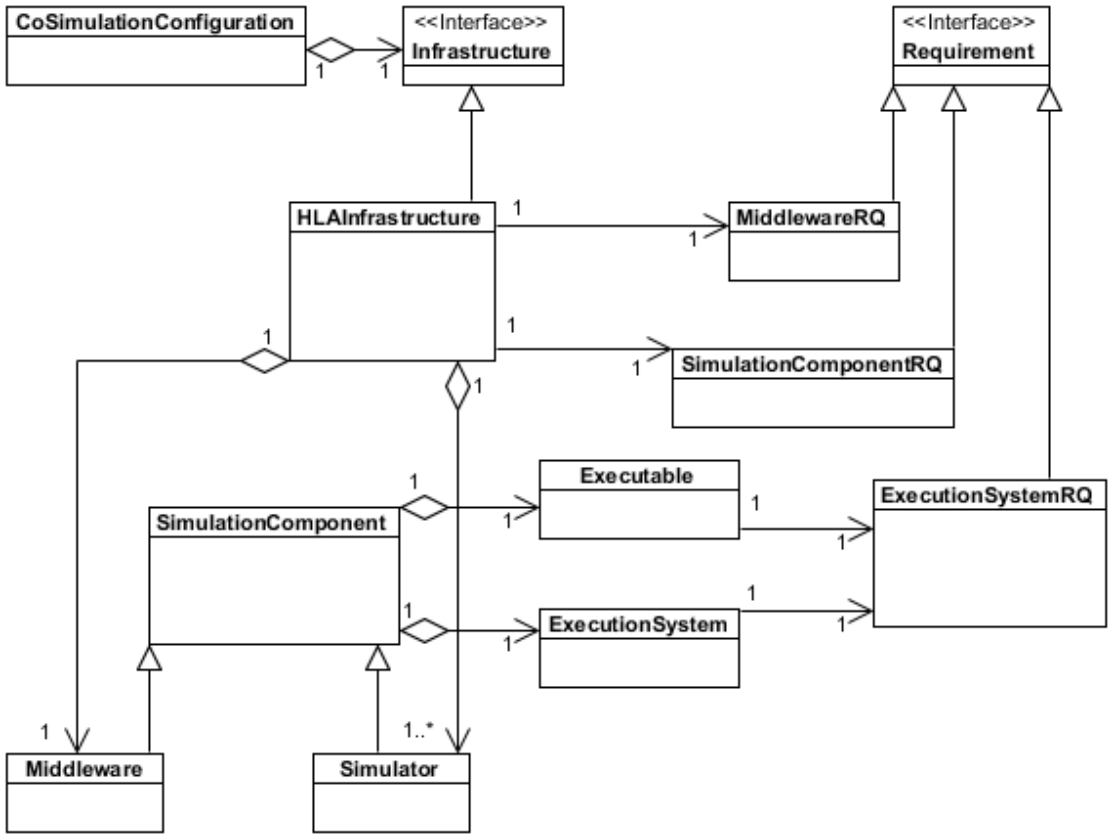


Abbildung 4-3 Aufbau einer Co-Simulation als vereinfachtes Klassendiagramm.

4.1.3 Beschreibung der Simulationskomponenten

Die verwendeten Simulationskomponenten wie z.B. Middleware und Simulatoren setzen sich aus *Executable*- und *ExecutionSystem*-Objekten zusammen. Das *Executable*-Objekt beschreibt das einem Simulator zu Grunde liegende, ausführbare Programm, während das *ExecutionSystem*, ein Client-System im Co-Simulations-Netzwerk beschreibt.

Ein Simulationsprogramm hat Anforderungen, die erfüllt sein müssen damit es auf einem System ausgeführt werden kann. Zur Verfügung stehende Systeme haben Fähigkeiten, die das Gegenstück zu den Anforderungen der Simulationsprogramme darstellen. Ein Beispiel dafür wäre die Anforderung eines Simulationsprogramms nur unter dem Windows Betriebssystem und einer 64bit Architektur lauffähig zu sein.

Zusätzlich können Anforderungen an benötigte Konfigurationsdateien angegeben werden die vom Simulationsprogramm verlangt werden.

Um die Entwicklung von Co-Simulationen zu unterstützen, wird dabei neben der Verteilung von fertig kompilierten Simulationsprogrammen auch die Möglichkeit gegeben, auf dem lokalen System Java Projekte im Debug-Modus auszuführen.

In Listing 4-1 ist eine Beispiel-Beschreibung für ein genutztes Simulationsprogramm für ein Windows 64-Bit-System zu sehen.

```
<executable:ProgramConfiguration name="LiftSupervisorSim"
path2Executable="lss.exe" successFullExecutionOutput="running"
time2WaitForExecutionSuccess="60">
    <requirements>
        <requirementsList key="os" value="win"/>
        <requirementsList key="architecture" value="64"/>
        <requirementsList
      xsi:type="executable:ResourceRequirement"
      upperBound="-1" fileExtension="conf"/>
    </requirements>
</executable:ProgramConfiguration>
```

Listing 4-1 Beispiel für eine Executable-Konfiguration mit Anforderungen an das ausführende System und die benötigten Ressourcen.

Speziellere Anforderungen, wie die nach bestimmter Hardware des ausführenden Systems können dabei ebenfalls über den Anforderungsmechanismus gestellt werden. In Listing 4-2 ist die Beschreibung für ein ausführendes System dargestellt, das neben den Fähigkeiten die das LiftSupervisor-Simulatorprogramm fordert, auch eine Anforderung bzgl. des zur Verfügung stehenden Arbeitsspeichers erfüllt. Jedes System, das der Co-Simulation zur Verfügung steht, startet dafür einen speziellen Dienst, der die Erreichbarkeit und Konfigurierbarkeit des ausführenden Systems für DistriCT ermöglicht.

```
<ExecutionEnvironmentConfiguration showGui="true"
javaBin="C:\Programme\Java\jdk1.7\bin\java.exe"
allowRemoteControl="true" allowScreenCapture="true">
      <fullfilledRequirements>
            <requirementsList key="os" value="win"/>
            <requirementsList key="architecture" value="32"/>
            <requirementsList key="RAM" value="1000"/>
      </fullfilledRequirements>
</ExecutionEnvironmentConfiguration>
```

Listing 4-2 Konfiguration des ausführenden Systems (*ExecutionEnvironment*).

Ein entwickeltes Programm (*Executable*), welches als Simulator innerhalb der Co-Simulationsumgebung verwendet werden soll, unterstützt zum einen die geforderte

Infrastruktur der Co-Simulation (vgl. Anforderung [A_S1]) zum anderen muss es die in Anforderung [A_S4] vorgegebene Kontrollfunktionalität unterstützen. Hierfür wird das HLA Konzept der Interaktionen verwendet. Wobei jede Kontrollfunktionalität als Interaktion in den OMT Dateien beschrieben ist.

Der automatisch generierte Part, der die Kontrollfunktionalität als HLAInteractions angibt, ist in Listing 4-3 dargestellt. Die jeweilige Implementierung dieser Interactions, inklusive der Funktionalität den aktuellen Zustand zu laden und zu speichern (vgl. Anforderung [A_S3]), muss dabei ein Bestandteil des Simulationsprogramms sein.

```
<interactionClass name="StartComponent" order="Receive"
sharing="Subscribe" transportation="HLAreliable">
      <parameter dataType="HLAString" name="SaveStateID"/>
      <parameter dataType="HLAString" name="CommandID"/>
</interactionClass>
<interactionClass name="StopComponent" order="Receive"
sharing="Subscribe" transportation="HLAreliable">
      <parameter dataType="HLAString" name="CommandID"/>
</interactionClass>
<interactionClass dimensions="NA" name="ShutDownComponent"
order="Receive" sharing="Subscribe" transportation="HLAreliable">
      <parameter dataType="HLAString" name="CommandID"/>
</interactionClass>
<interactionClass dimensions="NA" name="SaveComponentState"
order="Receive" sharing="Subscribe" transportation="HLAreliable">
      <parameter dataType="HLAString" name="SaveStateID"/>
      <parameter dataType="HLAString" name="CommandID"/>
</interactionClass>
<interactionClass dimensions="NA" name="PauseComponent"
order="Receive" sharing="Subscribe" transportation="HLAreliable">
      <parameter dataType="HLAString" name="CommandID"/>
</interactionClass>
<interactionClass dimensions="NA" name="ResumeComponent"
order="Receive" sharing="Subscribe" transportation="HLAreliable">
      <parameter dataType="HLAString" name="CommandID"/>
</interactionClass>
<interactionClass dimensions="NA" name="Done" order="Receive"
sharing="Publish" transportation="HLAreliable">
      <parameter dataType="HLAString" name="ComponentName"/>
      <parameter dataType="HLAString" name="InteractionName"/>
      <parameter dataType="HLABoolean" name="Success"/>
      <parameter dataType="HLAString" name="CommandID"/>
</interactionClass>
```

Listing 4-3 Beschreibung der Controlling Interactions im Object Model Template.

Um eine erleichterte Entwicklung und Anbindung neuer Simulatoren an die Co-Simulationsumgebung zu gewährleisten, wurde die HLA Infrastruktur als genutzte

Kommunikationsinfrastruktur der Co-Simulation mit Hilfe verschiedener Techniken unterstützt, die in den folgenden Abschnitten vorgestellt werden.

Die Simulatoren folgen einem, durch die Anforderungen von DistriCT und HLA vorgegebenen, Ablauf.

1.) Initialisierung

    a. Laden der Konfigurations- und Modelldaten (SOM, Parameterkonfiguration und zusätzliche Konfigurationsdateien)

    b. Setzen des Startzustands und der aktuellen Parametersettings

2.) Start

    a. Registrierung bei der RTIG

3.) Ausführung

    a. Warten auf die nächste Zeitzusage

    b. Ausführen der Simulationslogik

4.) Beenden

    a. Abmeldung von der RTIG

    b. Stoppen des Simulationsprogramms

Zusätzlich horchen die Simulatoren während der Ausführung auf Speicher- und Ladeanfragen, bezüglich ihres eigenen Zustands, auf DistriCT.

4.1.4 Unterstützung der Co-Simulationsinfrastruktur

Die Unterstützung der Co-Simulationsinfrastruktur setzt sich zusammen aus der entwickelten HLA Hilfsbibliothek, und Tools und Methoden zur Verwendung des genutzten Datenmodells mit der HLA Infrastruktur.

Die Simulatoren benötigen, wie auch die verwendete Middleware, eine Beschreibung über die Daten, die gesendet und abonniert werden sollen. Dies geschieht über die Angabe der OMT Dateien. Des Weiteren arbeiten die Simulatoren jedoch auf Klassenobjekten des gemeinsamen Datenmodells weswegen eine Zuordnung und Umwandlung dieser durchgeführt werden muss.

Zusätzlich verwenden die HLA Implementierungen XML Dateien, um die OMTs zu beschreiben. Aus diesem Grund werden Techniken aus der Model Driven Architecture (vgl. [BrCW12, SoOt00]) verwendet, um Plattform spezifische OMT Dateien aus dem Plattform unabhängigen Datenmodell zu generieren.

Durch eine Zuordnung von HLA Objekten und Objekten des gemeinsamen Datenmodells kann verhindert werden, dass zwei Versionen desselben Models gepflegt werden müssen.

Im Fall der OMT Dateien wird Thumper - eine „Model to Text"-Transformation basierend auf dem XPand-Projekt der Eclipse Modeling Tools - genutzt (vgl. [DiHS14]).

In Listing 4-4 befindet sich ein Beispiel für eine Thumper Datei, die zur Generierung von OMT Dateien für die verwendeten Simulatoren genutzt wird. Hierfür wird angegeben, auf welchen Daten die OMTs basieren in dem die dementsprechenden Modelldateien des Datenmodells angegeben werden (*#import*). Dazu reicht ein Verweis auf die entsprechende Ecore-Klasse und die verwendeten Attribute dieser. Durch die zusätzliche Angabe, welcher Simulator welche Objekte und Attribute der Systemmodellinstanz dieser veröffentlicht und abonniert, können mittels Thumper die benötigten OMT Dateien für die genannten Simulatoren generiert werden.

```
#import
"platform:/resource/OffshoreSimulation/model/OffshoreLoading.ecore"

Model1516_2010 OffshoreLoadingScenario {
	Simulators {
		Simulator PWSFederate {}
		Simulator LiftSupervisorFederate{}
	}

	HLAObject OffshoreLoadingSimulation.Cargo{
		sharings {
			Publish: PWSFederate
			Subscribe: LiftSupervisorFederate
		}

	OffshoreLoadingSimulation.PhysicalObject.identification;
		OffshoreLoadingSimulation.PhysicalObject.pose;
		OffshoreLoadingSimulation.PhysicalObject.velocity;
	}

	HLAObject OffshoreLoadingSimulation.LiftSupervisor{
		sharings {
			Publish: LiftSupervisorFederate
			Subscribe: PWSFederate
		}


	OffshoreLoadingSimulation.PhysicalObject.identification;
		OffshoreLoadingSimulation.PhysicalObject.pose;
		OffshoreLoadingSimulation.PhysicalObject.velocity;
	}

}
```

Listing 4-4 Beispiel für eine Thumper-Datei zur Generierung von OMTs ausgehend von Ecore-Dateien zum vorgestellten Verladeszenario.

Um die Angaben ausgehend von den verwendeten Modelldateien durchzuführen, existiert ein Wizard der den Co-Simulationsentwickler bei der Generierung der OMT Dateien unterstützt.

Dort lassen sich die importierten Modelldefinitionen angeben, die Föderationseinstellungen wie die Zeitschritte der Federates und das Lookahead. Die Sharing Settings (s. Abbildung 4-4) erlauben die Auswahl der veröffentlichten und abonnierten Daten pro Federate, wobei automatisch eine Mitauswahl geerbter Attribute oder erweiterter Objekte erfolgt. Nach Abschluss der Einstellungen erfolgt die Generierung der in Listing 4-4 beispielhaft angegebenen Thumper Datei.

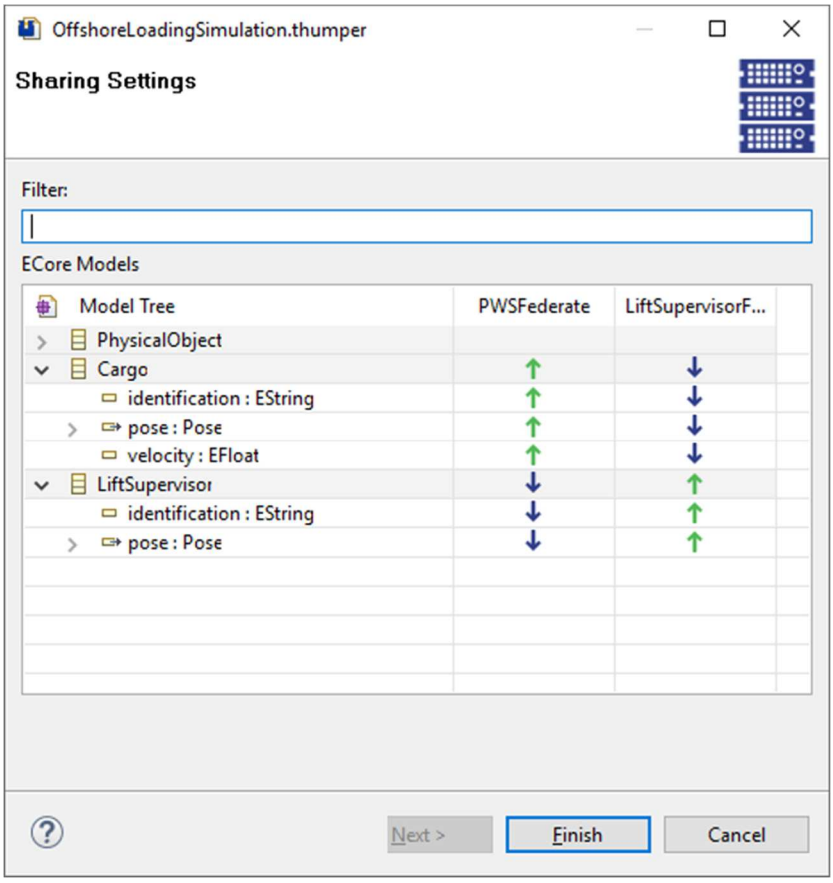


Abbildung 4-4 Darstellung der Sharing Settings zur Auswahl der veröffentlichten und abonnierten Daten pro Federate im Thumper Edit Wizard.

Zusätzlich zur Generierung der OMT Dateien wird die Thumper Datei in einem nächsten Schritt benutzt um dem Simulationsentwickler ein Konfigurationspaket zu erzeugen mit der die Co-Simulation ausgeführt werden kann.

Damit DistriCT und neu erstellte Simulatoren auf den Klassenobjekten des verwendeten Datenmodells arbeiten können, wurde eine Bibliothek entwickelt, welche die relevanten Objekte während der Laufzeit aus den empfangenen HLA-Objekten umwandelt und vice versa.

Die Bibliothek stellt dabei ein Binding zwischen Ecore-Objekten und HLA-Objekten dar und nutzt dafür die von Thumper erstellten OMT Dateien.

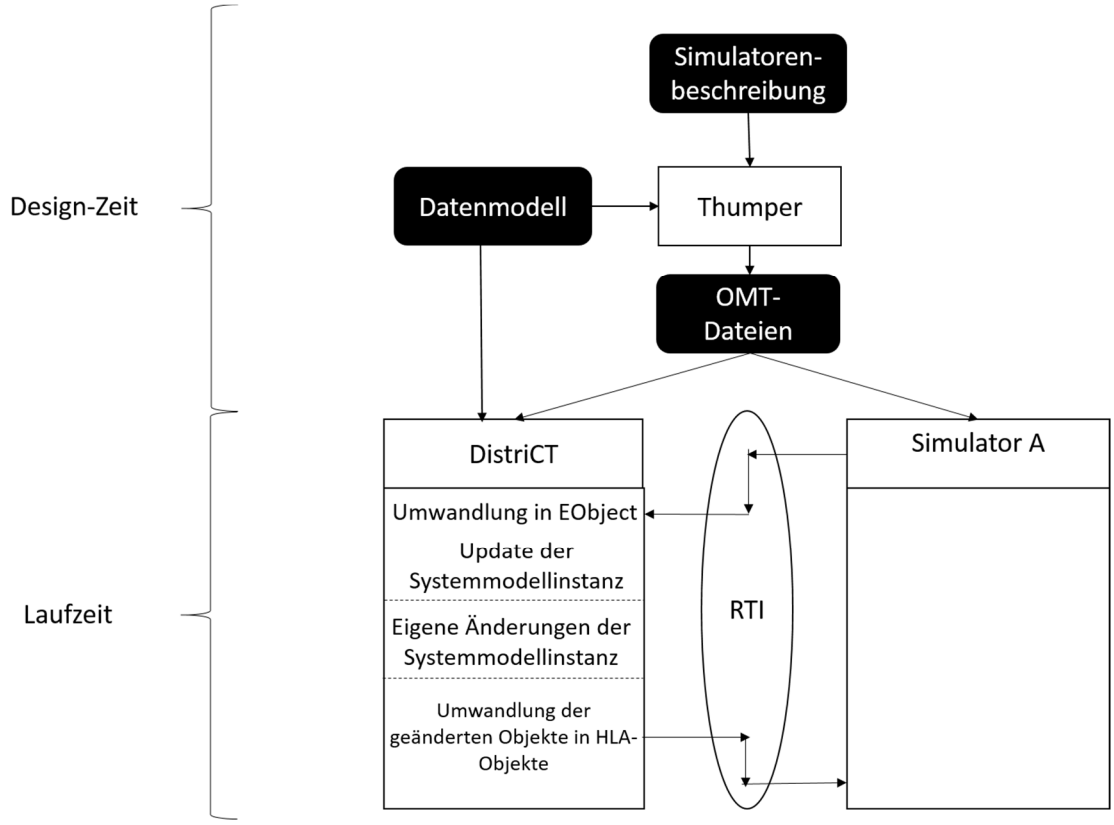


Abbildung 4-5 Zusammenarbeit der verschiedenen Komponenten zur Design- und Laufzeit. Es wird aufgezeigt welche Daten zwischen Thumper (zur Beschreibung der auszutauschenden Daten), Simulatoren und dem DistriCT-Controller ausgetauscht werden. Zusätzlich ist der Ablauf des Data-Binding zur Laufzeit der Co-Simulation sichtbar.

In Abbildung 4-5 ist der vereinfachte Ablauf und die Verbindung zwischen Thumper, dem verwendeten Datenmodell und der Hilfsbibliothek, die das Binding zwischen Datenmodell und HLA-Objekten übernimmt, zu sehen. Dabei ist zu erwähnen, dass eine weitere Verwendung mit Standardkonformen externen Simulatoren möglich ist.

Hierfür sind zwei Methoden in der Hilfsbibliothek integriert. Die erste beinhaltet das Object Handling. Das Management der Objekte findet dabei über zwei zu aktualisierende Listen statt. Die erste Liste beinhaltet alle simulationsrelevanten Objekte, wobei jedem Objekt eine eindeutige ID zugeordnet ist welche durch die RTI vergeben wird (*ObjectInstanceHandle*). Die zweite Liste beinhaltet die IDs aller Objekte, die geändert wurden und die den anderen Co-Simulationsteilnehmern aktualisiert zur Verfügung gestellt werden sollen. Erhält ein Simulator die Zusage der RTIG die nächsten Zeitschritte zu simulieren, wird mittels der Listen überprüft, welche Objekte geändert wurden und bereitet diese für den Versand über HLA vor. Die zweite Methode beschreibt das Object Mapping. Bei der Ausführung des

Simulators liegen die Objekte in Form von EObjects auf Basis des gemeinsamen Datenmodels, das auf dem Eclipse Modeling Framework (EMF) basiert, vor. Für die Kommunikation über HLA werden jedoch Objekte verwendet, die auf den beschriebenen OMT-Dateien basieren (HLAObject). Um den Wechsel zwischen den Objekttypen EObject und HLAObject zu realisieren, stellt das Objekt Mapping vier Funktionen bereit.

1. Erstellung eines HLAObject auf Basis eines EObject
2. Erstellung eines EObject auf Basis eines HLAObject
3. Aktualisierung der Attribute eines bereits erstellten EObjects anhand eines HLAObject
4. Aktualisierung der Attribute eines bereits erstellten HLAObject anhand eines EObject

Die notwendigen Informationen für den Wechsel zwischen den Objekttypen werden durch die durch Thumper vorgegebenen qualifizierten Namen der HLAObjekte beschrieben. Auf Basis dieser werden beim Objekt Mapping zur Laufzeit die Objekte erstellt. Über eine zusätzliche Angabe der identifizierenden Attribute eines Systemelements können die empfangenen und umgewandelten EObjects der verwendeten Systemmodellinstanz zugeordnet werden.

4.2 Konfiguration der Co-Simulation

Durch die Beschreibung der zu verwendenden Co-Simulation (vgl. Abschnitt 4.1.2), inklusive der Modellierung der genutzten Simulationskomponente lässt sich die Co-Simulation für das DistriCT-Framework konfigurieren, um einen Simulationsverlauf erstellen zu können.

In Abbildung 4-6 ist ein Überblick über die Realisierung des Zugriffs auf die Co-Simulation zu sehen, sowie die Anbindung des Konfigurationskonzepts an dieses. Dabei stellt DistriCT eine Schnittstelle bereit über die eine Erweiterbarkeit der Funktionalität gegeben wird (Command Provider). Jeder Command Provider registriert die zur Verfügung gestellten Kommandos bei DistriCT über den Command Holder.

In einem Simulationsplan, der sowohl einen Prozess- als auch Datenflussgraphen darstellt, lassen sich die Kommandos, die im Command Holder hinterlegt sind nutzen, um den Ablauf einer Co-Simulation zu beschreiben und die Zustände einer Co-Simulation zu bewerten.

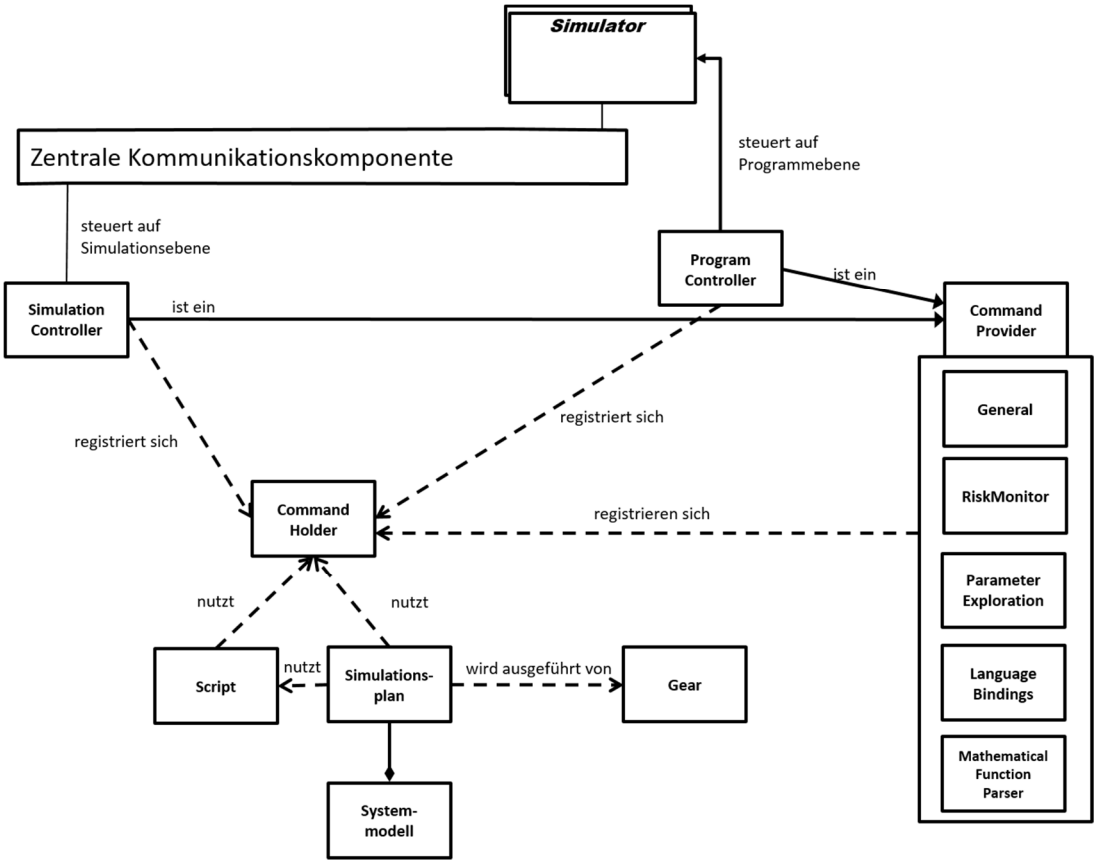


Abbildung 4-6 Übersicht über die Realisierung des Zugriffs auf die Co-Simulation.

Ein Simulationsplan folgt dabei immer den gleichen drei Phasen (s. Abbildung 4-7). Die erste Phase besteht aus der Verteilung und Konfiguration der Programme und einer ggf. vorhandenen Kommunikationsmiddleware über die Programmkontrolle. Danach werden die Simulatoren - im Sinne der mit der Kommunikationsmiddleware verbundenen Programme - über die Simulatorkontrolle gestartet. Hieraufhin beginnt die eigentliche Analyse, Beobachtung und Bewertung der Simulationssituationen. Da die beschriebene Logik der Simulation Zugriff auf die Simulatorkontrolle hat, kann diese entscheiden, ob eine Situation gespeichert werden soll, ein Simulationslauf aus einer gespeicherten Situation gestartet wird oder die Simulation gestoppt wird. Dabei hat auch die Simulatorkontrolle Zugriff auf die darüber liegende Phase (Programmkontrolle) und kann neben den Simulatoren auch die dahinterliegenden Programme auf den verteilten Rechnern stoppen.

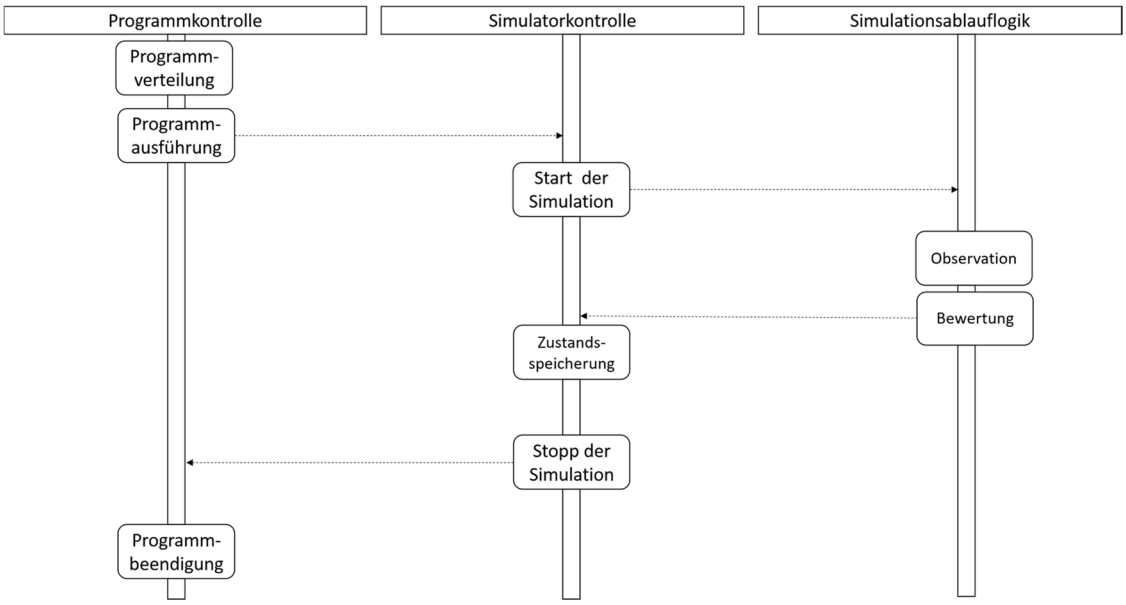


Abbildung 4-7 Ablaufsequenz einer Simulationsplaninstanz.

Die Konfiguration der Simulationsläufe über einen gerichteten Graphen wurde dabei gewählt, um das Erstellen und Kombinieren verschiedener Verfahren einfach und schnell durchführen zu können und ggf. vorher definierte Teilgraphen wiederzuverwenden. Der Graph besteht dabei aus einer beliebigen Menge Knoten, die über Kanten miteinander verbunden sind. Beim Simulationsplan dürfen Ausgangsports (Outports) in beliebig vielen Kanten vorkommen, während Eingangsports (Inports) nur in einer Kante vorhanden sein können.

Die Knoten des Simulationsplans können aus Inports, Outports und Parametern bestehen, deren Funktionalität im Folgenden näher beschrieben wird.

Inport Über Inports erhält ein Knoten Daten, wie beispielsweise Elemente einer Systemmodellinstanz auf denen der Knoten weitere Berechnungen durchführt. Inports können dabei als optional deklariert werden. Ist ein Inport optional, muss kein Datum anliegen, damit der Knoten ausgeführt wird. Ein Beispiel für einen optionalen Inport ist ein „Reset“-Inport, der bei Aktivierung die internen Knotendaten zum Ende eines Simulationslaufs zurücksetzt.

Outport Über die Outports verschickt der Knoten die verarbeiteten oder neu berechneten Daten an andere verbundene Knoten.

Parameter Über Parameter lassen sich Änderungen an verwendeten Variablen innerhalb der Knoten Ausführung vornehmen, so dass von außen auf die Funktionalität eines Knoten Einfluss genommen werden kann, ohne den Programmcode zu ändern.

Knoten können jedoch auch als einfache Prozessknoten verwendet werden, die ausgeführt werden, sobald eine eingehende Kante mit einem Datum belegt ist. In Abbildung 4-8 ist ein ausgeführter Simulationsplan in der DistriCT Umgebung zu sehen. Die Knoten innerhalb des Simulationsplans stellen genutzte, zur Verfügung stehende Kommandos dar, welche teilweise nur nacheinander abgearbeitet werden aber zum Teil auch einen Datenaustausch nutzen.

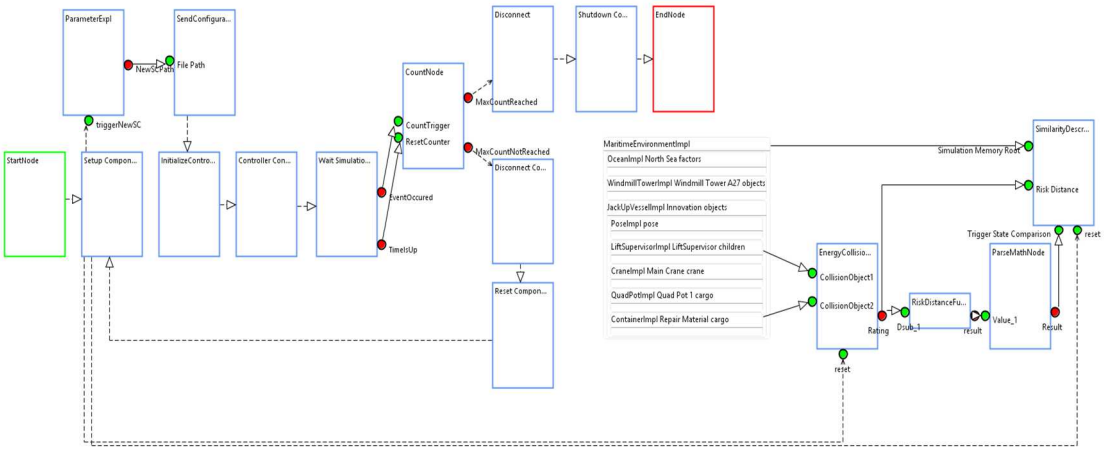


Abbildung 4-8 Beispielhafter Simulationsplan des DistriCT-Frameworks.

Zusätzlich existieren DistriCT-Skripte (s. Abbildung 4-9), die ebenfalls den Ablauf einer Simulation beschreiben können aber bei denen kein Datenaustausch zwischen den Kommandos beschrieben wird. Über diese lassen sich einfache wiederkehrende Aufgaben, wie das Aufsetzen und Beenden einer Co-Simulation auslagern, was eine Wiederverwendung vereinfacht und eine übersichtlichere Darstellung des Simulationsplans erlaubt. In Abbildung 4-9 ist ein Setup-Skript zu sehen, in welchem die Verteilung der benötigten Simulationsprogramme ausgeführt wird. Dabei werden zunächst globale Parameter definiert, welche in den im Skript aufgerufenen Befehlen (*Commands)* genutzt werden können (z.B. die IP-Adresse des Clients auf dem die HLA Runtime Infrastructure ausgeführt werden soll).

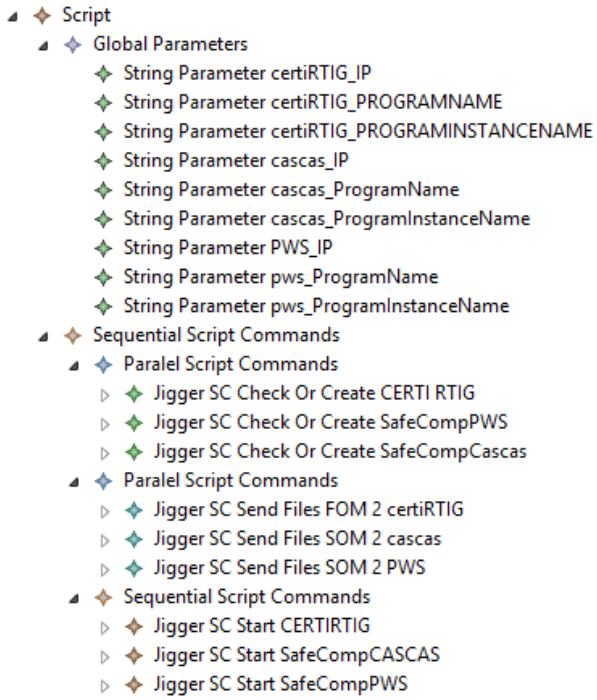


Abbildung 4-9 Beispiel für ein DistriCT-Skript, in dem die benötigten Komponenten und die beteiligten Simulatoren verteilt und gestartet werden.

Zur Ausführung der DistriCT-Skripte wird der Quartz Scheduler[14] verwendet. Dieser ermöglicht unter anderem eine einfache Einbindung parallel ausgeführter *Commands*.

Um den Vorgang der Konfiguration zu beschleunigen und sich wiederholende Arbeiten wie die Skript-Erstellung zu erleichtern wurden ein geführter Assistent in DistriCT integriert (s. Abbildung 4-10), der die Erstellung einer Co-Simulation mit ihren spezifischen Anforderungen erleichtert. Dieser greift dabei auf die generierte Thumper Datei zu und fasst die Sammlung der benötigten Informationen, Konfigurationsdateien und die Zuordnung der ausführenden Programme zusammen. Abschließend werden mit Hilfe der angegebenen Informationen die Ausführungsskripte und das Simulationsplan-Gerüst generiert.

[14] QUARTZ Job Scheduler http://quartz-scheduler.org [zuletzt abgerufen am 03.03.2017]

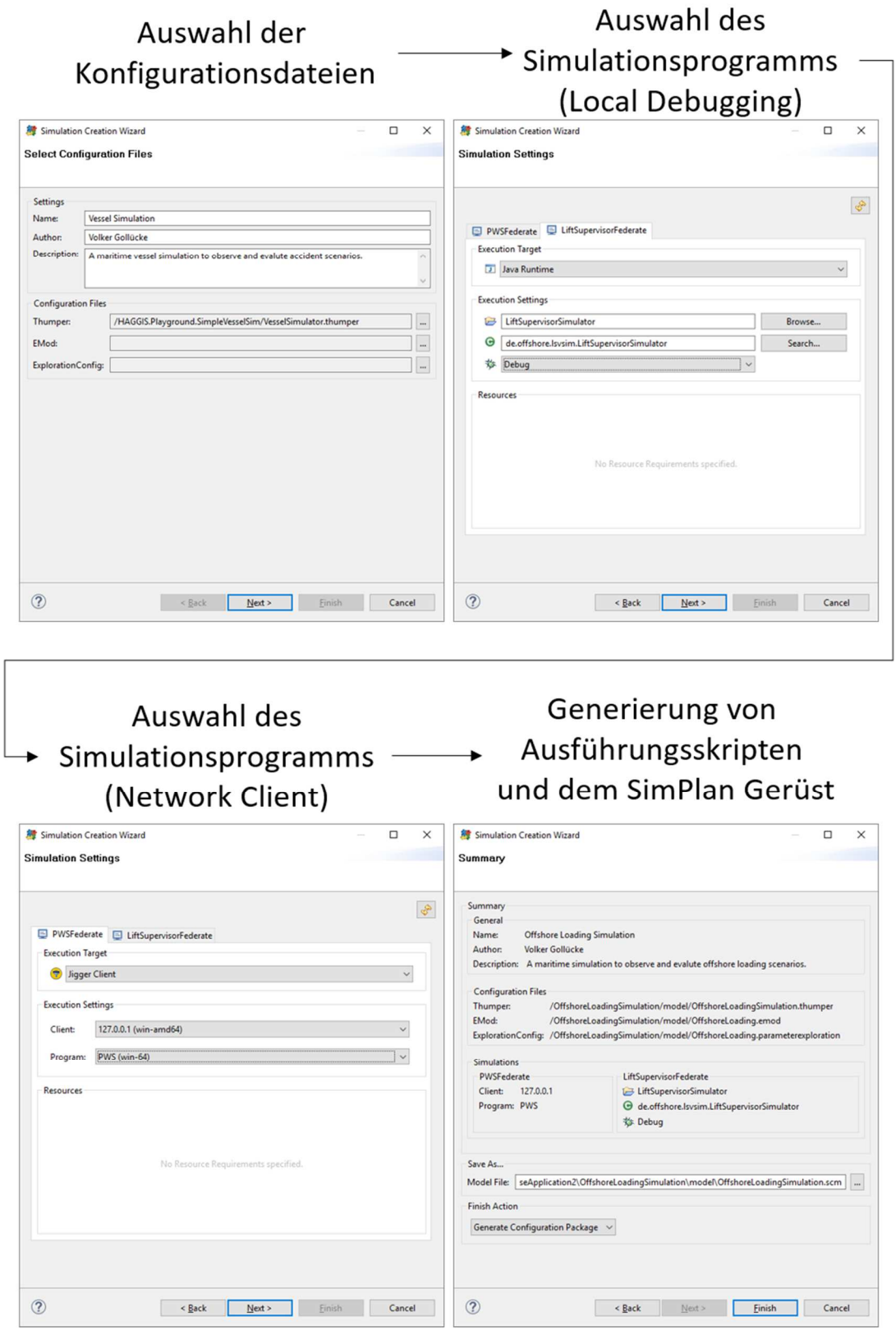


Abbildung 4-10 Aufbau des Simulation Creation Wizard – Ausgehend von der erstellten Thumper Datei, der verwendeten Systemmodellinstanz und der Parameterexplorationskonfiguration werden die Ausführungsskripte und das Simulationsplan-Gerüst erzeugt.

4.3 Steuern der Co-Simulation

Der Simulationsplan ermöglicht eine Steuerung auf Programm, Simulations- und Explorationsebene (vgl. Anforderung [A_C3]). Dies bedeutet, dass sowohl die Verteilung der Programme auf Rechner im Netzwerk, wie auch das Starten und Stoppen dieser Programme mit ihren jeweiligen benötigten Kommandozeilenparametern eingerichtet werden kann. Die Steuerung auf Simulationsebene besagt, dass die gestarteten und mit der RTIG verbundenen Programme gesteuert werden können. Dies findet über die Verwendung der Controlling Interactions (s. Tabelle 3-4 u. Listing 4-3) statt, mit denen unter anderem der Co-Simulationszustand gespeichert und geladen werden kann. Die Steuerung auf Explorationsebene beschreibt, wie eine systematische Exploration von Attributen innerhalb einer Co-Simulation durchgeführt werden kann.

In Abbildung 4-11 werden die drei unterschiedlichen Typen der Steuerung, einer von DistriCT kontrollierten Co-Simulation innerhalb eines Simulationsplans genutzt. Zunächst wird eine erstellte Explorationskonfiguration (s. Abschnitt 3.3.4) an einen Parameterexplorationsknoten übergeben. In diesem kann die Explorationstechnik ausgewählt werden, wie zum Beispiel: Depth First oder Breadth First. Sobald an dem *Request New Setting*-Inport (vgl. Abbildung 4-11) ein Datum anliegt, wird über den System Model Instance-Outport der Pfad zur nächsten Parameterkonfiguration, hinterlegt als Systemmodel-Datei, übermittelt. Die Systemmodel-Datei stellt dabei eine serialisierte Form der Systemmodellinstanz dar. In dem hier beschriebenen Beispiel wird die Parameterkonfiguration dann als Datei mittels der Programmsteuerung an das über die Einstellungen angegebene Simulationsprogramm übermittelt. Im Sinne der Anforderung [A_S5] sollten verwendete Simulatoren per Datei oder über ein Update der Systemmodellinstanz über neue Parameter informiert werden können. Im nächsten Schritt findet eine Steuerung auf Simulationsebene statt. Der Start Simulation-Knoten startet die Co-Simulation mittels der vorgestellten Start-Funktion der Controlling Interactions.

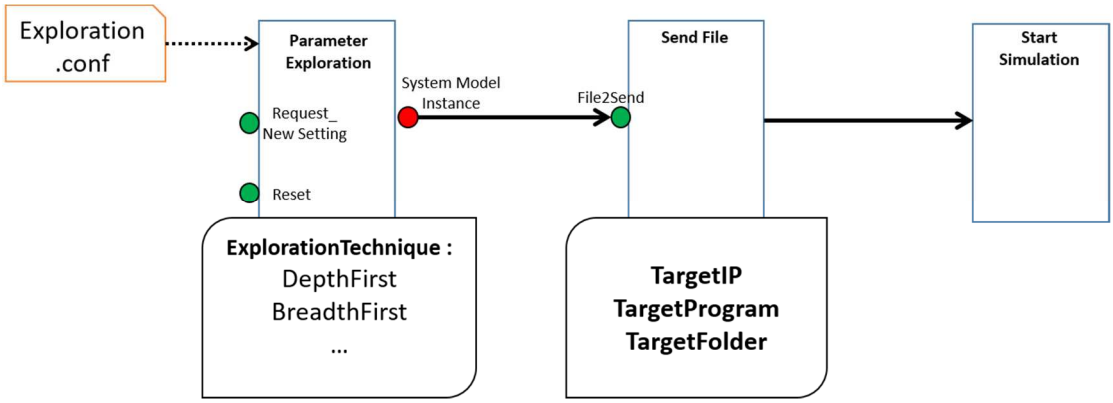


Abbildung 4-11 Ausschnitt eines Simulationsplans der die drei Steuerungsarten des DistriCT-Frameworks verwendet

4.3.1 Zusätzliche Kontrolle über die verteilten Programme

Eine zusätzliche Steuerung der verteilten Simulatoren und deren ausgeführter Programme besteht in der manuellen Kontrolle über alle laufenden Programminstanzen, deren Ausgaben und schlussendlich deren direkter Kontrolle. In Abbildung 4-12 ist ein Screenshot des Programm-Controller Parts zu sehen über den nachvollzogen werden kann, welche Systeme im Netzwerk und welche Simulations-Programme zur Verteilung und Ausführung zur Verfügung stehen. Dabei lassen sich über das Interface alle notwendigen Informationen zu dem Status der verteilten Systeme und Programme abfragen, unter anderem auch ob eine Programminstanz auf einem System korrekt oder fehlerhaft ausgeführt wird.

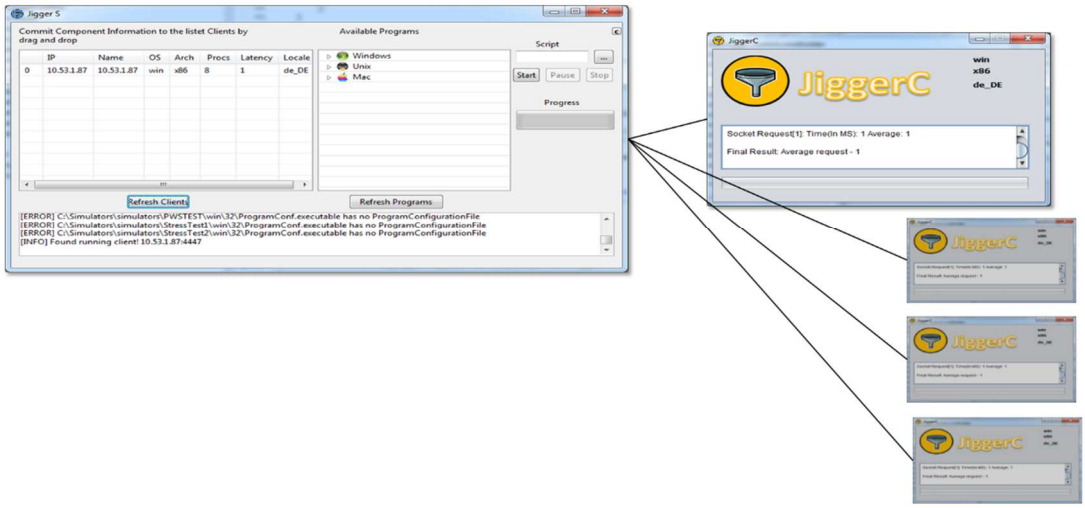


Abbildung 4-12 Benutzeroberfläche des Servers und verschiedener-Clients zur Programmkontrolle.

Um die Betrachtung des Zustands der ausgeführten Programme zu ermöglichen, wurde in der Programmausführungskomponente eine Weiterleitung des Programminstanz-Status eingerichtet, der aus Konsolenausgaben, des ausgewählten Programms und Screenshots des ausführenden Systems besteht und an die laufende DistriCT-Instanz weitergeleitet wird (s. Abbildung 4-13). Um Probleme oder Einstellungen auf den verteilten Systemen durchzuführen, besteht zusätzlich die Möglichkeit, mittels Fernsteuerung auf das entfernte System, sofern das entfernte System dies zulässt, zuzugreifen.

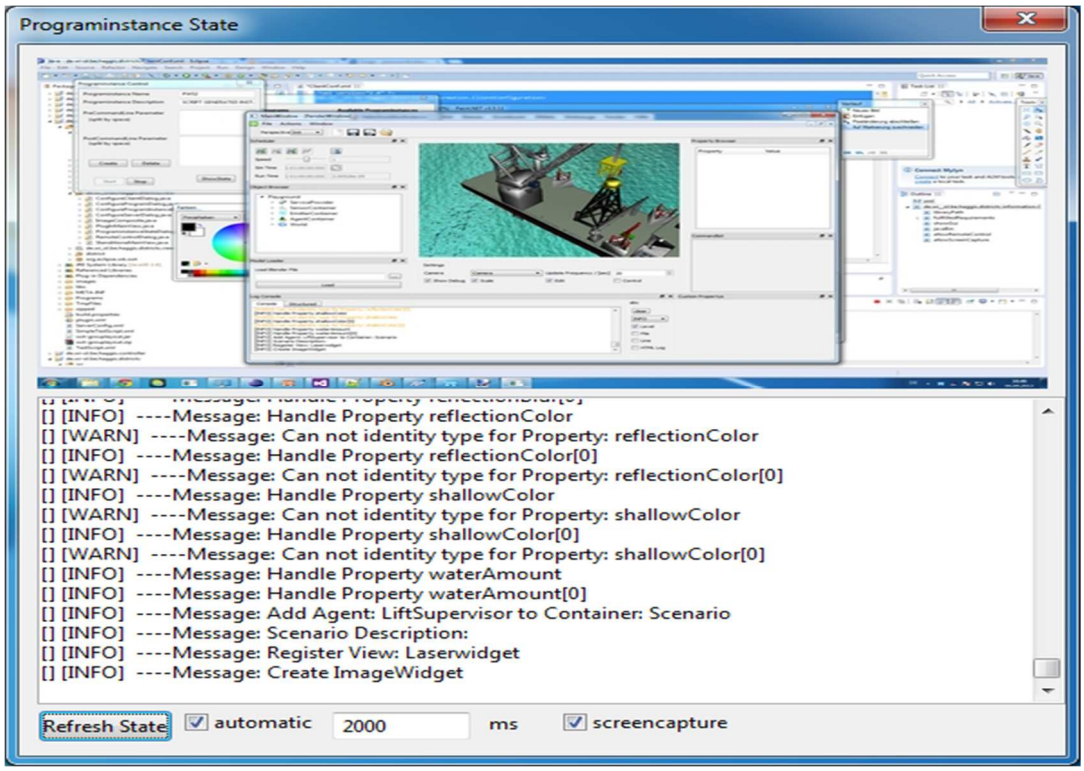


Abbildung 4-13 Beispiel für die Weiterleitung des Status einer Programminstanz.

4.4 Analyse der Co-Simulation

Neben den kommunizierenden Simulatoren besitzt auch DistriCT eine Repräsentation der Systemmodellinstanz. Damit der Controller immer ein aktuelles Bild des Systems hat, registriert dieser sich automatisch als Abonnent (”Subscriber”) für alle kommunizierten Objekte der Co-Simulation (vgl. Anforderung [A_C1]).

Die Verwendung der Analyse kann in Abbildung 4-14 gesehen werden. Dabei findet automatisch eine Weiterleitung der Attribute an die Analysekomponenten statt, sobald ein Update der Systemmodellinstanz durchgeführt wird.

Für die Analyse lassen sich sowohl komplexe Objekte als auch Attribute verwenden, wobei bewusst auf eine streng getypte Verwendung der Ports verzichtet wurde. Es lassen sich dabei eingehende Daten weiterverarbeiten (zur Berechnung), um Zähler zu aktivieren (unabhängig vom eingehenden Objekttyp), oder um diese als Auslöser für die Ausführung angeschlossener Knoten zu verwenden. Es muss dabei durch die Benennung der Ports und die Beschreibung der Analysekomponenten dem Anwender klar sein, welche Datentypen von welchem Port erwartet werden.

Die Sammlung von Analysekomponenten lassen sich leicht um neue Komponenten erweitern und innerhalb von DistriCT nutzen. Dabei folgen die Analysekomponenten aber auch die im nächsten Abschnitt vorgestellten Steuerungskomponenten dem gleichen Aufbau (s. Listing 4-5). Bei diesen wird die zu implementierende *setUp*-Methode genutzt, um die Eingangs- und Ausgangsports zu erstellen. Die *execute*-Methode beinhaltet die eigentliche Ausführung der Knotenalgorithmen, wobei dort die Verarbeitung der eingegangenen Daten durchgeführt werden kann.

Um die Daten einer Analyse zu verarbeiten, existiert bereits eine Auswahl an Basis-Funktionen. Hierzu gehören sowohl generelle Funktionen zur Entscheidungsfindung aber auch Funktionen zur Berechnung komplizierterer Zusammenhänge, um die Komplexität bei der Einbindung und Entwicklung weiterer Simulatoren und Komponenten zu verringern.

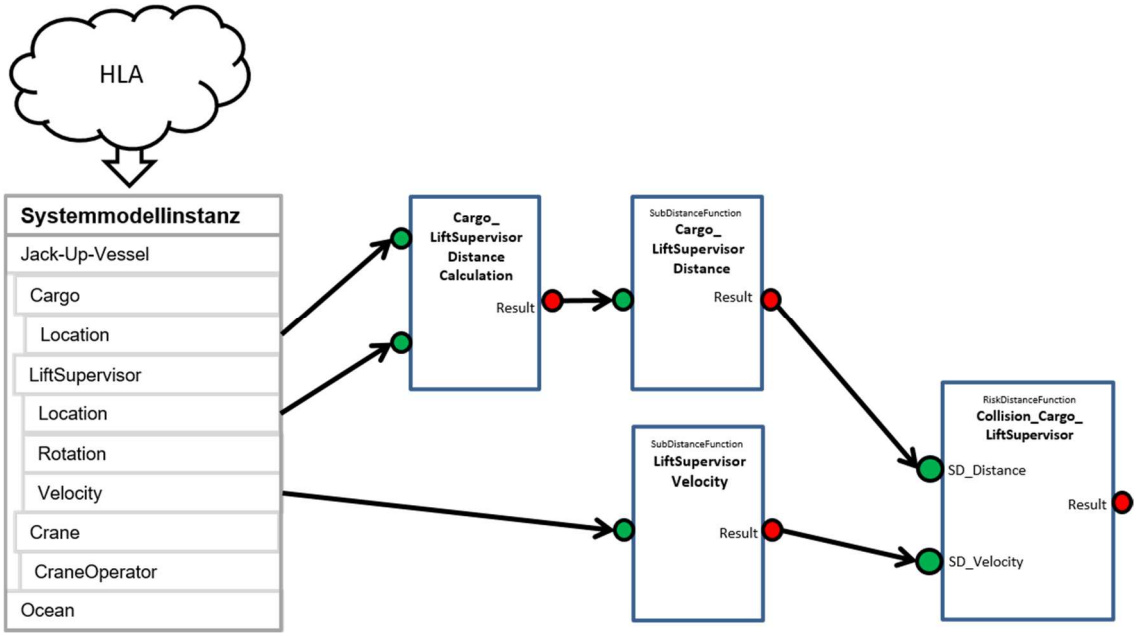


Abbildung 4-14 Beispiel für Nutzung der Systemmodellinstanz zur Analyse der Co-Simulation im Simulationsplan.

Generelle Funktionen

Zu den generellen Funktionen die im Simulationsplan verwendet werden können, gehören unter anderem Zählfunktionen, Komponenten zur Skript-Ausführung und Entscheidungsknoten. Ein wichtiger zu nennender Bestandteil ist zudem die Logger-Funktionalität (vgl. Anforderung [A_C2]), die um Zeitstempel erweiterte Einträge aller, vom Ersteller des Simulationsplans geforderten kommunizierten Daten als CSV-Datei zur späteren weiteren Analyse anlegen kann.

Parser für mathematische Funktionen

Um mathematische Funktionen auszuwerten, existiert eine auf, der mvel-Programmbibliothek[15] aufsetzende Komponente. An dieser kann eine dynamische Anzahl von Inports, über die unterschiedliche Werte übermittelbar sind, angefügt werden, die in einer angegebenen Funktion zur Berechnung genutzt und das Ergebnis ausgegeben werden kann. Die mvel-Bibliothek bietet dabei zusätzlich den Vorteil direkt auf Java-Objekten arbeiten zu können, so auch Methodenaufrufe auf einem eingehenden Objekt durchzuführen.

[15] MVFLEX Expression Language https://github.com/mvel/mvel [zuletzt abgerufen am 03.03.2017]

Unterstützung von Python Skripten

Um kompliziertere Zusammenhänge zu berechnen, lassen sich ebenfalls Python Skripte einbinden. Auch dieser Knoten, der auf Jython[16] basiert, kann über beliebig viele Eingänge verfügen, welche innerhalb des auszuführenden Python-Skriptes als Variablen verwendet werden können.

```
public class Node extends INode {
    @Override
    public void execute() {
        Object input1 = getInport("inport1").waitForInput();
        Object input2 = getInport("inport2").waitForInput();
        //An dieser Position können die Daten verarbeitet werden
        getOutport("outport1").send(result);
    }

    @Override
    public setUp(){
        addInport("inport1");
        addInport("inport2");
        getInport("inport2").setOptional(true);
        addOutport("outport1");
    }
```

Listing 4-5 Beispiel Aufbau der Knotenklassen die zur Steuerung und Analyse der Simulation genutzt werden können.

4.5 Risikodistanzberechnung innerhalb des Distributed Controlling Toolkits

Neben den Basis-Funktionen wurde auch die eigene Methodik zur Risikodistanzberechnung als Analysekomponente umgesetzt. Diese besteht dabei aus den Elementen *RiskDistanceFunction*, *SubDistanceFunction* und *RiskStateDB*.

Zur Berechnung der Risikodistanz zu einer Kollision zwischen Ladeoffizier und Ladung wird zunächst der *RiskDistanceFunction*-Knoten erstellt mit dem Fehlerbaum zum Kollisionsrisiko als Input.

[16] The Jython Project http://www.jython.org/ [zuletzt abgerufen am 03.03.2017]

Um die Fehlerbäume in für den Simulationsplan verwendbare Knoten-Elemente umzuwandeln, wird ein automatisches Mapping durchgeführt, welches im folgenden Abschnitt vorgestellt wird.

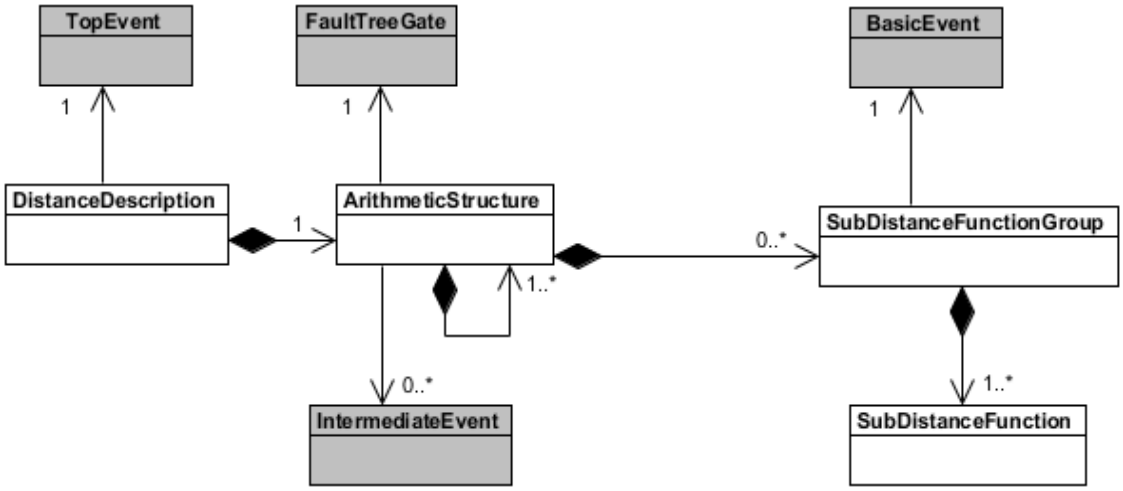


Abbildung 4-15 Zuordnung der Fehlerbaumelemente zu den Elementen der Distanzfunktionsstruktur. Elemente des Fehlerbaums sind grau eingefärbt, während Elemente der Distanzbeschreibung weiß dargestellt sind.

In Abbildung 4-15 ist die Zuordnung von Fehlerbaumelementen zu den verschiedenen Elementen der entwickelten Risikodistanzbeschreibung zu sehen. Dabei sind die Elemente des Fehlerbaums grau eingefärbt, während die Elemente der Distanzbeschreibung weiß dargestellt sind. Intermediate Events also Zwischenelemente und Fehlerbaum Gates wie ”Und” und ”Oder” werden für den Aufbau der arithmetischen Strukturen innerhalb der Distanzbeschreibung genutzt, während Basic Events, also die Blattelemente des Fehlerbaums, für den Aufbau von Subdistanzfunktionsgruppen genutzt werden. Die Distanzbeschreibung selber kennt das Top Event-Element des Fehlerbaums, welches den Namen und die textuelle Beschreibung des Risikos liefert.

In Abbildung 4-16 ist ein Ausschnitt des Simulationsplans zu sehen in dem die Berechnung der Risikodistanz einer Kollision zwischen Ladung und Ladeoffizier erfolgt. Es wird auf das Location-Attribut der Ladung (Cargo) und des Ladeoffiziers (Lift Supervisor), sowie das Velocity-Attribut des Ladeoffiziers in der Systemmodellinstanz zugegriffen.

Zur Extraktion und weiteren Berechnung der Analysedaten wurden Python Skripte und mathematische Funktionen angegeben. Die aktuelle Geschwindigkeit des Ladeoffiziers kann dabei gleich an die Subdistanzfunktionsberechnung weitergeleitet werden während mittels

der Location-Eigenschaft des Ladeoffiziers und der Ladung zunächst die Entfernung zwischen diesen berechnet wird, bevor eine Weiterleitung an die entsprechende Subdistanzfunktion erfolgt. Die Ergebnisse der Subdistanzfunktionen werden dann an den aus dem hinterlegten Fehlerbaum generierten Risikodistanzfunktionsknoten zur Kollision weitergeleitet.

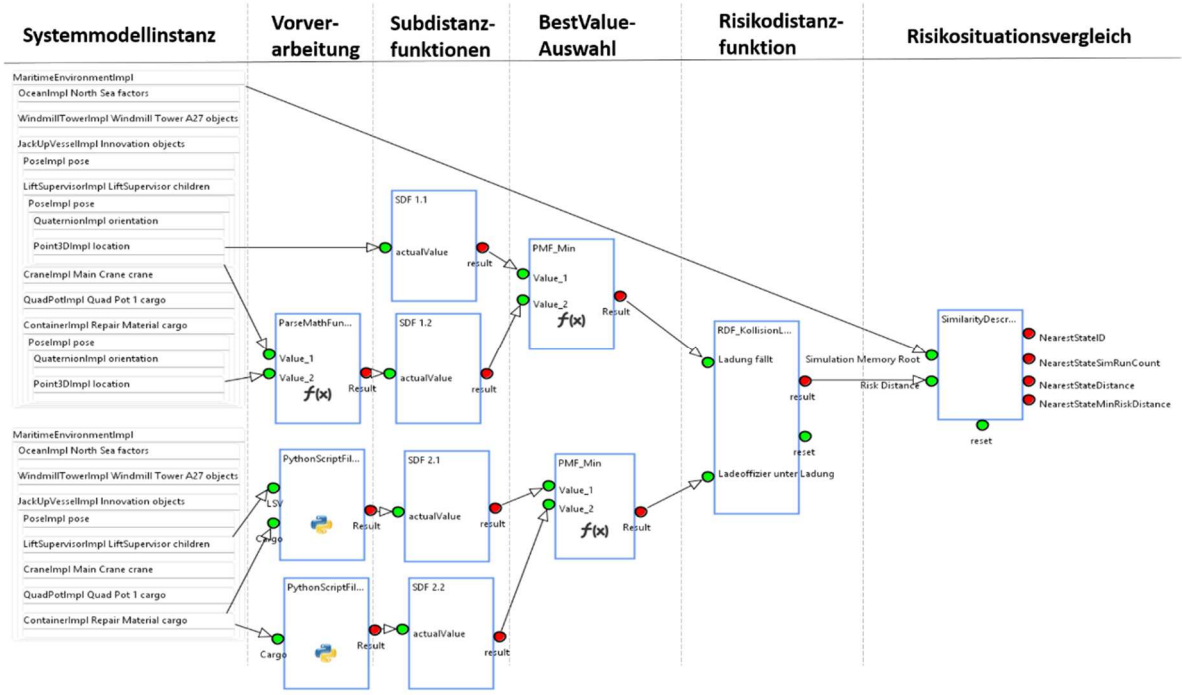


Abbildung 4-16 Ausschnitt eines Simulationsplans. In diesem wird gezeigt wie die Berechnung der Distanz zur Kollision zwischen Ladung und Ladeoffizier an Bord eines Jack-Up-Vessel konfiguriert wurde.

4.6 Zusammenfassung

In diesem Kapitel wurde die Beschreibung der Co-Simulation eingeführt. Diese unterteilte sich in die Beschreibung der verwendeten Infrastruktur, den Aufbau der verwendeten Co-Simulationen, die Beschreibung der Simulationskomponenten und der entwickelten Hilfsbibliothek, die als Binding zwischen HLAObjekten und Objekten des Datenmodells innerhalb von DistriCT genutzt wird.

Aufbauend auf der Definition der Co-Simulation und deren Simulatoren wurde im Anschluss auf die Entwicklung des DistriCT Framework eingegangen.

Hierbei wurde auf die Erfüllung der übergeordneten Anforderungen mittels des entwickelten DistriCT Frameworks eingegangen. Dabei wurde die Konfiguration der Co-Simulation vorgestellt, die über einen gerichteten Graphen (Simulationsplan) durchgeführt werden kann und Mittel zur Kontrolle der Programme, Simulatoren und Simulationslogik zur Verfügung stellt.

Anschließend wurde auf die Durchführung der Analyse der Co-Simulation eingegangen in dem gezeigt wurde, wie mittels DistriCT auf die aktuelle Instanz des Systemmodells zugegriffen wird, um unter anderem die Risikodistanzfunktion anzubinden. Zusätzlich wurde gezeigt, welche Basis-Funktionalitäten für die Analyse in DistriCT zur Verfügung stehen (z.B. Parser für mathematische Funktionen und Python Skripte).

Im folgenden Abschnitt erfolgte dann die Erläuterung wie DistriCT genutzt werden kann, um die Co-Simulation aber auch einzelne Programme zu steuern. Dabei wurde darauf eingegangen, wie Ergebnisse der (Risiko-)Analyse in die Steuerung einfließen können, um so das Speichern und Laden von Simulationszuständen zu triggern und damit die Importance Splitting/RESTART Methode zu unterstützen. Zusätzlich wurde die Kontrolle der verteilten Programme vorgestellt.

Abschließend wurde die Umsetzung des eigenen Ansatzes mit DistriCT präsentiert, wobei ein komplexer Simulationsplan gezeigt wurde, in dem die Risikodistanz ausgehend von dem zu analysierenden Beispielszenario vorgestellt wurde (s. Beispiel 1, S. 39). Die DistriCT-Perspektive mit den wichtigsten Elementen zur Risikobewertung ist in Abbildung 4-17 zu sehen.

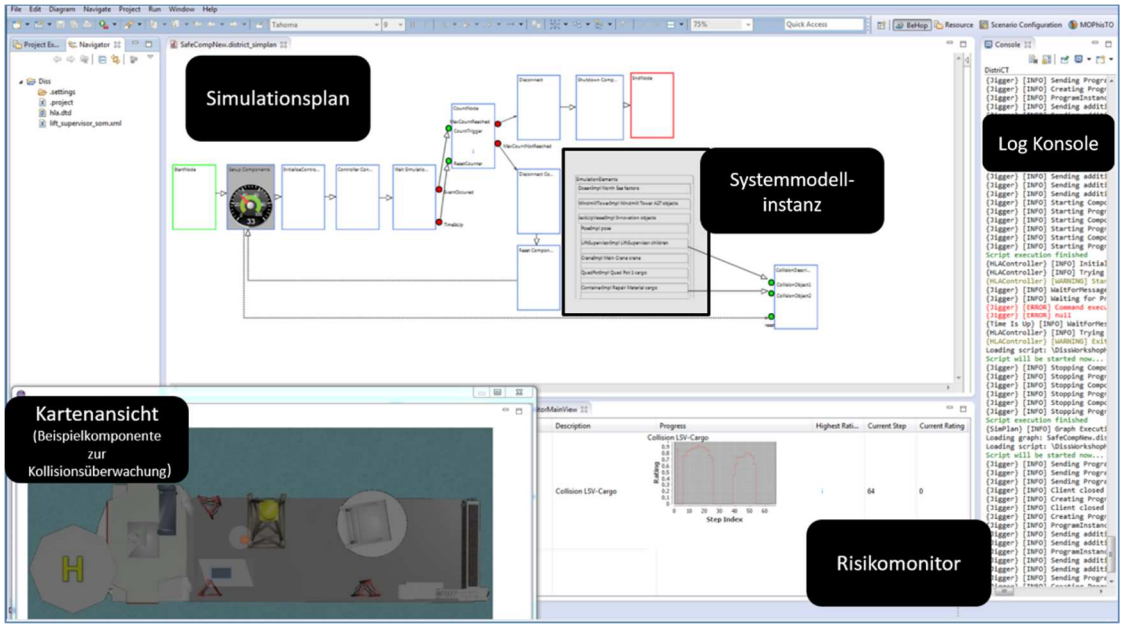


Abbildung 4-17 Die DistriCT–Perspektive mit den Elementen Simulationsplan, Systemmodellinstanz und Risikomonitor in der Eclipse Runtime Umgebung.

5 Evaluation der entwickelten Methodik und Simulationsunterstützung

Um die Beantwortung der wissenschaftlichen Fragestellung zu vervollständigen wurde einer Evaluation der Anforderungen und der entwickelten Methodik nachgekommen. Dafür wurde die Überprüfung der einzelnen Bestandteile der Methodik hinsichtlich ihrer Anwendbarkeit und Funktionalität durchgeführt. Dabei wurde die Berücksichtigung der ermittelten Anforderungen innerhalb des entwickelten Frameworks DistriCT betrachtet.

Das gewählte Szenario für die erste von drei Überprüfungen basiert dabei auf den vorgestellten Beispielelementen zur Analyse der Kollisionsgefahr zwischen einer transportierten Ladung und einem Ladeoffizier an Bord eines Errichter Schiffes (Jack-Up-Vessel) während einer Verladeoperation (s. Beispiel 1 aus Kapitel 3). Dieses wird genutzt, um festzustellen ob mit Hilfe der formulierten Methodik eine Erstellung von Risikodistanzfunktionen möglich ist und diese zur Bewertung von Simulationsverläufen angewendet werden kann. Zusätzlich wurde mit diesem ersten Szenario kontrolliert, ob die entwickelte Methodik verwendbar ist um eine Überprüfung von Fehlerbäumen hinsichtlich ihrer Unvollständigkeit durchzuführen.

Da die erste Evaluation noch nicht die Frage nach der Korrelation zwischen bewerteter und wirklicher Risikonähe der risikoreichen Situationen mit einschließt wurde dieses zusätzlich anhand eines weiteren Szenarios durchgeführt, um diesen Teil der wissenschaftlichen Fragestellung zu beantworten. Im zweiten Szenario, in dem ein reales Unfallszenario betrachtet wurde dessen Verlauf bekannt war, wurde eine maritime Verkehrssimulation verwendet. Das Wissen über den Verlauf und die, mittels Expertenwissen über die maritime Domäne, erstellten Risikodistanzfunktionen wurden dabei auf eine Korrelation hin untersucht.

Die letzte Evaluation befasste sich mit der Steuerung von Simulationen mittels der erstellten Risikodistanzfunktionen. Hierfür wurde ein Szenario gewählt in dem das Auftreten einer sehr unwahrscheinlichen Kollision zwischen zwei Schiffen naiv und geführt mittels Risikodistanzfunktionen untersucht und gegenübergestellt wurde.

5.1 Evaluation der DistriCT-Funktionalität und Vollständigkeitsüberprüfung von Fehlerbäumen anhand eines Verladeszenarios an Bord eines Jack-Up-Vessel

In Kapitel 3 „*Bewertung von Simulationszuständen in Co-Simulationen zur beschleunigten simulativen Analyse*“ wurde das Beispiel einer maritimen Ladeoperation vorgestellt, das auch im Kontext dieser Evaluation verwendet wurde, um die Anwendbarkeit des eigenen Ansatzes zu evaluieren.

Im Rahmen des Experiments wurde, der aus Prozessmodell und annotierten Gefahren generierte Fehlerbaum verwendet und auf eine Unvollständigkeit überprüft. Alle Ereignisse die während der Ausführung einer Simulation auftreten werden gespeichert. Dies erlaubt es einem Domänenexperten zu überprüfen ob zuvor identifizierte Gefahren auftreten nachdem die ebenfalls zuvor definierten Ursachen eingetreten sind. Ist dies nicht der Fall, also tritt eine Gefahr auf, ohne dass die im Fehlerbaum angegebenen Ursachen eingetreten sind, besteht die Möglichkeit, dass eine Ursache die zu der relevanten Gefahr führt übersehen wurde oder der Fehlerbaum auf andere Weise fehlerhaft ist [GPLG14].

5.1.1 Aufbau des Evaluationsexperiments

Um übersehene Ursachen für Gefahren mittels Simulation zu finden wird DistriCT als Simulationssetup und Kontrolltool verwendet. Dafür wird die Parameterexploration zusammen mit einer Beschreibung der benötigten Simulatoren verwendet, um einen Simulationsplan zu generieren. Dieser beschreibt dabei die konfigurierte Sequenz von Simulationsläufen und kann auf die beschriebene Systemmodellinstanz (s. Beispiel 2 - S. 41) zugreifen.

In dem betrachteten Experiment werden zwei Simulatoren verwendet. Einer ist der LiftSupervisor-Simulator, der für die Bewegung des Ladeoffiziers auf dem Schiffsdeck verantwortlich ist. Der Ladeoffizier verwendet verschiedene Pfade, um die Ladung zu beobachten, welche eine freie Sicht auf die Ladung und den Kranführer gewährleisten. Der simulierte Ladeoffizier wählt zufällig welchen Pfad er nimmt. Der zweite Simulator ist der Physical World Simulator (PWS), welcher in dem Experiment die Umwelt und die Kranbewegung kontrolliert. Die PWS wird weiterhin verwendet, um ein 3D-Modell des Szenarios und der physikalischen Effekte und Umwelteinflüsse bereitzustellen. Physikalische Effekte sind zum Beispiel die Kollision von Objekten oder Soft Body Effekte, die benutzt werden, um das Schwingen des Kranseils zu realisieren. Umwelteinflüsse sind

Partikeleffekte wie Regen oder Schnee, die gegebenenfalls die Sicht beeinflussen. Die PWS besitzt zusätzlich eine integrierte Visualisierung welche für Testzwecke und eine manuelle Beobachtung sowie zusätzlich zur Erstellung der Screenshots vom Experiment genutzt werden konnte. Eine detailliertere Beschreibung der PWS befindet sich in den Arbeiten von Schweigert et al. [ScDH12] und Läsche et al. [LäGH13].

In dem verwendeten Simulationsplan (s. Abbildung 5-1) werden zunächst das Runtime Infrastructure Gateway (RTIG), der LiftSupervisor-Simulator und die PWS auf einem konfigurierten System aufgesetzt. Dabei werden alle benötigten Konfigurationsdateien übertragen und die Komponenten in der richtigen Reihenfolge gestartet.

Danach wird der DistriCT-Controller verbunden und ein Startsignal an alle Federates gesendet (s. Abbildung 5-1:1). Während die Simulation ausgeführt wird, horcht DistriCT auf das Auftreten von Gefahren und die abgelaufene Simulationszeit (s. Abbildung 5-1:2). Eine Logger Komponente schreibt die Informationen über die aufgetretenen Ursachen und Gefahren in eine Datei und informiert einen Zähler (s. Abbildung 5-1:3), sobald eine Gefahr auftritt oder eine vorgegebene maximale Simulationszeit überschritten wurde. Ist eine maximale Anzahl von Simulationsläufen hintereinander ohne Auftreten einer Gefahr gezählt worden, wird die Simulation gestoppt (s. Abbildung 5-1:5). Der Zähler wird wieder auf 0 Simulationsläufe in einer Reihe ohne Auftreten einer Gefahr zurückgesetzt, sobald dieser über das weitere Auftreten einer Gefahr informiert wird. Auf der anderen Seite wird die Simulation zurückgesetzt und neu gestartet, wenn die maximale Anzahl von sequentiell ausgeführten Simulationsläufen ohne Auftreten einer Gefahr nicht erreicht wird (s. Abbildung 5-1:4).

Für die Evaluation des Ansatzes wurde die Risikodistanzfunktion genutzt, um herauszufinden ob die untersuchte Gefahr eingetreten ist (vgl. Anforderung [A_D4]). In dieser Evaluation wurden Eintrittsraum und Relevanzraum angegeben. Es wurde jedoch noch nicht auf die Nähe zur Gefahr eingegangen, sondern nur das Auftreten einer Gefahr untersucht.

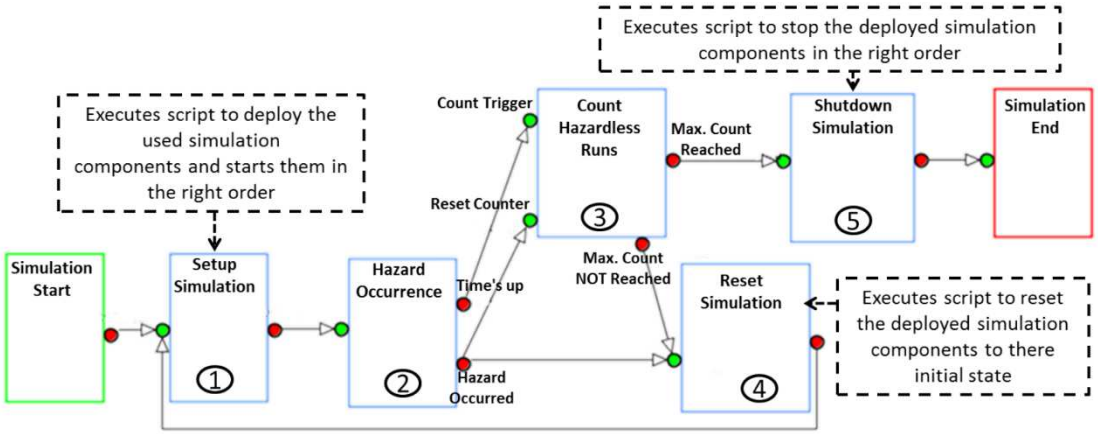


Abbildung 5-1 Darstellung eines Simulationsplans, der den Ablauf der Simulation für das Kollisionsszenario beschreibt.

5.1.2 Ablauf des Experiments

Am Beginn der Operation (s. Abbildung 5-2:1) steht der Ladeoffizier neben der Ladung und wartet auf den Beginn der Ladeoperation. Während der Ladeoffizier auf dem Weg zum nächsten Beobachtungspunkt ist, beginnt der Kranführer mit dem Anheben der Ladung. Im Beispielfall hebt der Kranführer die Ladung nicht hoch genug an und beginnt die Rotation der Kranbasis zu früh (s. Abbildung 5-2:2). In Abbildung 5-2:3 ist die fatale Entwicklung der Situation, die als nicht kritisch erkannt wurde, zu sehen. Die Ladung befindet sich zu niedrig über dem Schiffsdeck und bewegt sich auf die Position des Ladeoffiziers zu. Abbildung 5-2:4 zeigt den Unfall bei der der Ladeoffizier von der Ladung getroffen wird. Die Schwierigkeit für den Kranführer die Gesamtsituation komplett zu überblicken kann in den vier kleineren Bildern an den Ecken der großen Bilder gesehen werden. Diese zeigen die schlechte Sicht aus dem Cockpit des Krans während der Operation.

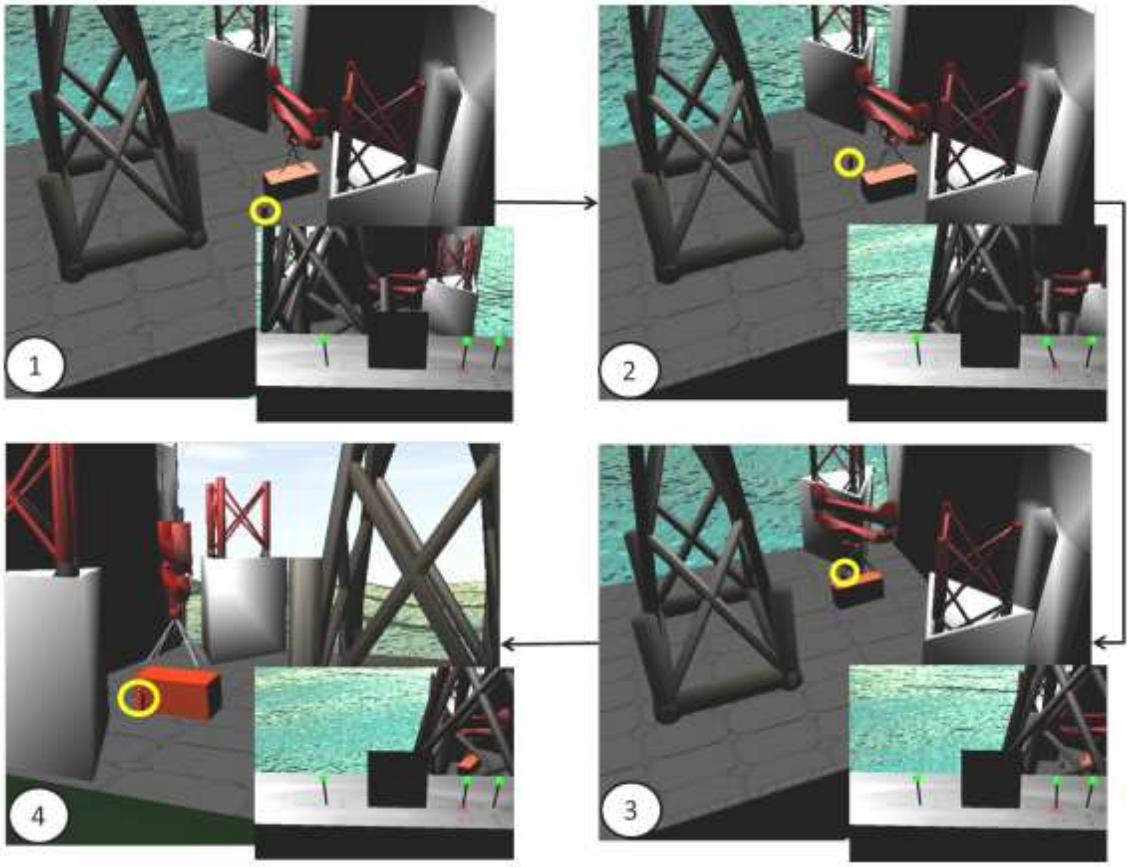


Abbildung 5-2 Fall einer Kollision beim Verladeszenario. Große Bilder: Verschiedene Ansichten des Ladebereichs kurz vor eintreten des Unfalls (1-3) und während des Auftretens der Kollision (4). Kleine Bilder: Sicht aus dem Krancockpit aufgenommen zur selben Zeit wie die korrespondierenden großen Bilder.

Bei der Untersuchung der Simulationsverläufe zeigte sich, dass es Ursachen gab, die zu der Gefahr führten die nicht berücksichtigt wurden bei der Erstellung des ersten Fehlerbaums (s. Abbildung 5-3 a). Die Analyse hat ergeben, dass die Gefahr auch Eintritt, wenn sich der Ladeoffizier nicht unterhalb der Ladung befindet (s. Abbildung 5-3 b:1). Zusätzlich haben die Ergebnisse gezeigt das die Rotation des Krans während die Ladung angehoben wird ebenfalls zur Gefahr führt (s. Abbildung 5-3 b:2). Diese Ergebnisse wurden genutzt, um den Fehlerbaum manuell anzupassen (s. Abbildung 5-3 b).

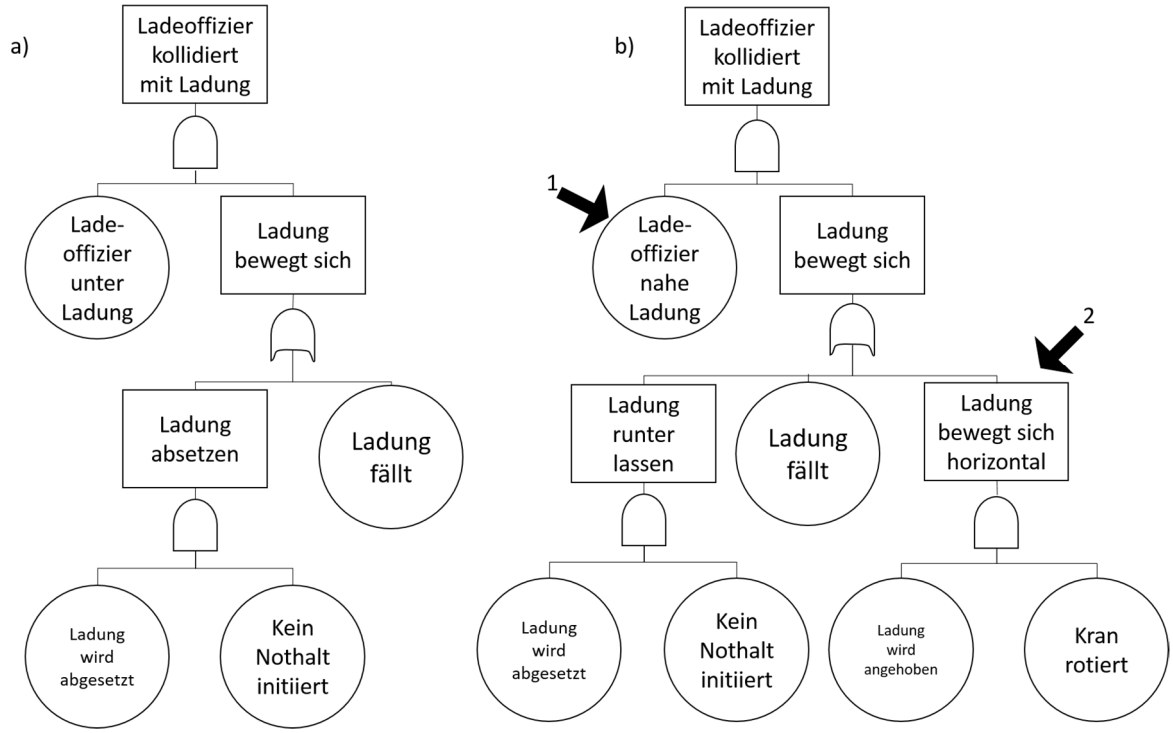


Abbildung 5-3 Die erstellten Fehlerbäume zum Verladeszenarios. a) Fehlerbaum vor der simulativen Analyse b) Durch die Ergebnisse der Simulationsanalyse angepasster Fehlerbaum. 1: Angepasstes existierendes Fehlerbaumelement 2: Neue Ursachen-Kombination entdeckt durch die Simulation.

5.1.3 Auswertung des Experimentes

Das DistriCT-Framework wurde evaluiert in dem die Anzahl an Simulationsläufen gemessen wurde, die benötigt waren, um die fehlenden Ursachen zu finden, die zu der übergeordneten Gefahr (Ladung kollidiert mit Ladeoffizier) führen.

Anhand einer empirischen Mittelung über 50 Iterationen (50-mal wurden nacheinander neue Simulationen aufgesetzt und gestartet) wurde festgestellt, dass durchschnittlich 6,4 Simulationsläufe benötigt wurden, um die nicht beschriebenen oder fehlenden Ursachen innerhalb eines Simulationslaufs zu beobachten.

In diesem Experiment wurde zu Demonstrationszwecken bewusst eine hohe Auftrittswahrscheinlichkeit gewählt, um die Evaluation der DistriCT Funktionalität durchzuführen und die Anwendung der Distanzfunktionen häufiger beobachten zu können.

Als Ergebnis des Experimentes ergab sich, dass das Anlegen der Distanzfunktionen im entwickelten Framework ohne Probleme möglich war. Zusätzlich konnte gezeigt

werden, dass die Anbindung der Simulatoren und die Kontrolle auf Programmebene mit verhältnismäßig wenig Aufwand funktionieren. Die letzte in dieser Evaluation zu kontrollierende Frage, nach der Überprüfung von Fehlerbäumen hinsichtlich ihrer Unvollständigkeit, konnte ebenfalls positiv beantwortet werden in dem mit durch die Nutzung des eigenen Ansatzes und Frameworks das Finden von nicht betrachteten Ursachen unterstützt wurde.

5.2 Evaluation der Erstellung und korrekten Korrelation der Risikodistanzfunktion am Beispiel des realen Unfalls zwischen dem Frachtschiff Marti Princess und dem Containerschiff Renate Schulte

Im folgenden Evaluationsabschnitt wird gezeigt, dass mit Hilfe der entwickelten Methodik Risikodistanzfunktionen erstellt werden können, mit denen sich der richtige Trend in Richtung der untersuchten Gefahr erkennen lässt.

Dafür wurde ein historisches Szenario ausgewählt dessen Ausgang und Verlauf bekannt ist. Aus dem verwendeten Szenario und dem Hergang des Unfalls wurden dabei die Ursachen abgeleitet welche zum Unfall führten und untersucht ob Unfallbericht und Bewertung durch die Risikodistanzfunktion korrelieren. Das heißt ob die Risikodistanzfunktion den richtigen Trend, also das über die Simulationszeit sich nähernde Risiko der Kollision, im Verlauf der Risikodistanzwerte zu erkennen ist (vgl. Anforderung [A_A2]).

5.2.1 Szenario Beschreibung

Im Juni 2009 waren das Frachtschiff Marti Princess und das Containerschiff Renate Schulte an einem Unfall in der Ägäis in der Nähe der Insel Bozcaada beteiligt. Eine Annäherung an die Kurse der involvierten Schiffe ist in Abbildung 5-4 zu sehen. Laut Unfallbericht (vgl. [Bund12]) war die Sichtweite zwischen fünf und sieben Meilen während eine ruhige See ohne Wellengang und eine Windgeschwindigkeit von 12-19 km/h (Windstärke 3) angegeben wurde.

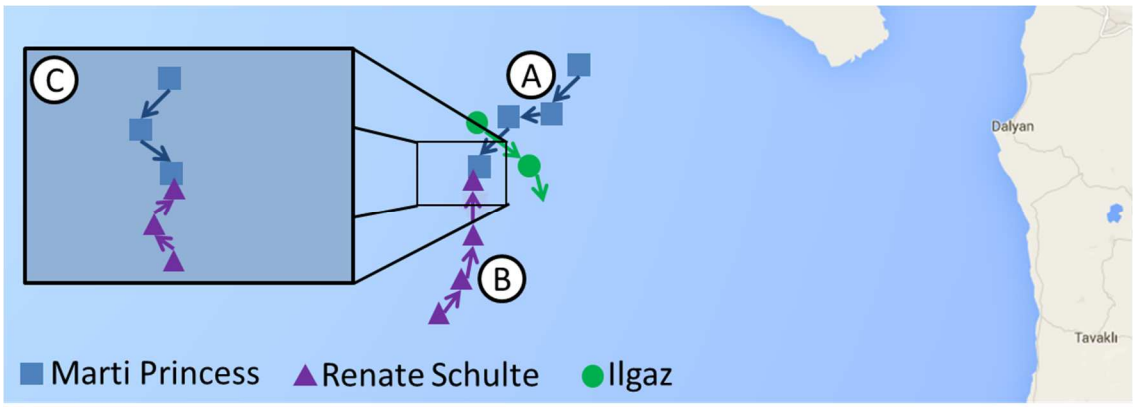


Abbildung 5-4 Kurse der drei Unfallteilnehmer Marti Princess, Renate Schulte und der Ilgaz.

Auf der Marti Princess hatten der wachhabende Offizier und der Kapitän eine Anpassung ihres Kurses vorgenommen, um hinter dem Heck der Ilgaz, wie vorgeschrieben durch die internationalen Kollisionsverhütungsregeln (vgl. [Orga03]), zu passieren und danach wieder zurück auf den ursprünglichen Kurs zu kommen (s. Abbildung 5-4:A). An Bord der Renate Schulte beobachtete der wachhabende Offizier die Ilgaz während das Schiff seiner Route weiter folgte (s. Abbildung 5-4:B).

Die Marti Princess wurde auf dem Radar vom wachhabenden Offizier der Renate Schulte bemerkt, als diese zwischen vier und fünf Seemeilen entfernt war. Als die Schiffe noch ca. zwei Seemeilen entfernt waren versuchte die Renate Schulte eine Funkverbindung mit der Marti Princess aufzubauen, um ihre Absichten zu melden. Es wurde jedoch keine Antwort der Marti Princess gehört. Als beide Schiffe schließlich begannen ausweichende Maßnahmen zu ergreifen (s. Abbildung 5-4:C) war nicht mehr genügend Zeit und Abstand vorhanden, um mit einem Manöver des letzten Augenblickes eine Kollision zu vermeiden (s. Abbildung 5-5) (vgl. [Mari12]). Um 22:10 Uhr kam es zur Kollision zwischen der Marti Princess und Renate Schulte.

Abbildung 5-5 Die drei am Unfall beteiligten Schiffe (Renate Schulte, Marti Princess, Ilgaz) und ein Bild des Unfallschadens(vgl. [Bund12]).

5.2.2 Aufbau des Evaluationsexperiments

Eine maritime Verkehrssimulation, wie die Maritime Traffic Simulation (MTS) (vgl. [HGBS15]), kann für die Analyse von maritimen Systemen hinsichtlich der Effizienz und Sicherheit des weltweiten Schiffsverkehrs genutzt werden. Die Simulationsumgebung wird unter anderem dazu genutzt, um das Verhalten mehrerer Schiffe zu implementieren, auszuführen und beobachten zu können.

Um das vorgestellte Kollisionsszenario zwischen den Schiffen Marti Princess und Renate Schulte zu evaluieren wurde die MTS so konfiguriert, dass sie die betroffenen Schiffe mit ihren physikalischen Dynamik- und Verhaltensmodellen simulieren kann. Die Ilgaz nimmt eine passive Rolle innerhalb des Kollisionsszenarios ein. Daher konnte diese mit einem einfachen Verhaltens- und Physikmodell konfiguriert werden. Das einfache Modell folgt dabei einem vorgegebenen Pfad dessen Wegpunkte aus dem Unfallbericht abgeleitet werden konnten. Die kollidierenden Schiffe (Marti Princess und Renate Schulte) benötigen, auf der anderen Seite, ein komplexeres Verhaltensmodell, um neben dem einfachen Folgen der Routenwegpunkte auch auf die anderen beteiligten Schiffe reagieren zu können. Ein Beispiel hierfür ist das Umfahren der Ilgaz durch die Marti Princess, welches innerhalb des Verhaltens der Marti Princess umgesetzt wurde.

Innerhalb der MTS werden die benötigten Sensordaten generiert, die zur Risikodistanzberechnung benötigten werden. Im Falle des vorgestellten Unfallszenarios wurden die beteiligten Schiffe mit einem AIS (Automatic Identification System) Sensor ausgestattet (vgl. [Schw16, SGHB14]), welche die symbolischen Messungen für die Risikoauswertung generieren. Der hierbei angewendete Ansatz wurde dabei in der wissenschaftlichen Veröffentlichung „Virtual test bed for maritime safety assessment“ vorgestellt und in dieser Arbeit zur weiteren Evaluation verwendet (vgl. [HGBS15])

5.2.3 Ablauf des Experimentes

Im Kontext des präsentierten Unfallszenarios wird das entwickelte DistriCT-Tool genutzt, um die verwendeten Simulatoren aufzusetzen. Diese setzen sich dabei aus den drei betrachteten Schiffssimulatoren der MTS und dem DistriCT-Controller, der die Risikoauswertung vornimmt, zusammen.

Die Distanzfunktion, die zur Berechnung des Kollisionsrisikos angewendet wird, basiert auf den geteilten Systemeigenschaften, wie beispielsweise den aktuellen Routen, der Geschwindigkeit, der Rotationsgeschwindigkeit und der aktuellen Position der beteiligten Schiffe.

Um die Risikodistanzfunktion aufzubauen, welche zur Bewertung des Risikos einer Kollision zwischen der Renate Schulte und der Marti Princess genutzt wird, wurde in diesem Fall zunächst eine Literaturrecherche durchgeführt, um mögliche Ursachen für eine Kollision und deren logische Verknüpfungen zu ermitteln.

Als Grundlagen für die Recherche wurden die Ausweich- und Fahrregeln der Kollisionsverhütungsregeln [Orga03] und die Arbeiten „Schuldverteilung bei Schiffskollisionen“ [Bier04], „Collisions and their Causes“ [CaBr02] sowie „Managing Collision Avoidance at Sea: A Practical Guide“ [LePa07] zum Aufbau des zugrunde liegenden Fehlerbaums und zur Erstellung der Subdistanzfunktionen betrachtet.

Dabei wurde zunächst eine Klassifikation der zu untersuchenden unterschiedlichen Kollisionsszenarien durchgeführt, wobei folgende drei identifiziert wurden:

1. Eine Schiffskollision als Ergebnis eines Überholvorganges,
2. eine Frontalkollision
3. und eine Schiffskollision während des Kreuzens.

Im nächsten Schritt wurden begünstigende Faktoren für Schiffskollisionen und Fehler, die zu diesen führen, aus den genannten Arbeiten abgeleitet, in denen Schiffskollisionen und deren Urteilssprüche inklusive deren Schuldaufteilungen betrachtet wurden. Ein gekürzter Ausschnitt dieser Analyse ist in Abbildung 5-6 dargestellt. Dabei fand eine Unterteilung in generelle begünstigende Faktoren, wie eine hohe Verkehrsdichte und schlechte Umweltbedingungen, Manövrierprobleme, wie eine nicht vermiedene Nahbereichslage und eine mangelhafte Informationslage (Beispiel: eine verminderte Sicht), statt.

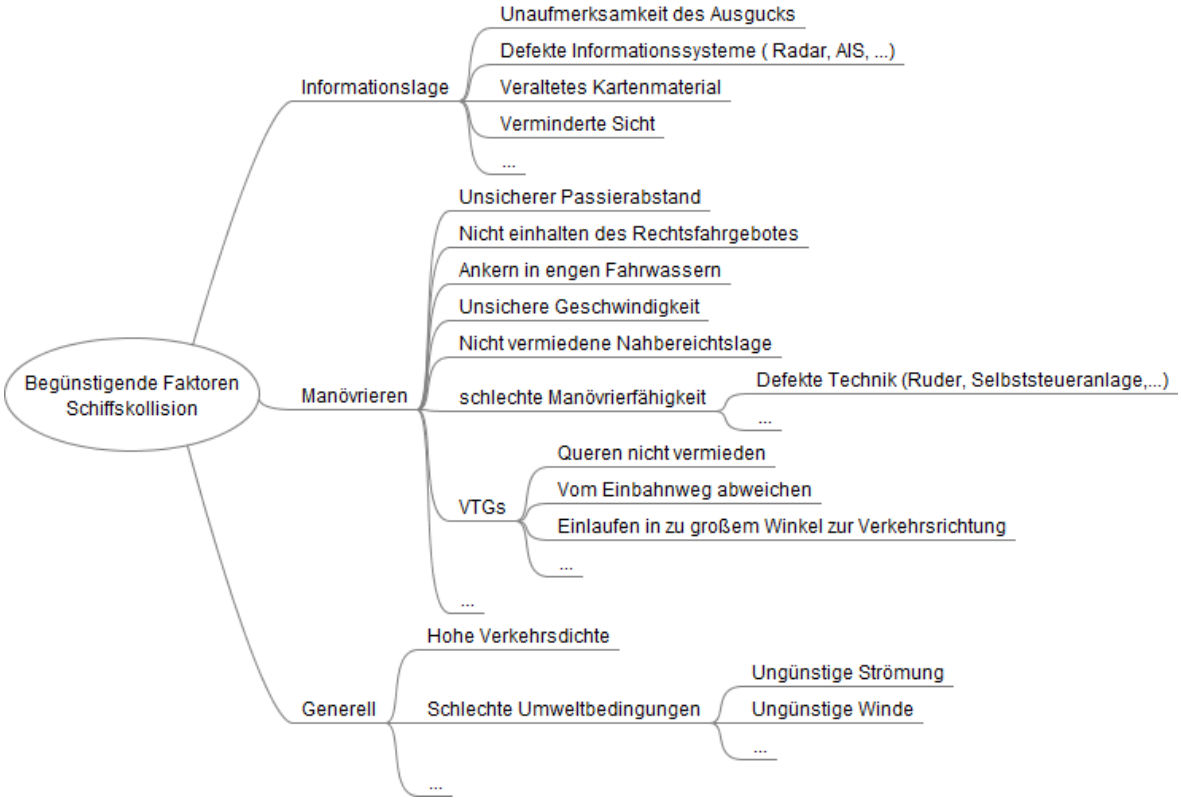


Abbildung 5-6 Ausschnitt der begünstigenden Faktoren für Schiffskollisionen. Abgeleitet aus den Quellen [Orga03, Bier04, CaBr02, LePa07].

Die Fehler und vorgenommene Klassifikation betrachtend wurde der in der Risikodistanzfunktion zu verwendende Fehlerbaum erstellt, welcher in Abbildung 5-7 dargestellt ist.

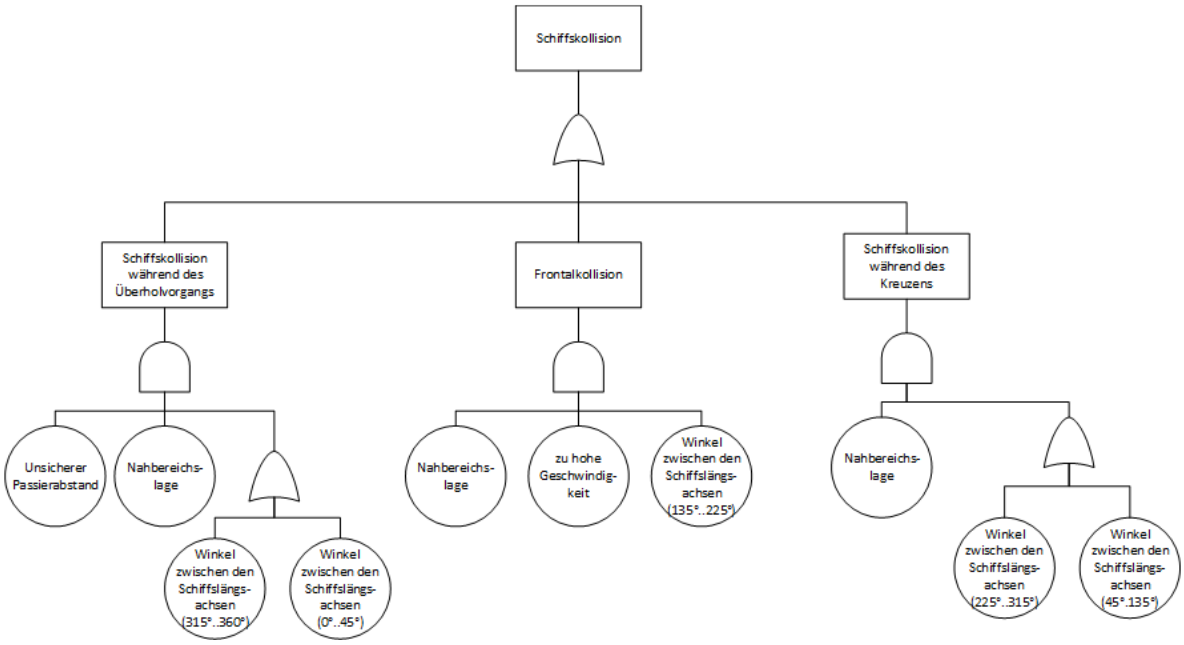


Abbildung 5-7 Fehlerbaum zur Bewertung von Kollisionsrisiken zwischen zwei beteiligten Schiffen.

Aufbauend auf der in Abschnitt 3.2.3 beschriebenen Herleitung der Subdistanzfunktionen wurden für die Basic Events des Fehlerbaums die in der Simulation bewerteten Indikatoren beschrieben.

Als Beispiel für die Bewertung der Nahbereichslage wurde als Indikator die kürzeste Distanz zwischen zwei Schiffsdomänen eingeführt, welche im folgenden Abschnitt erklärt wird.

Um eine Auswertung des Experiments durchzuführen ist das Konzept der Schiffsdomänen, welches 1971 von Fujii und Tanaka eingeführt wurde, verwendet worden [FuTa71]. Dieses ermöglicht eine Analyse von Beinahe-Zusammenstößen (engl.: near collisions) durchführen zu können. Eine bekannte Definition von Schiffs Domänen wurde dabei von Goodwin gegeben, welche übersetzt eine Schiffsdomäne als

„das umgebende Wasser welches der Navigator eines Schiffes frei von anderen Schiffen oder fixen Objekten halten will“

beschreibt (vgl. [Good75]).

Die Schiffsdomäne wird durch eine Ellipse beschrieben deren Zentrum durch die Position des Schiffes definiert ist. Die Halbachsen der Ellipse werden dabei als 4 fache Schiffslänge L für die Hauptachse a und 1,6 fache Schiffslänge L für die Nebenachse b angegeben (s. Abbildung 5-8).

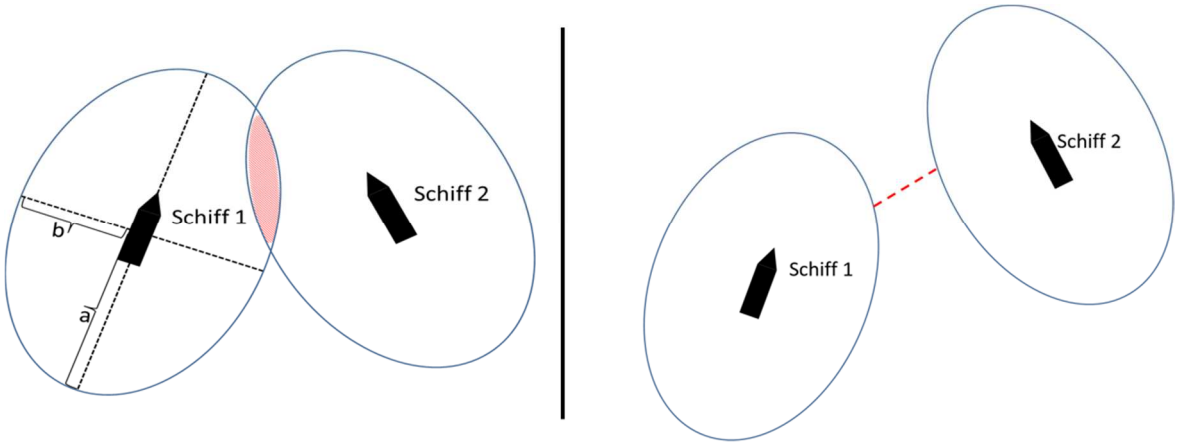


Abbildung 5-8 Beispiel für den Einsatz und die Beschreibung von Schiffsdomänen.

Um die Beinahe Zusammenstöße zu berechnen werden die Schiffsdomänen für alle Schiffe basierend auf ihren Positionen und dem Kurs über Grund zu jedem Zeitpunkt berechnet. Eine Überlappung mehrerer Schiffsdomänen zeigt dabei einen Beinahe Zusammenstoß an.

Zusätzlich zur Überlappung wurde die nächste Entfernung zwischen zwei Schiffsdomänen als Subdistanzfunktion eingeführt. Die analytische Berechnung ist nicht trivial, daher kommt der algorithmische Ansatz von Ik-Sung Kim zum Einsatz (vgl. [Iksu06]), welcher durch eine intelligente Auswahl der zu überprüfenden Ellipsen Segmente die kürzeste Distanz mit einer Wahrscheinlichkeit von 95% in ausreichend schneller Zeit findet. Die Berechnung der kürzesten Distanz zwischen mehreren Schiffsdomänen wurde dabei als virtueller Sensor umgesetzt welche als Eigenschaft der einzelnen Schiffe gepflegt wird.

Als Eintrittsraum wurde dabei eine Schiffsdomänendistanz von 0m gewählt während für den Relevanzraum eine Distanz von 4980m zwischen den Schiffsdomänen als Beginn gewählt wurde (s. Tabelle 5-1). Die Distanz von 4980m basiert dabei auf der Definition der Planned/Normal Zone (vgl. [LePa07], S.130) welche 10 Schiffslängen beträgt. Dabei wurde die Renate Schulte, als das größere Schiff, mit 166m Länge in der Angabe des Relevanzraumes verwendet und mit einer zusätzlichen Gewichtung von 3 versehen, um schon eine frühzeitige Annäherung zu bewerten. ($166m * 10 * 3 = 4980m$).

Nahbereichslage	$E_D 1a_{max} = 0\text{m}$	$R_D 1a_{max} = 4980\text{m}$

Tabelle 5-1 Grenzen des Eintritts- und Relevanzraums für die Bewertung der Nahbereichslage.

Die daraus resultierende Subdistanzfunktion ist in Formel (7) zu sehen.

$$d_{nb} = \begin{cases} 0, wenn\ Distanz = 0m \\ 1, wenn\ Distanz \geq 4980m \\ \dfrac{\text{Distanz}}{4980\text{m}}, sonst \end{cases} \quad (7)$$

Als Indikatoren für die Winkel der Schiffslängsachsen zur Bestimmung der betrachteten Kollisionssituation (Überholen, Frontal, Kreuzen) konnte auf die Ausrichtung (engl. heading) der Schiffe zugegriffen werden (s. Tabelle 5-2), wobei der Relevanzraum für die Frontalkollision innerhalb der Eintrittsräume für eine Schiffskollision während des Kreuzens liegt (s. Abbildung 5-7). Die entsprechenden Grenzen für eine Kollision zwischen kreuzenden und überholenden Schiffen sind ebenfalls in Tabelle 5-2 angegeben.

Winkel zwischen den Schiffslängsachsen (Frontalkollision)	$E_D 2a_{min} = 135°$ $E_D 2a_{max} = 225°$	$R_D 2a_{min} = 100°$ $R_D 2a_{max} = 250°$
Winkel zwischen den Schiffslängsachsen (Überholvorgang A)	$E_D 3a_{min} = 315°$ $E_D 3a_{max} = 360°$	$R_D 3a_{min} = 285°$
Winkel zwischen den Schiffslängsachsen (Überholvorgang B)	$E_D 4a_{min} = 0°$ $E_D 4a_{max} = 45°$	$R_D 4a_{max} = 80°$
Winkel zwischen den Schiffslängsachsen (Kreuzen A)	$E_D 5a_{min} = 225°$ $E_D 5a_{max} = 315°$	$R_D 5a_{min} = 190°$ $R_D 5a_{max} = 350°$
Winkel zwischen den Schiffslängsachsen (Kreuzen B)	$E_D 6b_{min} = 45°$ $E_D 6b_{max} = 135°$	$R_D 6b_{min} = 10°$ $R_D 6b_{max} = 170°$

Tabelle 5-2 Grenzen des Eintritts- und Relevanzraums für die Bewertung der Winkel zwischen den Schiffslängsachsen. Betrachtung der Fälle: Frontalkollision, Überholvorgang und Kreuzen.

Die Subdistanzfunktionen für die jeweiligen Indikatoren sind in den Formeln (8),(9),(10),(11) und (12) dargestellt.

$$d_{wif} = \begin{cases} 0, wenn\ Winkel \geq 135° \wedge Winkel \leq 225° \\ 1, wenn\ Winkel \leq 100° \vee Winkel \geq 250° \\ \min\left(\frac{(Winkel - 135°)}{100° - 135°}, \frac{(Winkel + 225°)}{250° - 225°}\right), sonst \end{cases} \quad \textbf{(8)}$$

$$d_{wi\ddot{u}a} = \begin{cases} 0, wenn\ Winkel \geq 0° \wedge Winkel \leq 45° \\ 1, wenn\ Winkel \geq 80° \\ \frac{Winkel}{80° - 45°}, sonst \end{cases} \quad \textbf{(9)}$$

$$d_{wi\ddot{u}b} = \begin{cases} 0, wenn\ Winkel \geq 315° \wedge Winkel \leq 360° \\ 1, wenn\ Winkel \leq 285° \\ \frac{Winkel - 315°}{285° - 315°}, sonst \end{cases} \quad \textbf{(10)}$$

$$d_{wika} = \begin{cases} 0, wenn\ Winkel \geq 225° \wedge Winkel \leq 315° \\ 1, wenn\ Winkel \leq 190° \vee Winkel \geq 350° \\ \min\left(\frac{(Winkel - 225°)}{190° - 225°)}, \frac{(Winkel + 315°)}{350° - 315°}\right), sonst \end{cases} \quad \textbf{(11)}$$

$$d_{wikb} = \begin{cases} 0, wenn\ Winkel \geq 45° \wedge Winkel \leq 135° \\ 1, wenn\ Winkel \leq 10° \vee Winkel \geq 170° \\ \min\left(\frac{(Winkel - 45°)}{10° - 45°)}, \frac{(Winkel + 135°)}{170° - 135°}\right), sonst \end{cases} \quad \textbf{(12)}$$

Zur Beschreibung des unsicheren Passierabstands bei einem Überholvorgang wurde die kürzeste geometrische Distanz zwischen den Bounding-Boxen der am Überholvorgang beteiligten Schiffe verwendet (s. Abbildung 5-9). Zur weiteren Angabe wurde die maximale Schiffsbreite des überholenden Schiffes (Overtaking Vessel Beam ($OVBeam$)) herangezogen (s. Tabelle 5-3).

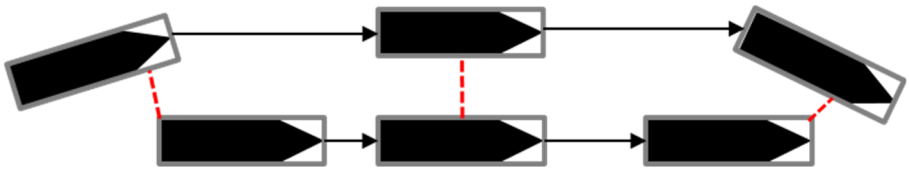


Abbildung 5-9 Beispiel zur Bestimmung des Passierabstandes zweier Schiffe. Die grauen Kästen um die Schiffe stellen die Bounding-Boxen dar während die rote Linie den Passierabstand repräsentiert.

Unsicherer Passierabstand	$E_D 5a_{max} = OV\text{beam}/2$	$R_D 5a_{max} = OV\text{beam}/2 + \text{OVbeam}$

Tabelle 5-3 Grenzen des Eintritts- und Relevanzraums für die Bewertung des unsicheren Passierabstands bei einem Überholvorgang.

Die Subdistanzfunktion für die Bewertung des unsicheren Passierabstands ist in Formel (13) dargestellt.

$$d_{up} = \begin{cases} 0, & wenn\ Passierabstand \leq OVbeam/2 \\ 1, & wenn\ Passierabstand \geq OVbeam/2 + OVbeam \\ \dfrac{Passierabstand + OVbeam/2}{OVbeam}, & sonst \end{cases} \quad \textbf{(13)}$$

Als letzte zu beschreibende Subdistanzfunktion wurde die in den COLREGs beschriebene sichere Geschwindigkeit, welche bei Missachtung ein Manöver des vorletzten und letzten Augenblicks bei einer Nahbereichslage unmöglich machen kann, betrachtet. Hierfür wurden

die Geschwindigkeiten der beteiligten möglichen Unfallgegner beobachtet und Ereignisraum und Relevanzraum festgelegt.

Um ein anwendbares Maß zu finden wurde in diesem Fall ebenfalls auf Urteile ähnlicher Konstellationen wie im Renate Schulte Fall zurückgegriffen. Unter anderem wurde ein Urteil welches die Kollision zwischen den Schiffen „Hel“ und „Skyron“ im Jahr 1987 zum Thema hatte betrachtet (vgl. [Bier04]). Bei dieser Kollision, die in der Nordsee stattfand, gab das zuständige Gericht bekannt, dass weder die „Hel“ noch die „Skyron“ mit einer sicheren Geschwindigkeit gefahren waren. Die "Hel" fuhr mit 16kn, die "Skyron" mit 12,5kn. Andere Urteile von ähnlichen Situationen (offene See, gleiche oder ähnliche Schiffstypen, gleiche oder ähnliche Umweltbedingungen) gaben ebenfalls ähnliche Geschwindigkeiten als nicht sicher an.

Unsichere (zu hohe) Geschwindigkeit	$E_D 6a_{min} = 12{,}5kn$	$R_D 6a_{min} = 10kn$

Tabelle 5-4 Grenzen des Eintritts- und Relevanzraums für die Bewertung der unsicheren (zu hohen) Geschwindigkeit.

Die Formel für die Bewertung der unsicheren Geschwindigkeit eines Schiffes ist in Formel (14) zu sehen.

$$d_{ug} = \begin{cases} 0, wenn\ \ Geschwindigkeit \geq\ 12{,}5kn \\ 1, wenn\ Geschwindigkeit \leq\ 10kn \\ \dfrac{Geschwindigkeit - 12{,}5kn}{10kn - 12{,}5kn}, \qquad sonst \end{cases} \tag{14}$$

Den Fehlerbaum für Schiffskollisionen nutzend wurde der, in Abbildung 5-10 gezeigte, Simulationsplan erstellt. Im oberen Teil der Abbildung sind die im Simulationsplan verwendeten Knoten zum Aufsetzen und Starten der Co-Simulation sowie zu dessen Stoppen zu sehen. Der untere Teil stellt einen Ausschnitt der Risikobewertung mittels Zugriff auf die in der Co-Simulation übertragenen Daten dar.

Die verwendete Funktion zur Berechnung des Kollisionsrisikos zwischen der Renate Schulte und Marti Princess, welche aus der Fehlerbaumstruktur abgeleitet wurde, ist dabei in Formel (15) zu sehen, während die komplette zusammengesetzte Risikodistanzfunktion in Formel (16) dargestellt ist.

$$D_K = \min \begin{pmatrix} 1 - \left(\left(1 - d_{up}\right) \times \left(1 - d_{nb}\right) \times \left(1 - min(d_{wi\ddot{u}a}, d_{wi\ddot{u}b})\right) \right), \\ 1 - \left(\left(1 - d_{nb}\right) \times \left(1 - d_{ug}\right) \times \left(1 - d_{wif}\right) \right), \\ 1 - \left(\left(1 - d_{nb}\right) \times \left(1 - \min(d_{wika}, d_{wikb})\right) \right) \end{pmatrix} \quad \textbf{(15)}$$

$$
\begin{aligned}
D_K = \min\Bigg[& 1 - \Bigg(\Bigg(1 - \begin{cases} 0, & wenn\ Passierabstand \leq 0m \\ 1, wenn\ Passierabstand \geq OVbeam \div 2 + OVbeam \\ \dfrac{Passierabstand + OVbeam \div 2}{OVbeam}, & sonst \end{cases} \Bigg) \\
& \times \Bigg(1 - \begin{cases} 0, wenn\ Distanz = 0m \\ 1, wenn\ Distanz \geq 4980m \\ \dfrac{Distanz}{4980m}, sonst \end{cases} \Bigg) \\
& \times \Bigg(1 - min \Bigg(\begin{cases} 0, wenn\ Winkel \geq 0° \wedge Winkel \leq 45° \\ 1, wenn\ Winkel \geq 80° \\ \dfrac{Winkel}{80° - 45°}, sonst \end{cases}, \\
& \begin{cases} 0, wenn\ Winkel \geq 315° \wedge Winkel \leq 360° \\ 1, wenn\ Winkel \leq 285° \\ \dfrac{Winkel - 315°}{285° - 315°}, sonst \end{cases} \Bigg) \Bigg) \Bigg) \Bigg], \\
1\text{-}\Bigg(& \Bigg(\Bigg(1 - \begin{cases} 0, wenn\ Distanz = 0m \\ 1, wenn\ Distanz \geq 4980m \\ \dfrac{Distanz}{4980m}, sonst \end{cases} \Bigg) \\
& \times \Bigg(1 - \begin{cases} 0, wenn\ Geschwindigkeit \geq 12{,}5kn \\ 1, wenn\ Geschwindigkeit \leq 10kn \\ \dfrac{(Geschwindigkeit - 12{,}5kn)}{(10kn - 12{,}5kn)}, & sonst \end{cases} \Bigg) \\
& \times \Bigg(1 - \begin{cases} 0, wenn\ Winkel \geq 135° \wedge Winkel \leq 225° \\ 1, wenn\ Winkel \leq 100° \vee Winkel \geq 250° \\ \min\left(\dfrac{(Winkel - 135°)}{(100° - 135°)}, \dfrac{(Winkel + 225°)}{(250° - 225°)} \right), sonst \end{cases} \Bigg) \Bigg), \\
1 - \Bigg[& \Bigg(1 - \begin{cases} 0, wenn\ Distanz = 0m \\ 1, wenn\ Distanz \geq 4980m \\ \dfrac{Distanz}{4980m}, sonst \end{cases} \Bigg) \\
& \times \Bigg(1 - min \Bigg(\begin{cases} 0, wenn\ Winkel \geq 225° \wedge Winkel \leq 315° \\ 1, wenn\ Winkel \leq 190° \vee Winkel \geq 350° \\ min\left(\dfrac{(Winkel - 225°)}{(190° - 225°)}, \dfrac{(Winkel + 315°)}{(350° - 315°)} \right), sonst \end{cases}, \\
& \begin{cases} 0, wenn\ Winkel \geq 45° \wedge Winkel \leq 135° \\ 1, wenn\ Winkel \leq 10° \vee Winkel \geq 170° \\ min\left(\dfrac{(Winkel - 45°)}{(10° - 45°)}, \dfrac{(Winkel + 135°)}{(170° - 135°)} \right), sonst \end{cases} \Bigg) \Bigg) \Bigg) \Bigg]
\end{aligned}
\tag{16}
$$

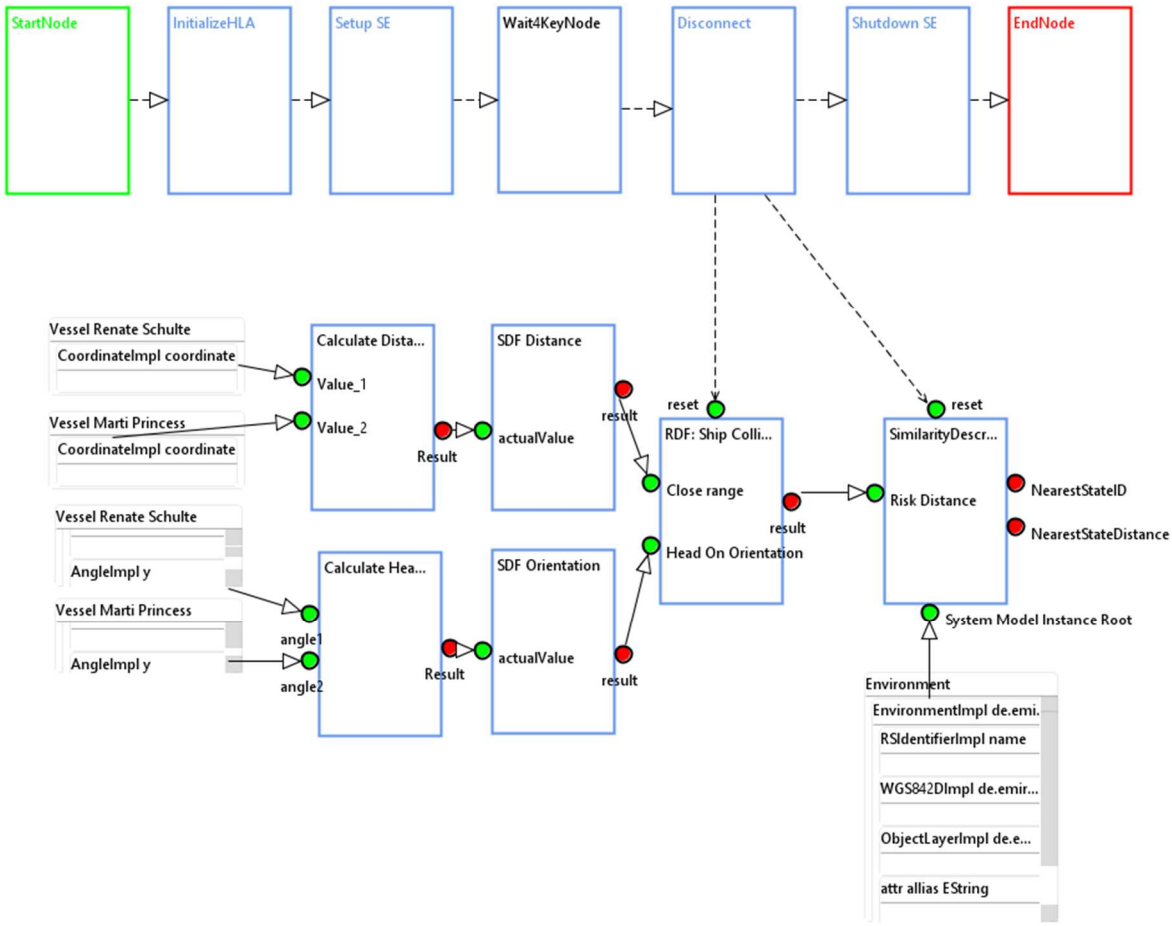


Abbildung 5-10 Ausschnitt des erstellten Simulationsplans für die simulative Bewertung des Kollisionsrisikos zweier Schiffe. Die Risikodistanzfunktion wurde mit Hilfe des erstellten Fehlerbaums generiert. Auch zu sehen sind die eingehenden Subdistanzfunktionen und die Anfrage an die Situationsdatenbank.

Im Simulationsplan fand ausgehend von den über HLA übertragenen Daten der Schiffe Renate Schulte und Marti Princess die Auswertung der sicheren Geschwindigkeit und Distanz der Schiffsdomänen sowie die Ausrichtung der beiden Schiffe zueinander statt, welche in die Auswertung durch die Risikodistanzfunktion einflossen.

Während der Simulation konnte der Verlauf des Risikos mit Hilfe der Ausgabe des Risikomonitors verfolgt werden (s. Abbildung 5-11). Die genaue Auswertung der Simulationsläufe und der dazugehörigen Risikodistanzbewertung sowie deren Vergleich erfolgt im nächsten Abschnitt („Auswertung des Experimentes“).

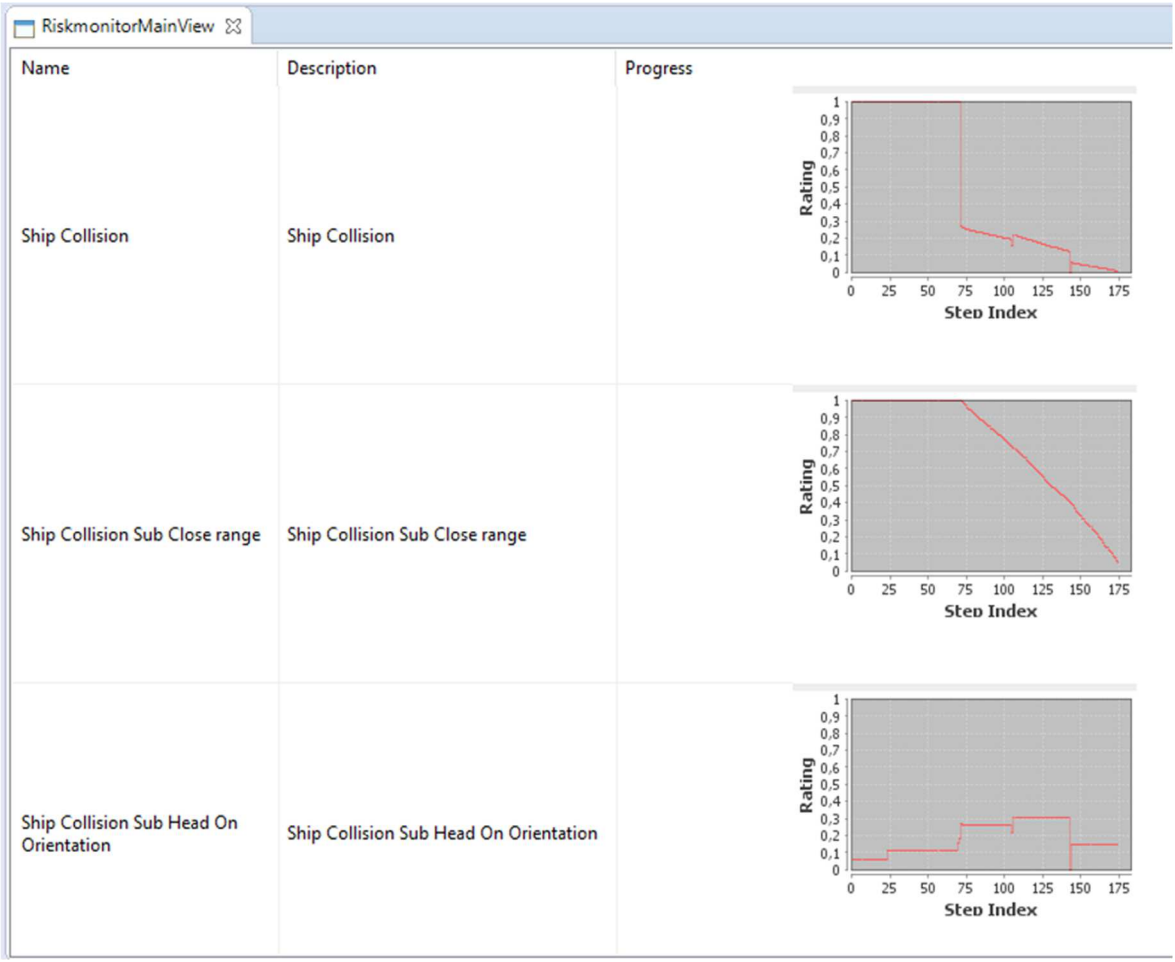


Abbildung 5-11 Ausschnitt der Darstellung des Risikoverlaufs im DistriCT-Framework. Zusätzlich zum Gesamtrisiko werden auch die Verläufe der einzelnen Bestandteile der Risikodistanz angezeigt.

5.2.4 Auswertung des Experimentes

Um die Auswertung des Experimentes durchzuführen wurde der Fahrtenverlauf (s. Abbildung 5-15) der beteiligten Schiffe und die Gesamtrisikoentwicklung (s. Abbildung 5-12) miteinander verglichen, um eine Korrelation zwischen Risikodistanz und Unfallhergang als Evaluationsgrundlage zu nutzen.

Der Verlauf der Risikodistanzfunktion ist dargestellt in Abbildung 5-12, während die Ergebnisse der Subdistanzfunktionen aufbauend auf dem Konzept der Schiffsdomänen und ihrer kürzesten Distanz sowie der unsicheren Geschwindigkeit in Abbildung 5-13 zu sehen ist. Die Bewertung durch die Ausrichtung der Schiffe zueinander wird in Abbildung 5-14 dargestellt.

Während die Fehlerbaum Ereignisse „Schiffskollision während des Überholvorgangs“ und „Schiffskollision während des Kreuzens“ nicht ausgelöst worden, konnte die erwartete Veränderung des Risikos für das Ereignis „Frontalkollision“ gemessen werden.

Im ersten Drittel des Verlaufs ergab sich noch keine Risikoannäherung, da die Schiffe (Renate Schulte und Marti Princess) weder auf einem Kollisionskurs lagen noch die Nähe der Schiffsdomänen einen Risikoanstieg hervorriefen (s. Abbildung 5-15:1 und 2). Dies änderte sich jedoch, als die Marti Princess die Ilgaz hinter deren Heck passiert hatte und wieder auf ihren alten Kurs einschwenkte (s. Abbildung 5-15:3). Durch die Anpassung des Kurses (Schwarz markierte Kreise in Abbildung 5-12 und Abbildung 5-14) konnte kurzfristig eine Verringerung des Risikos -also eine Vergrößerung der Risikodistanz- gesehen werden. Jedoch evaluierte die Risikodistanz mit der Kollision (s. Abbildung 5-15:4) schlussendlich zu nahe 0.

Zusätzlich fand eine Überprüfung der Funktionalität der Situationsdatenbank statt (vgl. Abschnitt 3.3.5.1 – Einbindung der Risikodistanzfunktion). Ausgehend von der eingehenden Risikodistanz wurde dabei ab einem bestimmten Schwellwert überprüft ob eine Anfrage an die Situationsdatenbank bzgl. der gleichen oder stark ähnlichen Situation gestellt wird. Der Schwellwert wurde dabei dynamisch angepasst, beginnend bei einer Risikodistanz von 0,7 und einer Schrittweite von 0,1 wurde dieser nach Erreichen weiter herabgesetzt. Die Systemmodellinstanz, auf der auch der Situationsdeskriptor basiert, beinhaltete dabei die Positionen, Orientierungen und Geschwindigkeiten der beteiligten Schiffe sowie die statischen Charakteristiken der Schiffe wie zum Beispiel die Größe. Die Anfragen lieferten dabei Ergebnisse über die ähnlichste gefundene Situation in der Datenbank, wie oft diese Situation schon erreicht wurde über alle bis zum Anfragezeitpunkt durchgeführten Simulationsläufe und die dementsprechende ID der Situation, um ein eindeutiges Speichern und Laden dieser wiederdurchzuführen.

Aus der näheren Betrachtung der Anfragen innerhalb der mehrere Simulationsläufe andauernden Evaluation konnten dabei zwei Ergebnisse sichergestellt werden.

1.) Der Detailgrad des Systemmodells beeinflusst die Güte der Anfrageergebnisse. Jedoch reicht schon ein relativ kleiner Detailgrad, wie der hier betrachtete, um verwertbare Situationsvergleiche durchzuführen.

2.) Die Nutzung der Situationsdatenbank erlaubt es über mehrere Simulationsläufe hinweg Informationen über die Risikonähe bestimmter Situationen zu persistieren. Während die Risikodistanzfunktion zwar die aktuelle Situation bewertet erlaubt die Anfrage an die Situationsdatenbank Wissen aus vorherigen Simulationsläufen mit in die zukünftigen Entscheidungen des Simulationsablaufs mit einzubeziehen.

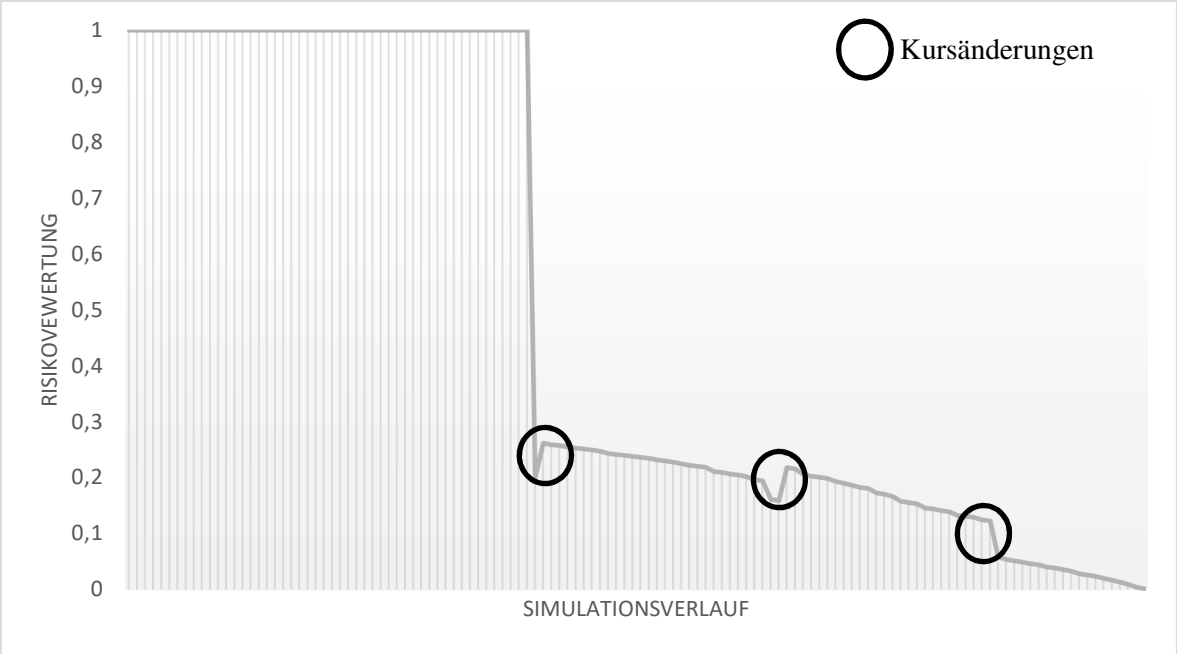


Abbildung 5-12 Gesamtrisikobewertung mittels der Risikodistanzfunktion während eines Simulationslaufs zur Schiffskollision zwischen der Renate Schulte und Marti Princess.

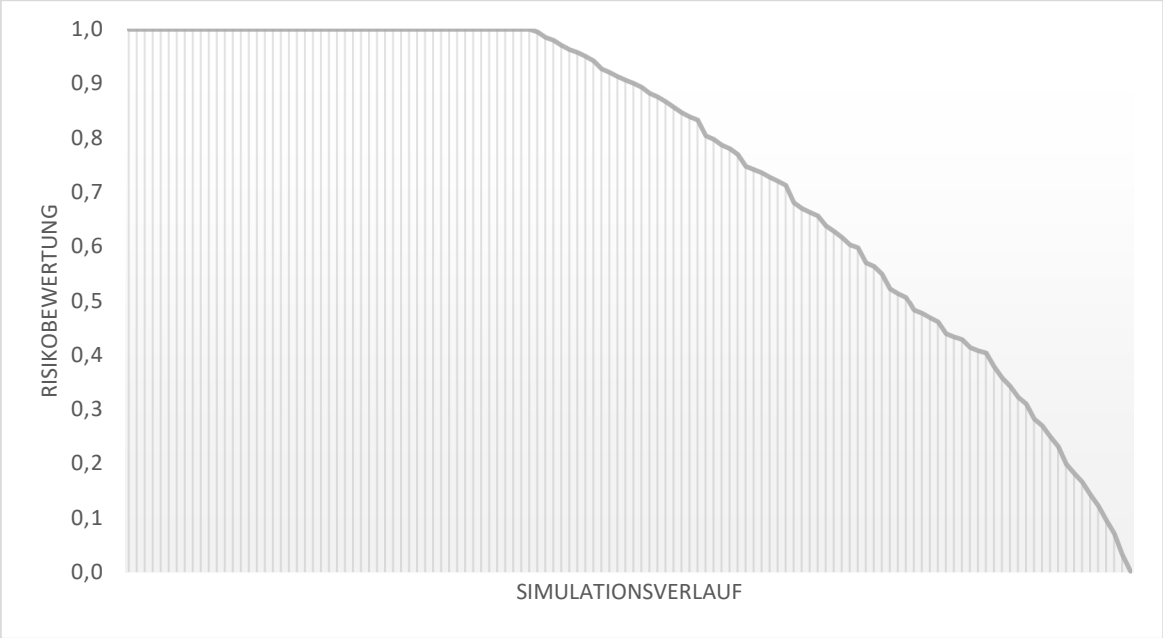


Abbildung 5-13 Risikobewertung während eines Simulationslaufs zur Schiffskollision zwischen der Renate Schulte und Marti Princess für die Subdistanzfunktion „Nahbereichslage“.

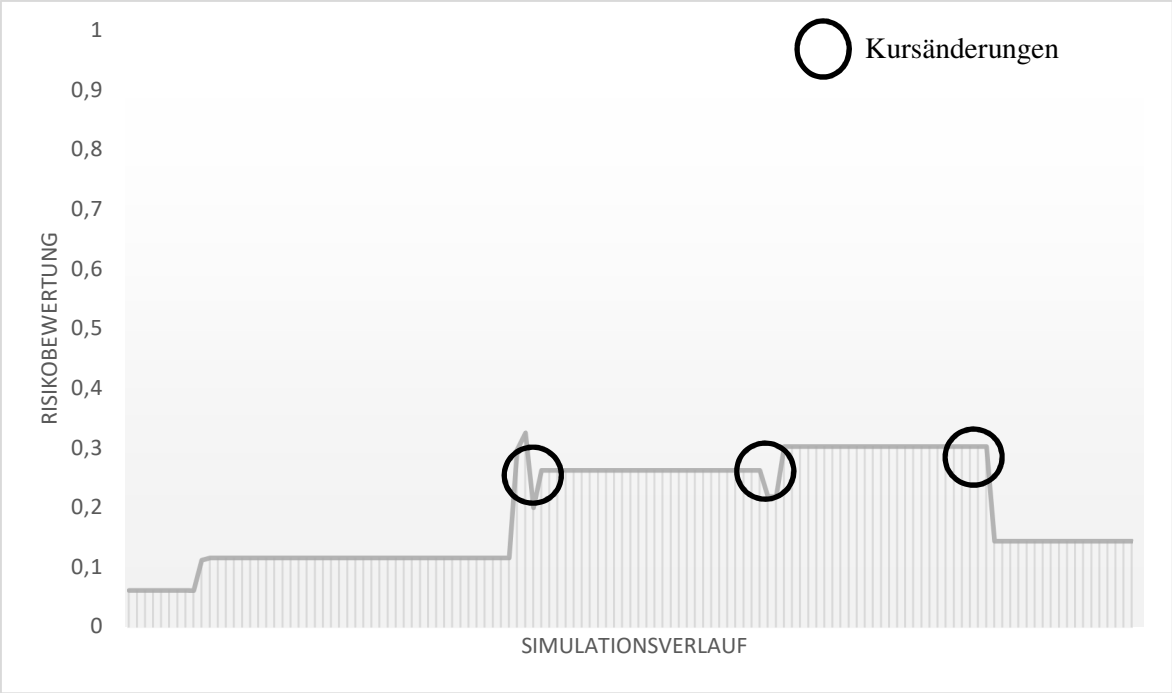


Abbildung 5-14 Risikobewertung während eines Simulationslaufs zur Schiffskollision zwischen der Renate Schulte und Marti Princess für die Subdistanzfunktion „Winkel zwischen den Schiffslängsachsen (Frontalkollision)“.

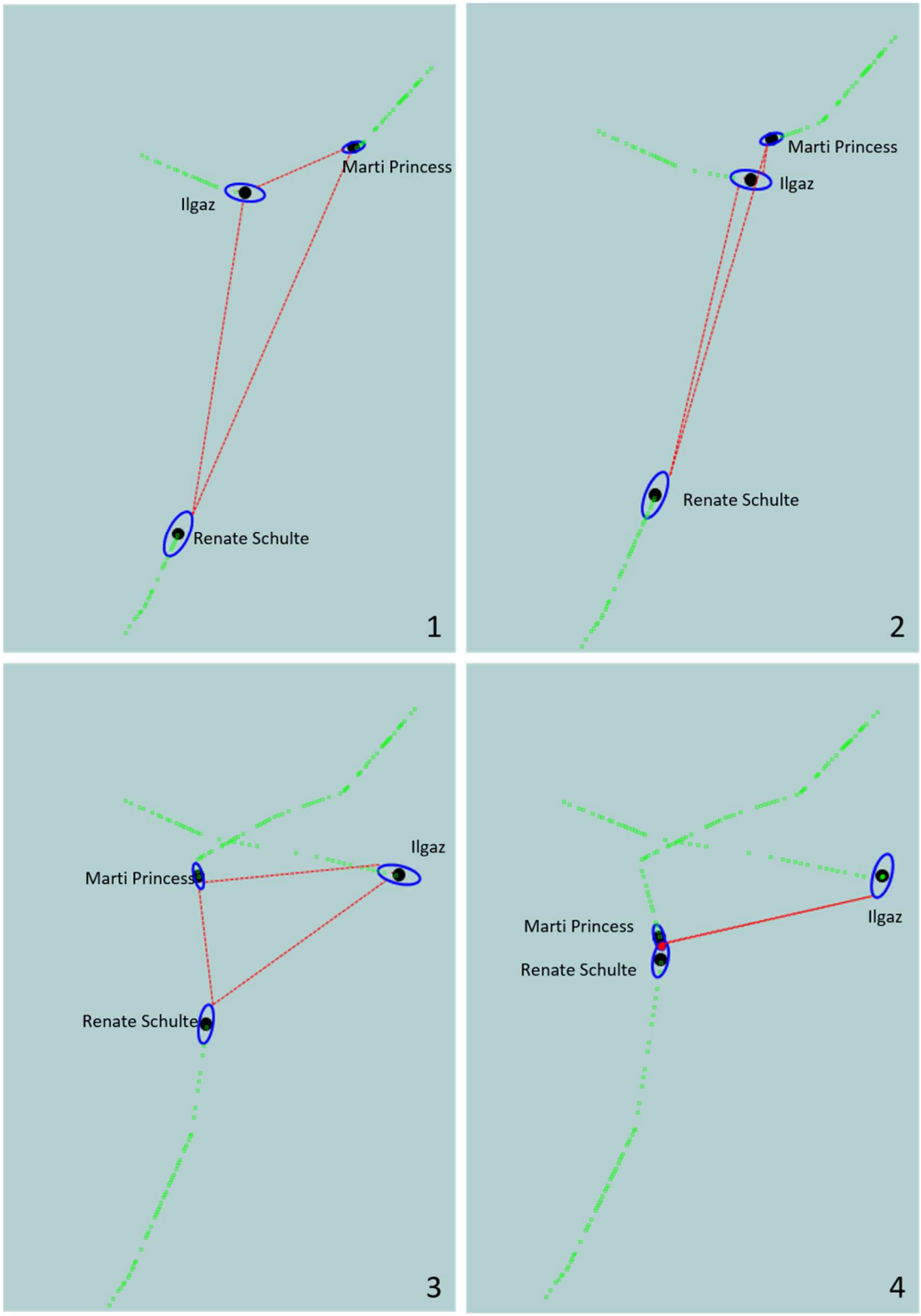


Abbildung 5-15 Beispielhafter Fahrtenverlauf während eines Simulationslaufs – die hellgrüne Spur zeigt den zurückgelegten Weg der drei Schiffe an, die rot gestrichelte Linie die kürzeste Distanz zwischen den Schiffsdomänen.

Aus dem Verlauf der Risikobewertung lässt sich ableiten, dass -das benötigte Expertenwissen vorliegend- eine den korrekten Risikotrend wiederspiegelnde Risikofunktion erstellt werden kann. Diese ist ebenfalls online (mittels Ausgabe des Risikomonitors) mit verfolgbar, um notwendige Anpassungen, durch neues Expertenwissen gegeben, vorzunehmen.

Die Betrachtung des Verlaufs und der Risikobewertung zeigt, dass das Annähern des Risikos schon einige Zeit vor der eigentlichen Kollision und dem Erreichen der Nahbereichssituation erkannt werden konnte. Auch die Kursänderungen spiegelten sich in der Entwicklung der Gesamtrisikodistanz wieder, wobei der Trend zur Schiffskollision weiterhin sichtbar war.

Daraus ableitend lässt sich schließen, dass mit dem notwendigen Expertenwissen die erstellten Risikodistanzfunktionen genutzt werden können, um reale Szenarien hinsichtlich ihres Risikos zu bewerten.

Die Situationsdatenbank stellte sich dabei zusätzlich als einsetzbares Werkzeug heraus, die schon ab einem relativ niedrigem Detailgrad des Systemmodells verwertbare Ergebnisse über mehrere Simulationsläufe hinweg liefert.

Neben der erstellten Risikodistanzfunktion ist jedoch auch die Wahl der richtigen Schwellwerte, um zu bestimmen wann ein Simulationszustand gespeichert werden soll, eine Entscheidung die vom betrachteten Szenario abhängig ist. Die Methodik und das entwickelte DistriCT-Tool bieten jedoch die Möglichkeit eine dynamische Anpassung des Schwellwertes durchzuführen sobald der vorherige erreicht wurde.

Zusammenfassend lässt sich sagen, dass mit Hilfe der Risikodistanzfunktion, der Situationsdatenbank sowie der Simulationskontroll- und Programkontrollfunktionalität (DistriCT) eine der Importance Splitting/RESTART Methode ähnliche Führung der Simulation in Richtung untersuchter risikoreicher Situationen durchgeführt werden kann. Die Bewertung der Simulationszustände durch die Risikodistanzfunktion und die weitere Bewertung der erreichten Situationen mittels Situationsdatenbank kann dabei als Entscheidungsgrundlage genutzt werden, um das Speichern (und neu Explorieren) aussichtsreicher Situationen durchzuführen oder unterbrochen werden, um die Zeit der Analyse zu verkürzen.

5.3 Evaluation der Simulationsführung mittels Risikodistanzfunktion

Im letzten Evaluationsabschnitt soll die Simulationsführung mittels Risikodistanzfunktionen bewertet werden (vgl. Anforderung [A_A3]). Hierfür wurde ein Szenario mit zwei beteiligten Schiffen gewählt, welche in entgegengesetzter Richtung auf einem Fluss unterwegs sind. Es soll dabei eine Kollision zwischen diesen beiden Schiffen beobachtet werden, welche jedoch durch die in den nachfolgenden Abschnitten vorgestellten Bedingungen nur sehr selten zu erreichen ist. Um dennoch eine häufigere Beobachtung einer Kollision zu erhalten, wurden die in dieser Arbeit vorgestellten Ansätze und Konzepte verwendet, um die Simulation zu steuern.

5.3.1 Aufbau und Ablauf des Evaluationsexperimentes

In dem betrachteten Experiment werden zwei Simulatoren verwendet, die die beiden beteiligten Schiffe steuern. Beide Schiffe steuern mit einer hohen Wahrscheinlichkeit ihren Zielwegpunkt an und weichen nur mit einer sehr geringen Wahrscheinlichkeit von dem direkten Kurs auf ihren Zielwegpunkt ab (s. Abbildung 5-16).

Als Abbruchbedingung für einen Simulationslauf wurde das Erreichen der Zielwegpunkte durch beide Schiffe bestimmt. Das Ereignis, welches in dieser Evaluation untersucht wurde, war die Kollision zwischen beiden Schiffen, welche innerhalb dieser Evaluation 100mal unter den jeweils angewandten Simulationseinstellungen beobachtet werden sollte.

Die Simulationseinstellungen bestanden dabei aus einer naiven und einer geführten Simulation, bei der es drei Varianten gab wie die Festlegung des nächsten Splitting Points zu erfolgen hat. Zum einen wurde ein adaptives Verfahren gewählt, welches in jedem Simulationsschritt überprüft, ob eine niedrigere Risikodistanzfunktionsbewertung erreicht wurde als zuvor und zum anderen ein Verfahren, welches die Splitting Points in vorgegebenen Abständen angelegt hat (0.01 und 0.05).

Bei allen Verfahren, außer bei der naiven Simulation, wurden bei einer neuen niedrigeren Risikodistanz beim adaptiven Verfahren oder dem Erreichen eines neuen Risikolevels, der Zustand gespeichert und aus diesem der nächste Simulationslauf durchgeführt

Es wurden alle Simulationsläufe gezählt in denen keine Kollision aufgetreten ist sowie die Realzeit gemessen bis zum Zeitpunkt des Eintritts einer Kollision. Dieser Vorgang wurde 100mal durchgeführt, wobei keine parallele Ausführung stattfand.

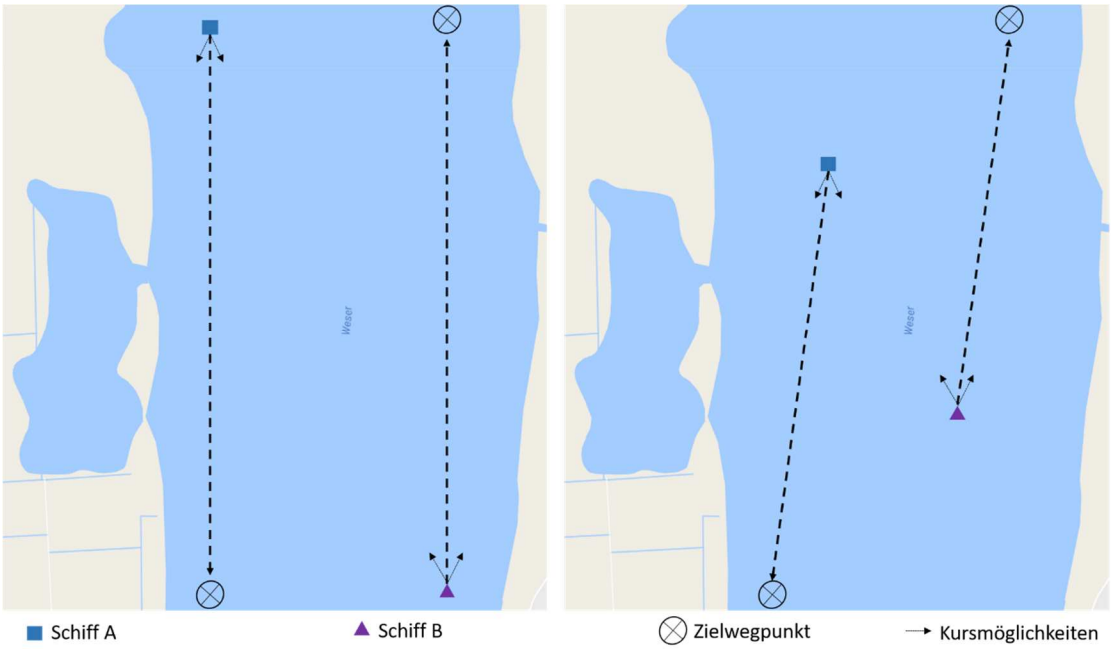


Abbildung 5-16 Evaluationsszenario 3: Die beiden beteiligten Schiffe versuchen ihren Zielwegpunkt anzufahren. Mit einer sehr geringen Wahrscheinlichkeit weichen die Schiffe nach links oder rechts von Ihrem Kurs ab. Mit einer sehr hohen Wahrscheinlichkeit fahren sie nach einem Abweichen wieder ihren Zielwegpunkt an.

Als Risikodistanzfunktion wurde dabei die Funktion aus dem Evaluationsabschnitt 5.2 übernommen (s. Formel (16)), welche die Nähe, Geschwindigkeit und Ausrichtung der beiden Schiffe zueinander für die Bestimmung der Risikodistanz nutzt. Die beiden Schiffe fuhren während des Experiments mit einer Wahrscheinlichkeit von 99,9% auf ihren Zielwegpunkt zu, während sie mit einer Wahrscheinlichkeit von 0,1% ihren Kurs nach links oder rechts abänderten. Die Entscheidung den Kurs zu ändern wurde dabei von den beiden Schiffen in jedem Simulationsschritt neu berechnet.

5.3.2 Auswertung des Experimentes

Die Durchführung des Experiments ergab, dass die naive Simulation ca. 152 Minuten benötigte um 100 Kollisionen zu beobachten, während die mittels Risikodistanzfunktion geführten Simulationsläufe 100 Kollisionen bereits nach ca. 3 Sekunden (Abstand 0,05), 95 Millisekunden (Abstand 0,01) und 92 Millisekunden (Adaptiv) erreichten.

Während bei der naiven Simulation 44.076.619 Simulationsläufe benötigt wurden bis 100 Kollisionen beobachtet werden konnten, betrug die Anzahl der Simulationsläufe für den Abstand der Splitting Points von 0,05 5.374.930 Läufe, für den Abstand der Splitting Points von 0,01 59.551 Läufe und für den angepassten Abstand 53.150 Läufe.

In Abbildung 5-17 ist ein beispielhafter Verlauf der Risikobewertung bis zum Eintritt einer Kollision bei Anwendung des adaptiven Abstandes zu sehen. Auffällig ist dabei die schnelle Annäherung auf eine Risikodistanz von 0,3, die sich durch das Entgegenkommen der Schiffe und die sich dadurch verringernde Distanz ergab, und das dann langsamere annähern der beiden Schiffe zueinander hin.

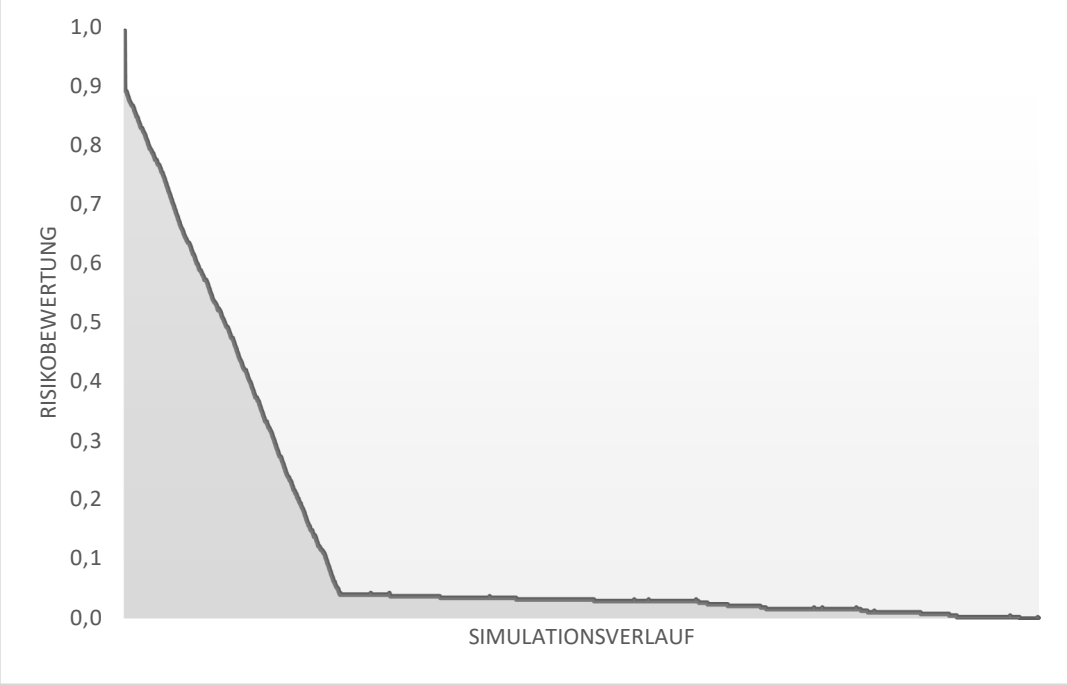


Abbildung 5-17 Beispielhafter Verlauf der Risikobewertung für die Anwendung des adaptiven Schwellwertes bis zum Eintritt einer Kollision.

Im Falle dieses Experiments stellte sich das Verfahren mit dem adaptiven Abstand als das Beste, hinsichtlich der Summe der Simulationsläufe als auch der gemessenen Realzeit, heraus bis 100 unabhängige Kollisionen beobachtet werden konnten. In Tabelle 5-5 ist eine Übersicht mit den ermittelten Werten zur Gesamtlaufzeit und Summe der Simulationsläufe aber auch dem Mittelwert über die Anzahl benötigter Simulationsläufe, bis zum Auftreten der Kollision zwischen den beteiligten Schiffen mit und ohne Anwendung der Steuerung mittels Risikodistanzfunktion, sowie die maximale und minimale Anzahl an benötigten Simulationsläufen zu sehen.

Tabelle 5-5 Ergebnisse des dritten Evaluationsexperiments im Überblick

	Naiv	RDF (Adaptiv)	RDF (Abstand 0,01)	RDF (Abstand 0,05)
Gesamtlaufzeit (Millisekunden)	9176670	92	95	3.153
Mittelwert (Anzahl Läufe)	440766,19	531,5	595,51	53.749,3
Max (Anzahl Läufe)	1.929.412	1.074	1.161	149.351
Min (Anzahl Läufe)	8.254	4	6	6
Summe (Anzahl Läufe)	44.076.619	53.150	59.551	5.374.930

Der Vergleich zwischen der naiven und mittels Risikodistanzfunktion gesteuerten Simulation zeigt, dass die Verwendung der Risikodistanzfunktion einen erheblichen Vorteil gegenüber der Anwendung einer naiven Simulation bringen kann. Zusätzlich zeigte sich in diesem Experiment, dass die adaptive Festlegung von Splitting Points über das herabsetzen der nächsten zu erreichenden Risikodistanz, zu den besten Ergebnissen führte

6 Zusammenfassung und Schlussfolgerungen für den eigenen Beitrag

In diesem Abschnitt erfolgt ein abschließender Überblick über die Bestandteile der Arbeit, deren Zusammenfassung und Schlussfolgerungen für den eigenen Beitrag so wie ein Ausblick über mögliche Erweiterungen des Frameworks und Weiterentwicklungen der Methodik und Risikodistanzmessung.

6.1 Zusammenfassung

In dieser Arbeit wurde auf Grund der immer angespannteren Lage in der maritimen Domäne eine Motivation (s. Kapitel 1 „*Einleitung*“) für die simulative Risikoanalyse gegeben. Dabei wurde gerade die auf Grund von fehlender Zeit vernachlässigte Untersuchung von schweren aber selten auftretenden Gefahren erläutert.

Im Kapitel 2 „*Stand der Wissenschaft und Technik zur Risikoanalyse durch Simulation*“ wurden aktuell verfügbare Methoden und Techniken, sowie Software zur simulativen Risikoanalyse, Simulationsoptimierung und Simulationsframeworks vorgestellt, die ähnliche Ziele wie die dieser Arbeit umsetzen oder aber Teilziele davon. Dabei wurden zusätzlich die probabilistische Risikoanalyse, Modellierung & Simulation und das Thema Simulationsoptimierung und im speziellen Rare Event Simulation vorgestellt. Das Kapitel schließt mit dem ermittelten Handlungsbedarf inklusive der für den eigenen Ansatz notwendigen übergeordneten Anforderungen.

Den Handlungsbedarf und die aufgestellten übergeordneten Anforderungen kennend wurde der eigene Ansatz in Kapitel 3 „*Bewertung von Simulationszuständen in Co-Simulationen zur beschleunigten simulativen Analyse*“ präsentiert. Hierbei wurde zunächst die Erstellung einer System-, Verhaltens-, und Gefahrenbeschreibung vorgestellt, auf welche die entwickelte Methodik zur Bewertung von Simulationszuständen aufsetzt. Diese unterteilte sich dabei in eine Methodik zur Ermittlung von Risikodistanzfunktionen durch Nutzung des semiprobabilistischen Sicherheitskonzeptes und die weitere Einbindung dieser in eine simulative Analyse. Dabei wurde beschrieben, wie die Distanzfunktionen in Co-Simulationsumgebungen eingesetzt werden können, um Rare-Event-Technologien wie

RESTART/Importance Splitting, die Simulationsführung betreffend, zu unterstützen. Des Weiteren wurde erläutert wie Teile der Information Retrieval Technik genutzt werden, um simulationslaufübergreifende Informationen zu Simulationszuständen zu erhalten.

Im folgenden Kapitel 4 „*DistriCT - Ein Framework zur Konfiguration, Kontrolle und Analyse von Co-Simulationen*“ wurde zum einen die Definition der Co-Simulationen und Simulatoren und zum anderen das DistriCT Framework, welches die Definition von Distanzfunktionen aber auch das Aufsetzen, Analysieren und Steuern der Simulationsumgebung unterstützt, präsentiert. Zusätzlich wurde in diesem Kapitel beschrieben, wie die Einbindung von Risikodistanzfunktionen, zur Steuerung des Simulationsverlaufs, mit Hilfe der entwickelten Toolchain erfolgen kann.

Abschließend wurde die Methodik hinsichtlich des Einsatzes in Co-Simulationen evaluiert und in Kapitel 5 „*Evaluation der entwickelten Methodik und Simulationsunterstützung*“ vorgestellt. Wobei drei verschiedene Szenarien dafür genutzt wurden. Das erste Szenario war dabei eine Verladeoperation an Bord eines Errichterschiffes, mit deren Hilfe die generelle Anwendbarkeit und der Einsatz zur Überprüfung der Unvollständigkeit von Fehlerbäumen gezeigt wurde. Das zweite Szenario war die Evaluation der Risikodistanzbewertung anhand eines realen Unfallszenarios mit dem der Risikotrend mit einem zu Grunde liegenden Unfallbericht verglichen wurde und auf die Nutzbarkeit der Situationsdatenbank und die Verwertbarkeit der ermittelten Ergebnisse für die Anwendung bei Rare-Event-Simulationen eingegangen wurde. Das letzte Szenario war ein Vergleich zwischen naiver und mittels Risikodistanzfunktion gesteuerter Simulation mit dem das schnellere Erreichen eines risikoreichen Ereignisses gezeigt wurde.

6.2 Schlussfolgerungen

Der in dieser Arbeit entwickelte Ansatz, bestehend aus einer Methodik zur Beschreibung einer Risikobewertung und der Verwendung dieser in Co-Simulationen unterscheidet sich in einigen Punkten von aktuellen Programmen und Konzepten in der Wissenschaft und Technik.

Insbesondere der Weg zur Bewertung eines Risikos wird neu betrachtet in dem ausgehend von einem Systemmodell, Prozessmodell und Fehlerbäumen, der auch für Nichtinformatiker nachvollziehbare Aufbau von Risikodistanzfunktionen möglich ist und damit die Frage „*Wie kann die Distanz zu risikoreichen Situationen in den Läufen einer Co-Simulation definiert und ermittelt werden?*“ beantwortet. Zusätzlich wird in der Arbeit beschrieben, wie die bestimmten Risikodistanzfunktionen genutzt werden können, um mittels Ansätzen aus der

Rare Event Simulation und dem Information Retrieval eine Annäherung an die risikoreiche Situation -auch bei der Verwendung von Black-Box Simulatoren- zu erreichen. So können Schwellwerte genutzt werden, um die simulativ zu erreichenden Level für den betrachteten Ansatz zum Importance Splitting zu generieren. Eine Situationsdatenbank hilft simulationslaufübergreifende Informationen über erreichte Simulationszustände zu speichern und für die weitere Steuerung der betrachteten Co-Simulation zu nutzen. Dies bezieht sich dabei auf die Beantwortung der zweiten Frage der wissenschaftlichen Fragestellung „*Wie können die ermittelten Distanzen zur beschleunigten Erreichung der zu analysierenden risikoreichen Situationen genutzt werden?*“

Es konnte gezeigt werden, dass sich der entwickelte Ansatz mit Erfüllung der aufgestellten Anforderungen an die verwendeten Simulatoren (s. Abschnitt 3.3.1) anwenden ließ und richtige Risikotrends zeigt. Durch falsch und/oder unvollständig formulierte Distanzfunktionen kann es vorkommen, dass sich die Distanz entfernt obwohl sich das untersuchte Risiko nähert. Allerdings wird die immer von Experten und ihrem Wissen abhängige Risikoanalyse nicht verschlechtert durch den Einsatz von Risikodistanzfunktionen sondern durch die Möglichkeit übersehene Ursachen simulativ ausfindig zu machen verbessert.

Es existieren fünf weitere zu untersuchende Aspekte die in zukünftigen Arbeiten von Interesse sein könnten und möglicherweise eine weitere Verbesserung dieser Arbeit ergeben könnten.

1. ***Sandbox Nutzung für Simulatoren die keine Zustandskontrolle erlauben***

   Eine wichtige zu erfüllende Anforderung an die verwendeten Simulatoren damit die entwickelte Methodik mit der Importance Splitting Technik funktioniert ist, dass diese ihren Zustand speichern und laden können und dies über eine HLAInteraction ausgelöst werden kann. Um diese Anforderung auszuklammern wäre eine mögliche zukünftige Erweiterung das Nutzen von speziellen Sandbox Umgebungen (Bsp.: VirtualBox[17]) für diese Simulatoren bei denen die Zustandsspeicherung über die aktuelle Belegung des virtuellen Arbeitsspeichers erfolgen könnte.

[17]Virtualisierungssoftware des US-amerikanischen Unternehmens Oracle (URL: https://www.virtualbox.org/ [zuletzt abgerufen am 03.03.2017])

2.) Berücksichtigung von risikominimierenden Maßnahmen

In der aktuellen Version der Risikodistanz werden das Risiko begünstigende Fehler und Indikatoren berücksichtigt. In zukünftigen Arbeiten könnte die Betrachtung von risikominimierenden Maßnahmen eine noch genauere Analyse ermöglichen.

3.) Übertragbarkeit der entwickelten Risikodistanzfunktionen auf Assistenzsysteme

Eine weitere Betrachtung der zur simulativen Bewertung genutzten Risikodistanzfunktionen hinsichtlich der Entwicklung von realen Assistenzsystemen könnte ein richtiger Schritt für zukünftige Arbeiten sein. Lassen sich die Indikatoren aus der simulativen Betrachtung durch reale Sensoren ersetzen, um Risiken besser einschätzen zu können?

4.) Optimierung des Situationsvergleichs

Da der momentane Situationsvergleich voraussetzt, dass die zu vergleichenden Systemzustände gleich groß sind damit Standarddistanzmaße eingesetzt werden können, wäre hier eine Verwendung und Evaluation des Ansatzes Gardner et al. ([GKDS14]) angebracht. Dieser unterstützt nach eigener Aussage den Vergleich von unsortierten Deskriptoren mit unterschiedlichen Größen und ist nicht anfällig gegenüber Ausreißern. Bei positiver Evaluation könnten zusätzliche Objekte (zum Beispiel Schiffe) während eines Simulationslaufs zum Systemmodell hinzugefügt werden ohne dadurch den Situationsvergleich zu verfälschen.

5.) Sensitivitätsanalyse der Operatoren

In dieser Arbeit wurde ausgehend von Ansätzen zur Wahrscheinlichkeitsbestimmung von Risiken eine Vorauswahl bzgl. der verwendeten Operatoren für „Und“- und „Oder“ Gates getroffen. In zukünftigen Arbeiten wäre mittels einer Sensitivitätsanalyse zu betrachten ob eine unterschiedliche Auswahl für unterschiedliche untersuchte Szenarien nützlich ist bzw. bessere Ergebnisse liefert.

Abschließend lässt sich sagen, dass mit dem entwickelten Ansatz eine Möglichkeit geschaffen wurde, Distanzfunktionen strukturiert zu erstellen und gezeigt wurde wie die Distanz zu risikoreichen Situationen in den Läufen einer Co-Simulation ermittelbar ist. Zusätzlich ließ sich die Frage beantworten wie die ermittelten Distanzen in einer Co-Simulation mittels des entwickelten DistriCT Frameworks verwendet werden können, um die zu analysierenden risikoreichen Situationen schneller zu erreichen.

Literaturverzeichnis

[Also00] ALSOC: *ALSOC Project homepage*. URL http://www.enq.ufrgs.br/trac/alsoc. - abgerufen am 2015-06-14. — ALSOC Project homepage

[Aust04] AUSTRALIAN DEFENCE SIMULATION OFFICE: Distributed Simulation Guide (2004)

[Aven14] AVEN, TERJE: *Uncertainty in risk assessment: the representation and treatment of uncertainties by probabilistic and non-probabilistic methods*, 2014 — ISBN 978-1-118-76303-2

[Bank10] BANKS, J.: *Discrete-event System Simulation* : Prentice Hall, 2010. — LCCN: 2010286363 — ISBN 978-0-13-606212-7

[Benk03] BENKER, H.: *Mathematische Optimierung mit Computeralgebrasystemen: Einführung für Ingenieure und Naturwissenschaftler und Wirtschaftswissenschaftler unter Anwendung von MATHEMATICA, MAPLE, MATHCAD, MATLAB und EXCEL*, *Engineering online library* : Springer Berlin Heidelberg, 2003 — ISBN 978-3-540-44118-2

[BGKW08] BETTER, MARCO ; GLOVER, FRED ; KOCHENBERGER, GARY A. ; WANG, HAIBO: Simulation Optimization: Applications in Risk Management. In: *International Journal of Information Technology and Decision Making* Bd. 7 (2008), Nr. 4, S. 571–587

[BiAz13] BIDGOLY, AMIR JALALY ; AZGOMI, MOHAMMAD ABDOLLAHI: A new technique for rare-event simulation based on partition of the region. In: *Turkish Journal of Electrical Engineering & Computer Sciences* Bd. 21 (2013), Nr. 2, S. 309–329

[Bier04] BIERWIRTH, MICHAEL: Schuldverteilung bei Schiffskollisionen (Diplomarbeit (FH)). Hochschule Bremen, 2004

[Bmwi16] BMWI: *Schlaglichter der Wirtschaftspolitik*, *Schlaglichter der Wirtschaftspolitik* (Monatsbericht Nr. 3/16) : Bundesministerium für Wirtschaft und Energie (BMWi), 2016

[Boss04] BOSSEL, H.: *Systeme, Dynamik, Simulation:* : Books on Demand, 2004 — ISBN 978-3-8334-0984-4

[BrCW12] BRAMBILLA, MARCO ; CABOT, JORDI ; WIMMER, MANUEL: Model-driven software engineering in practice. In: *Synthesis Lectures on Software Engineering* Bd. 1 (2012), Nr. 1, S. 1–182

[BüCC10] BÜTTCHER, STEFAN ; CLARKE, CHARLES ; CORMACK, GORDON V.: *Information Retrieval: Implementing and Evaluating Search Engines* : The MIT Press, 2010 — ISBN 0-262-02651-1

[Bund12] BUNDESSTELLE FÜR SEEUNFALLUNTERSUCHUNG: *Gemeinsame Untersuchung zur Kollision des unter maltesischer Flagge fahrenden Frachtschiffs MARTI PRINCESS mit dem unter deutscher Flagge fahrenden Containerschiffs RENATE SCHULTE vor der Insel Bozcaada 27. Juni 2009, Untersuchungsbericht zur Sicherheit auf See* (Untersuchungsbericht Nr. 03/2012) : Bundestelle für Seeunfalluntersuchung, 2012

[CaBr02] CAHILL, R.A. ; BRITAIN), NAUTICAL INSTITUTE (GREAT: *Collisions and Their Causes* : Nautical Institute, 2002 — ISBN 978-1-870077-60-6

[CaMa97] CARSON, YOLANDA ; MARIA, ANU: Simulation optimization: methods and applications. In: *Proceedings of the 29th conference on Winter simulation* : IEEE Computer Society, 1997, S. 118–126

[CrCS11] CRAIN, BEN ; CHEN, CHUN-HUNG ; SHORTLE, JOHN F.: Combining simulation allocation and optimal splitting for rare-event simulation optimization. In: *Simulation Conference (WSC), Proceedings of the 2011 Winter* : IEEE, 2011, S. 3998–4007

[Dapd12] DAPD NACHRICHTENAGENTUR: *Wieder tödlicher Unfall in Offshore-Windpark*. URL http://www.t-online.de/nachrichten/panorama/id_56148058/toedlicher-unfall-in-offshore-windpark.html. - abgerufen am 2016-08-09. — www.t-online.de

[DiHS14] DIBBERN, CHRISTOPH ; HAHN, AXEL ; SCHWEIGERT, SÖREN: Interoperability In Co-Simulatons Of Maritime Systems. In: *ECMS*, 2014, S. 71–77

[Din03] DIN (Hrsg.): *DIN 1055-100 Einwirkungen auf Tragwerke - Teil 100: Grundlagen der Tragwerksplanung - Sicherheitskonzept und Bemessungsregeln*. Bd. DIN 1055–100, Ausgabe 2001–03, 2003. — 00000

[Din06] DIN: *DIN EN 60812 Analysetechniken für die Funktionsfähigkeit von Systemen – Verfahren für die Fehlzustandsart- und –auswirkungsanalyse (FMEA)*. Berlin und Frankfurt am Main, Germany : Deutsches Institut für Normung e.V. (DIN) und Verband der Elektrotechnik Elektronik Informationstechnik e.V. (VDE), 2006

[Din12] DIN (Hrsg.): *DIN EN 1990 Eurocode: Grundlagen der Tragwerksplanung*. Bd. DIN EN 1990 Ausgabe 2010–12, 2012. — 00004

[Din81] DIN25424: *Fehlerbaumanalyse: DIN 25424. Methode und Bildzeichen*, 1981

[Din90] DIN: *Fehlerbaumanalyse: Fault tree analysis : DIN 25424. Handrechenverfahren zur Auswertung eines Fehlerbaumes*, *DIN-Normen: Deutsches Institut für Normung* : Beuth, 1990

[Dinn12] DINNA (Hrsg.): *DIN EN 1990/NA Nationaler Anhang - National festgelegte Parameter - Eurocode: Grundlagen der Tragwerksplanung*. Bd. DIN EN 1990/NA, Ausgabe 2010–12, 2012. — 00000

[DLSB12] DROSTE, RAINER ; LÄSCHE, CHRISTOPH ; SOBIECH, CILLI ; BÖDE, ECKARD ; HAHN, AXEL: Model-based risk assessment supporting development of HSE plans for safe offshore operations. In: *Formal Methods for Industrial Critical Systems* : Springer, 2012, S. 146–161

[Dros16] DROSTE, RAINER: *Modellbasierte Planung und Analyse von Offshore-Operationen (unveröffentlicht)*. Oldenburg, Carl-von-Ossietzky Universität Oldenburg, Dissertation, 2016

[Drpe06] DR. PETER LANGER: *DIN 1055-100 Einwirkungen auf Tragwerke Teil 100: Grundlagen der Tragwerksplanung, Sicherheitskonzept und Bemessungsregeln*, *Xella Neues Bauen* (Technischer Bericht Nr. 041/2006) : Xella Technologie- und Forschungsgesellschaft mbH, 2006

[Emso04] EMSO: *EMSO Environment for Modelling, Simulation, and Optimization*. URL http://www.vrtech.com.br/rps/emso.html. - abgerufen am 2015-06-14. — EMSO Environment for Modelling, Simulation, and Optimization

[Eric05] ERICSON, C.A.: *Hazard Analysis Techniques for System Safety* : Wiley, 2005 — ISBN 978-0-471-73941-8

[Ferb03] FERBER, R.: *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web* : dpunkt-Verlag, 2003 — ISBN 978-3-89864-213-2

[Flur90] FLURY, BERNARD D: Acceptance-rejection sampling made easy. In: *SIAM Review* Bd. 32 (1990), Nr. 3, S. 474–476

[FuTa71] FUJII, YAHEI ; TANAKA, KENICHI: Traffic Capacity. In: *Journal of Navigation* Bd. 24 (1971), Nr. 04, S. 543–552

[GKDS14] GARDNER, ANDREW ; KANNO, JINKO ; DUNCAN, CHRISTIAN ; SELMIC, RASTKO ; OTHERS: Measuring distance between unordered sets of different sizes. In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on* : IEEE, 2014, S. 137–143

[Glas93] GLASSERMAN, PAUL: Filtered monte carlo. In: *Mathematics of Operations Research* Bd. 18 (1993), Nr. 3, S. 610–634

[GlLa13] GLOVER, FRED ; LAGUNA, MANUEL: Tabu Search∗. In: PARDALOS, P. M. ; DU, D.-Z. ; GRAHAM, R. L. (Hrsg.): *Handbook of Combinatorial Optimization* : Springer New York, 2013 — ISBN 978-1-4419-7996-4, S. 3261–3362

[Good75] GOODWIN, ELISABETH M.: A Statistical Study of Ship Domains. In: *Journal of Navigation* Bd. 28 (1975), Nr. 03, S. 328–344

[GöSc96] GÖRG, CARMELITA ; SCHREIBER, FRIEDRICH: The RESTART/LRE method for rare event simulation. In: *Proceedings of the 28th conference on Winter simulation* : IEEE Computer Society, 1996, S. 390–397

[GPLG14] GOLLÜCKE, VOLKER ; PINKOWSKI, JAN ; LÄSCHE, CHRISTOPH ; GERWINN, SEBASTIAN ; HAHN, AXEL: Simulation-based Completeness Analysis and Adaption of Fault Trees. In: *SIMUL 2014, The Sixth International Conference on Advances in System Simulation*. Nice, France, 2014 — ISBN 978-1-61208-371-1, S. 228 to 235

[Gran04] GRANOWETTER, LEN: IEEE 1516 Compliance – Will the Real C++ API Please Stand Up? In: *MÄK Technologies* (2004)

[HeGt10] HENDRIX, ELIGIUS MT ; G.-TÓTH, BOGLÁRKA: *Introduction to nonlinear and global optimization*. Bd. 37 : Springer, 2010

[Heid95] HEIDELBERGER, PHILIP: Fast simulation of rare events in queueing and reliability models. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* Bd. 5 (1995), Nr. 1, S. 43–85

[HGBS15] HAHN, AXEL ; GOLLÜCKE, VOLKER ; BUSCHMANN, CARSTEN ; SCHWEIGERT, SÖREN: Virtual test bed for maritime safety assessment. In: *Scientific Journal of the Maritime University of Szczecin*. Szczecin, 2015

[HoKl79] HOPMANS, A.C.M. ; KLEIJNEN, J.P.C.: Importance sampling in systems simulation: a practical failure? In: *Mathematics and Computers in Simulation (MATCOM)* Bd. 21 (1979), Nr. 2, S. 209–220

[Iala10] IALA: *IALA TRAINING SEMINAR ON RISK MANAGEMENT: PAWSA, IWRAP Mk2 & SIMULATION*. St Germain en Laye, France : IALA, 2010

[Ieee00] IEEE: *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*. 345 East 47th Street, New York, NY 10017, USA : The Institute of Electrical and Electronics Engineers, Inc., 2000

[Ieee10] IEEE: *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*. 345 East 47th Street, New York, NY 10017, USA : The Institute of Electrical and Electronics Engineers, Inc., 2010

[Iksu06] IK-SUNG, KIM: AN ALGORITHM FOR FINDING THE DISTANCE BETWEEN TWO ELLIPSES. In: *Commun. Korean Math. Soc 21*, 2006, S. 559–567

[JeLS13] JEGOUREL, CYRILLE ; LEGAY, AXEL ; SEDWARDS, SEAN: Importance splitting for statistical model checking rare properties. In: *Computer Aided Verification* : Springer, 2013, S. 576–591

[JuSh06] JUNEJA, SANDEEP ; SHAHABUDDIN, PERWEZ: Rare-event simulation techniques: an introduction and recent advances. In: *Handbooks in operations research and management science* Bd. 13 (2006), S. 291–350

[KaBu09] KAMISKE, G.F. ; BURCKHARDT, W.: *Qualitätstechniken für Ingenieure* : Symposion, 2009 — ISBN 978-3-939707-62-2

[KaOl07] KABIRIAN, ALIREZA ; OLAFSSON, SIGURDUR: Allocation of simulation runs for simulation optimization. In: *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come* : IEEE Press, 2007, S. 363–371

[KBKM12] KRUSE, R. ; BORGELT, C. ; KLAWONN, F. ; MÖWES, C. ; RU\S S, G. ; STEINBRECHER, M.: *Computational Intelligence:*, *Computational Intelligence* : Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2012 — ISBN 978-3-8348-8299-8

[Klug07] KLUG, YVETTE: *Lastannahmen nach neuen Normen: Grundlagen, Erläuterungen, Praxisbeispiele ; Einwirkungen auf Tragwerke aus: Eigen- und Nutzlasten, Wind- und Schneelasten, Erdbebenlasten*, 2007 — ISBN 978-3-89932-130-2

[Köni13] KÖNIGS, H.P.: *IT-Risikomanagement mit System: Praxisorientiertes Management von Informationssicherheits- und IT-Risiken*, *Edition kes* : Springer Fachmedien Wiesbaden, 2013 — ISBN 978-3-8348-2165-2

[LäBP12] LÄSCHE, CHRISTOPH ; BÖDE, ECKARD ; PEIKENKAMP, THOMAS: A method for guided hazard identification and risk mitigation for offshore operations. In: *Computer Safety, Reliability, and Security* : Springer, 2012, S. 37–48

[LäGH13] LÄSCHE, CHRISTOPH ; GOLLÜCKE, VOLKER ; HAHN, AXEL: Using An HLA Simulation Environment For Safety Concept Verification Of Offshore Operations. In: *ECMS*, 2013, S. 156–162

[LaGP11] LAUER, CHRISTOPH ; GERMAN, REINHARD ; POLLMER, JENS: Fault tree synthesis from UML models for reliability analysis at early

design stages. In: *ACM SIGSOFT Software Engineering Notes* Bd. 36 (2011), Nr. 1, S. 1–8

[LePa07] LEE, G.W.U. ; PARKER, J.: *Managing Collision Avoidance at Sea: A Practical Guide* : Nautical Institute, 2007 — ISBN 978-1-870077-86-6

[LPGD14] LÄSCHE, CHRISTOPH ; PINKOWSKI, JAN ; GERWINN, SEBASTIAN ; DROSTE, RAINER ; HAHN, AXEL: Model-Based Risk Assessment of Offshore Operations. In: *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering* : American Society of Mechanical Engineers, 2014, S. V01BT01A010–V01BT01A010

[MaLG06] MARTÍ, RAFAEL ; LAGUNA, MANUEL ; GLOVER, FRED: Principles of scatter search. In: *European Journal of Operational Research* Bd. 169 (2006), Nr. 2, S. 359–372

[Mari12] MARINE SAFETY INVESTIGATION UNIT: *Joint investigation into the collision between the Maltese registered general cargo MARTI PRINCESS and the German registered container ship RENATE SCHULTE off Bozcaada Island* (Marine Safety Investigation Report Nr. 03/2012) : Marine Safety Investigation Unit, 2012

[Mari97] MARIA, ANU: Introduction to modeling and simulation. In: *Proceedings of the 29th conference on Winter simulation* : IEEE Computer Society, 1997, S. 7–13

[MeDD05] MERRICK, JASON R. W. ; VAN DORP, J. RENE ; DINESH, VARUN: Assessing Uncertainty in Simulation-Based Maritime Risk Assessment. In: *Risk Analysis* Bd. 25 (2005), Nr. 3, S. 731–743

[MoPG10] MORIO, JÉRÔME ; PASTEL, RUDY ; LE GLAND, FRANÇOIS: An overview of importance splitting for rare event simulation. In: *European Journal of Physics* Bd. 31 (2010), Nr. 5, S. 1295–1303

[Nakh13] NAKHAEIZADEH, G.: *Data Mining: Theoretische Aspekte und Anwendungen*, *Beiträge zur Wirtschaftsinformatik* : Physica-Verlag HD, 2013 — ISBN 978-3-642-86094-2

[NiSN01] NICOLA, VICTOR F ; SHAHABUDDIN, PERWEZ ; NAKAYAMA, MARVIN K: Techniques for fast simulation of models of highly dependable systems. In: *Reliability, IEEE Transactions on* Bd. 50 (2001), Nr. 3, S. 246–264

[Optq11] THE OPTQUEST ENGINE: *The OptQuest Engine*. URL http://www.opttek.com/OptQuest. - abgerufen am 2015-06-14. — OptQuest | OptTek Systems, Inc.

[Orga03] ORGANIZATION, INTERNATIONAL MARITIME: *COLREG: Convention on the International Regulations for Preventing Collisions at Sea, 1972*, *IMO Publication* : International Maritime Organization, 2003 — ISBN 978-92-801-4167-2

[PaDu02] PAI, G.J. ; DUGAN, J.B.: Automatic synthesis of dynamic fault trees from UML system models. In: *Software Reliability Engineering, 2002. ISSRE 2003. Proceedings. 13th International Symposium on*, 2002, S. 243 – 254

[PaLZ04] PADOVITZ, AMIR ; LOKE, SENG WAI ; ZASLAVSKY, ARKADY: Towards a theory of context spaces. In: *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on* : IEEE, 2004, S. 38–42

[Paul15] PAULWEBER, MICHAEL: ENABLE-S3 (2015)

[Pink15] PINKOWSKI, JAN: *Prozessgetriebene Risikoanalyse zur Bewertung maritimer Operationen*. Oldenburg, Carl-von-Ossietzky Universität Oldenburg, Dissertation, 2015

[Plan15] PLANT SIMULATION: *Plant Simulation*. URL http://www.plant-simulation.de/?gclid=Cj0KEQjwzPSrBRC_oOXfxPWP6t0BEiQARqav2OQQo8_Xb5B-iBS-zLZQ0TyqNfXTt7NH0KGFPEggQI8aAm-58P8HAQ. - abgerufen am 2015-06-14. — Simulation mit Plant Simulation

[PLHH13] PAGE, B. ; LIEBERT, H. ; HEYMANN, A. ; HILTY, L.M. ; HÄUSLEIN, A.: *Diskrete Simulation: Eine Einführung mit Modula-2*, *Springer-Lehrbuch* : Springer Berlin Heidelberg, 2013 — ISBN 978-3-642-76862-0

[POFC12] PUCH, STEFAN ; OSTERLOH, JAN-PATRICK ; FRÄNZLE, MARTIN ; CHRISTOPH LÄSCHE: Rapid Virtual–Human–in–the–Loop Simulation with the High Level Architecture. In: *Proceedings of Summer Computer Simulation Conference 2012 (SCSC 2012)*, *Simulation Series Vol* : Curran Associates, Inc., 2012 — ISBN 978-1-61839-984-7, S. 44–50

[Pres13] PRESSEMITTEILUNG RIFFGAT: *Tödlicher Unfall im Offshore-Windpark-Riffgat*. URL http://www.windkraft-journal.de/2013/07/15/toedlicher-unfall-im-offshore-windpark-riffgat/. — www.windkraft-journal.de

[PWFP13] PUCH, STEFAN ; WORTELEN, BERTRAM ; FRÄNZLE, MARTIN ; PEIKENKAMP, THOMAS: Evaluation of Drivers Interaction with Assistant Systems Using Criticality Driven Guided Simulation. In: DUFFY, V. (Hrsg.): *Digital Human Modeling and Applications in Health, Safety, Ergonomics, and Risk Management. Healthcare and Safety of the Environment and Transport*, *Lecture Notes in Computer Science*. Bd. 8025 : Springer Berlin Heidelberg, 2013 — ISBN 978-3-642-39172-9, S. 108–117

[Rayc08] RAYCHAUDHURI, SAMIK: Introduction to Monte Carlo Simulation. In: *Proceedings of the 40th Conference on Winter Simulation*, *WSC*

'08. Miami, Florida : Winter Simulation Conference, 2008 — ISBN 978-1-4244-2708-6, S. 91–100

[RBSJ12] REIJSBERGEN, D. P. ; DE BOER, PIETER-TJERK ; SCHEINHARDT, W. R. W. ; JUNEJA, SANDEEP: Some advances in importance sampling of reliability models based on zero variance approximation (2012)

[Rene13] RENEWABLEUK: *Guidelines for Onshore and Offshore Wind Farms – Health & Safety in the Wind Energy Industry Sector*. London : RenewableUK, 2013

[RoTa60] ROGERS, DAVID J ; TANIMOTO, TAFFEE T: A computer program for classifying plants. In: *Science* Bd. 132 (1960), Nr. 3434, S. 1115–1118

[RoWa97] ROSS, KEITH W. ; WANG, JIE: Implementation of Monte Carlo integration for the analysis of product-form queueing networks. In: *Performance Evaluation* Bd. 29 (1997), Nr. 4, S. 273 – 292

[ScBG99] SCHMIDT, ALBRECHT ; BEIGL, MICHAEL ; GELLERSEN, HANS-W.: There is more to context than location. In: *Computers & Graphics* Bd. 23 (1999), Nr. 6, S. 893–901

[ScDH12] SCHWEIGERT, SÖREN ; DROSTE, RAINER ; HAHN, AXEL: Multi–Agenten basierte 3D Simulation für die Evaluierung von Offshore Operationen. In: Go-3D, 2012.

[Schw16] SCHWEIGERT, SÖREN: *Simulative Überprüfung von Sensordatenverarbeitungssystemen (unveröffentlicht)*. Oldenburg, Carl-von-Ossietzky Universität Oldenburg, Dissertation, 2016

[SGHB14] SCHWEIGERT, SÖREN ; GOLLÜCKE, VOLKER ; HAHN, AXEL ; BOLLES, ANDRÉ: HAGGIS: A modelling and simulation platform for e-Maritime technology assessment. In: . Istanbul, Turkey, 2014, S. 10

[Shah95] SHAHABUDDIN, PERWEZ: Rare Event Simulation in Stochastic Models. In: *Proceedings of the 27th Conference on Winter Simulation*, *WSC '95*. Washington, DC, USA : IEEE Computer Society, 1995 — ISBN 0-7803-3018-8, S. 178–185

[ShBe06] SHYAM, SMITHA ; BERTACCO, VALERIA: Distance-guided hybrid verification with GUIDO. In: *Proceedings of the conference on Design, automation and test in Europe: Proceedings* : European Design and Automation Association, 2006, S. 1211–1216

[SoOt00] SOLEY, RICHARD ; OTHERS: Model driven architecture. In: *OMG white paper* Bd. 308 (2000), S. 308

[Spec07] SPECIFICATION, OMG ADOPTED: Meta Object Facility (MOF) Core Specification. In: *Object Management Group pct/07-08-04* (2007)

[Stam03] STAMATIS, D.H.: *Failure Mode and Effect Analysis: FMEA from Theory to Execution* : ASQ Quality Press, 2003. — LCCN: 2003005126 — ISBN 978-0-87389-598-9

[Stän11] STÄNDER, TOBIAS: *Eine modellbasierte Methode zur Objektivierung der Risikoanalyse nach ISO 26262*. Braunschweig, Technische Universität Braunschweig, 2011

[Swis15] SWISS RE: *Weltweite Versicherungsschäden verursacht durch maritime Katastrophen bis 2014*, *Sigma - Natur- und Man-made-Katastrophen* (Statistik Nr. 2 / 2015) : Swiss Re, 2015

[ThMC96] TH.W.CH. HUIBERS ; M. LALMAS ; C.J. VAN RIJSBERGEN: Information Retrieval and Situation Theory (1996)

[Thom14] THOMSEN, K.: *Offshore Wind: A Comprehensive Guide to Successful Offshore Wind Farm Installation* : Elsevier Science, 2014 — ISBN 978-0-12-409594-6

[Usde98] US DEPARTMENT OF DEFENSE: *High-Level Architecture Rules, Version 1.3*. 345 East 47th Street, New York, NY 10017, USA : The Institute of Electrical and Electronics Engineers, Inc., 1998

[VDFM02] VESELY, W. ; DUGAN, J. ; FRAGOLA, J. ; MINARICK ; RAILSBACK, J.: *Fault Tree Handbook with Aerospace Applications* (Handbook). Washington, DC : National Aeronautics and Space Administration, 2002

[ViVi94] VILLEN-ALTAMIRANO, M. ; VILLEN-ALTAMIRANO, J.: RESTART: a straightforward method for fast simulation of rare events. In: : IEEE, 1994 — ISBN 0-7803-2109-X, S. 282–289

[VMGF94] VILLÉN-ALTAMIRANO, M. ; MARTÍNEZ-MARRÓN, A. ; GAMO, J. ; FERNÁNDEZ-CUESTA, F.: Enhancement of the Accelerated Simulation Method RESTART by Considering Multiple Thresholds. In: *Teletraffic Science and Engineering*. Bd. 1 : Elsevier, 1994 — ISBN 978-0-444-82031-0, S. 797–810

[Wagn12] WAGNER, SARAH: *Ausbau der Windenergie forderte schon drei Tote*. URL http://www.focus.de/politik/deutschland/atomausstieg/drei-tote-durch-windparks-atomausstieg-kann-toedlich-sein_aid_741092.html. - abgerufen am 2016-09-08. — www.focus.de

[WuSA02] WU, HUADONG ; SIEGEL, MEL ; ABLAY, SEVIM: Sensor fusion for context understanding. In: *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*. Bd. 1 : IEEE, 2002, S. 13–17

Eidesstattliche Erklärung

Hiermit versichere ich, Volker GOLLÜCKE, dass ich die von mir vorgelegte Arbeit mit dem Titel, *'Bewertung von Simulationszuständen für eine gezielte Analyse risikoreicher Systeme'* selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

---------- ----------

Datum Volker GOLLÜCKE