Weiterentwicklung der Methodik zur automatisierten Integration übergreifender Einwirkungen in PSA-Modelle der Stufe 1

GRS - 454

Gesellschaft für Anlagen-
und Reaktorsicherheit
(GRS) gGmbH

Weiterentwicklung der Methodik zur automatisierten Integration übergreifender Einwirkungen in PSA-Modelle der Stufe 1

Technischer Fachbericht

Nadine Berner
Joachim Herb

März 2017

Anmerkung:

Das diesem Bericht zugrunde liegende F&E-Vorhaben wurde mit Mitteln des Bundesministeriums für Wirtschaft und Energie (BMWi) unter dem Kennzeichen RS 1539 durchgeführt.

GRS - 454
ISBN 978-3-946607-36-6

Deskriptoren

Automatisierung, Brand, Duplikation, Fehlerbaummodellierung, Integration, Modifikation, PSA-Werkzeuge, Überflutung, übergreifende Einwirkungen

Kurzfassung

Im Rahmen des vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderten Forschungs- und Entwicklungsprojekts RS1539 wurde die Methodik zur automatisierten Integration von übergreifenden Einwirkungen in PSA-Modelle der Stufe 1 weiterentwickelt. Das dabei erarbeitete Analysewerkzeug pyRiskRobot bietet die methodischen Grundlagen, um ein im Prinzip generisches Spektrum übergreifender Einwirkungen von innen und außen auf komplexe PSA-Anlagenmodelle abzubilden.

Die Reimplementierung des Werkzeugs in die Programmiersprache Python erweitert die Einsatzmöglichkeiten und erhöht die Anwenderfreundlichkeit von pyRiskRobot im Vergleich zur Ruby-basierten Vorgängerversion RiskRobot. Ebenso wurde die methodische Weiterentwicklung der Funktionen zur topologischen Modellierung von Fehlerbäumen und zur probabilistischen Spezifikation modifizierter Fehlerbaumelemente weitergeführt. In Folge dieser Reimplementierung und Weiterentwicklungen können Fehlerbaummodellierungen verschiedener Komplexität systematisch erzeugt, flexibel in bestehende PSA-Modelle integriert und zusammenhängende Topologien automatisiert dupliziert werden. Damit ermöglicht pyRiskRobot die effiziente und zuverlässige Realisierung einwirkungsspezifischer, meist arbeitsintensiver Modifikationen von PSA-Modellen.

Des Weiteren wurde pyRiskRobot zu einer dynamischen Schnittstelle zwischen der Datensammlung der möglichen Auswirkungen einer übergreifenden Einwirkung auf PSA-relevante Komponenten und der Datenbank eines PSA-Anlagenmodells erweitert. Durch diese Konzeptionierung können neben dem Zugriff auf die Daten auch weiterführende Analysen der Daten vor Integration in die PSA-Topologie durchgeführt werden.

Die entwickelten Funktionen von pyRiskRobot wurden anhand von Referenzanwendungen, z. B. zur Modellierung eines anlageninternen Brandes, im Vergleich zur Vorgängerversion RiskRobot erprobt. Basierend auf den Anforderungen weiterer Anwendungen, u. a. zur Modellierung einer anlagenexternen Überflutung, wurde die bestehende Methodensammlung zur automatisierten Modifikation von Fehlerbaum-Topologien erweitert. Die fallspezifisch erarbeiteten Modellierungsansätze führten dabei zu einer kontinuierlichen Konvergenz hin zu methodisch-strukturierten Funktionenklassen.

Die Weiterentwicklungsarbeiten erlauben es, insbesondere für die Anwendung in periodischen Sicherheitsüberprüfungen relevante Fragestellungen zu untersuchen, deren adäquate Modellierung einen hohen Detaillierungsgrad erfordert. Das Analysewerkzeug pyRiskRobot bietet dabei die notwendigen Methoden und weiterführende Strategien, um übergreifende Einwirkungen und deren Kombinationen in PSA-Anlagenmodelle realistischer Komplexität integrieren zu können.

Abstract

In the course of the research and development project RS1539 funded by the German Federal Ministry for Economics and Energy (BMWi) the methodology for the automated integration of hazards in Level 1 PSA models has been enhanced. Thereby, the analysis tool pyRiskRobot provides the methodological framework for mapping a generic spectrum of internal and external hazards onto complex PSA plant models.

The reimplementation of the software tool via the programming language python extends the applicability and facilitates the handling of pyRiskRobot in comparison to the previous Ruby-based version RiskRobot. Moreover, the development of functions to perform the topological modelling of fault trees and the probabilistic specification of modified fault tree elements have been continued. Due to the reimplementation and further developments, the tool enables to systematically generate fault trees of varying complexity, to flexibly integrate fault trees in existing PSA models and to automatically duplicate interconnected topologies. Thus, pyRiskRobot allows the efficient and traceable realization of hazard specific, usually laborious modifications of PSA models.

In addition, pyRiskRobot has been extended to serve as a functional interface between the data compilations comprising the potential influences of hazards on PSA relevant components and the data base of a PSA plant model. Based on this conceptual design, additional analyses of the data can be carried out *prior* to the integration within the PSA model topology. The reimplemented functionalities of pyRiskRobot have been validated with respect to reference applications, such as the modelling of an internal fire scenario, against the previous version RiskRobot. The existing method collection for the automated modification of fault tree topologies has been extended based on the requirements for further applications, among others the modelling of an external flooding scenario. The deduced hazard specific modelling approaches have been used to pursue a continuous convergence towards methodologically structured function classes.

The performed developments provide the framework to investigate relevant scenarios that need to be addressed in the context of periodic safety reviews and that may require an adequate modelling at a high level of detail. The analysis tool pyRiskRobot provides the necessary methods and practical strategies in order to integrate hazards and hazard combinations in PSA plant models of realistic complexity.

Inhaltsverzeichnis

1 Einleitung

Die Methodik der probabilistischen Sicherheitsanalyse (PSA) erlaubt es, Risikomaße für komplexe Anlagensysteme unter der Annahme verschiedener Szenarien bzw. auslösender Ereignisse abzuschätzen. Die Berechnung der Beiträge der einzelnen Teilsysteme zum Gesamtrisiko bietet die analytische Grundlage, um sicherheitsrelevante Systeme zu optimieren und so das Sicherheitsniveau von Anlagensystemen kontinuierlich zu verbessern.

Einen aktuellen Forschungsschwerpunkt der PSA für Kernkraftwerke stellt die systematische Berücksichtigung übergreifender Einwirkungen (Englisch: hazards) und Ereigniskombinationen mit solchen Einwirkungen dar. Dafür werden im Folgenden PSA-Modelle der Stufe 1 betrachtet und somit das Verhalten einer Anlage für einwirkungsbedingte auslösende Ereignisse mit der Zielsetzung untersucht, die Kern- bzw. Brennstabschadenshäufigkeit (Englisch: core damage frequency, CDF bzw. fuel damage frequency, FDF) pro Jahr zu ermitteln. Eine übergreifende Einwirkung bezeichnet dabei eine Einwirkung von innen oder außen, die das Potenzial für redundanzübergreifende Ausfälle hat und somit zu einem auslösenden Ereignis führen bzw. die Zuverlässigkeit mehrerer sicherheitsrelevanter Systeme beeinflussen kann. Das Gefahrenpotenzial einer übergreifenden Einwirkung von innen (z. B. anlageninterner Brand, Explosion, Überflutung) oder außen (naturbedingte Einwirkungen, wie z. B. Erdbeben, anlagenexterne Überflutung, oder zivilisatorische Einwirkungen, wie z. B. Explosionsdruckwelle, Flugzeugabsturz) kann auf Grund ihrer jeweils charakteristischen Wirkmechanismen die gesamte Anlage beeinflussen /FAK 05/.

Um den Effekt einer übergreifenden Einwirkung auf ein Anlagensystem probabilistisch untersuchen zu können, ist es erforderlich, die Auswirkungen auf sicherheitstechnisch relevante Systeme, Strukturen und Komponenten (Englisch: systems, structures and components, SSCs) in das PSA-Modell der Anlage zu integrieren. Eine detaillierte Abbildung solcher Auswirkungen in Modelle realistischer Komplexität erfordert in der Regel eine enorme Menge schematisch ähnlicher Fehlerbaummodifikationen. Solche Modifikationen können beispielsweise die Ergänzung eines Fehlerbaums um ein zusätzliches Basisereignis oder um eine zusätzliche Fehlerbaumstruktur sein. Die große Anzahl an Veränderungen ergibt sich insbesondere für die PSA von Kernkraftwerken, die nur durch PSA-Modelle hoher Komplexität adäquat beschrieben werden können. Des Weiteren besteht dabei oftmals die Notwendigkeit, bestimmte Aspekte übergrei-

fender Einwirkungen einschließlich Ereigniskombinationen solcher Einwirkungen mit einem hohen Detaillierungsgrad zu berücksichtigen.

Damit solche arbeitsintensiven Modellierungen mit sinnvollem Aufwand umsetzbar sind, hat die GRS Methoden und Werkzeuge entwickelt, um die Modifikation von PSA-Modellen automatisiert und nachvollziehbar durchführen zu können /HER 15a/. Basierend auf diesen Arbeiten wurde eine weiterentwickelte Version der Software zur automatisierten Fehlerbaummodifikation erstellt, um auf die wachsenden Anforderungen an das zukünftige Einsatzspektrum für das Analysewerkzeug zu reagieren.

Das nun Python-basierte Programm pyRiskRobot ist eine konsequente Weiterentwicklung der Ruby-basierten Vorgängerversion RiskRobot und umfasst die bisherigen Funktionen der Fehlerbaummodifikation eines PSA-Anlagenmodells. Grundlegende Voraussetzung ist hierbei weiterhin, dass das zu bearbeitende PSA-Modell mit Hilfe der Analysesoftware RiskSpectrum® realisiert wurde. Diese Abhängigkeit ist eine Folge der angewandten Methodik von (py)RiskRobot, direkt auf die Datenbank des PSA-Anlagenmodells zuzugreifen. Aufgrund der fehlenden Programmierschnittstelle (Englisch: application programming interface, API) ist ein direkter Zugriff auf die MSSQL-Datenbank von RiskSpectrum® notwendig. Die angewandte Strategie bietet dabei eine dynamische, effiziente und nachzuvollziehende Vorgehensweise, um auch komplexe Aspekte im Kontext einer statischen PSA zu modellieren.

Um diese Modellierungsarbeiten kontrolliert und robust im PSA-Anlagenmodell durchführen zu können, wurde bei der Reimplementierung von pyRiskRobot ein objektorientiertes Programmierungskonzept verfolgt. Darüber hinaus wurden die generischen Funktionalitäten auf frei verfügbaren Python-Bibliotheken aufgebaut, welche bereits verbreitet Einsatz finden und deshalb einen hohen Entwicklungsgrad erreicht haben. Basierend auf den bei der Entwicklung von RiskRobot und den bisherigen einwirkungsspezifischen Anwendungsbeispielen gesammelten Erfahrungen wurde pyRiskRobot so umgesetzt, dass es je nach Bedarf als funktionale Schnittstelle zwischen den Datensammlungen der übergreifenden Einwirkungen und der Datenbanken des PSA-Anlagenmodells dient. Im Rahmen dieser vielfältigen Anwendungen konnte pyRiskRobot um weitere Funktionen zur effizienten Fehlerbaummodifikation erweitert werden, welche in modularer Weise realisiert wurden, und in Folge dessen künftig relativ unkompliziert gepflegt und weiterentwickelt werden können.

Das Gesamtziel des vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderten Forschungs- und Entwicklungsvorhabens RS1539 besteht darin, die vorhandenen Methoden und Werkzeuge für PSA so weiterzuentwickeln, dass im Prinzip ein generisches Spektrum von übergreifenden Einwirkungen in PSA-Modelle integriert werden kann. Die Weiterentwicklung des Analysewerkzeugs pyRiskRobot ist Voraussetzung für die technische Umsetzbarkeit methodischer Ansätze, um übergreifende Einwirkungen von innen wie außen, einschließlich Einwirkungskombinationen automatisiert auf das Anlagensystem abzubilden. Die hier beschriebenen Entwicklungsarbeiten dienen sowohl der praktischen Erweiterung des Wissensstands als auch der Untersuchung möglicher technischer Grenzen bei der Umsetzung komplexer Modellierungsansätze innerhalb eines PSA-Modells. Letztendlich bietet pyRiskRobot Methoden und Strategien, um aktuelle wissenschaftliche Fragestellungen der probabilistischen Sicherheitsbewertung zu behandeln, die sich mit dem Einfluss übergreifender Einwirkungen und Einwirkungskombinationen beschäftigen und deren praktische Umsetzung ohne Automatisierung aufgrund des hohen manuellen Arbeitsaufwands meist nicht realisierbar wäre.

2 Methodik zur automatisierten Integration übergreifender Einwirkungen in PSA-Modelle der Stufe 1

Die GRS verwendet, wie die meisten deutschen und viele ausländische Institutionen, die probabilistische Sicherheitsanalysen durchführen, das Programm RiskSpectrum® /SCA 12/ zur Durchführung von PSA der Stufe 1. Diese Software ist auch bei Gutachtern und Aufsichtsbehörden weit verbreitet und dient daher innerhalb der GRS als Analyse- und Modellierwerkzeug für eine PSA der Stufe 1. Im Allgemeinen basiert die PSA-Modellierung auf der Kombination von Fehler- und Ereignisbäumen. Dabei beschreiben die Fehlerbäume die Zuverlässigkeit und Abhängigkeiten von SSCs und geben somit die fachlich-technischen Hypothesen bezüglich der sicherheitsrelevanten Anlagensysteme wieder. Die Einbindung der Fehlerbäume in spezifische Ereignisablaufdiagramme erlaubt die probabilistische Analyse des abgebildeten Anlagenverhaltens für die angenommenen Ereignisabläufe.

Viele aktuelle wissenschaftliche PSA-Fragestellungen erfordern teils komplexe Modellierungsansätze neuer und/oder komplexe Modifikationen bestehender Anlagenmodelle. Dies ist zum Beispiel bei einer realistischen Berücksichtigung übergreifender Einwirkungen von innen oder außen in der PSA, insbesondere bei der Untersuchung von Ereigniskombinationen, die auch übergreifende Einwirkungen beinhalten, erforderlich. Um auch bei solch aufwendigen Aufgaben eine effiziente Modellierung zu gewährleisten, wurde von der GRS ein methodischer Ansatz entwickelt, welcher eine automatisierte Modifikation von PSA-Modellen – realisiert mittels RiskSpectrum® – ermöglicht. Eine schematische Übersicht der dazu entwickelten Analysewerkzeuge ist in Abb. 2.1 dargestellt.

Das Seitens der GRS entwickelte Analysewerkzeug RiskRobot /HER 15a/ ist in der Programmiersprache Ruby implementiert und ermöglicht eine automatisierte Modifikation von Fehlerbäumen. Die Makrosprache (Englisch: domain specific language) Risk Lang ermöglicht dabei den direkten Zugriff auf die MSSQL-Datenbank des elektronischen Anlagenmodells (d. h. RiskSpectrum®-Projektdatei) mittels der frei verfügbaren Ruby-Bibliothek ActiveRecord /MAR 07/. Basierend auf den abgeleiteten, objektrelationalen Abbildungen (Englisch: object relational mapping, ORM) der Datenbank erlaubt RiskLang per Anwendungsskript Operationen direkt in der Datenbank auszuführen und somit die darin verwalteten Fehlerbäume zu verändern. In einem Anwendungsskript können artgleiche Änderungen zu Algorithmen zusammengefasst und iterativ abgearbeitet werden.

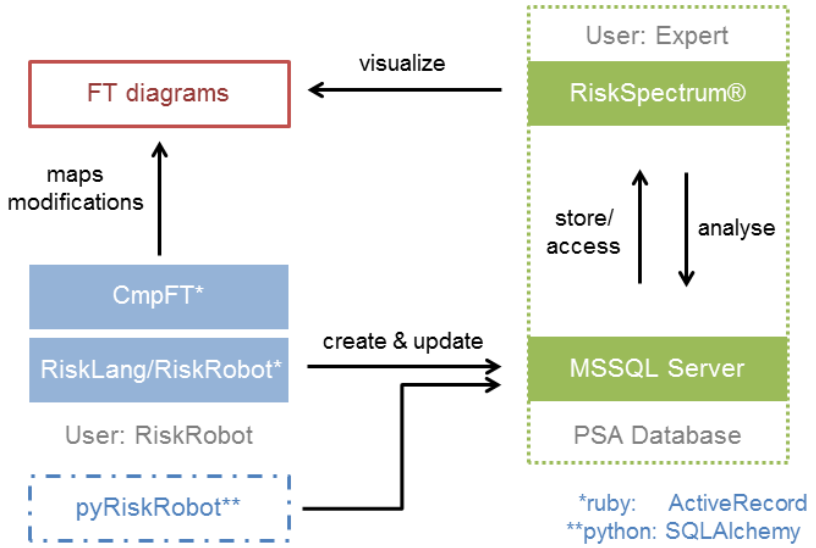

Abb. 2.1 Schematische Darstellung der GRS-Werkzeuge RiskLang, (py)RiskRobot und CmpFT zur automatisierten Modifikation und zum automatisierten Vergleich von PSA-Modellen

Diese Form der prozeduralen Modifikation kann somit effizient, da automatisiert, ausgeführt werden. Neben der Einsparung der Bearbeitungszeit ergibt sich durch die Automatisierung auch eine Minimierung der möglichen manuellen Fehlerquellen (z. B. des Verwechselns redundanzspezifischer Basisereignisse bei der manuellen Modellierung). Aus diesem Grund führt eine Automatisierung im Allgemeinen zusätzlich zu einer verminderten Fehleranfälligkeit bei der praktischen Umsetzung der PSA-Modellierung. Die Modifikationen der Fehlerbäume werden unter einem bestimmten Nutzernamen (z. B. „*RiskRobot*“) in der Datenbank ausgeführt und somit ebenso protokolliert wie die Arbeiten am PSA-Modell die Nativ über die Benutzeroberfläche (Englisch: graphical user interface, GUI) von RiskSpectrum® ausgeführt werden. Dadurch können die Veränderungen jederzeit zurückverfolgt und individuell freigegeben werden. Diese Vorgehensweise bietet die praktische Grundlage, um komplexe, strukturell ähnliche Modellierungen im Rahmen eines PSA-Modells zu leisten, indem manuelle Modellierungen teilweise von einem RiskRobot-Anwendungsskript übernommen werden.

Um die große Menge an ausgeführten Modifikationen in PSA-Modellen hoher Komplexität, wie sie für eine realistische Modellierung notwendig ist, nachzuvollziehen und auf ihre Konsistenz zu überprüfen, wurde das GRS-Werkzeug CmpFT (*Comp*are *F*ault *T*rees) entwickelt /HER 11/. Dabei werden zwei PSA-Modelle (d. h. Referenzmodell und modifiziertes Modell) automatisiert auf Abweichungen voneinander untersucht. Es werden dabei sowohl Unterschiede in der Struktur der Fehler- bzw. Ereignisbäume als

auch unterschiedliche Basisereignisse bzw. Systemfunktionen erkannt. Die erfassten Änderungen werden durch farbliche Markierung in den als PDF-Dokument ausgegebenen Fehlerbaum- bzw. Ereignisablaufdiagrammen dargestellt. Die ausgeführten Modifikationen konnten zum Zeitpunkt der Werkzeugentwicklung innerhalb der GUI von RiskSpectrum® weder systematisch nachverfolgt werden, noch konnten verschiedene PSA-Modelle einander gegenübergestellt werden. Daher ist CmpFT ein wichtiges Werkzeug zur automatisierten und visuellen Validierung durchgeführter Fehlerbaummodifikationen hoher Komplexität.

Im Rahmen von Forschungs- und Entwicklungsvorhaben hat sich die GRS eine umfassende Kompetenz in der PSA-Modellierung übergreifender Einwirkungen von innen und außen erarbeitet /BAB 11/, /TUE 15/, /TUE 15a/, /TUE 15b/. Für eine effiziente und zuverlässige Umsetzung der erarbeiteten Modellierungsansätze bieten die Werkzeuge RiskRobot und CmpFT die methodisch-technischen Grundlagen. Mit der zunehmenden Modellkomplexität zur Beantwortung aktueller wissenschaftlicher Fragestellungen, zum Beispiel die Berücksichtigung von Ereigniskombinationen mit Beteiligung übergreifender Einwirkungen, steigt allerdings auch der Anspruch an die Flexibilität und Handhabbarkeit der verfügbaren Analysewerkzeuge. Deshalb wurde das Konzept von RiskRobot weiterentwickelt, indem es in der Programmiersprache Python als pyRiskRobot reimplementiert, überarbeitet und anhand verschiedener einwirkungsspezifischer Modellierungsaufgaben erweitert wurde.

2.1 Weiterentwicklung der Methodik zur automatisierten Integration übergreifender Einwirkungen

Im Prinzip kann ein PSA-Modell der Stufe 1 als eine Menge von miteinander verknüpften Topologien, bestehend aus Fehler- und Ereignisbäumen, betrachtet werden. Um die Zuverlässigkeit der Anlage zu modellieren, dienen Fehlerbäume als Systemmodelle zur Berechnung der Nichtverfügbarkeit der einzelnen sicherheitsrelevanten SSCs. Die Fehlerbäume kombinieren alle zugeordneten Basisereignisse, die zur Nichtverfügbarkeit eines/r SSC führen können. Diese Basisereignisse werden mittels probabilistischer Modelle spezifiziert. Durch Propagation der resultierenden Wahrscheinlichkeiten aller verknüpften Basisereignisse durch die Baumlogik wird die Nichtverfügbarkeit eines/r SSC berechnet. Das Verhalten der Anlage wird durch Ereignisbäume modelliert, um fehlerfreie und fehlerhafte Ereignisabläufe, bestimmt durch Verfügbarkeit oder Nichtverfügbarkeit relevanter Systeme, zu berücksichtigen. Die Ereignisbäume umfassen al-

le potenziellen Sequenzen ausgehend vom auslösenden Ereignis (Englisch: initiating event, IE) eines Szenarios bis zu den zugeordneten Konsequenzen. Die einzelnen Ereignisse bzw. Abzweigungen des Ereignisbaumes stellen dabei die Verknüpfungspunkte zu den Fehlerbäumen dar. Die Kombination von Fehler- und Ereignisbäumen ergibt das eigentliche PSA-(Anlagen-)Modell, welches die Risikoanalyse einer Anlage für verschiedene betrachtete Szenarien erlaubt. Die zur PSA-Modellierung notwendigen Aspekte der topologischen Modifikation und probabilistischen Spezifikation beschreiben die beiden grundlegenden Aufgabenbereiche des GRS-Analysewerkzeugs pyRiskRobot wie in Abb. 2.2 skizziert.

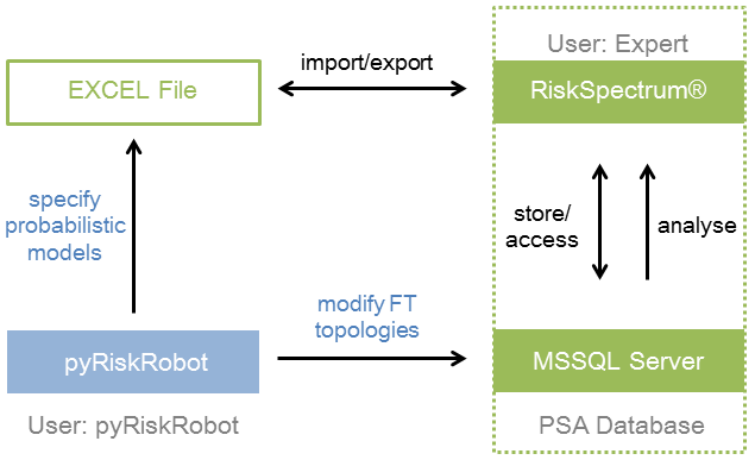

Abb. 2.2 Darstellung der grundsätzlichen Aufgaben der topologischen Modifikation und probabilistischen Spezifikation des Werkzeugs pyRiskRobot bei der automatisierten PSA-Modellierung

Die Hauptaufgabe von pyRiskRobot stellt die automatisierte Modifikation der Fehlerbaum-Topologien dar. Die dafür notwendigen Operationen werden direkt auf der Datenbank des PSA-Modells ausgeführt, basierend auf derselben Vorgehensweise wie RiskRobot. Die Spezifikation der modifizierten Basisereignisse, d. h. Typ und Parametrisierung der angenommenen probabilistischen Modelle, wird dann mittels einer MS EXCEL®-Datei über die Importfunktion von RiskSpectrum® eingelesen. Somit besteht eine weitere Aufgabe von pyRiskRobot in der Vorbereitung der Importdatei eines PSA-Modells entsprechend der zuvor ausgeführten topologischen Modifikationen. Diese Vorgehensweise hat den Vorteil, dass die Arbeitsschritte Modifikation und Spezifikation voneinander entkoppelt sind und schrittweise durchgeführt werden. Mögliche Inkonsistenzen in der Namensgebung werden beim Importprozess erkannt, als Fehlermeldungen in der GUI von RiskSpectrum® ausgegeben und folglich nicht in die Datenbank übernommen.

Basierend auf einem entsprechend modifizierten und konsistent spezifizierten Modell können dann die PSA wie auch weiterführende manuelle Modellierungen Nativ in RiskSpectrum® ausgeführt werden. Dies bedeutet, dass komplexe Modellierungsaufgaben, die sich sinnvoll in Skriptform abstrahieren lassen, mit den Mitteln von pyRiskRobot ausgeführt werden können. Alle anderen Aufgaben können dann weiter mit den von RiskSpectrum® zur Verfügung gestellten Funktionen und Methoden innerhalb der GUI umgesetzt werden. Diese Vorgehensweise ermöglicht insbesondere eine effiziente und sukzessive Erhöhung der Modellierungstiefe (d. h. des Detaillierungsgrades) einer oder mehrerer übergreifender Einwirkungen im Kontext komplexer PSA-Modelle.

2.2 Entwicklung des Analysewerkzeugs pyRiskRobot

Im Folgenden werden sowohl die Arbeitsumgebung als auch die objektorientierte Programmierung (OOP) von pyRiskRobot beschrieben. Bei der Reimplementierung des Werkzeugs mittels der Interpretersprache Python wurde der dynamische Zugriff auf die Datenbank konsequent von den topologischen Modellierungsmethoden getrennt, so dass die Methoden modular in den Anwendungsskripten eingebunden werden können. Dies bildet die Voraussetzung für ein interaktives Arbeiten mit pyRiskRobot im Rahmen eines Notebookformates und für die Konzeptionierung von pyRiskRobot als funktionale Schnittstelle zwischen den Datensammlungen der übergreifenden Einwirkungen und der Datenbank des PSA-Modells der Stufe 1.

2.2.1 Implementierung mittels der Programmiersprache Python

Das prinzipielle Vorgehen bei der Umstellung auf pyRiskRobot als Python-basiertes Programm orientierte sich an den Erfahrungen und Strategien, die im Rahmen der Ruby-basierten Vorgängerversion RiskRobot erarbeitet wurden. Der allgemeine Aufbau der formulierten Klassenstruktur von pyRiskRobot spiegelt dabei die Abstraktionsschichten von RiskRobot wieder /HER 15a/. In Abb. 2.3 ist die hierarchische Klassenstruktur von pyRiskRobot in Form eines Software-Layer-Diagrammes dargestellt. Die Aufgaben der einzelnen Schichten werden im Folgenden, ausgehend von der untersten bis zur obersten Abstraktionsebene erläutert.

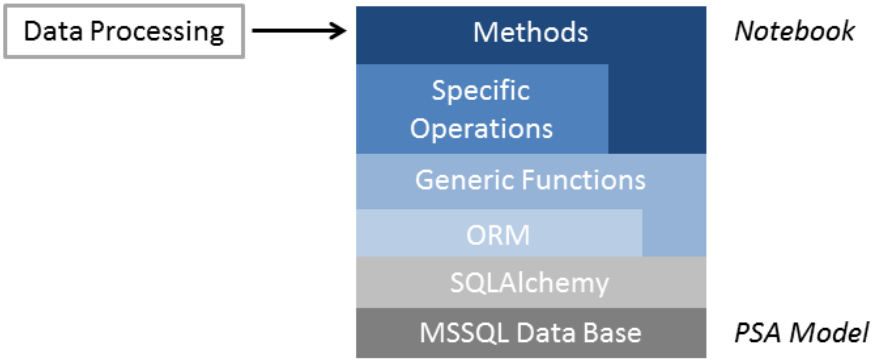

Abb. 2.3 Software-Layer-Diagramm des Analysewerkzeugs pyRiskRobot

SQLAlchemy: Zugriff auf die Datenbank

Um die Verbindung zur MSSQL-Datenbank des PSA-Modells herzustellen, verwendet pyRiskRobot die frei verfügbare Python-Bibliothek SQLAlchemy /SQL 17/. Diese Bibliothek ermöglicht es, auf der Ebene von Python-Objekten Informationen aus der PSA-Datenbank zu extrahieren sowie in der Datenbank Einträge zu erstellen, zu ändern und zu löschen. Die Bibliothek verwaltet die Verbindung zum Datenbankserver und übernimmt die Erstellung der expliziten SQL (Englisch: structured query language)-Kommandos, die an den Datenbankserver geschickt werden.

ORM: Objektrelationale Abbildung der Datenbank

Damit die einzelnen Einträge in den Tabellen der Datenbank korrekt als untereinander referenzierte Klassenobjekte interpretiert werden können, werden die von der Datenbank verwendeten Objektabbildungen in pyRiskRobot formuliert. Die explizite Aufschlüsselung des ORM wurde bereits im Rahmen von RiskRobot erarbeitet. Da die verwendete Bibliothek SQLAlchemy das ORM als Persistente Python-Objekte abbildet, kann pyRiskRobot mit den Einträgen der Datenbank in Form regulärer Python-Objekte arbeiten. Diese Tatsache stellt den eigentlichen Vorteil der Reimplementierung von pyRiskRobot dar, da nun alle Aspekte der Interpretersprache Python für das Arbeiten mit den Datenbankeinträgen genutzt werden können.

Bei der Implementierung von pyRiskRobot wurden die Aufgabenbereiche der expliziten SQL-Kommandos und der Aufschlüsselung des ORM konsequent voneinander getrennt. Die strikte Unterteilung der Klassen erlaubt es, die Vorteile einer OOP optimal ausnutzen zu können: Abkapselung, Wiederverwendung und Vererbung. Die Abkapselung ist vor allem in Bezug auf den kritischen Zugriff via SQL-Kommandos auf die Da-

tenbank essentiell. Da solche direkten Zugriffe die Datenbank korrumpieren können, ist es wichtig, diese möglichst einfach und robust zu gestalten. Dadurch können potenzielle Fehler im Rahmen der Softwareentwicklung rasch ausfindig gemacht und behoben werden.

Generic Functions: Semantische Verarbeitung der Datenbankobjekte

Aufbauend auf der adäquaten Abbildung der Datenbankeinträge als Python-Objekte besteht eine Hauptaufgabe von pyRiskRobot in der Bereitstellung einer generischen Funktionensammlung zur Fehlerbaummodellierung. Letztendlich entspricht dieser Kern von pyRiskRobot einer semantischen Schicht, die die generischen Funktionen der Fehlerbaummodellierung konsistent in der Semantik des Datenbank-ORM formuliert. Diese Klasse enthält die notwendigen Basisfunktionen zum Erstellen, Bearbeiten und Löschen einer Fehlerbaum-Topologie. Hierbei umfasst der Begriff Fehlerbaum-Topologie die Gesamtheit aller Elemente, die in der Semantik des ORM einen Fehlerbaum definieren, d. h. ein Fehlerbaum-Objekt, eine Menge an Gatter-Objekten und eine Menge an Basisereignis-Objekten. Die relativen Verknüpfungen dieser Elemente untereinander und innerhalb eines Fehlerbaum-Objekts sind als Attribute der Python-Objekte hinterlegt und beschreiben die topologische Struktur des Fehlerbaums. Ab dieser Abstraktionsschicht ermöglicht pyRiskRobot mit den Einträgen der Datenbank in Form von topologischen Elementen (d. h. Fehlerbaum, Gatter und Basisereignis) zu arbeiten, und verfügt damit über eine interne API zum PSA-Anlagenmodel der Stufe 1.

Specific Operations: Funktionen zur Modellierung von Fehlerbaum-Topologien

Basierend auf den generischen Funktionen können nun die spezifischen Operationen zur Fehlerbaummodellierung als Funktionenklassen weiterentwickelt werden. Die grundlegenden Operationen werden dabei aus den Gemeinsamkeiten der zur Integration von verschiedenen übergreifenden Einwirkungen erforderlichen Funktionalitäten abgeleitet. Anhand der bisherigen pyRiskRobot Anwendungen konnten drei generische Funktionenklassen zusammengefasst werden, die die Generation, Modifikation und Duplikation von Fehlerbaum-Topologien erlauben.

Methods: Funktionale Schnittstelle zwischen PSA-Modell und Datensammlung für die übergreifende Einwirkung

Die oberste Abstraktionsebene umfasst alle für eine konkrete pyRiskRobot-Anwendung notwendigen Arbeitsschritte in Form eines Python-Skripts. Mit Hilfe der generischen Funktionenklassen können Methoden zur einwirkungsspezifischen topologischen Modellierung formuliert und ausgeführt werden. Zusätzlich können hier separat entwickelte Methoden eingebunden werden, um beispielsweise automatisiert die MS EXCEL®-Eingabedatei zur probabilistischen Spezifikation des PSA-Modells zu erstellen oder die Datensammlungen einer übergreifenden Einwirkung zu analysieren (siehe Abschnitt 3.1.2). In diesem Sinne dient pyRiskRobot als funktionale Schnittstelle zwischen dem PSA-Anlagenmodell und der Datensammlung für die jeweils zu betrachtende übergreifende Einwirkung.

Primär dient pyRiskRobot zur Unterstützung komplexer PSA-Modellierungen innerhalb einer RiskSpectrum®-Datei. Die Umstellung auf die Interpretersprache Python erlaubt eine intuitive Arbeitsweise mit pyRiskRobot mittels interaktiven Anwendungsskripten im Rahmen von Jupyter Notebooks /JUP 17/. Ein Jupyter Notebook ist ein Python-basiertes Dokument, das in einem beliebigen Internet-Browser geöffnet und ausgeführt werden kann. Das Notebook erlaubt es ein Anwendungsskript in unabhängige Skript-Zellen zu unterteilen, die nacheinander und auch mehrfach ausgeführt werden können. Dabei wird pro Zelle das Ergebnis des jeweils letzten abgearbeiteten Rechenschrittes ausgegeben oder darin generierte Abbildungen dargestellt. Des Weiteren kann zwischen den Skript-Zellen Text eingebunden werden, um beispielsweise die Rechenschritte der jeweiligen Anwendung direkt im Notebook zu dokumentieren. Die Vorteile des Jupyter Notebookformats sind exemplarisch in Abb. 2.4 dargestellt.

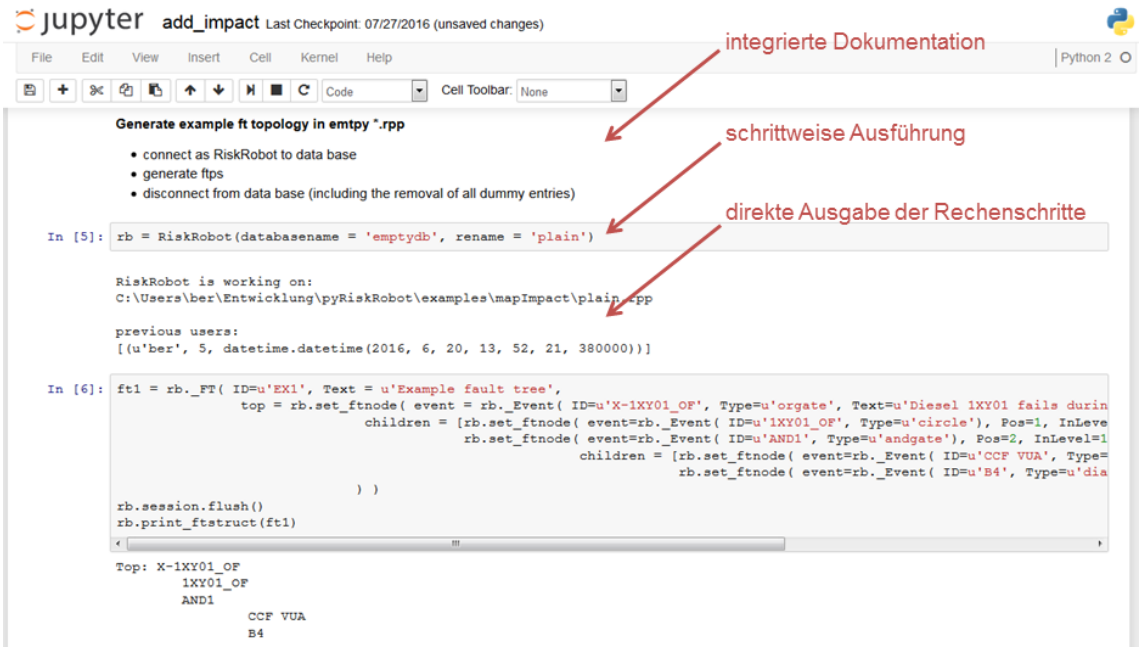

Abb. 2.4 Anwendung des Analysewerkzeugs pyRiskRobot im Rahmen eines Jupyter Notebooks

Jedoch sind nicht alle von pyRiskRobot angebotenen Operationen mehrfach ausführbar, da ein wiederholter Aufruf unter anderem zu einer irreversiblen Korrumpierung der PSA-Datenbank führen kann. Im Allgemeinen bietet das Notebookformat aber eine übersichtliche und einfach zu bedienende Arbeitsweise, um mit Hilfe von pyRiskRobot interaktiv auf die Datenbank des PSA-Modells zuzugreifen, Fehlerbaummodellierungen auszuführen oder vorliegende Datensammlungen von übergreifenden Einwirkungen zu untersuchen.

2.2.2 Konzeptionierung als Schnittstelle zwischen Datenbanken

Das Notebookformat erweist sich im Einsatz als relativ unkomplizierter Zugang zu den Anwendungs- und Erweiterungsmöglichkeiten von pyRiskRobot. Die konkrete Hauptanwendung von pyRiskRobot umfasst dabei die Methoden zur automatisierten Modifikation von Fehlerbaum-Topologien und zur automatisierten Spezifikation der probabilistischen Modelle relevanter Basisereignisse. Mit Hilfe der frei verfügbaren Python-Bibliothek OpenPyXL /OPY 17/ können dabei die MS EXCEL®-Eingabedateien zur Spezifikation des PSA-Anlagenmodells automatisiert erstellt werden. Da die MS EXCEL®-Eingabedateien von RiskSpectrum® eine vorgegebene Struktur besitzen, konnten hierbei notwendige Funktionen standardisiert und als zusätzliche Methoden zusammengefasst werden. Somit steht pyRiskRobot eine zusätzliche Funktionenklasse

zur automatisierten Spezifikation des PSA-Modells via MS EXCEL®-Eingabedateien zur Verfügung.

Des Weiteren bietet das Notebookformat auch eine effiziente Möglichkeit, die zusammengestellten Datensammlungen, die zum Beispiel eine übergreifende Einwirkung auf ein Anlagensystem beschreiben, auszulesen, zu untersuchen und je nach aktueller PSA-Fragestellung anzupassen. Diese szenarienspezifischen Vorarbeiten können innerhalb eines separaten Notebooks erarbeitet werden und stellen eine zusätzliche modulare Methodensammlung dar. Damit pyRiskRobot diese Vorarbeiten nutzen kann, wurde eine weitere Funktionenklasse entwickelt. Diese erlaubt es separate Notebooks als Module in das Notebook der pyRiskRobot-Hauptanwendung zu importieren und als reguläre Python-Klasse zu verwenden.

Basierend auf der zentralen Stellung von pyRiskRobot bezüglich der verwendeten Datenstrukturen bietet das Analysewerkzeug eine funktionale Schnittstelle zwischen dem PSA-Anlagenmodell und der Datensammlung einer übergreifenden Einwirkung. Diese Konzeptionierung von pyRiskRobot ist schematisch in Abb. 2.5 dargestellt.

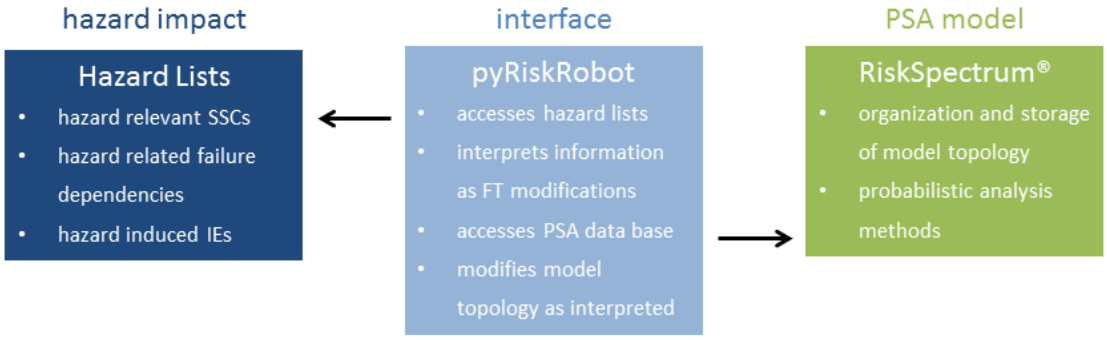

Abb. 2.5 Konzeptionierung von pyRiskRobot als Schnittstelle zwischen der Datensammlung für eine übergreifende Einwirkung und der Datenbank eines PSA-Anlagenmodells

Im Allgemeinen basieren die einwirkungsspezifischen Methoden zur topologischen Modifikation auf den Anforderungen der jeweils für eine PSA-Fragestellung entwickelten Modellierungsansätze. Im Rahmen mehrerer Forschungs- und Entwicklungsvorhaben hat die GRS Konzepte und Strategien zu einer möglichst realistischen Modellierung übergreifender Einwirkungen von innen und außen erarbeitet (siehe dazu /BAB 11/, /TUE 15/, /TUE 15a/, /TUE 15b/ und /BAB 17/). In diesem Zusammenhang wurden auch Datensammlungen erstellt, welche Informationen für spezifische Einwir-

kungen, wie anlageninternen Brand oder anlagenexterne Überflutung, bezüglich eines Referenzmodells beinhalten.

Eine wichtige Voraussetzung für die direkte, d. h. automatisierte Übertragbarkeit der Informationen in den Datensammlungen der übergreifenden Einwirkungen ist die Konsistenz der Namensgebung der zu modifizierenden Fehlerbaum-Elemente. Das bedeutet, dass die Namen (Englisch: *id*entifiers, IDs) der betreffenden Fehlerbäume, Gatter und Basisereignisse konsistent zwischen den Datensammlungen und dem PSA-Anlagenmodell sein müssen. Sobald eine Einwirkung in ein bestehendes PSA-Modell integriert werden soll, müssen die jeweiligen IDs bekannt sein, damit pyRiskRobot als funktionale Schnittstelle fungieren kann. Wird hingegen eine Fehlerbaum-Topologie unabhängig von der bereits bestehenden PSA-Topologie modelliert, so kann dies unabhängig von der jeweils vergebenen IDs erfolgen.

2.3 Funktionenklassen zur automatisierten Integration übergreifender Einwirkungen

Im Prinzip sollen die von pyRiskRobot zur Verfügung gestellten Methoden sicherstellen, dass die erforderlichen Fehlerbaummodellierungen praktisch, effizient und technisch umsetzbar sind. Dabei basieren verschiedene Modellierungsansätze oft auf ähnlichen topologischen Operationen, die teilweise standardisiert und zu generischen Funktionenklassen zusammengefasst werden können.

Die Abstraktion der Methoden in generische Funktionenklassen kann im Folgenden am Besten im Rahmen einer vereinfachten, generischeren Syntax der Fehlerbaum-Topologien beschrieben werden. In allgemeiner Form entspricht eine Fehlerbaum-Topologie einem azyklisch, gerichteten Graphen /ALE 82/, wie in Abb. 2.6 dargestellt. Basierend auf dieser Interpretation können die Elemente (d. h. Basisereignisse und Gatter) eines Fehlerbaums vereinfacht als Knoten interpretiert werden, deren Kanten (relative Anordnung zueinander bzw. paarweise Verknüpfungen miteinander) die Topologie des Graphen bzw. Fehlerbaums festlegen. Da der Graph gerichtet ist, kann zwischen oben und unten, d. h. den Richtungen top-down und bottom-up, unterschieden werden. Das oberste Element einer betrachteten zusammenhängenden Fehlerbaum-Topologie wird somit als Wurzelknoten (Englisch: root node), die untersten Elemente werden als Blätter (Englisch: leaf nodes) bezeichnet.

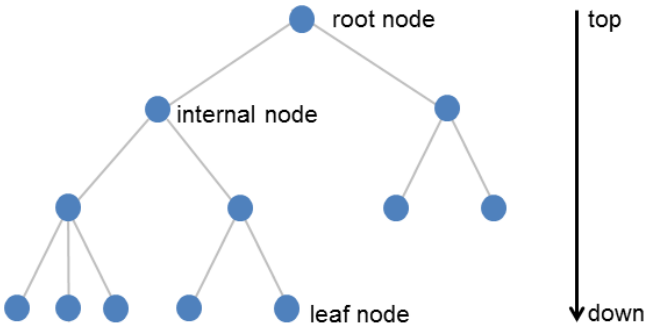

Abb. 2.6 Vereinfachte Darstellung einer Fehlerbaum-Topologie als azyklisch, gerichteter Graph bestehend aus Knoten (blau) und Kanten (grau)

Diese schematische Betrachtungsweise einer Fehlerbaum-Topologie erleichtert die Beschreibung und Unterteilung der entwickelten generischen Funktionenklassen. Die im Rahmen der Weiterentwicklung und Erprobung des Analysewerkzeugs pyRiskRobot abgeleiteten Funktionenklassen umfassen bisher

- das Generieren neuer Fehlerbaum-Topologien,
- das Modifizieren bestehender Fehlerbaum-Topologien und
- das Duplizieren zusammenhängender Fehlerbaum-Topologien.

Basierend auf diesen Operationen können dann je nach Bedarf der jeweiligen Anwendung komplexere Methoden abgeleitet werden, um die einwirkungsspezifischen Veränderungen in den relevanten Fehlerbäumen automatisiert auszuführen. Eine schematische Übersicht über die generischen Funktionenklassen ist in Abb. 2.7 gegeben. Die Implementierung dieser Funktionen wurde anhand verschiedener Anwendungsbeispiele erprobt und mittels existierender Referenzanwendungen gegen RiskRobot validiert. Um einen umfassenden Überblick über das Potenzial von pyRisk-Robot zur automatisierten Modellierung komplexer PSA-Fragestellungen zu geben, werden im Folgenden die grundlegenden Funktionsklassen und deren Einsatzgebiete genauer beschrieben.

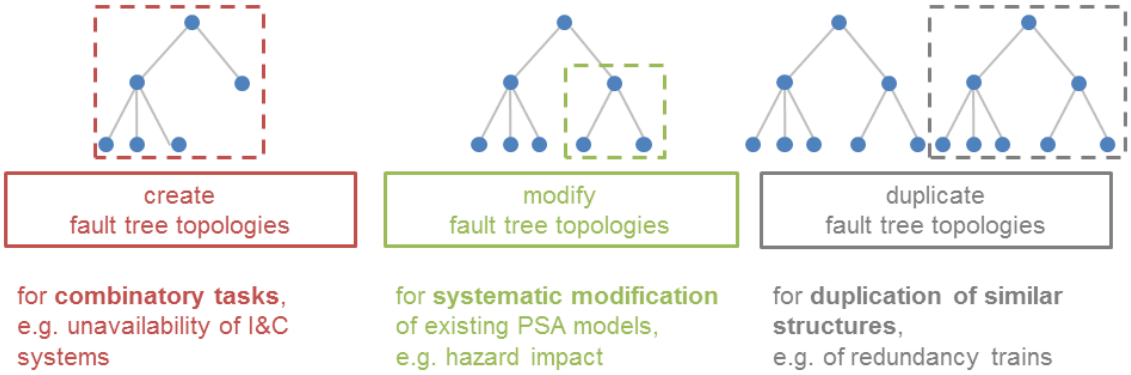

Abb. 2.7 Übersicht über die generischen Funktionenklassen topologischer Operationen und deren typischen Modellierungsaufgaben des Analysewerkzeugs pyRiskRobot

2.3.1 Generieren neuer Fehlerbäume

Die grundsätzliche Voraussetzung für die Ausführung aller topologischen Operationen von pyRiskRobot ist ein bereits vorhandenes elektronisches Anlagenmodell in Form einer RiskSpectrum®-Datei, welches aber noch keine Einträge enthalten muss. Innerhalb des vorgegebenen PSA-Modelles erlaubt pyRiskRobot die Erzeugung neuer Fehlerbaum-Topologien. Zum Beispiel kann durch den in Abb. 2.8 gezeigten Skriptausschnitt eine Fehlerbaum-Topologie in die PSA-Datenbank eingetragen und dann über die GUI von RiskSpectrum® als Fehlerbaum-Diagramm, wie in Abb. 2.9 gezeigt, dargestellt werden.

Der Skriptausschnitt verdeutlicht, dass ein Fehlerbaum in textueller Form top-down beschrieben wird. Die prozedurale Erstellung eines Fehlerbaums als referenzierter Eintrag in der Datenbank erfolgt auf Grund der ineinander verschachtelten Python-Funktionen bottom-up. Deshalb muss für die Erzeugung eines Gatters in Zwischenschritten je ein temporärer Hilfsfehlerbaum und ein temporäres Hilfsbasisereignis als Vaterknoten zur Verfügung gestellt werden, damit ein Gatter eineindeutig in eine Topologie eingebettet werden kann. Im Verlaufe der prozeduralen Erstellung werden alle Hilfsobjekte mit den expliziten Objekten top-down referenziert und somit eine zusammenhängende Fehlerbaum-Struktur aufgebaut. In Folge dieser Strategie müssen nach dem Generieren eines vollständigen neuen Fehlerbaums alle obsoleten Hilfsobjekte aus der Datenbank gelöscht werden. Um dies effizient auszuführen, wird das einzige explizite SQL-Kommando auf dieser Abstraktionsebene von pyRiskRobot verwendet, um alle Hilfsobjekte simultan aus der Datenbank zu löschen.

```
ft=rr.set_FT(
        ID='EX1',
        Text='Example fault tree',
        top=rr.set_ftnode(
                    event=rr.set_event(ID='TOP1',
                                       Type='orgate',
                                       CalcType=1,
                                       Text='TOP of EX1'),
                    Pos=1, InLevel=0,
                    children=[rr.set_ftnode(
                                       event=rr.set_event(ID='B1',
                                                          Type='circle',
                                                          Model=3,
                                                          CalcType=1,
                                                          Text='Basic element 1'),
                                       Pos=1, InLevel=1),
                              rr.set_ftnode(
                                       event=rr.set_event(ID='B2',
                                                          Type='circle',
                                                          Model=3,
                                                          CalcType=1,
                                                          Text='Basic element 2'),
                                       Pos=1, InLevel=1)
                             ]
                    )
            )
```

Abb. 2.8 Skriptausschnitt einer pyRiskRobot-Anwendung zur Erzeugung einer Fehlerbaum-Topologie

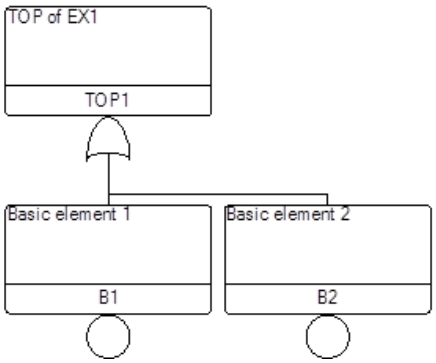

Abb. 2.9 Fehlerbaum-Topologie erzeugt durch den Skriptausschnitt in Abb. 2.8 und visualisiert mit Hilfe der RiskSpectrum®-GUI

Die Notwendigkeit der automatisierten Erstellung von Fehlerbaum-Topologien ergibt sich vor allem im Rahmen umfangreicher kombinatorischer Aufgaben. Diese können automatisiert von pyRiskRobot abgearbeitet und in Form zugeordneter Fehlerbaum-Topologien generiert werden. Dies ist beispielsweise bei der Modellierung komplexer Systeme der Mess- und Regeltechnik erforderlich /HER 12/. Neben der Komplexität eines Fehlerbaummodelles stellt die praktische Umsetzbarkeit eines theoretischen Modellierungsansatzes eine weitere Herausforderung bei der Erzeugung von Fehlerbaum-Topologien dar. Zum Beispiel kann bei der Untersuchung eines Brandausbreitungsszenarios die Abhängigkeit unter den Brandräumen theoretisch durch eine NOT-Logik

beschrieben werden /BER 16/. Allerdings unterstützt die PSA-Software diesen Logikoperator nicht ausreichend beim internen Aufbau der kompletten PSA-Topologie, welcher jeder Analysemethode vorausgeht. Jedoch ermöglicht pyRiskRobot die effiziente Umsetzung einer expliziten Modellierungsstrategie der Raumabhängigkeiten innerhalb des PSA-Anlagenmodells, wie in Abschnitt 3.1 näher beschrieben.

Somit bietet das Analysewerkzeug pyRiskRobot eine effiziente Methode zur Erzeugung aufwendiger Fehlerbaum-Topologien und ermöglicht dadurch die Untersuchung komplexer Systemmodelle, die sonst nur mit großem manuellem Aufwand realisierbar wären. Die Methodik von pyRiskRobot zeichnet sich durch ihre Effizienz und Flexibilität aus und ermöglicht folglich auch alternative Modellierungsansätze (siehe Abschnitt 3.1.1).

2.3.2 Modifizieren bestehender Fehlerbäume

Die von pyRiskRobot generierten Fehlerbaum-Topologien können ein Modell eines komplexen Teilsystems oder eine spezifische Auswirkung einer übergreifenden Einwirkung darstellen. Damit die so unabhängig voneinander modellierten Teilaspekte in einer PSA berücksichtigt werden, müssen diese in das bestehende PSA-Modell eingebunden werden. Die PSA-Topologie wird dabei um die unabhängig erstellten Fehlerbaum-Topologien erweitert, so dass eine umfassende probabilistische Auswertung unter Berücksichtigung des neu modellierten Aspekts durchgeführt werden kann.

Die automatisierte Modifikation von Fehlerbaum-Topologien ist insbesondere für die Modellierung solcher übergreifender Einwirkungen notwendig, deren Auswirkungen mit einem hohen Detaillierungsgrad berücksichtigt werden müssen. Beispielsweise müssen für eine realistische Modellierung eines anlageninternen Brandes mittels des PSA-Modells alle sicherheitsrelevanten SSCs den jeweils zugehörigen Brandräumen zugeordnet werden /HER 15b/, /BER 16/. Bei dieser automatisierten Zuordnung von Brandräumen zu Komponenten, gegeben durch Datenlisten (sogenannte Hazard Equipment Lists, HEL /TUE 15a/), müssen bestehende Fehlerbaum-Topologien vielfach erweitert werden, wie näher in Abschnitt 3.1 beschrieben. Ein anderes Beispiel ergibt sich bei der Untersuchung verschiedener Überflutungsszenarien eines angenommenen Einwirkungsspektrums im Rahmen eines PSA-Modells /BER 17/. Dabei müssen die Fehlerbaum-Topologien der von der übergreifenden Einwirkung betroffenen SSCs um die Topologien der modellierten Überflutungsszenarien erweitert werden, wie näher in Abschnitt 3.2 beschrieben. Damit die Erweiterung bestehender Fehlerbaum-Topologien

automatisiert erfolgen kann, müssen alle zu modifizierenden topologischen Elemente mit deren IDs, d. h. mit der exakten Namensgebung wie im PSA-Anlagenmodell verwendet, aus zur Verfügung gestellten Datenlisten bekannt sein.

Als funktionale Schnittstelle zwischen den Datenlisten und der PSA-Datenbank bietet pyRiskRobot eine automatisierte Methodik, um PSA-Modelle auch um aufwendige Fehlerbaum-Topologien zu erweitern. Durch die Standardisierung der topologischen Operationen von pyRiskRobot bietet die Methodik die Funktionalitäten, um ein im Prinzip generisches Spektrum übergreifender Einwirkungen von innen und außen (einschließlich Einwirkungskombinationen) in komplexe PSA-Anlagenmodelle zu integrieren.

2.3.3 Duplizieren zusammenhängender Fehlerbäume

Die zentrale Aufgabe ist die Erzeugung und Modifikation von Fehlerbaum-Topologien, um neuartige Aspekte in einer PSA zu berücksichtigen, die zuvor noch nicht im Model abgebildet waren. Eine weitere Aufgabe von pyRiskRobot kann aber auch sein, redundante Aspekte von bereits modellierten Teilsystemen zu berücksichtigen, d. h. vorhandene, zusammenhängende Fehlerbaumstrukturen zu vervielfältigen. Der Begriff „zusammenhängend“ bezieht sich dabei auf Fehlerbaum-Topologien, die sich über referenzierende Gatter (d. h. TRANSFER-Gatter) hinweg erstrecken. Durch Spezifikation der duplizierten Topologie und Einbindung in das bestehende PSA-Modell können redundante Aspekte automatisiert mittels pyRiskRobot in ein Modell integriert werden.

Das Duplizieren von Fehlerbaum-Topologien kann zum Beispiel dazu verwendet werden, den Einfluss verschiedener Überflutungsszenarien auf mehrere redundante Stränge eines sicherheitsrelevanten Systems zu modellieren, wie näher in Abschnitt 3.2 beschrieben. Außerdem können topologisch identische Überflutungsszenarien durch automatisiertes Duplizieren eines manuell erstellten Referenzszenarios modelliert werden /BER 17/.

Die generische Funktionenklasse des Duplizierens ist eine konkrete Weiterentwicklung von pyRiskRobot, die auf den Möglichkeiten der Interpretersprache Python und den Funktionalitäten der SQLAlchemy-Bibliothek beruht. Die Grundlage des Duplizierens bildet das automatisierte top-down Auslesen aller Knoten einer zusammenhängenden Fehlerbaum-Topologie aus der PSA-Datenbank ausgehend vom angegebenen Wurzelknoten bis zu allen Blättern der Topologie. Das Auslesen wird von pyRiskRobot iterativ ausgeführt und resultiert in einer Liste aller IDs der in der Topologie enthaltenen

Elemente. Durch Vorgabe eines Schemas zur automatisierten Umbenennung (z. B. durch Ersetzen eines Teilnamens oder Zufügen eines Teilnamens) können die zu erzeugenden Elemente dann automatisiert umbenannt werden. Durch erneutes top-down-Auslesen der Topologie erzeugt pyRiskRobot sukzessive alle Elemente der Referenz-Fehlerbaum-Topologien – ohne den Umweg einer textuellen Form – und vergibt für die duplizierten Elemente IDs wie über das Namensschema vorgegeben. Dabei werden zunächst der Wurzelknoten und alle darunter liegenden Knoten der Wurzelknoten-Fehlerbäume dupliziert. Die Information über die Verknüpfung der Knoten untereinander, d. h. deren relative Lage zueinander, wird durch die Angabe von Vater- und Kindsknoten an das duplizierte Objekt weitervererbt. pyRiskRobot arbeitet nun alle darunterliegenden Fehlerbaum-Topologien der Reihe nach ab. Das heißt, dass zuerst die leeren Fehlerbäume erzeugt und dann top-down alle enthalten Knoten erzeugt werden, bis alle Blätter des aktuellen Fehlerbaums erreicht sind. Dieses Vorgehen wird solange wiederholt, bis alle Blätter der zusammenhängenden Fehlerbaum-Topologie erreicht sind. In einem letzten Schritt kann dann die duplizierte Topologie (sogenannter Klon) an einem geeigneten Knoten in das PSA-Modell eingebunden werden.

Nach diesem Prinzip können komplexe Strukturen dupliziert werden, indem nur die IDs des Wurzelknoten-Fehlerbaums (sogenannter Harbor FT) und des Wurzelknoten selbst (sogenanntes Anchor Event) angegeben werden müssen, wie in Abb. 2.10 skizziert. Hierbei sei angemerkt, dass nur die IDs der duplizierten Knoten umbenannt werden, nicht aber der Beschreibungstext der einzelnen Knoten. Dieser kann über den Export und Import der MS EXCEL®-Datei von RiskSpectrum® bei Bedarf angepasst werden.

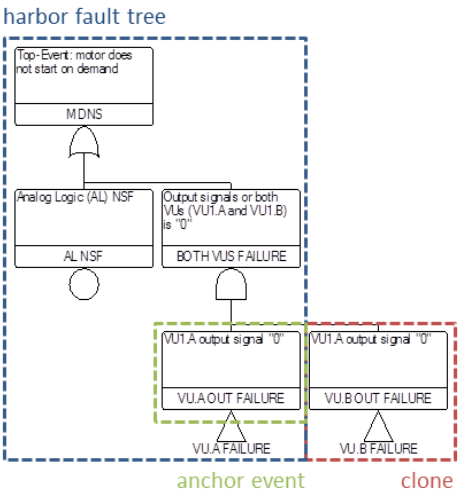

Abb. 2.10 Prinzip der automatisierten Duplikation von Fehlerbaum-Topologien mittels pyRiskRobot basierend auf der Angabe des Harbor Fehlerbaums und des Anchor Ereignisses und resultierend in einer integrierten Klon-Topologie

Insbesondere beim Duplizieren zeigt sich das Notebookformat als vorteilhaft bei der pyRiskRobot-Anwendung. Hier können die ausgelesen IDs einer zu duplizierenden Topologie interaktiv bearbeitet werden und somit das Namensschema flexibel und auch abweichend von einer strengen Syntax erzeugt werden. Somit bietet das automatisierte Duplizieren eine wichtige Ergänzung der generischen Funktionenklassen von pyRisk Robot und ermöglicht die explizite Modellierung von redundanzübergreifenden Aspekten.

3 Erprobung von pyRiskRobot anhand von Referenzanwendungen für übergreifende Einwirkungen

Das Analysewerkzeug pyRiskRobot wurde anhand von Referenzbeispielen zur Modellierung übergreifender Einwirkungen von innen und außen erprobt. Mit Hilfe bereits durchgeführter Anwendungen konnte die Implementierung als Python-basierte Version gegen die Ruby-basierte Version validiert werden. Eine erfolgreiche Validierung bezieht sich dabei auf die automatisierte Erstellung identischer Fehlerbaum-Topologien mittels pyRiskRobot im Vergleich zu RiskRobot. Potenzielle topologische Abweichungen vom Referenzmodell wurden für komplexe Beispiele mit Hilfe des Werkzeugs CmpFT automatisiert überprüft. Des Weiteren wurden – soweit ein Referenzbeispiel vorlag – die numerischen Ergebnisse der Konsequenzanalysen in RiskSpectrum® miteinander verglichen. Die hier vorgestellten Anwendungen wurden im Notebookformat erarbeitet und dienen somit als Anwendungsbeispiele, welche die Funktionalitäten von pyRiskRobot veranschaulichen und eine schrittweise Dokumentation der Methodik beinhalten. Bei den Validierungsrechnungen konnten alle bisher mit RiskRobot realisierten PSA-Modelle erfolgreich nachgebildet werden.

Um die weiterentwickelten Funktionalitäten von pyRiskRobot zu dokumentieren, werden im Folgenden auch erweiterte bzw. neue Anwendungsbeispiele beschrieben, für die kein direktes Referenzbeispiel, basierend auf dem Vorgänger-Werkzeug, existiert. Zum einen wurde die bereits mit RiskRobot berechnete Einwirkung „anlageninterner Brand“ umgesetzt und um den Aspekt der Brandausbreitung zwischen benachbarten Brandräumen erweitert /BER 16/. Somit konnten die numerischen Fehlerbaumanalysen der Raumabhängigkeiten in Abhängigkeit der Länge der Brandausbreitungspfade bzw. Modellierungstiefe des Brandes direkt in einer RiskSpectrum®-Datei untersucht werden. Das erarbeitete Vorgehen dient als Fallbeispiel, wie der Einfluss des Detaillierungsgrades einer spezifischen Fehlerbaummodellierung quantitativ abgeschätzt werden kann. Zum anderen wurde die übergreifende Einwirkung von außen „anlagenexterne Überflutung“ modelliert, wobei der Einfluss verschiedener Szenarien der Überflutung auf alle redundante Systeme eines Anlagengebäudes berücksichtigt wurde /BER 17/. Das Vorgehen dient als Fallbeispiel, um den Einfluss eines Spektrums verschiedener Szenarien von Einwirkungen von innen oder außen in ein PSA-Anlagenmodell zu integrieren und dessen Auswirkungen auf das gesamte Analyseergebnis der CDFs untersuchen zu können.

3.1 Anlageninterner Brand mit Raumabhängigkeiten

3.1.1 Modellierungsansatz des Brandszenarios

Ausgangspunkt der hier beschriebenen Modellierung ist die automatisierte Integration der übergreifenden Einwirkung von innen „anlageninterner Brand“ auf die Brennelement-Beckenkühlung einer Referenzanlage mit Hilfe von RiskRobot /HER 15b/. Zum Zweck der Validierung werden dieselben Modellierungsschritte von pyRiskRobot ausgeführt. Ein brandbedingter Ausfall von SSCs wird dabei nur dann unterstellt, wenn ein Brand in einem Raum oder Anlagenbereich (als Brandraum bezeichnet) auftritt, in dem sich diese Komponenten oder Kabel dieser Komponenten befinden. Die in Bezug auf diese Einwirkung wesentlichen sicherheitsrelevanten Komponenten der betrachteten Referenzanlage wurden im Rahmen eines Forschungs- und Entwicklungsvorhabens in Datenlisten zusammengefasst /FRE 08/. Basierend darauf können diejenigen Komponenten identifiziert werden, die von dem angenommen Brandszenario betroffen sind. Die Fehlerbaum-Topologien dieser Komponenten wird im ursprünglichen PSA-Modell um ein HOUSE-Event zur optionalen Berücksichtigung eines Brandes erweitert. Durch die Zuordnung von Komponenten zu Brandräumen werden die jeweiligen Fehlerbaum-Topologien um alle relevanten Brandräume als Basisereignisse eines Brandeintritts ergänzt, wie schematisch in Abb. 3.1 dargestellt.

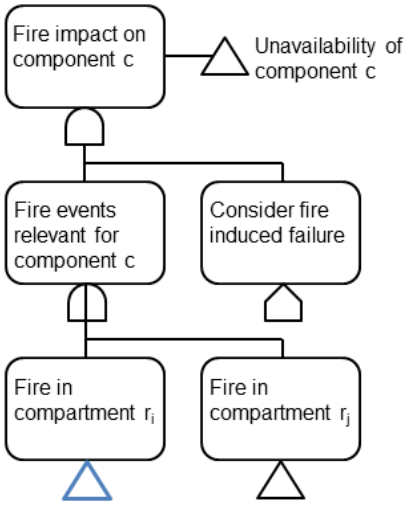

Abb. 3.1 Erweiterung des Fehlerbaums einer Komponente um die optionalen Ereignisse eines Brandeintritts in jedem zugeordneten Brandraum

Um auch die Brandausbreitung zwischen den Brandräumen zu berücksichtigen, endet die Topologie nicht bei den Basisereignissen des Brandes (wie in /HER 15b/), sondern führt per TRANSFER-Gatter in die Brandraum-spezifischen Fehlerbäume. Diese bilden die Abhängigkeiten der Brandräume untereinander ab. Zunächst muss berücksichtigt werden, ob der Brand im Raum selbst entstanden ist oder, falls das nicht der Fall ist, ob sich der Brand von einem benachbarten Raum ausgebreitet hat, wie in Abb. 3.2 (A) dargestellt. Für alle nächsten Raumnachbarn des betrachteten Raumes kann nun die Brandausbreitung aus dem Raum und wiederum die Entstehungsquellen berücksichtigt werden, wie in Abb. 3.2 (B) gezeigt. Die dazu notwendigen Informationen zu den Raumabhängigkeiten und den Brandübergangswahrscheinlichkeiten zwischen den Räumen werden denselben Datenlisten entnommen, die auch die Brandraum-Komponenten-Zuordnung enthalten. Die topologischen Erweiterungen werden automatisiert mit pyRiskRobot ausgeführt. Die probabilistischen Spezifikationen der für das angenommene Szenario relevanten Räume, d. h. die Übergangswahrscheinlichkeiten zwischen den Räumen und die Brandeintrittshäufigkeiten in den Räumen, werden automatisiert mit pyRiskRobot aus den Datenlisten ausgelesen und in die MS EXCEL®-Importdatei des PSA-Modells eingetragen. Mit Hilfe der MS EXCEL®-Importfunktion von RiskSpectrum® werden die neuen Elemente der PSA-Topologie spezifiziert.

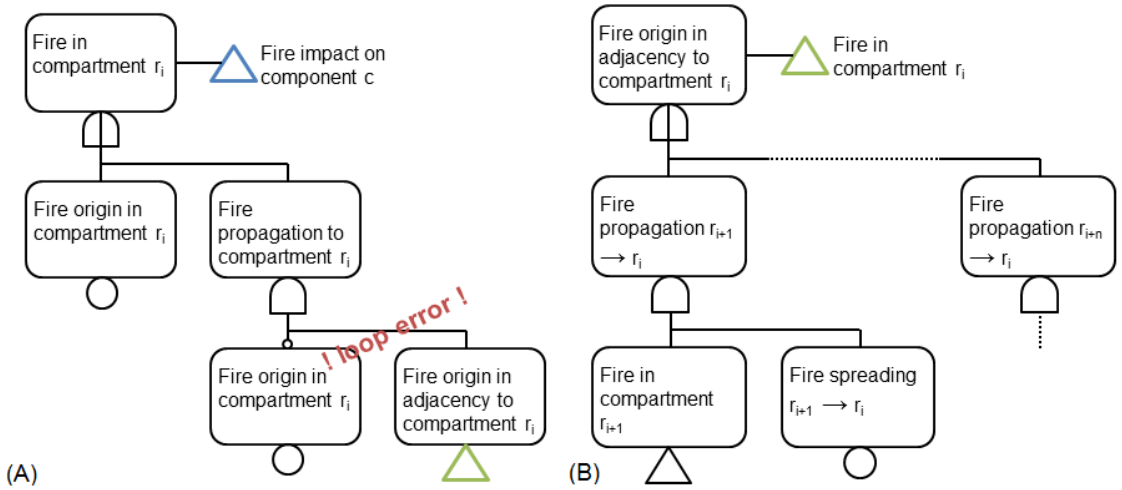

Abb. 3.2 Fehlerbaummodellierung einer Brandausbreitung: (A) Unterscheidung der möglichen Quellen der Brandentstehung und (B) Propagation des Brandes durch die benachbarten Brandräume

Alle beschriebenen Arbeitsschritte sind mit Hilfe von pyRiskRobot durchführbar und ergeben ein PSA-Modell, welches einen fehlerfreien Import der probabilistischen Spezifikationen erlaubt. Allerdings bedingen die Fehlerbäume der Raumabhängigkeiten bei der Ausführung der Konsequenzanalysen eine Endlosschleife (Englisch: loop error)

und führen letztendlich zum Abbruch der numerischen Analyserechnungen. Als Ursache dafür konnte im Fehlerbaum der Brandentstehung die NOT-Logik identifiziert werden (vgl. Abb. 3.2 (A)), die beim internen Verarbeitungsprozess der PSA-Topologie in RiskSpectrum® nicht korrekt aufgelöst werden kann. Auf Grund der nicht berücksichtigten NOT-Logik beschreiben die modellierten Topologien der Raumabhängigkeiten eine zyklische Topologie. Das Zusammensetzen des PSA-Modells basiert aber auf der Grundannahme, dass nur Fehlerbaum-Topologien, also azyklische, gerichtete Graphen vorliegen. Somit kann die gesamte Topologie des PSA-Modells nicht mehr analysiert werden.

3.1.2 Untersuchung der Raumabhängigkeiten für Brandausbreitungen verschiedener Modellierungstiefe

Zur Untersuchung realisierbarer Modellierungsansätze der Brandausbreitung wurden die Methoden von pyRiskRobot angewandt. Unter der Annahme, dass ein Brand im Laufe seiner Ausbreitung jeden Raum maximal nur einmal durchqueren kann, können die zyklischen Abhängigkeiten in den Fehlerbaum-Topologien verhindert werden. Somit können die Ausbreitungspfade entlang der miteinander verbundenen Brandräume explizit formuliert werden. Diese explizite Formulierung ermöglicht eine passende topologische Modellierung, wobei die Pfadtiefe aus Gründen der Umsetzbarkeit beschränkt werden muss. Das Prinzip der Nachbarschaftsordnung oder Pfadtiefe, d. h. die Anzahl der Räume, die vom Brand maximal einmal durchlaufenen werden, ist vereinfacht in Abb. 3.3 skizziert.

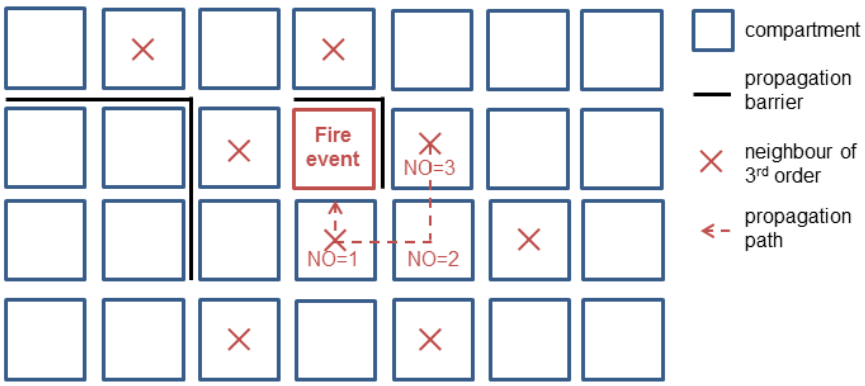

Abb. 3.3 Schematische Darstellung der expliziten Modellierung mittels der Brandausbreitungspfade (roter Pfeil) aus den Brandräumen (blau) der dritten Nachbarschaftsordnung (rotes Kreuz) zu einem betrachteten Brandraum (rot)

Dadurch ist es mit pyRiskRobot möglich, alle potentiellen Feuerwege automatisiert bis zu einer vorgegebenen Pfadtiefe zu modellieren. Die Brandraumabhängigkeiten werden von den geometrischen Anordnungen der Brandräume, d. h. den tatsächlichen Räumen die eine Komponente enthalten und den ihr zugeordneten Kabeln, als auch den Annahmen für die erlaubten Ausbreitungspfade bestimmt. Zu berücksichtigen ist, dass Abb. 3.3 eine starke Vereinfachung der realen Abhängigkeiten zwischen verschiedenen Brandräumen darstellt. Um den Einfluss der Nachbarschaftsordnung auf die die modellierte Brandausbreitung zu untersuchen, wurden die realen Brandraumkorrelationen der Referenzanlage, wie sie in den Datenlisten angegeben sind, genauer analysiert. Eine übersichtliche Möglichkeit bietet die Visualisierung der Raumabhängigkeiten in Form von Netzwerken, wie in Abb. 3.4 gezeigt. Mit Hilfe der frei verfügbaren Python-Bibliothek Plotly /PLY 17/ kann jeder Brandraum als Knoten in einem Netzwerk dargestellt und durch Kanten mit seinen nächsten Nachbarn verbunden werden. Die Anzahl der nächsten Nachbarn wird durch die Farbkodierung der Knoten angegeben.

Eine Möglichkeit, die Komplexität der PSA-Modellierung zu reduzieren, wäre die Vereinfachung der Netzwerke basierend auf angenommenen Randbedingungen für jeden Knoten bzw. jede Kante. Zum Beispiel können alle Kanten herausgefiltert werden, deren assoziierte Übergangswahrscheinlichkeit zwischen den jeweiligen Räumen einen vorgegebenen Schwellenwert unterschreitet und somit die berücksichtigten Raumabhängigkeiten reduziert werden. Diese Voranalysen der Datenlisten mittels pyRiskRobot bieten die Möglichkeit, komplexe Modellierungsansätze bereits außerhalb des PSA-Modells zu vereinfachen.

In diesem Zusammenhang wurden Leistungsfähigkeit und Grenzen von pyRiskRobot sowie RiskSpectrum® im Rahmen komplexer Modellierungen näher untersucht. Von besonderem Interesse war dabei die Untersuchung des Einflusses der Modellierungstiefe oder des Detaillierungsgrads der abgebildeten Raumabhängigkeit auf die resultierende Wahrscheinlichkeit eines Brandeintritts in einem Brandraum bzw. die Nichtverfügbarkeit eines Brandraums auf Grund eines Brandeseintritts darin. Aus Tab. 3.1 wird ersichtlich, dass sich die Komplexität der Raumabhängigkeiten für die Anlagengebäude stark unterscheiden. Durch die Brute-Force-Berechnung aller möglichen Brandpfade, unter der Bedingung, dass jeder Raum maximal einmal durchlaufen wird, wird die maximale Brandpfadtiefe pro Gebäude ermittelt. Aufgrund des hohen Rechenaufwands ist dies nicht für alle Gebäude umgesetzt worden.

Dabei wird deutlich, dass die Beschränkung auf eine maximal berücksichtigte Pfadtiefe oder Nachbarschaftsordnung zur Machbarkeit des expliziten Modellierungsansatzes notwendig ist. Um die zusätzliche Annahme einer maximal berücksichtigten Pfadtiefe zu begründen, wird der Einfluss der modellierten Pfadtiefe auf die resultierende Brandwahrscheinlichkeit in einem Raum, d. h. die Analyse des Fehlerbaums in Abb. 3.2 (B), qualitativ abgeschätzt. Zu diesem Zwecke wurden die expliziten Fehlerbaum-Topologien der Brandausbreitung für alle Brandräume automatisiert für verschiedene Nachbarschaftsordnungen mittels pyRiskRobot erzeugt. Durch die unabhängige Analyse der einzelnen Brandraum-Fehlerbäume in RiskSpectrum® wurde so die Nichtverfügbarkeit eines Raumes aufgrund eines Brandes für die Nachbarschaftsordnungen 0 bis 4 berechnet. Dabei entspricht die Pfadtiefe 0 dem Brand ohne Ausbreitung, die Pfadtiefe 1 einem Brand, der sich auch von den nächsten Nachbarn ausgebreitet haben kann etc.

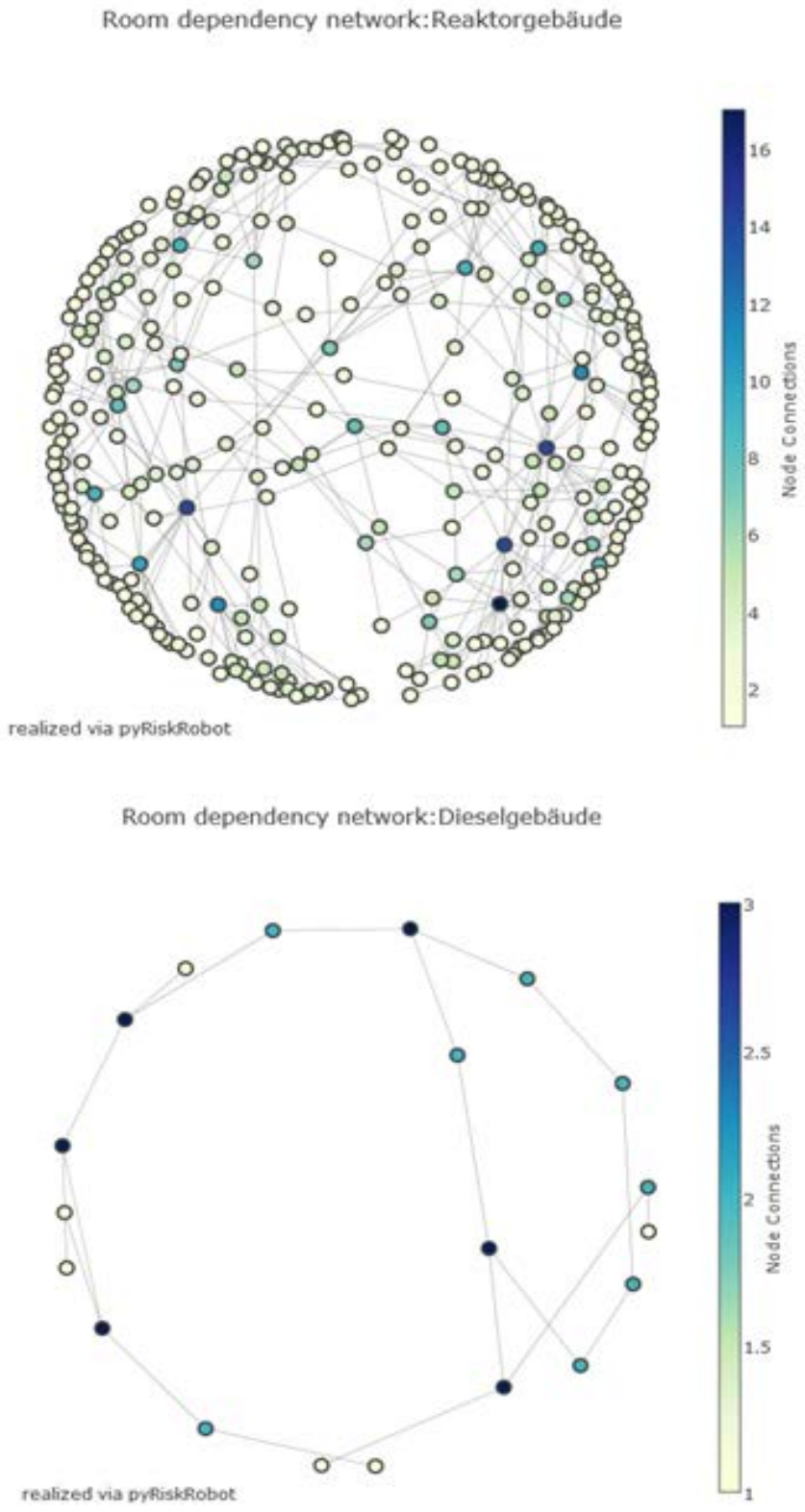

Abb. 3.4 Darstellung der Brandraumabhängigkeiten in Form von Netzwerken realisiert mit Hilfe des Analysewerkzeugs pyRiskRobot

Tab. 3.1 Charakteristika der Brandraumabhängigkeiten für verschiedene Gebäude einer Referenzanlage (basierend auf /FRE 08/)

Gebäude	Anzahl abhängiger Brandräume	Maximale Anzahl nächster Nachbarn	Maximale Pfadtiefe
Reaktorgebäude	342	17	*nicht berechnet*
Schaltanlagengebäude	176	11	87
Maschinenhaus	99	10	57
Dieselgebäude	20	3	14
Unabhängiges Notstandsgebäude	35	8	19

Die Fehlerbaum-Analysen aller Anlagengebäude in Abb. 3.5 machen deutlich, dass sich die Brandwahrscheinlichkeiten eines Brandraumes auch noch zwischen Pfadtiefen höherer Ordnungen ändern können. Dies ist insbesondere der Fall, wenn ein Raum mit einer extrem hohen Brandeintrittswahrscheinlichkeit und hoher Übergangswahrscheinlichkeit in die berücksichtigte Nachbarschaft eingeschlossen wird. Dabei ist festzustellen, dass für die Mehrzahl der Räume aller Gebäude der größte Zuwachs in den Brandwahrscheinlichkeiten bis zur 3. Ordnung auftritt.

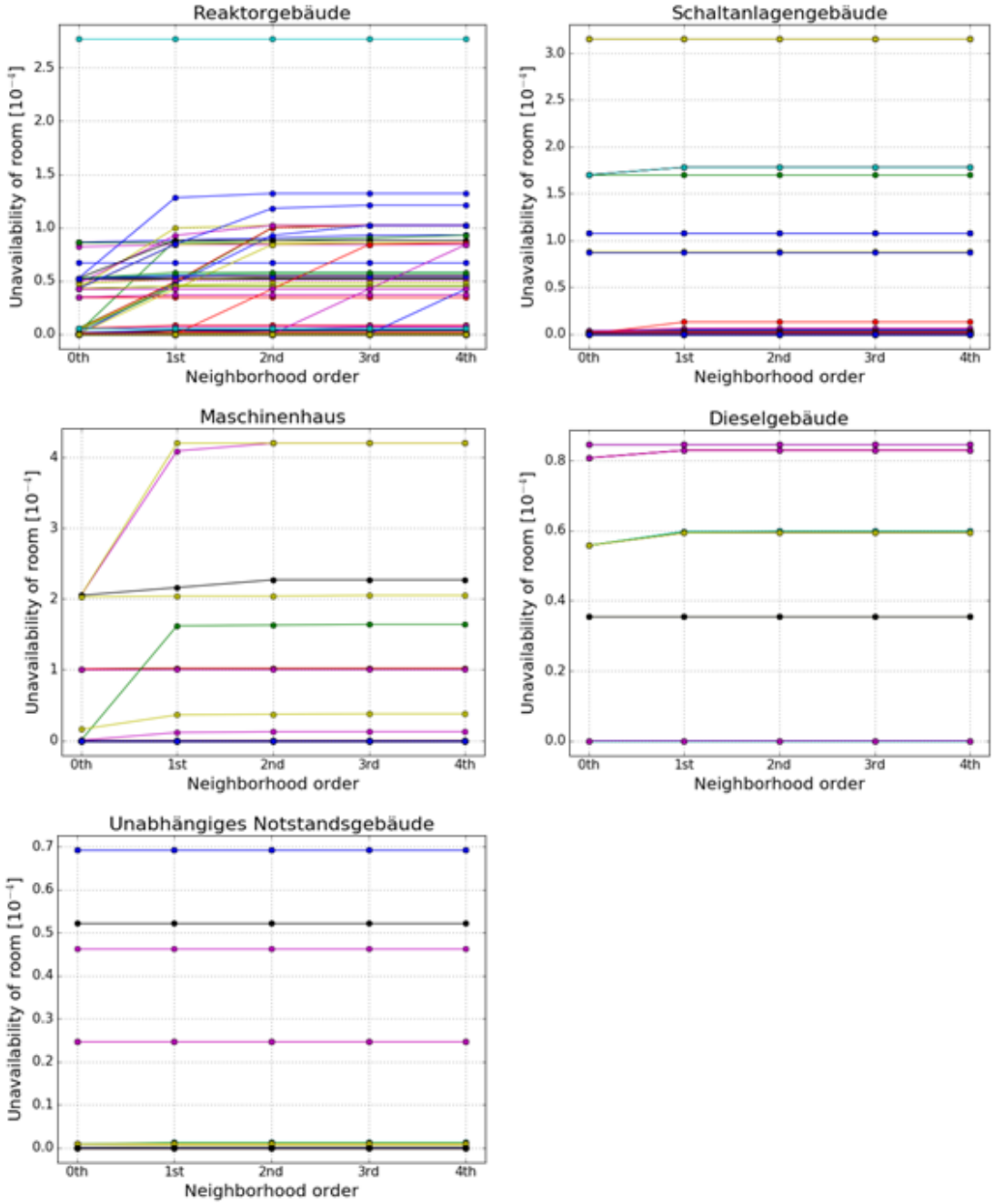

Abb. 3.5 Brandwahrscheinlichkeit der Brandräume (d. h. Nichtverfügbarkeit durch Brand) pro Anlagengebäude in Abhängigkeit der modellierten Nachbarschaftsordnungen 0 bis 4 der spezifischen Brandausbreitungspfade

3.1.3 Analyseergebnisse und Leistungsverhalten der pyRiskRobot-Modellierung

Als Kompromiss zwischen Machbarkeit und Genauigkeit wird somit für die Modellierung der Brandausbreitung eine Pfadtiefe der Ordnung 3 als ausreichend detailliert angenommen. Mit Hilfe von pyRiskRobot werden die expliziten Ausbreitungspfade automatisiert in das PSA-Anlagenmodell integriert. In den Konsequenzanalysen führte diese Erweiterung des PSA-Modells zu einer mittleren Erhöhung der CDF um 3,34 % (Min. 0,46 % und Max. 9,90 %) im Vergleich zu den Referenzanalysen des PSA-Modells mit „anlageninternem Brand“ ohne Brandausbreitung. Dabei wiesen einige Basisereignisse des Brandübergangs zwischen den Räumen relativ hohe Importanzwerte auf. Der kumulative Beitrag zur Gesamthäufigkeit blieb dabei allerdings gering.

Die Zeitdauer der pyRiskRobot-Anwendung für die Modellierung des Brandszenarios ohne Raumabhängigkeiten (d. h. ohne Analyserechnungen) in das PSA-Anlagenmodell betrug weniger als zwei Minuten. Dabei wurden 405 Komponenten um die optionale Berücksichtigung eines Brandes (d. h. per HOUSE-Event) aus 172 Brandräumen ergänzt. Die automatisierte Modellierung der betrachteten Nachbarschaftsordnungen 0 bis 4 aller Gebäude in einer separaten RiskSpectrum®-Datei dauerte insgesamt etwa 308 Minuten. Die Integration aller relevanten Raumabhängigkeiten in das komplette PSA-Anlagenmodell erstreckte sich über etwa 52 Minuten. Dabei wurden für 26 der 172 berücksichtigten Brandräume die Raumabhängigkeiten explizit bis zur 3. Nachbarschaftsordnung abgebildet.

3.2 Externe Überflutung mit redundanzübergreifendem Einfluss

3.2.1 Modellierungsansatz des Überflutungsszenarios

Die hier vorgestellte Modellierung beschreibt die automatisierte Integration einer übergreifenden Einwirkung in Folge einer „anlagenexternen Überflutung“ auf eine Referenzanlage mit Hilfe von pyRiskRobot /BER 17/. In dem modellierten Szenario wird ein lokales Starkregenereignis unterstellt, welches auf Grund der topographischen Lage zu einer Sturzflut und zur Überflutung der Anlage führen kann. Infolge der Überflutung des Maschinenhauses kommt es zum Ausfall des Hauptkühlwassersystems und des Kondensatsystems, was der Transiente „Ausfall der Hauptwärmesenke“ entspricht. Dies wird durch den Reaktorschutz erkannt und führt zum Turbinenschnellschluss (TUSA) und zur Reaktorschnellabschaltung (RESA). Sofern das Überflutungsszenario auch

den Ausfall des konventionellen Nebenkühlwasser- bzw. Zwischenkühlsystems bedingt, stehen auch die Hauptspeisewasserpumpen nicht zur Verfügung. Weiterhin kann es auf Grund von Kurzschlüssen in der Eigenbedarfsversorgung der Anlage und Störungen bei der Umschaltung der Eigenbedarfsversorgung auf das Reservenetz zur Anforderung der Notstromversorgung kommen. Daher wird bei der Modellierung abdeckend ein dem Hochwasser überlagerter Notstromfall angenommen. Zum Schutz gegen Überflutung sind im Betriebshandbuch der Anlage temporäre Schutzmaßnahmen für das Notstromdieselgebäude (z. B. Setzen von Schotten, englisch: stop locks) vorgesehen. Aufgrund der Annahme einer Sturzflut wird unterstellt, dass die benötigte Zeit für die Durchführung entsprechender Maßnahmen nicht ausreichen könnte und somit die redundant verfügbaren Notstromdieselgeneratoren (Englisch: emergency diesel generators, EDG) als nicht verfügbar angenommen werden müssen (siehe Abb. 3.6 (A)).

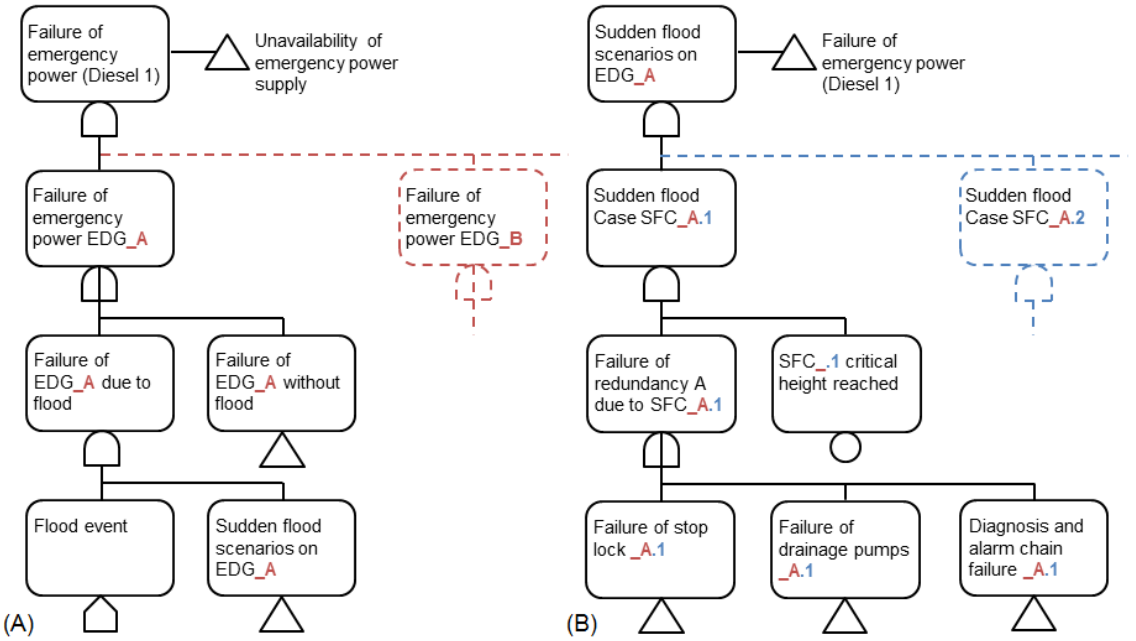

Abb. 3.6 Fehlerbaummodellierung einer Überflutung des Notstromdieselgebäudes: (A) Einfluss auf die redundanten Notstromdiesel und (B) Modellierung des Ausfallverhaltens eines Notstromdiesels infolge verschieden starker Sturzflutszenarien

Die probabilistische Modellierung naturbedingter Einwirkungen bzw. die Ermittlung deren standortspezifischen Risikos und daraus resultierende Gefährdungshäufigkeiten gestaltet sich als schwierige Aufgabe. Hierzu werden die übergreifenden Einwirkungen in Bezug auf ihr Gefahrenpotenzial kategorisiert. Je nach angenommenem Szenario können sich im PSA-Modell die Anzahl der einwirkungsbedingt beeinträchtigten SSCs

(d. h. die Topologie) als auch die probabilistische Parametrisierung (d. h. die Spezifikation) der übergreifenden Einwirkungen erheblich voneinander unterscheiden. Eine möglichst umfassende Berücksichtigung des Einflussspektrums der naturbedingten Einwirkung ist die Modellierung der verschiedenen, anzunehmenden Szenarien. Somit kann anstelle eines ausgewählten Einzelszenarios ein Spektrum mehrerer Szenarien infolge der jeweiligen Einwirkung in einem gemeinsamen PSA-Anlagenmodell abgebildet und untersucht werden.

In Tab. 3.2 werden drei Szenarien eines Starkregenereignisses (die sich durch ihre Niederschlagsrate unterscheiden) angenommen, die eine Überflutung des Notstromdieselgebäudes und somit der Notstromdieselgeneratoren verursachen können. Je größer die Niederschlagsrate, desto schneller erreicht die resultierende Sturzflut das Anlagengelände und desto größer ist die Flussrate der jeweiligen Sturzflut. Das Ausfallverhalten der Notstromdieselgeneratoren wird aus den menschlichen Handlungsabläufen abgeleitet, welche im Rahmen der vorgesehenen Notfallmaßnahmen ausgeführt werden müssen. Diese Notfallmaßnahmen, d. h. die Diagnose und Alarmierung, die Installation temporärer Schotten und das Starten der Entwässerungspumpen sind dabei teilweise voneinander abhängig. Insbesondere sind die Handlungen aber von den spezifischen Überflutungsszenarien abhängig, wie aus Abb. 3.6 (B) ersichtlich. Die modellierten Ausfälle unterscheiden sich auf Grund der Überflutungsszenarien nur bezüglich der Stärke der angenommenen Starkregenereignisse und somit deren probabilistischen Spezifikation, nicht aber bezüglich der topologischen Struktur der beschreibenden Fehlerbäume. Somit kann pyRiskRobot dazu verwendet werden, um die formulierten Überflutungsszenarien automatisiert durch Modifikation und Duplikation in das bestehende PSA-Modell der Referenzanlage redundanzübergreifend zu integrieren.

Tab. 3.2 Angenommene Starkregenereignisse für die Spezifikation der modellierten Überflutungsszenarien

Überflutungsszenario	**Niederschlagsrate [l/m^2/h]**	**Eintrittshäufigkeit [10^{-6}/J]**
1	100	65
2	200	30
3	400	5

3.2.2 Verschachtelte Duplikation von Fehlerbäumen in Abhängigkeit von Überflutungsszenarien und Redundanzen

Die Modellierungsarbeit des Analysewerkzeugs pyRiskRobot besteht bei der Integration der „anlagenexternen Überflutung“ aus zwei ineinander verschachtelten Arbeitsschritten:

- Erstellung und Einbindung der drei Überflutungsszenarien für die angenommenen Starkregenereignisse,
- Berücksichtigung der modellierten Szenarien in allen modellierten redundanten Strängen der Notstromdieselgeneratoren.

Basierend auf einer entweder in RiskSpectrum® oder per pyRiskRobot-Skript erstellten Fehlerbaum-Topologie für das Referenzszenario 1, können alle weiteren Szenarien 2 und 3 durch Duplikation erstellt werden. Die entsprechenden Fehlerbäume lassen sich dann, wie in Abb. 3.6 (B) gestrichelt dargestellt, für die Redundanz A in das PSA-Modell integrieren. Für alle restlichen redundanten Stränge B bis D werden die Fehlerbäume der Überflutungsszenarien 1 bis 3 der Redundanz A per Duplikation automatisiert erstellt und in deren Topologie integriert, wie in Abb. 3.6 (A) gestrichelt angedeutet.

Um diese ineinander verschachtelte Modellierung effizient umsetzen zu können, wird bei der Erstellung des Referenzszenarios der Überflutung folgendes Namensschema der Fehlerbaumelemente eingehalten:

- „_A.1“ bezeichnet Elemente, die spezifisch für Redundanz und Szenario sind,
- “_A.“ bezeichnet Elemente, die spezifisch für eine Redundanz und unabhängig vom Szenario sind, und
- „_.1“ bezeichnet Elemente, die unabhängig von der Redundanz und spezifisch für ein Szenario sind.

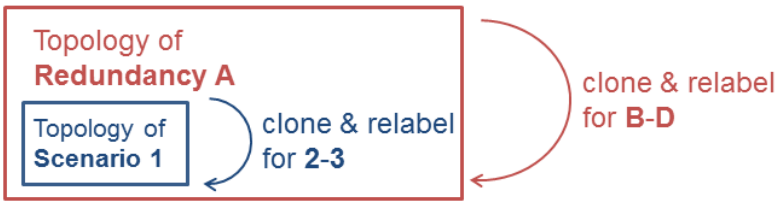

Abb. 3.7 Prinzip der automatisierten verschachtelten Duplikation von Fehlerbaum-Topologien zur Modellierung verschiedener Überflutungsszenarien (blau) mit redundanzübergreifendem Einfluss (rot)

Das generische Namensschema _A.1 bezeichnet dabei die Redundanz A und das Überflutungsszenario 1. Dieses Namensschema erlaubt die prozedurale Ausführung der Modellierungsschritte als ineinander verschachtelte Schleifen, so dass die topologischen Operationen komplett automatisiert werden können, wie in Abb. 3.7 skizziert. Die probabilistische Spezifikation der modifizierten topologischen Elemente erfolgt wiederum per Einlesen einer MS EXCEL®-Eingabedatei mittels der Import-Funktion von RiskSpectrum®.

3.2.3 Analyseergebnisse und Leistungsverhalten der pyRiskRobot-Modellierung

Der beschriebene Modellierungsansatz zur Integration einer „anlagenexternen Überflutung“ resultiert in der automatisierten Duplikation von 568 topologischen Elementen in der Datenbank des PSA-Anlagenmodells. Diese wurden mittels pyRiskRobot in weniger als zwei Sekunden ausgeführt. Die Konsequenzanalysen des resultierenden PSA-Modells ergab einen relativen Zuwachs von 80 % der (allerdings geringen) CDF auf Grund der redundanzübergreifenden Einwirkung der angenommenen Überflutungsszenarien auf die SSCs des Notstromdieselgebäudes unter den unterstellten Ausfallannahmen.

Im Rahmen dieser Anwendung ermöglicht das Analysewerkzeug pyRiskRobot die systematische Betrachtung angenommener Szenarien einer übergreifenden Einwirkung. Die dazu erarbeiten Funktionen und Strategien bieten die methodischen Grundlagen, um ein prinzipiell generisches Spektrum von übergreifenden Einwirkungen einschließlich Ereigniskombinationen mit solchen Einwirkungen auf komplexe PSA-Anlagenmodelle effektiv abbilden zu können.

4 Zusammenfassung und Ausblick

Im vorliegenden Bericht ist die vom Bundesministerium für Wirtschaft und Energie (BMWi) im Rahmen des Forschungs- und Entwicklungsprojektes RS1539 geförderte Weiterentwicklung der Methodik zur automatisierten Integration übergreifender Einwirkungen in PSA-Modelle der Stufe 1 dargestellt. Um die Auswirkungen übergreifender Einwirkungen von innen und außen effizient und nachvollziehbar in komplexen PSA-Anlagenmodellen abzubilden, wurde das Analysewerkzeug pyRiskRobot als Python-basierte Software reimplementiert und weiterentwickelt.

Die Methodik von pyRiskRobot basiert auf einem dynamischen Zugriff auf das PSA-Anlagenmodell einer RiskSpectrum®-Datei und erlaubt die Ausführung komplexer Fehlerbaummodifikationen. Die grundlegenden Aufgaben des Analysewerkzeugs pyRiskRobot sind dabei die topologische Modellierung von Fehlerbäumen und die probabilistische Spezifikation modifizierter Fehlerbaumelemente. Auf Grundlage der einwirkungsspezifischen Anwendungen konnten drei generische Funktionenklassen zur topologischen Modellierung abgeleitet werden, die die automatisierte Generation, Modifikation und Duplikation von Fehlerbäumen ermöglichen.

Die Reimplementierung des GRS-Analysewerkzeugs in der Interpretersprache Python ermöglicht die Verwendung frei verfügbarer, leistungsstarker Bibliotheken, die zum Beispiel für den Zugriff auf die Datenbank des PSA-Modells genutzt werden können. Ein weiterer Vorteil der Umstellung auf Python stellt die unkomplizierte und daher anwenderfreundliche Arbeitsumgebung eines Jupyter Notebookformats dar. Im Rahmen von Notebooks bietet die Arbeitsumgebung von pyRiskRobot auch Methoden, um bestehende Datensammlungen auszulesen, zu analysieren und für die jeweilige Anwendung vorzubereiten. Somit dient pyRiskRobot als funktionale Schnittstelle zwischen der Datensammlung einer übergreifenden Einwirkung (z. B. in Form von MS EXCEL®-Tabellen) und der Datenbank eines PSA-Anlagenmodells (in Form einer MSSQL-Datenbank von RiskSpectrum®).

Anhand von Anwendungsbeispielen für verschiedene übergreifende Einwirkungen wurde pyRiskRobot erprobt und weiterentwickelt. Zum einen wurde ein anlageninterner Brand mit Brandausbreitung zwischen benachbarten Brandräumen einer Referenzanlage modelliert. Dabei ermöglicht pyRiskRobot die sukzessive Erhöhung der Modellierungstiefe der Brandausbreitung innerhalb eines komplexen PSA-Anlagenmodells und die Untersuchung von dessen Einfluss auf die numerischen Analyseergebnisse. Zum

anderen wurden eine anlagenexterne Überflutung und deren Auswirkungen auf redundante Einrichtungen des Sicherheitssystems einer Referenzanlage modelliert. Dabei ermöglicht pyRiskRobot, basierend auf einem vorgegebenen Referenzszenario, alternative Überflutungsszenarien zu duplizieren und dieses Spektrum von Szenarien in alle redundanten Stränge des jeweils betroffenen Anlagengebäudes zu integrieren.

Die weiterentwickelte Version des GRS-Analysewerkzeugs pyRiskRobot umfasst die methodischen Grundlagen und weitere praktische Strategien, um verschiedenartige übergreifende Einwirkungen auf die Topologie von PSA-Anlagenmodellen der Stufe 1 automatisiert abbilden zu können. Somit erlaubt pyRiskRobot die Realisierung komplexer Modellierungsansätze, um im Prinzip ein generisches Spektrum übergreifender Einwirkungen in probabilistischen Sicherheitsanalysen zu berücksichtigen.

Zukünftig sollen die topologischen Modellierungsmethoden von pyRiskRobot weiterentwickelt werden, um insbesondere auch Kombinationen von Ereignissen mit übergreifenden Einwirkungen automatisiert integrieren zu können. Des Weiteren soll pyRisk Robot als funktionale Schnittstelle zwischen Datensammlungen für übergreifende Einwirkungen und der Datenbank des PSA-Anlagenmodells weiter ausgebaut werden. So sollen auch die Analysemethoden der Daten vor der Integration in das PSA-Modell erweitert werden, um beispielsweise die Korrelationsstruktur von Raumabhängigkeiten basierend auf bestimmten Kriterien noch vor der eigentlichen Modellierung zu reduzieren.

Weiterhin ist es geplant, Anwendungsbeispiele zur Integration einwirkungsspezifischer Auswirkungen im Jupyter Notebookformat zu erstellen, um die Möglichkeiten des Analysewerkzeugs interaktiv zu dokumentieren.

Literaturverzeichnis

/ALE 82/ Alesso, H. P.: Some Fundamental Aspects of Fault-Tree and Diagraph-Matrix Relationships for a Systems-Interaction Evaluation Procedure, Report, Lawrence Livermore National Laboratory (LLN), Livermore, CA, USA, Februar 1982.

/BAB 11/ Babst, S., et al.: Methoden zur Durchführung von Brand-PSA im Nichtleistungsbetrieb, GRS-A-3579, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, Köln, Januar 2011, http://www.grs.de/publikation/grs-A-3579.

/BAB 17/ Babst, S., et al:, Bestimmung des standortspezifischen Risikoseines Kernkraftwerks, GRS-A-3888. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Köln, 2017 (in Vorbereitung).

/BER 16/ Berner, N., J. Herb: Generic framework for the automated integration of impacts from hazards in PSA models, Risk, Reliability and Safety: Innovation Theory and Practice; Walls, Revie and Bedford (Eds.), in: Proceedings of the 26th European Safety and Reliability Conference 2016 (ESREL 2016), Glasgow, Großbritannien, 2016.

/BER 17/ Berner, N., M. Utschick, G. Gänssmantel, M. Röwekamp: Systematic Integration of Hydrological Hazards by Automatically Extending PSA Models, in: Proceedings of the 27th European Safety and Reliability Conference 2017 (ESREL 2017), Portoroz, Slovenia, 2017 (in Vorbereitung).

/FAK 05/ Facharbeitskreis (FAK) Probabilistische Sicherheitsanalyse für Kernkraftwerke: Methoden zur probabilistischen Sicherheitsanalyse für Kernkraftwerke, Stand: August 2005, BfS-SCHR-37/05, Bundesamt für Strahlenschutz (BfS), Salzgitter, Germany, Oktober 2005, http://doris.bfs.de/jspui/handle/urn:nbn:de:0221-201011243824.

/FRE 08/ Frey, W., et al. : Methoden zur Abschätzung des Risikobeitrags redundanzübergreifender Brandschäden, Technischer Bericht, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-A-3425, Köln, Dezember 2008, https://www.grs.de/publikation/grs-A-3425.

/HER 11/ Herb, J., J. von Linden: Procedures and Tools Comparing PSA in the Frame of Periodic Safety Reviews, in: Proceedings of ANS PSA 2011 International Topical Meeting on Probabilistic Safety Assessment and Analysis, Wilmington, NC, March 13-17, 2011, on CD-ROM, American Nuclear Society, LaGrange Park, IL, USA, 2011, S. 1364, http://toc.proceedings.com/11651webtoc.pdf.

/HER 12/ Herb, J.: Fault Tree Auto-Generator: How to Cope with Highly Redundant Systems, in: 11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference 2012 (PSAM11 ESREL 2012), ISBN: 978-1-62276-436-5, Curran Associates, Inc., Red Hook, NY, 2012, S. 2704.

/HER 15a/ Herb, J., et al.: Entwicklung eines Ansatzes zur Analyse der Netzwerktechnologien in sicherheitsrelevanten Leittechniksystemen hinsichtlich Verbreitung und Auswirkung postulierter Fehler – Technischer Bericht, Gesellschaft für Anlagen- und Reaktorsicherheit: (GRS) gGmbH, GRS-377, Köln, Juni 2015, http://www.grs.de/sites/default/files/pdf/grs-377.pdf.

/HER 15b/ Herb, J., et al.: Automatic integration of a Fire PSA model in a Level 1 PSA. In: Proceedings of Safety and Reliability of Complex Engineered Systems: (ESREL 2015) – Podofillini et al. (Eds.), 2015. Taylor & Francis Group, London, ISBN 978-1-138-02879-1, S. 429-434.

/JUP 17/ The Jupyter Notebook, Offizielle Homepage: http://jupyter.org/, (zuletzt aufgerufen am 07.03. 2017).

/MAR 07/ Marshall, K., et al.: Pro Active Record Databases with Ruby and Rails, Apress. Berkeley, CA, USA, 2007.

/OPY 17/ OpenPyXL: A Python library to read/write Excel 2010 xlsx/xlsm files, Offizielle Homepage: http://openpyxl.readthedocs.io/en/default/, (zuletzt aufgerufen am 06.04.2017).

/PLY 17/ Plotly: Collaboration Platform for modern data science, Offizielle Homepage: https://plot.ly/feed, (zuletzt aufgerufen am 06.03. 2017).

/SCA 12/ Scandpower AB: RiskSpectrum Analysis Tools User's Manual, Version 3.2.1, 2012, elektronische Dokumentation.

/SQL 17/ SQLAlchemy: The Python SQL Toolkit and Object Relational Mapper, Offizielle Homepage: https://www.sqlalchemy.org/, (zuletzt aufgerufen am 06.03. 2017).

/TUE 15/ Türschmann, M., H. Holtschmidt, M. Röwekamp: Recent Research on Hazards PSA, IEM8 - International Experts' Meeting on Strengthening Research and Development Effectiveness in the Light of the Accident at the Fukushima Daiichi Nuclear Power Plant, IAEA, Wien, 16-20 February 2015.

/TUE 15a/ Türschmann, M., M. Röwekamp, S. Babst: Concept for Comprehensive Hazards PSA and Fire PSA Application, Progress in Nuclear Energy, Volume 84, Special Issue: EUROSAFE 2013, S. 36-40, 2015, http://www.sciencedirect.com/science/article/pii/S0149197015000876.

/TUE 15b/ Türschmann, M, S. Sperbeck, W. Frey: "Methodische Ansätze zur Durchführung einer standortspezifischen PSA zu den Auswirkungen übergreifender Einwirkungen", GRS-A-3838, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Köln, August 2015.

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen und Begriffe

API	Programmierschnittstelle (Englisch: application programming interface)
BE	Basisereignis (Englisch: basic event)
EB	Ereignisbaum (Englisch: event tree, ET)
EDG	Notstromdiesel (Englisch: emergency diesel generator, EDG)
FB	Fehlerbaum (Englisch: fault tree, FT)
GUI	Benutzeroberfläche (Englisch: graphical user interface)
AE	Auslösendes Ereignis (Englisch: initiating event, IE)
CDF	Kernschadenshäufigkeit (Englisch: core damage frequency)
FDF	Brennstabschadenshäufigkeit (Englisch: fuel damage frequency)
NDG	Notstromdieselgebäude (Englisch: emergency diesel generator building, EDG building)
OOP	Objektorientierte Programmierung
ORM	Objektrelationale Abbildung (Englisch: object relational mapping)
PSA	Probabilistische Sicherheitsanalyse
RESA	Reaktorschnellabschaltung
SQL	Structured Query Language
SSCs	Systeme, Strukturen und Komponenten (Englisch: systems, structures and components)
TUSA	Turbinenschnellschluss
ÜE	Übergreifende Einwirkung (Englisch: hazard)

Brute-Force-Methode

Eine Methode, die auf dem Ausprobieren aller möglichen (oder einer großen Zahl von) Fällen basiert und das Ziel verfolgt, Probleme aus beispielsweise der Graphen- oder Spieltheorie zu lösen (auch als Exhaustionsmethode bezeichnet).

Interpretersprache

Eine Programmiersprache, deren Quellcode (im Gegensatz zu nativen Programmiersprachen) nicht in ein eigenständig ausführbares Programm übersetzt wird, sondern durch einen sogenannten Interpreter zur Laufzeit eingelesen, analysiert, interpretiert und ausgeführt wird.

Jupyter Notebook

Ein Dokument, das sowohl ausführbare Programmskripte als auch Textelemente (z. B. zur Dokumentation) enthält. Durch die Unterteilung der Skripte in unabhängig voneinander ausführbare Zellen können Rechenschritte interaktiv bearbeitet, ausgeführt und die Ergebnisse direkt im Notebook textuell oder graphisch ausgegeben werden.

Nativ

Der Begriff nativ ausgeführt (in einem Programm) bedeutet, dass Arbeiten innerhalb einer Programmumgebung und nur mit deren zur Verfügung gestellten Methoden ausgeführt werden.

Persistent

Der Begriff persistente Objekte beschreibt Objekte, die einer Programmumgebung über einen langen Zeitraum mit all ihren zugehörigen Informationen zur Verarbeitung zur Verfügung stehen.

Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH

Schwertnergasse 1
50667 Köln
Telefon +49 221 2068-0
Telefax +49 221 2068-888

Boltzmannstraße 14
85748 Garching b. München
Telefon +49 89 32004-0
Telefax +49 89 32004-300

Kurfürstendamm 200
10719 Berlin
Telefon +49 30 88589-0
Telefax +49 30 88589-111

Theodor-Heuss-Straße 4
38122 Braunschweig
Telefon +49 531 8012-0
Telefax +49 531 8012-200

www.grs.de

ISBN 978-3-946607-36-6