

Experimentplanung und -durchführung mit BPMN-Modellen im Windpark-Simulationswerkzeug DESMO-Windpark-Studio

Tilman Stehle¹

Abstract

Im Umfeld der Energieerzeugung auf im Meer errichteten Windkraftanlagen außerhalb von Küstenzonen, sogenannten Offshore Windparks, erschließt sich ein neues Forschungsgebiet der Logistik. Im Rahmen des Projekts SystOp Offshore Wind sollen Betriebs- und Instandhaltungsprozesse des Leistungssystems Offshore Windpark optimiert werden. Dazu gehören beispielsweise die Versorgung der Windkraftanlagen mit Ersatzteilen, oder Wartungsdienstleistungen, für die der Einsatz von Schiffen oder Hubschraubern an den Anlagen notwendig ist. Aufgrund der besonderen Umgebung mit extremen Wettereinflüssen und gesetzlichen Auflagen bezüglich des Passierens von Staatsgrenzen, sowie der Notwendigkeit des Zusammenspiels verschiedener Akteure ist hierbei stets ein komplexer logistischer Zusammenhang gegeben. Aus diesem Grund werden Techniken und Methoden der Simulation eingesetzt, mit denen Analysen über kritische Prozessfaktoren, Rückkopplungen und Engpässe der betroffenen Ressourcennutzer untersucht werden sollen, die direkt oder indirekt Einfluss auf Aspekte wie Sicherheitsrisiken der Prozesse, Verfügbarkeit der Anlagen, Stromgestehungskosten oder den Stromertrag nehmen. An der Universität Hamburg entsteht zu diesem Zweck der Werkzeugkasten DESMO-Windpark -Studio, eine Software zur Modellierung und Simulation der abzubildenden Wettereinflüsse, Ressourcen, Geschäftsprozesse und Windpark-Komponenten. Sie basiert auf der Plug-in-Rahmenanwendung Empinia und nutzt die quelloffene Simulationsbibliothek DESMO-J. Mit DESMO-Windpark-Studio soll das Optimierungsvorhaben unter Durchführung wiederholter Simulationsläufe und unter Variation der Modellparameter unterstützt werden. Eine besondere Herausforderung ist unter anderem in der Heterogenität der eingesetzten Modelle zu sehen, die gleichartig im Rahmen der Simulation parametrisiert und in mehrfache Simulationen eingebunden werden. DESMO-Windpark-Studio bietet inzwischen neben anderen Funktionalitäten den Import von BPMN-Modellen und bietet eine Experimentierumgebung, in der wiederholte Experimentläufe unterschiedlich konfigurierter Modelle geplant und durchgeführt werden können. Der vorliegende Beitrag beschreibt diese Erweiterung der Software, die das Experimentieren mit BPMN-Kollaborationsmodellen in mehrläufigen Experimenten unter Variation von Modellparametern ermöglicht. Basierend auf dem aktuellen Stand der Software stehen prototypische Experimente mit Geschäftsprozessmodellen in Aussicht. Dabei sollen durch die Universität Bremen erhobene Betriebs- und Instandhaltungsprozesse in DESMO-Windpark-Studio simuliert werden. An dieser prototypischen Simulation sollen die Funktionsfähigkeit von DESMO-Windpark-Studio, der Wert der erhobenen Modelle und gegebenenfalls erste Informationen über Optimierungspotentiale der Prozesse untersucht werden.

1. Einleitung

Mit dem Projekt SystOp Offshore Wind soll der ökonomisch und ökologisch verträgliche Ausbau der Windenergie auf See gefördert werden. Um den Erfolg der Energiegewinnung aus Windkraft auf See zu unterstützen, ist es sinnvoll, die durchzuführenden Betriebsprozesse und die zur Wartung und Instandhaltung der Anlagen notwendigen Vorgänge zu optimieren. Ein wichtiges Ziel besteht darin, die einzusetzenden finanziellen Mittel reduzieren zu können, sodass diese Form der ökologisch verträglichen Energiegewinnung wettbewerbsfähig bleiben kann. Einige der durchgeführten Instandhaltungsprozesse sind im Rahmen von Systop Offshore Wind bereits aufgenommen worden und liegen in Form von

¹Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Straße 30 D-22527 Hamburg

Geschäftsprozessmodellen der Geschäftsprozess-Modellierungssprache BPMN 2.0 vor. DESMO-Windpark-Studio ist bereits in der Lage, Simulationsexperimente mit BPMN-Modellen durchzuführen, sodass erste prototypische Simulationen vollzogen werden können, die zum einen der Validierung der Software bzw. der erhobenen Modelle und zum anderen der Gewinnung erster Erkenntnisse über Abhängigkeiten und Rückkopplungen zwischen Teilprozessen dienen sollen.

Die zur Simulation verwendete Software muss die wichtigen Informationen aus den Modellen zur Simulationszeit entnehmen und in Simulationsberichten ausgeben. Welche Informationen als wichtig zu erachten sind, ist abhängig vom Ziel der durchgeführten Simulationsstudie. Im Falle der Simulation der im Rahmen von SystOp Offshore Wind erhobenen Modelle liegt das Ziel in der Identifikation optimierungsfähiger Prozess-Abschnitte oder -Strukturen, die Einfluss auf die Verfügbarkeit der Anlagen, Stromgestehungskosten oder den Stromertrag nehmen (vgl. SystOp Offshore Wind 2010, 1).

Entscheidend für das Erzielen wertvoller Ergebnisse ist auch die wiederholte Ausführung der Simulationsexperimente. Angenommen es soll die Durchlaufzeit eines Instandsetzungsprozesses geschätzt werden. Aufgrund der Abhängigkeit der Simulation von eingesetzten stochastischen Verteilungen (siehe Abschnitt 2) wäre dann aus einem einzelnen Simulationslauf lediglich eine Punktschätzung zu gewinnen, deren Güte höchst fragwürdig ist, da keine Angaben über die Varianz dieser Kennzahl gemacht werden können. Um eine Grundlage für solide Simulationsexperimente zu legen, ist es notwendig, eine Experimentplanung zu ermöglichen, die mehrere Simulationsläufe unter gleichbleibender Modellkonfiguration erlaubt und diese aggregierend auswertet (vgl. Page 91, 4 S. 15).

Im Folgenden werden zunächst einige notwendige Grundlagen der Simulation gelegt. Daran anschließend wird die Prozessmodellierungs-Sprache BPMN 2.0 vorgestellt, die im Projekt SystOp verwendet wird. Es wird ihre Anreicherung um Simulationsinformationen anhand eines Minimalbeispiels motiviert und erläutert. Schließlich wird die Experimentplanungs-Komponente von DESMO-Windpark-Studio vorgestellt, ehe deren Nutzen abschließend beurteilt und ein Ausblick zu weiterführenden Entwicklungen gegeben wird.

2. Grundlagen der Simulation

Reale Systeme sind häufig zu komplex, um die Abhängigkeiten ihrer Komponenten und mögliche Konsequenzen eines Eingriffs am Original zu untersuchen. Wie eingangs erwähnt, ist dies auch bei der Instandhaltung von Offshore-Windkraftanlagen der Fall. Um ein solches System trotzdem handhabbar und verständlich zu machen, wird ein Abbild erstellt, das es auf die wesentlichen Aspekte der Untersuchung reduziert. Eine solche Abstraktion eines realen Systems nennt Wainer ein Modell (vgl. Wainer (2010), 5 S.24). Niemeyer erweitert den Begriff des Modells um die Eigenschaft der Manipulierbarkeit und fordert, “[...] daß eine experimentelle Manipulation der abgebildeten Strukturen und Zustände möglich ist“ (vgl. Niemeyer (1977), 6). Es sind also nicht nur die statischen Aspekte des Realsystems vereinfacht darzustellen. Vielmehr soll es möglich sein, ein Modell durch seinen Zustandsraum zu führen und aus seinem dynamischen Verhalten auf das des abgebildeten Systems rückzuschließen (vgl. Wainer (2010), 5 S.25). Genau diese dynamische Manipulation eines Modells nennt man Simulation.

In Optimierungs-Vorhaben, wie SystOp eines ist, ist es häufig sinnvoll, verschiedene Varianten des Realsystems im Abbild zu untersuchen und sie zu vergleichen. Um einen Doppelaufwand in der Modellierung des Systems zu vermeiden, gestaltet man ein Vergleichsmodell generisch und führt sogenannte Modellparameter ein, anhand derer ein Modell unterschiedlich konfiguriert werden kann. Das einmalige Umsetzen des konkreten Modellverhaltens über eine bestimmte Zeit bezeichnet man als Simulationslauf (vgl. Page (1991) 4 S.17). Eine Menge solcher Simulationsläufe zur Beantwortung einer Fragestellung bezeichnet man als Experiment (vgl. Wittmann (1993), 8 S.57). Als Simulationszeit wird

dabei die Zeit bezeichnet, die in der Zeitrechnung des Modells verbraucht wird, um das Verhalten zu simulieren. Die Rechenzeit ist dabei die Zeit, die in der Realwelt während der Berechnung der Simulation vergeht. Aufgrund der Komplexität des abgebildeten Systems und der Notwendigkeit, eindeutige Systemgrenzen zu definieren, können häufig nicht alle kausalen Zusammenhänge abgebildet oder überhaupt verstanden werden, die Einfluss auf das System nehmen. Um die davon abhängigen Entscheidungen im Modell trotzdem zu simulieren, nimmt man stochastische Verteilungen für solche Aspekte an, die durch Pseudo-Zufallszahlengeneratoren im Modell nachgebildet werden. Diese werden mit einer Zahl initialisiert, die man Seed nennt. Sie determiniert den Verlauf des erzeugten Zahlenstroms, sodass unter dem gleichen Seed gleiche Sequenzen von Zufallszahlen erzeugt werden. Die Simulationsläufe sind in der Folge ebenfalls deterministisch und rekonstruierbar. Parameter wie Seeds oder auch die vorgegebene zu simulierende Zeitspanne der Simulationsläufe, die die Rahmenbedingungen für ein Experiment und seine Simulationsläufe festlegen, nennt man Experimentparameter. Die wiederholte Simulation eines Modells mit unterschiedlichen Seeds und ansonsten gleichbleibenden Modell- und Experimentparametern nennt man Batchexperiment. Sie stellen eine Methode zur Erweiterung der erzeugten Simulationsstatistiken dar. Im Gegensatz zur Methode der Verlängerung eines Simulationslaufes wird die Statistik um Daten erweitert, die nicht von denen des ersten Simulationslaufes abhängen.

3. BPMN 2.0 und ihre Erweiterung zu Simulationszwecken

Ein Prozess im Sinne der BPMN ist gemäß Allweyer ein „Ablauf von Aktivitäten innerhalb einer Organisation“ (Allweyer (2009), 3 S. 5). Die BPMN (BusinessProcess Model and Notation, 2) ist eine grafische Modellierungssprache für Geschäftsprozesse, die in Form von Ereignissen, Aktivitäten, Sequenzflüssen und anderen Elementen der Prozessmodellierung ausgedrückt werden. Der Vorteil der BPMN gegenüber anderen Prozessbeschreibungssprachen wie BPEL (Business Process Execution Language) besteht in einer sehr einfachen Symbolik. Ferner handelt es sich bei BPMN um einen globalen, von der Object Management Group vorangetriebenen Standard. Die BPMN versteht sich als gemeinsame Prozesssprache von Laien und Experten über Organisationsgrenzen hinweg. Aus diesen Gründen wird die BPMN (in erweiterter Form) im Rahmen des Projektes SystOp Offshore Wind zur Modellierung der betrachteten Instandhaltungsprozesse verwendet. Um das Verständnis des Ansatzes von DESMO-Windpark-Studio zu ermöglichen, werden im Folgenden die wichtigsten Elemente der BPMN vorgestellt und ihre für die Simulation notwendigen Erweiterungen beschrieben.

3.1 Wichtige Elemente der BPMN

Die Prozessmodelle werden in DESMO-Windpark-Studio in sogenannten Kollaborationsdiagrammen abgebildet, die die Prozessteilnehmer und deren nachrichtenbasierte Kommunikation grafisch darstellen. Ihre wichtigsten Elemente, sind ohne ihre spezifischen Ausprägungen im Folgenden vorgestellt:

- Pools: Pools stellen einen Teilnehmer der Kollaboration dar. Teilnehmer können hierbei nicht nur einzelne natürliche Personen, sondern auch ganze Firmen sein. Soll ein Pool in sinnvolle Einheiten, wie beispielsweise eine Abteilung oder in einzelne Mitarbeiter einer Firma aufgeteilt werden, denen die Aktivitäten zuzuordnen sind, so kann dies durch sogenannte Swimlanes geschehen, die den Pool längs in Bereiche unterteilen.
- Aktivitäten: Aktivitäten sind diejenigen Elemente eines Prozesses, die die Durchführung zeitkonsumierender Tätigkeiten durch Menschen oder Maschinen darstellen. Sie werden in den Pool gezeichnet, der den ausführenden Teilnehmer darstellt. Innerhalb des Pools werden sie in die Swimlane gezeichnet, die diejenige Untereinheit des Pools repräsentiert, welche die Aktivität ausführt.

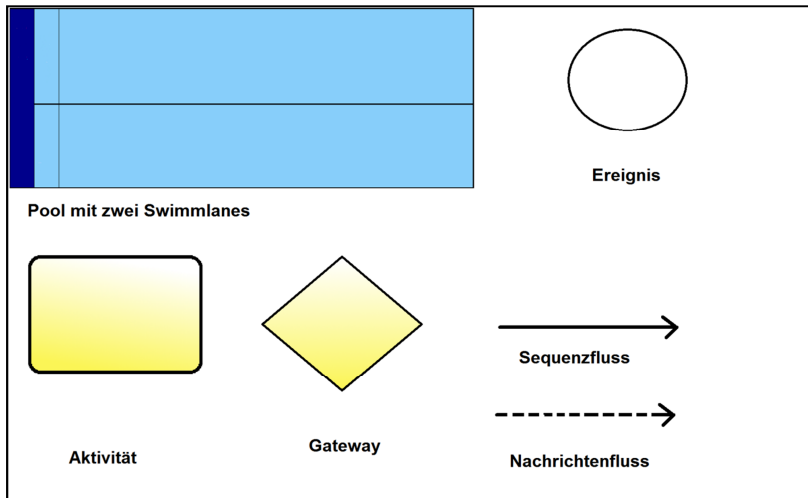


Abbildung 1:
Wichtige Elemente
der BPMN 2.0

- Sequenzflüsse: Sequenzflüsse verbinden die Ereignisse und Aktivitäten eines Prozesses, um so die Abfolge der Prozessausführung festzulegen.
- Gateways: Gateways verzweigen den Sequenzfluss oder führen ihn zusammen. Beispielsweise wird bei einem exklusiven Gateway der Sequenzfluss verzweigt. Mit Hilfe einer Regel wird dann anhand des Prozesszustands entschieden, welcher Pfad weiter verfolgt wird (siehe Beispielmodell im folgenden Abschnitt).
- Nachrichtenfluss: Ein Nachrichtenfluss stellt die Kommunikation zwischen zwei Teilnehmern dar.

3.2 Beispielmodell

Zur Veranschaulichung des Ansatzes wird hier ein Beispielmodell eingeführt, in dem die Instandsetzung einer Offshore-Windkraftanlage extrem vereinfacht abgebildet wird: Die Anlage gibt eine Störungsmeldung an ein Instandhaltungsunternehmen ab. In diesem entscheidet zunächst die Mitarbeiterin Tina Technik, ob es sich bei der Störungsmeldung um eine ernstzunehmende handelt oder um einen Fehlalarm. Abhängig von dieser Entscheidung löscht Frau Technik entweder den Fehler der Anlage, oder

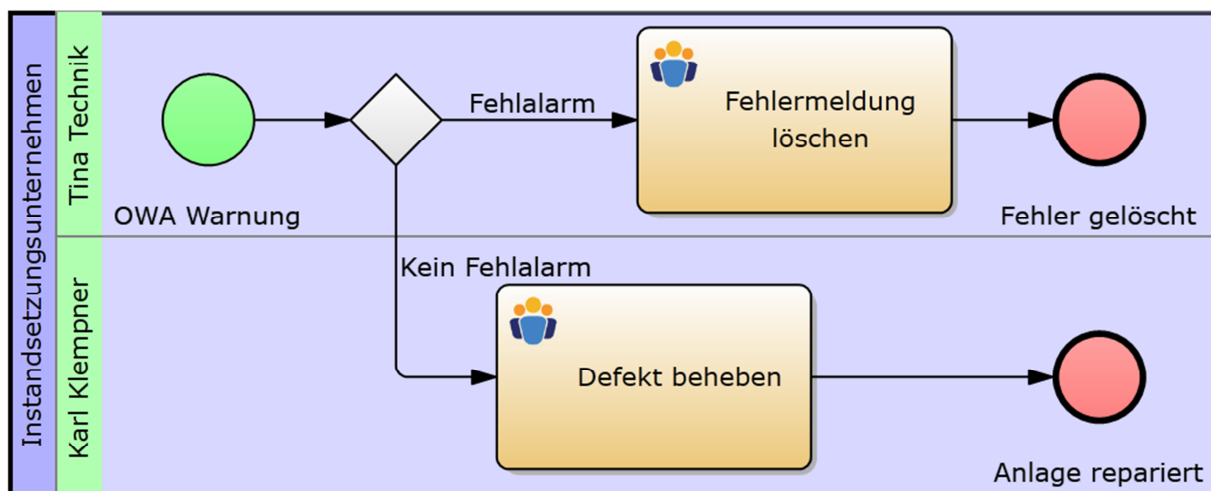


Abbildung 2: Beispielmodell

übergibt den Fall an ihren Kollegen Karl Klempner, der den Schaden behebt. In beiden Fällen endet der durchgeführte Prozess mit einem Endereignis. Im Gegensatz zu den in SystOp erhobenen Modellen ist von sämtlicher logistischer Komplexität abstrahiert und nur ein Teilnehmer in der Kollaboration abgebildet worden, um den Rahmen nicht zu sprengen

3.3 Erweiterung der BPMN zu Simulationszwecken

Wie am Beispielmodell zu erkennen, fehlen zur Simulation des Prozesses wichtige Informationen. Insbesondere zeitliche Ergebnisse, wie die durchschnittliche Durchlaufzeit des Prozesses sind maßgebliche Kennzahlen in Prozessoptimierungsszenarien für Instandhaltungsprozesse (vgl. Christophel, D. (2009), 7 S. 35). Um Informationen über diese gewinnen zu können, muss zunächst zeitliche Information im Modell enthalten sein. Aus diesem Grund wurde die BPMN im Projekt SystOp um vier wichtige Elemente erweitert: Erstens um Zwischenankunftszeiten von Ereignissen, sodass (auch als stochastische Verteilung) festgelegt werden kann, wie viel Zeit zwischen zwei Eintreten eines Ereignisses vergeht. Zweitens um Aktivitätsdauern, sodass ein Simulant angeben kann, wie viel Zeit für die Bearbeitung einer Aktivität in Anspruch genommen wird. Drittens um einen (optionalen) Zeitverzug bei der Übermittlung von Nachrichten zwischen zwei Prozessteilnehmern. Viertens wurde die Möglichkeit geschaffen, eigene Variablen an Ereignissen und Aktivitäten zu definieren und mit Werten zu belegen. Anhand dieser können Entscheidungen an Gateways reguliert werden. Diese Informationen sind als Modellparameter realisiert worden, sodass sie mit Hilfe der unter 5. beschriebenen Erweiterung variiert werden können.

Für das Beispielmodell sei für die Simulation eine Zwischenankunftszeit zwischen zwei Störungsmeldungen von Konstant zwei Stunden angenommen. Die Bearbeitung der Aktivität „Fehlermeldung löschen“ betrage eine Minute und das Beheben des Defekts benötige sechs Stunden. Um die Entscheidung über die Art des Fehlers zu treffen, bedarf es einer eigens definierten Variablen, anhand derer die Entscheidung am Gateway getroffen werden kann. Wir definieren eine Integer-Variable „Fehlalarm“, die am Startereignis den Wert 1 erhält, wenn es sich bei der eingehenden Warnung um einen Fehlalarm handeln soll und den Wert 0, wenn dem Alarm ein echter Defekt zugrunde liegt. Dies kann nun am Gateway geprüft werden, sodass eine der beiden möglichen Folgeaktivitäten eingeleitet wird.

4. DESMO-Windpark-Studio

Die Simulation der Modelle geschieht auf Basis des Simulationsrahmenwerkes DESMO-J, das sowohl in Forschung und Lehre, als auch in kommerziellen Anwendungen eingesetzt wird. Es bietet verschiedenste Komponenten zur Modellierung, Simulation und statistischen Auswertung des Verhaltens eines Realsystems im Rahmen sowohl ereignisorientierter als auch prozessorientierter und sogar hybrider Modellierung. Die Software DESMO-Windpark-Studio verdankt dem Simulationsframework ihren Namen. Sie setzt auf der Plug-in-Architektur der Rahmenanwendung Empinia auf. Dabei handelt es sich um ein Rahmenwerk zur Entwicklung von Anwendungen mit grafischer Oberfläche, das ursprünglich für betriebliche Umwelt-Informationssysteme entwickelt wurde. Die Rahmenanwendung stellt Erweiterungs-Schnittstellen, sogenannte Extension-Points zur Verfügung, die einen eindeutigen Namen tragen. Der Plug-in-Entwickler hängt den erzeugten Klassenbibliotheken zur Einbindung seiner Plug-ins eine XML-Datei mit dem Namen bundle.xml an. Diese enthält Informationen über die implementierten Erweiterungen. Insbesondere werden für jede implementierte Erweiterung der zugehörige Extension-Point und die implementierende Klasse genannt. Diese XML-Datei wird beim Start der Anwendung durchsucht und die entsprechenden Klassen werden von Empinia registriert. Um die Plug-ins zu nutzen, kann eine von Empinia instanziierte Klasse in ihrem Konstruktor einen Parameter vom Typ der Schnittstelle angeben, die von dem zugehörigen Extension Point ausgewiesen ist. Empinia prüft dann, ob eine

Entsprechende Erweiterung zu dem Extension Point angemeldet ist und instanziiert und übergibt das geforderte Objekt. Ein Plug-in kann seinerseits auch weitere Extension Points definieren, sodass auch hier eine weitergehende Erweiterung der Software vollzogen werden kann. Es stellt dazu das zugehörige Interface bereit, das von einem erweiternden Plug-in zu implementieren ist. Die Basisanwendung ist neben der hier beschriebenen Erweiterung auf diese Weise bereits um Editoren für Windpark- und Ressourcenmodelle sowie einen BPMN-Modelleditor und die Möglichkeit zur Durchführung einzelner BPMN-Experimente ohne Variation der Modellparameter erweitert worden.

5. Die Experimentierumgebung von DESMO-Windpark-Studio

Nachdem DESMO-Windpark-Studio Nutzern die Möglichkeit bot, Windpark-, Ressourcen-, sowie BPMN-Modelle zu erstellen, und Simulationen mit BPMN-Modellen durchzuführen, wurde es um eine Perspektive zur Planung und Durchführung von Batchexperimenten erweitert. Diese Batchexperimente vollziehen mehrere Wiederholungen von Simulationsläufen für jede Modellkonfiguration. Bei Batchexperimenten entstehen entsprechend der Anzahl der Wiederholungen mehrere Simulationsstatistiken pro Modellkonfiguration, die zu aggregierten Berichten über das Verhalten eines konfigurierten Modells über mehrere Simulationsläufe hinweg zusammenzufassen sind. Um nachträglich auch andere Modelle als BPMN-Kollaborationsmodelle in DESMO-Windpark-Studio simulieren zu können, wurde der Extension Point des Modellkonfigurators entwickelt. Implementierende Plug-ins dieses Extension Points konfigurieren ein Modell entsprechend des spezifischen Metamodells und einer anzugebenen Konfiguration der Modellparameter. Sie liefern im Rückgabewert die konfigurierte Version des Modells. Durch diese Trennung der Experimentplanung von metamodell-spezifischer Funktionalität bleibt sie generisch und überlässt solche Funktionalität den Plug-ins, die neue Modelle in die Software einbringen. Ein Plug-in für Windkraftanlagen muss somit seinen eigenen spezifischen Modellkonfigurator implementieren, ohne die Experimentplanung zu berühren.

Die Berechnung der Konfigurationen und der sich daraus ergebenden Simulationsläufe geschieht bei jeder Veränderung einer Modellparameter-Belegung durch eine Klasse namens *RunsFactory*. Wird ein Experiment angestoßen, so wird der Start in einem Objekt der Klasse *SimulationStarter* vollzogen. Diese verknüpft die einzelnen Simulationsläufe mit einem als *BatchdataCollector* bezeichneten Objekt. Dieses sammelt und aggregiert die Statistiken der zur selben Konfiguration gehörigen Simulationsläufe. Die Experimentier-Perspektive von DESMO-Windpark-Studio bietet die Möglichkeit, anhand des parametrisierten Modells einen Modellparameter anzuwählen und unterschiedliche stochastische Verteilungen zu hinterlegen (vgl. Abbildung 3). So kann für die in 3.2 eingeführte naive Version der Instandhaltung verglichen werden, wie sich die Prozesszeit bei einer Gleichverteilung von Fehlalarmen und ernstzunehmenden Störungen bzw. bei einer Empirischen Verteilung mit 90% ernstzunehmenden Fehlmeldungen verhält. Dazu wurde das Start-Ereignis angewählt, an dem die Variable „Fehlalarm“ hinterlegt wurde. Am rechten Rand können nun neue Verteilungen als Belegung für den Parameter konfiguriert und hinterlegt werden. Hier ist der Fall der empirischen Verteilung zu sehen, bei der beide Werte des Parameters gleich wahrscheinlich sind. Die Experimentparameter sind im verborgenen Reiter „Experiment parameter configuration“ eingestellt worden. Bei Änderungen an den Parametern wird die Menge der entstehenden Konfigurationen als Kreuzprodukt neu berechnet und am linken Rand im Reiter „Computed model configurations“ dargestellt. Hier können auch konkrete Seedwerte für bestimmte Konfigurationen vergeben werden. Ferner kann der Simulant hier ganze Konfigurationen aus der Simulation ausschließen, falls diese nicht simulationswürdig erscheinen(etwa, weil sie in der Realität nicht auftreten können).

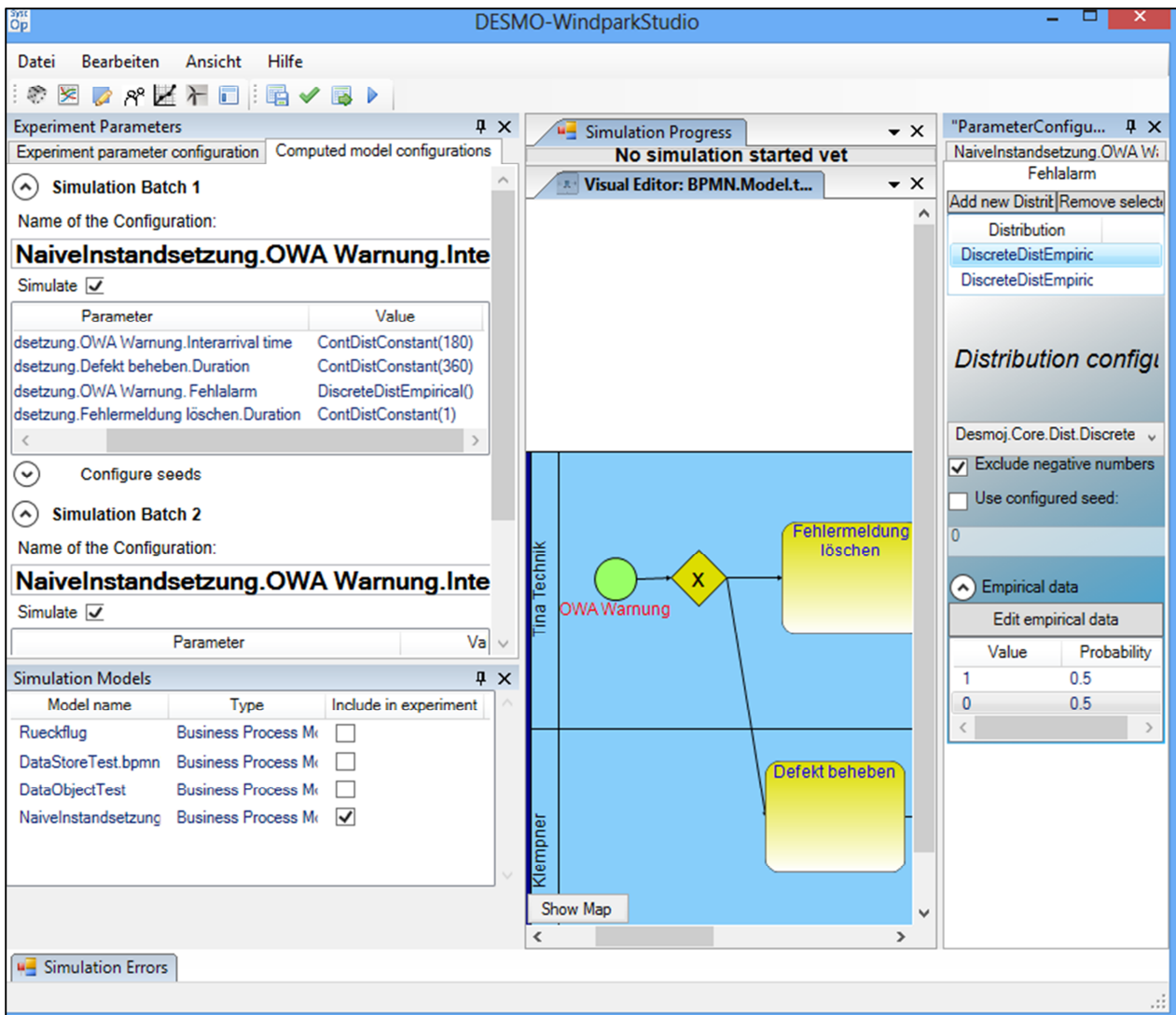


Abbildung 3: Screenshot der Experimentplanung in DESMO-Windpark-Studio

Wird der Start der Simulation veranlasst, so erzeugt der oben erwähnte *BatchdataCollector* ein Hauptmodell, in dem später alle Statistiken der Wiederholungsläufe aggregiert werden. Daran anschließend wird die statische Version des Modells in ein simulierbares Objekt der DESMO-J-Bibliothek umgewandelt und simuliert. Diese Umwandlung erfolgt für jede Wiederholung des Batchexperiments.

Nach Durchführung der Simulation werden die zusammengehörigen Statistiken der Wiederholungen aggregiert und Kennzahlen ihrer statistischen Verteilungen in einem Batchbericht zusammengeführt. Beispielsweise wird die durchschnittliche Durchlaufzeit des Prozesses für alle angesetzten Wiederholungsläufe zusammengeführt und die Kennzahlen der statistischen Verteilung über die Simulationsläufe hinweg im Bericht festgehalten, der nach der Simulation in html-Form angezeigt wird.

6. Fazit

Mit der Entwicklung einer Experimentplanungs-Komponente in DESMO-Windpark-Studio ist eine wichtige Grundlage für die Durchführung von Batchexperimenten im Projekt SystOp Offshore Wind gelegt worden. Sie erlaubt es den implementierenden Modellierungskomponenten, Modellparameter auszuweisen

und macht diese dem Anwender zugänglich. Simulanten sind auf dieser Basis in der Lage, Experimentparameter festzulegen und unterschiedliche Konfigurationen der Modelle gegenüberzustellen. Hierbei unterstützt die Experimentplanung die Wiederholung der Simulationsläufe mit unterschiedlicher Initialisierung von Zufallszahlengeneratoren. Die so durchführbaren Wiederholungen von Simulationsläufen werden nach der Simulation statistisch zusammengeführt, um eine statistische Basis für die Auswertung der Experimente zu legen. Durch die umgesetzte Kapselung der Experimentplanung von metamodell-spezifischen Funktionalitäten wird es möglich sein, auch Windpark- und Ressourcenmodelle in die bestehende Experimentierumgebung einzubinden.

7. Ausblick

Nachdem die technischen Grundlagen zur Durchführung von Simulationsexperimenten mit BPMN-Modellen in DESMO-Windpark-Studio gelegt sind, ist im Folgenden geplant, erste Exemplare der vorliegenden BPMN-Modelle in die von DESMO-Windpark-Studio unterstützte XML-Formatierung gemäß OMG-Standard (vgl. Object Management Group (2011), 3) zu überführen. Die Simulation einzelner Läufe wird daraufhin zeigen, in wie fern die Modelle standardkonform spezifiziert sind. Da es sich bei DESMO-Windpark-Studio um eine in der Entwicklung befindliche Software handelt, wird gleichzeitig zu prüfen sein, in wie fern diese für zweckdienliche Simulationsexperimente zu erweitern ist. Denkbar ist eine Erweiterung der Berichtsfunktion, die bislang die von DESMO-J erzeugten Berichte ungefiltert als html-Dokumente ausgibt. Eine stärkere Ausrichtung auf optimierungswürdige Prozesskennzahlen und eine grafische Aufbereitung der Berichte im Sinne der angestrebten Prozessoptimierung ist eine diskutierte Perspektive. Unverzichtbar ist außerdem das Einbinden von Ressourcen-, Wetter-, und Windparkmodellen in die Simulationsexperimente.

Literatur

- SystOp Offshore Wind (2010): Entwicklung eines Planungs- und Optimierungswerkzeugs zur systemumfassenden Optimierung des Leistungssystems Offshore-Windpark, Gesamtvorhabensbeschreibung
- Object Management Group (2011): Business Process Model and Notation (BPMN), OMG Document Number: formal/2011-01-03
- Allweyer, T. (2009): Kollaborationen, Choreographien und Konversationen in BPMN 2.0, Fachhochschule Kaiserslautern.
- Page, B. (1991). Diskrete Simulation: Eine Einführung mit Modula-2. Springer-Lehrbuch. Springer, Berlin.
- Wainer, G. A. (2010). Discrete-event modeling and simulation: Theory and applications. Computational analysis, synthesis, and design of dynamic systems. CRC Press, Boca Raton
- Niemeyer, G. (1977). Kybernetische System- und Modelltheorie: System dynamics. Systemstudium Wirtschaftsinformatik. Vahlen, München.
- Christophel, D. (2009). Key Performance Indicators zur Optimierung von Instandhaltungsprozessen. Diplomica Verlag, Hamburg.
- Wittmann, J. (1993). Eine Benutzerschnittstelle für die Durchführung von Simulationsexperimenten. Dissertation, Univ, San Diego, Erlangen-Nürnberg.