

# **Expect, ein industrielles Tool für Zuverlässigkeits-, Sicherheits- und Kostenanalysen komplexer Systeme**

Stefan Heller, Felix Engelhard  
{stefan.heller, felix.engelhard}@daimlerchrysler.com  
DaimlerChrysler AG, REI/AA

Graham Horton, Fabian Wickborn  
{graham, wickborn}@sim-md.de  
Universität Magdeburg, ISG

## **Kurzfassung**

Dieses Paper stellt den aktuellen Stand des Tools Expect und dessen Einsatzmöglichkeiten vor. Expect ist ein Tool zur Modellierung, Parametrierung und Auswertung verschiedener Modellierungsparadigmen, wie zum Beispiel Petri Netze, Fehlerbäume und Zustandsräume. Für die einzelnen Modelltypen stehen jeweils spezielle grafische Editoren zur Verfügung, die wiederum verschiedene Analytoren über eine eigene Schnittstelle anbinden können. Darüber hinaus bietet Expect die Möglichkeit, die Daten, die für die Modellparametrierung herangezogen werden, zu speichern und entsprechend aufzubereiten. Diese Daten können dann direkt oder aber verdichtet zu einer statistischen Verteilung zur Parametrierung von Modellen verwendet werden. Zusätzlich existiert eine eigene Skriptschnittstelle, die unabhängig vom verwendeten Modelltyp Reihenanalysen und Optimierungen unter zu Hilfenahme einer abstrakten Skript-Engine erstellen kann. Über diese Basisfunktionen hinaus ist es zusätzlich möglich, Expect in Verbindung mit anderen Tools zu betreiben. Ein Einsatzbereich ist zum Beispiel die Parametrierung von Modellen über eine Schnittstelle zu einer speziell entwickelten Web-Plattform für Felddaten. Diese Web-Plattform beinhaltet zum einen Felddaten über Bauteile und ermöglicht zusätzlich die komfortable Analyse der Daten. Zusätzlich ist es zukünftig auch möglich, Modelle in diese Plattform einzubinden und über die Skriptschnittstelle in der Web-Plattform zu parametrieren und auszuwerten.

## **1 Einleitung**

Durch die enorme Zunahme der Komplexität und Vernetzung heutiger Fahrzeugsysteme ist der Aufwand für Zuverlässigkeits-, Sicherheits- und Kostenanalysen immens gewachsen. Oftmals wird es nahezu unmöglich, solche Analysen ohne Unterstützung von speziellen Modellierungs- und Analysewerkzeugen zu erstellen. Darüber hinaus sind in vielen Bereichen spezielle Entwicklungs- und Entstehungsprozesse umgesetzt, die auf Grund von internen Vorgaben bis hin zu internationalen Normen nicht ohne weiteres verändert werden können. Aus diesem Grund sollte ein Werkzeug möglichst einfach integriert werden oder zumindest ohne großen Aufwand unterstützend eingesetzt werden können. Um diesen Anforderungen gerecht zu werden, wird in der DaimlerChrysler Forschung das Tool Expect in Kooperation mit der Universität Magdeburg entwickelt. Mit diesem Tool ist es möglich, Modelle für Fahrzeugsysteme zu erstellen und zu analysieren. Darüber hinaus kann es einfach an Produktentstehungs- und Sicherheitsprozesse angepasst werden.

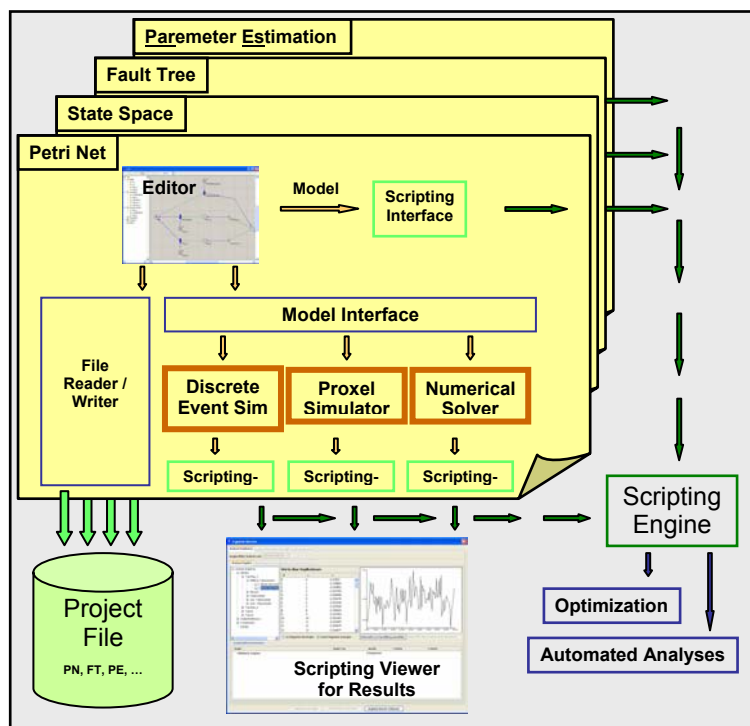
Die Idee zur Integration verschiedener Modelltypen in ein Tool ist nicht neu, zum Beispiel verfolgen dieses Ziel Tools wie SHARP [1], SMART [2] und Möbius [3]. Das Ziel dieser Tools ist es, verschiedene Typen von Modellen in einem Tool zu bearbeiten und miteinander zu verknüpfen. Dieses Ziel verfolgt auch Expect, um die Vorteile einzelner Modelltypen zu nutzen. Darüber hinaus verfolgt allerdings Expect noch weitere Ziele: Mit Expect ist es nicht nur möglich, mit dem Tool selbst zu arbeiten, sondern es ist auch möglich, sich mit anderen Tools über Schnittstellen zu verbinden. So ist es möglich die Parametrierung von Modellen zu erleichtern, oder aber auch die Erstellung von Analysen ohne einen offensichtlichen Einsatz von Expect durchzuführen. Darüber hinaus werden auch Reihenanalysen und Optimierungen über die integrierte Skriptschnittstelle unabhängig vom Modelltyp ermöglicht. Zusätzlich bietet die Skriptschnittstelle die Möglichkeit, Analysen von Modellen, die aus verschiedenen Modelltypen bestehen, durchzuführen.

In den folgenden Kapiteln wird zunächst in Kapitel 2 die allgemeine Architektur vorgestellt. In diesem Kapitel wird die Grundlage für die Erläuterung der Einsatzmöglichkeiten von Expect in Kapitel 3 geschaffen. Abschließend wird in Kapitel 4 der aktuelle Stand der Entwicklung vorgestellt, um dann die weiteren Arbeiten aufzuführen.

## **2 Die Architektur von Expect**

Expect ist ein Framework, durch das die Erstellung, Parametrierung, Komposition und Analyse von Modellen verschiedener Typen ermöglicht wird. Um dies zu realisieren, gibt es in Expect für jeden Modelltyp ein eigenes Modul. In Abbildung 1 links oben exemplarisch für Petri Netze, Zustandsräume und Fehlerbäume zu sehen. Da die Anforderungen der einzelnen Modelltypen extrem unterschiedlich sind, gibt es in jedem Typmodul einen eigenen Editor, durch den die speziellen Anforderungen und Modell-Umfänge exakt abgebildet werden können. Hierzu kann mit dem grafischen Teil des Editors die Struktur des Modells komfortabel und schnell erstellt und zusätzlich durch entsprechende Dialoge das Modell parametrierbar werden. Parallel zum Editor enthält jedes Modul ein eigenes Submodul zum Speichern und Lesen von Modellen. Dies ist notwendig, da jeder Modelltyp unterschiedliche Umfänge aufweist und es nur mit extrem hohem Aufwand möglich wäre, ein abstraktes Dateiformat für alle Typen zu definieren und umzusetzen. Das Speichermodul ist über eine Schnittstelle an das eigentliche Expect angebunden, wodurch die Verwaltung der einzelnen Modelle abstrakt durch das Framework erledigt werden kann. Zu diesem Zweck verwaltet Expect eine Projektdatei, in der beliebige Modelle eingefügt, bearbeitet, kopiert und gelöscht werden können. Über die eigentlichen Modelle hinaus können auch Felddaten und deren statistischen Auswertungen aus dem Pares-Modul mit verwaltet werden, die zur Parametrierung herangezogen werden können. Das in Expect enthaltene Pares-Modul ermöglicht die Verwaltung und statistische Analyse von Daten. Der Unterschied zwischen einem Modell und Felddaten ist hierbei aus Sicht des Frameworks unerheblich, wodurch Expect die einzelnen Bestandteile eines Projektes nur unterscheiden können muss, um sie dann an das eigentliche Modul weiterzuleiten. Durch diese Abstraktion von Modellen und Parametrierungsdaten in Expect ist es möglich geworden, alle notwendigen Daten für komplette Zuverlässigkeits-, Sicherheits- oder Kostenanalysen in einer einzigen Projektdatei zu verwalten, wodurch die Prozesssicherheit bei der Erstellung von Analysen erhöht wurde.

Mit der Projektverwaltung in Expect und den spezifischen Editoren in den einzelnen Modulen können neue Modelle angelegt, gespeichert und bearbeitet werden. Um die erstellten Modelle zu analysieren, wurde in jedem Modul eine eigene, vom Modelltyp sehr stark abhängige Schnittstelle entwickelt, über die entsprechende Analysatoren angebunden werden können. Durch diese Typ-spezifischen Schnittstellen werden die Analysatoren mit allen notwendigen Daten des Modells versorgt, wodurch dann Analysen erstellt werden können. Da aber eine reine Datenschnittstelle oftmals nicht ausreichend ist, da über sie nur Struktur und Parametrierungsinformation ausgetauscht wird, wurde in Expect die Schnittstelle erweitert, um die Implementierung von Analysatoren zu vereinfachen. Ein Beispiel hierfür ist ein Modul in Expect, mit dem Funktionscode in Modellen durch die Schnittstelle aufbereitet wird und dann für die Analysatoren fertig ausführbar zur Verfügung steht.



**Abbildung 1:** Architektur von Expect

Durch die Entscheidung zu diesem Design ist es allerdings nicht ohne weiteres möglich, einen umgesetzten Analysator eines speziellen Modelltyps für einen anderen Typ zu verwenden, wodurch sich Expect von anderen Tools wie zum Beispiel Möbius [3], bei dem verschiedene Typen von Modellen auf ein zentrales Format transformiert werden, unterscheidet. Der Vorteil bei diesem Design ist allerdings die Möglichkeit, sich auf spezielle Gegebenheiten von Modelltypen einzustellen, um auch unterschiedliche Ergebnistypen zu erhalten. So ist es zum Beispiel mit Expect möglich, bei einem Petri Netz die charakteristischen Größen von Stellen und Transitionen zu erhalten, wohingegen bei Fehlerbäumen Cutsets bestimmt werden können. Da bei beiden grundverschiedene

Analysen vorliegen, macht es auch keinen Sinn, aufwendige Routinen für die Transformation zu einem zentralen Format zu entwickeln, um dann wiederum auf verschiedene Analysatoren zu verzweigen. Zusätzlich kann sich bei den Analysatoren auf die spezifischen Anforderungen des Modells konzentriert werden, wodurch eine optimale Analyse und Performance gewährleistet werden. Um aber nicht für jeden Analysator alles vollständig neu implementieren zu müssen, wurden Grundbestandteile von Analysatoren als Module zur Verfügung gestellt. Einige Beispiele hierfür sind Zufallszahlen-generatoren für Simulatoren und ein Modul für die Verteilung von Berechnungen auf Rechenknoten.

Neben der zentralen Dateiverwaltung in Expect und den Modulen der Modelltypen ist in Expect eine zentrale Skriptschnittstelle umgesetzt. Mittels dieser Schnittstelle kann unabhängig vom Typ auf Parameter von Modellen und auf Analysatoren zugegriffen werden, wodurch es auf eine abstrakte Art und Weise möglich wird, zum Beispiel Reihenanalysen und Optimierungen zu generieren. Zusätzlich kann diese Schnittstelle bei der Komposition von Modellen für die Analyse verwendet werden. Um dies zu ermöglichen, sind von den Modell-Modulen einige Schnittstellen zu implementieren, wodurch der Modelltyp an das Skriptmodul angebunden wird. Eine dieser Schnittstellen ist die Schnittstelle der Modellparameter, mit der es möglich wird, abstrakt auf alle veränderlichen Parameter eines Modells zuzugreifen. Zusätzlich erlaubt diese Schnittstelle das temporäre setzen von Parametern, wodurch das Modell mit anderen Parametersätzen analysiert werden kann. Außerdem gibt es eine Schnittstelle für Analysatoren, die von jedem Analysator zu implementieren ist. Mittels dieser Schnittstelle können Analysatoren, unabhängig vom Typ und dem dahinter liegenden Modelltyp, parametrisiert und gestartet werden. Zusätzlich zur eigentlichen Analyseschnittstelle implementieren die Analysatoren auch eine abstrakte Schnittstelle für Ergebnisse. Mittels dieser Schnittstelle werden nach der Analyse die Ergebnisse zur Verfügung gestellt und können dann auch ohne großen Aufwand durch eine zentrale Darstellungskomponente visualisiert werden. Auch hierbei ist es unerheblich, von welchem Analysator und welchem Modelltyp die Ergebnisse erstellt wurden. Um diese Abstraktion jeweils zu ermöglichen, gliedern sich alle vorgestellten Schnittstellen in zwei Bereiche: Zum einen liefert eine Schnittstelle Metadaten, durch die alle zur Verfügung gestellten Daten bekannt gegeben werden, und zum anderen die eigentlichen Daten. Damit die Daten wiederum abstrakt übergeben werden können, wurden Basisdatentypen durch die Skriptschnittstelle definiert, auf die sich alle Schnittstellen beziehen. Durch die Implementierung der notwendigen Schnittstellen durch die Modellmodule können diese Modelle durch die Skriptschnittstelle angesprochen und somit erweiterte Analysen erstellt werden. Die zentrale Stelle hierzu ist das Skript-Modul, dem ein Modell in Form eines Skriptes übergeben werden kann und das dann die notwendigen Analysen oder Optimierungen erstellt. Das Skript-Modul kann sowohl direkt in Expect als auch von extern angesprochen werden, wodurch es möglich wird Analysen ohne die grafische Oberfläche von Expect zu erstellen. Durch diese Möglichkeit der Anbindung von Expect wurde es geschafft, dass andere Tools Expect verwenden können und Expect vollständig im Hintergrund Analysen erstellt. Zusätzlich gibt es in Expect auch eine Visualisierungskomponente, durch die Ergebnisse von Analysatoren dargestellt werden können. Durch die abstrakte Darstellung der Ergebnisse ist es so einfach möglich, Ergebnisse unterschiedlicher Modelltypen grafisch aufzubereiten.

### 3 Expect in Verbindung mit anderen Tools

Ein Hauptanliegen für Expect ist eine möglichst einfache Unterstützung bestehender Prozesse, die oftmals schon etabliert sind und sehr selten verändert werden können. Aus diesem Grund wurde bei Expect darauf geachtet, dass es zum einen möglich ist, sich ohne großen Aufwand in bestehende Datenprozesse einzuklinken, und zum anderen, dass Expect in bestehende Abläufe leicht einzubinden ist. Wie dies mit Expect möglich ist, wurde exemplarisch für einige Szenarios in Abbildung 2 dargestellt.

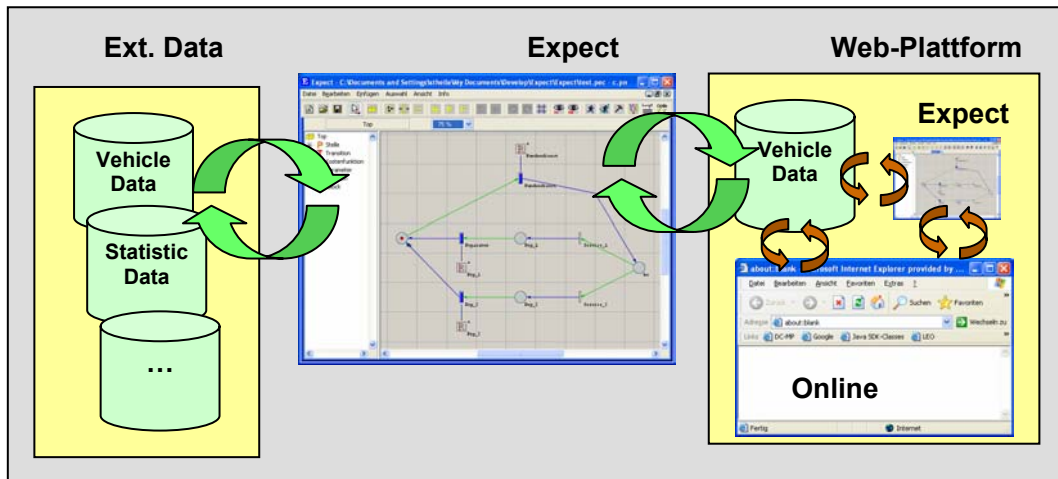


Abbildung 2: Expect mit anderen Tools

Die Anwendung von Expect gliedert sich grob in den „Stand alone“ Betrieb und den integrierten Betrieb. In Abbildung 2 ist dies mit dem Tool in der Mitte der Abbildung und rechts integriert in die Web-Plattform dargestellt. Wird Expect als Tool betrieben, wird mit der eigenen grafischen Oberfläche gearbeitet, wohingegen bei einem integrierten Einsatz bereits erstellte Modelle über die Skriptschnittstelle parametrisiert und analysiert werden. Für den eigentlichen Toolbetrieb ist die Hauptanforderung eine möglichst einfache Datenanbindung an bestehende Tools. Zu diesem Zweck gibt es in Expect verschiedene Möglichkeiten. Die einfachste ist ein vollständig autonomer Betrieb, bei dem Modelle über die Standarddialoge der Editoren vom Benutzer parametrisiert werden können. Darüber hinaus ist es möglich, die Daten, die für eine Parametrierung verwendet werden sollen, direkt in Expect einzulesen und sie dort mit dem Pares-Modul aufzubereiten. Eine weitaus komfortablere Methode ist es aber, sich direkt mit anderen Tools zu verbinden und so keine Datenduplizierung herbeizuführen. Dies ist in Abbildung 2 mit den grünen Pfeilen rechts von Expect zur Web-Plattform angedeutet. Durch diese Schnittstelle können Modell-Elemente direkt parametrisiert werden.

Für die zweite Betriebsart von Expect wird die Skriptschnittstelle verwendet, durch die es möglich ist, Modelle zu parametrisieren und auszuwerten. In Abbildung 2 ist dies durch die braunen Pfeile in der Web-Plattform dargestellt. Durch die Skriptschnittstelle ist es möglich, Expect von extern anzusprechen und so ein erstelltes Skript abarbeiten zu lassen. Hierbei kann zum einen eine Reihenanalyse oder aber zum anderen eine Optimierung erstellt werden. In beiden Fällen wird das Skript im Toolbetrieb von Expect

erstellt und kann dann für den integrierten Einsatz verwendet werden. Da das Skript in XML-Code vorliegt, kann es von externen Tools geparkt und angepasst werden. Im dargestellten Fall wird Expect in die bei DaimlerChrysler entwickelte Web-Plattform eingebunden und zu verschiedenen Analysen herangezogen. Von besonderem Interesse bei diesen Analysen ist das spezielle Feldverhalten von Fahrzeugkomponenten, die über das Skript analysiert werden können. Der größte Vorteil hierbei ist es, dass auch Benutzern die keine Modellierungsspezialisten sind, wie sie für Analysen im Toolbetrieb notwendig sind, diese Analysen zugänglich werden.

#### **4 Aktueller Stand und weitere Arbeiten**

In der aktuellen Version von Expect ist es möglich, Fehlerbäume, Zustandsräume und Petri Netze grafisch zu editieren und zu analysieren. Darüber hinaus wird momentan ein Modul für Reliability Block Diagramme entwickelt. Für Fehlerbäume steht die qualitative Cut-Set und die quantitative Analyse zur Verfügung, wobei alle gängigen Gattertypen unterstützt werden. Im Modul für Zustandsräume können beliebige Verteilungen für Zustandsübergänge modelliert werden und mit einem diskreten Simulator analysiert werden. Das Modul für Petri Netze ermöglicht die Erstellung erweiterter stochastischer Petri Netze, deren Modellierungsumfang zum Beispiel folgende Standarderweiterungen umfasst: Stellenkapazitäten, Guardfunktionen, verschiedene Verteilungen, Race-Policies und Prioritäten. Darüber hinaus können verschiedene Reward-Typen wie z.B. Impulsrewards definiert werden. Für die Analyse von Petri Netzen steht ein diskreter Simulator zur Verfügung, der auch verteilt auf verschiedene Rechenknoten eingesetzt werden kann. Zusätzlich ist die Analyse mit einem Proxelsimulator [5] und die numerische Analyse über Phasentypverteilungen [4] möglich. Aktuell wird an der Umsetzung eines Proxelsimulators für Zustandsräume gearbeitet und an der Anbindung aller Analysatoren und Modelltypen an die Skriptschnittstelle, um so die vollständige Kopplung von verschiedenen Modellen und Modelltypen zu ermöglichen. Parallel hierzu wird auch an die Einbindung in die Web-Plattform gearbeitet.

#### **5 Literatur**

- [1] *R. A. Sahner and K. S. Trivedi. Reliability modeling using SHARPE, in IEEE Transactions on Reliability, vol. R-36, no. 2 pp. 186-193, June 1987.*
- [2] *G. Ciardo und A. S. Miner. Smart: Simulation and markovian analyzer for reliability and timing. In Proceedings of IEEE International Computer Performance and Dependability Symposium Sep. 1996, p. 60.*
- [3] *G. Clark, T. Coutney, D. Daly, D. D. Deavours, S. Derisavi, J.M. Doyle, W. H. Sanders and P. G. Webster. The Möbius modelling tool. In Proceedings of Petri Nets and Performance Models (PNPM 2001), Aachen, Germany, Sept. 2001, pp. 241-250.*
- [4] *F. Engelhard, S. Heller and G. Horton. Safety Analyses with non-Markovian Stochastic Petri nets. Submitted to the 18. Symposium Simulationstechnik 2005.*
- [5] *F. Wickborn, G. Horton, F. Engelhard, S. Heller. A General-Purpose Proxel Simulator for an Industrial Application. Submitted to the 18. Symposium Simulationstechnik 2005.*